

COMPUTER MODEL FOR ESTIMATION OF TEACHING FACULTY REQUIREMENT

by

KRISHNASWAMY RANGANATHAN

M. Tech (Agricultural Engineering)

Indian Institute of Technology, Kharagpur, 1963

959-5358

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1972

Approved by:



Major Professor

20
2668
R4
1972
R36
C.2

Documents

TABLE OF CONTENTS

PAGE

I.	INTRODUCTION	1
II.	ANALYSIS OF THE PROBLEM.	3
	2.1 GENERAL METHODOLOGY	5
	2.1.1 Method 1	5
	2.1.2 Method 2	6
III.	ESTIMATION OF COURSE PARAMETERS.	9
	3.1 ESTIMATION OF COURSE PARAMETERS	9
	3.2 ESTIMATION OF COURSE CURRICULUM FRACTIONS FROM CURRICULUM SPECIFICATIONS.	1 .12
	3.3 DECISION RULE FOR SECTION SIZE.15
IV.	COMPUTATIONAL ALGORITHM.18
	4.1 PROGRAM COMPONENTS.18
	4.2 INPUT PARAMETERS.18
	4.3 OUTPUT VARIABLES.19
	4.4 GENERAL COMPUTATION PROCEDURE19
	4.5 USE OF THE COMPUTER PROGRAM19
V.	SENSITIVITY ANALYSIS22
	5.1 EFFECT OF INPUT PARAMETERS ON 'FTE ₁ '23
	5.2 EFFECT OF INPUT PARAMETERS ON 'FTE ₂ '23
	5.3 CUMULATIVE EFFECT OF INPUT PARAMETERS24
VI.	RESULTS AND DISCUSSION37
	6.1 RESULTS37
	6.2 DISCUSSION37

APPENDIXES.	39
A. EXAMPLES TO SHOW THE DEVIATION OF ESTIMATED STUDENT COUNT FROM ACTUAL STUDENT COUNT, WITH REGULARITY ASSUMPTION.	40
B. EXAMPLE OF HISTORICAL DATA.	41
C. MODIFICATION FACTOR FOR COURSE CURRICULUM FRACTIONS	42
D. FUNCTIONS OF SUBROUTINES	48
E. GLOSSARY OF TERMS	50
F. GENERAL FLOW CHART AND COMPUTER PROGRAM	58
G. SAMPLE COMPUTER OUTPUT.	84

CHAPTER I

INTRODUCTION

Personnel planning is part of any administrative set up in an organization. An administrator is required to forecast his personnel requirement and his estimate is based on his knowledge of the trend and behavioral pattern in the past, and a set of decision parameters. These decision parameters can be both qualitative and quantitative. The qualitative parameters are assigned certain reasonable weights or values before a decision is made.

In this report an attempt is made to build a computer model for estimation of teaching faculty requirement for the Engineering School of Kansas State University.

A faculty member is required to perform either one or a combination of the following duties:

1. Teaching
2. Research
3. Administration

The weight given to each of these functions vary with each faculty member. Each of these functions impart a certain workload on the faculty member. Hence the faculty requirement can be conceived as being constituted of the requirements for each of these functions. This functional categorization of faculty requirement also aids the administrator in his assignment of duties to individual faculty members. This model deals with the faculty requirement for only one such function namely teaching. Though the model is built for the Engineering School in particular, care is taken to see that

the procedure could be adapted to any school.

The following decision parameters were considered in the model.

1. Desired section size - Desired number of students in a section.
2. Minimum section size - Minimum number of students in a section.
3. Number of students in each curriculum at various levels - Freshmen, Sophomore, Junior and Senior.
4. Weight factors to quantify the proportion of the load imparted by the student count in a course, level of the course and the mode of instruction; the level being an indication of the difficulty associated with the course and the mode of instruction representing the way the course is taught, namely lecture, recitation, laboratory and individual instruction (called problems).
5. Full time equivalent credit hours for each faculty member.
6. Full time equivalent credit hours for the individual instructions.

The model also includes a decision rule to determine the number of sections resulting from the projected enrollment in a given course.

CHAPTER II

ANALYSIS OF THE PROBLEM

The first step in the analysis is to identify the objective of the study and the factors affecting it. In this problem the objective can be stated as:

1. Aiding the administrator in planning the faculty requirement.
2. Identification of the factors leading to wasteful utilization.
3. Sensitivity analysis of the factors affecting the model.

In most instances, the administrator would have developed a feel for the problem based on his past experience and built on approximate regression model for the faculty requirement, identifying certain common sense factors. He will be validating any new model in this perspective. Hence it is desirable to incorporate those features in the model which will aid him in the comparative study.

Teaching faculty requirement is comprised of the requirement for each individual course offered in each of the departments in the school. The work load induced by each course may be conceived of as being a function of the following course parameters.

1. Student count in the given course
2. Level of the course
3. Mode of instructions and the associated hours
4. Section size - Numbers of students in a given section
5. Speciality and importance of the course in relation with the other courses

Having agreed upon these course parameters it is a question of the subjective argument as to which of these factors is a major contributor to the work load of a teaching faculty. The following methods may be considered reasonable for calculating the full time equivalent teaching faculty for a given course.

1. A method based on the student count in a course, level of the course and the mode of instruction in conjunction with the associated hours.
2. A method which discards the mode of instruction and considers the work load as a function of the level of the course, and the student credit hours which is the product of the number of students and the credit hour associated with the course.
3. A method which considers the section size limitations and the work load as being constituted of two components. A constant work load which is dependent on the number of sections irrespective of the number of students in the section and a varying load induced by the student count in a particular section. Each of these components may be considered as varying with the level of the course and the mode of instruction, and may have to be weighted to denote the proportionate work load.

All these methods do not consider the speciality requirement of the course, since the speciality requirement will have erratic behavior at different times, and can usually be hand picked. A computer model which considers all of these features or capable of giving all relevant parameters for a comparative study seems to be most desirable. In these methods we had conceived of the possibility of varying the induced course load with different levels and modes of instruction. The weight given to these factors are

qualitative in nature depending on the judgement of the administrator and have to be quantified in the model. Since the judgement may vary with each administrator it is desirable to feed this as input variables to the model.

Since the method 2 does not take into consideration the mode of instruction, which is considered as a relevant factor in deciding the work load, it was not included in the present model, though student credit hours generated, are given for comparative purposes. The first method which is based upon only the number of students in a given course is formulated by D. P. Hoyt and has been widely used (2). The input parameters in the model for calculating the full time equivalent teaching faculty requirement by the last method were chosen to approximate the Hoyt model. The model also identifies the courses which do not fulfil the minimum course count requirements and calculates the deleted teaching faculty requirement for different classifications separately.

2.1 GENERAL METHODOLOGY

The general procedure employed in the model can best be described by the following flow chart, shown in Figure 2-1. The full time equivalent teaching faculty requirement is calculated by two methods. The input parameters under the jurisdiction of the administrator are marked with an asterik.

2.1.1 Method 1 (Hoyt's method)

$$FTE_{\text{school}} = \sum_{i=1}^{N_o} \sum_{j=1}^{N_i} \sum_{k=1}^4 \frac{K_j \times \omega_{jk} \times H_{jk}}{A}$$

2.1.2 Method 2

$$FTE_{\text{school}} = \sum_{i=1}^{N_o} \sum_{j=1}^{N_{ci}} \sum_{k=1}^3 S_j \times P_{jk} \times F_1 + \frac{K_j}{B_c} \times \omega_{jk} \times F_2 \times \frac{H_{jk}}{B}$$

+ problem load calculated by Method 1.

where

FTE_{school} = Full time equivalent teaching faculty requirement for the school.

N_o = Number of curricula comprising the school

N_{ci} = Number of courses in the i^{th} curriculum

k = Index representing the mode of instruction

$k=1$ represents lecture, $k=2$ represents recitation

$k=3$ represents laboratory, $k=4$ represents problems

ω_{jk} = Weighting factor for j^{th} course k^{th} mode, for calculation of teaching load due to student count.

H_{jk} = Number of contact hours for j^{th} course, k^{th} mode.

A = Base student credit hours constituting 1-FTE teaching load attributed to student count.

S_j = Number of sections for j^{th} course

P_{jk} = Constant load factor depending on number of sections for j^{th} course, with k^{th} mode instruction.

F_1 = Weighting factor representing the administrator's judgement of the proportion of load attributed to the number of sections irrespective of the section size.

F_2 = Weighting factor representing the administrator judgement of the proportion of load attributed to the student count in a given course ($F_1 + F_2 = 1.$)

- K_j = Student count in j^{th} course
- ** B_c = Base student count (non weighted) to represent full load attributed to course count.
- ** B = Base student credit hour to represent 1-FTE teaching faculty member.

The first method of FTE computation is a special case of the second method where

$$F_1 = 0 \quad F_2 = 1 \quad B_c \times B = A.$$

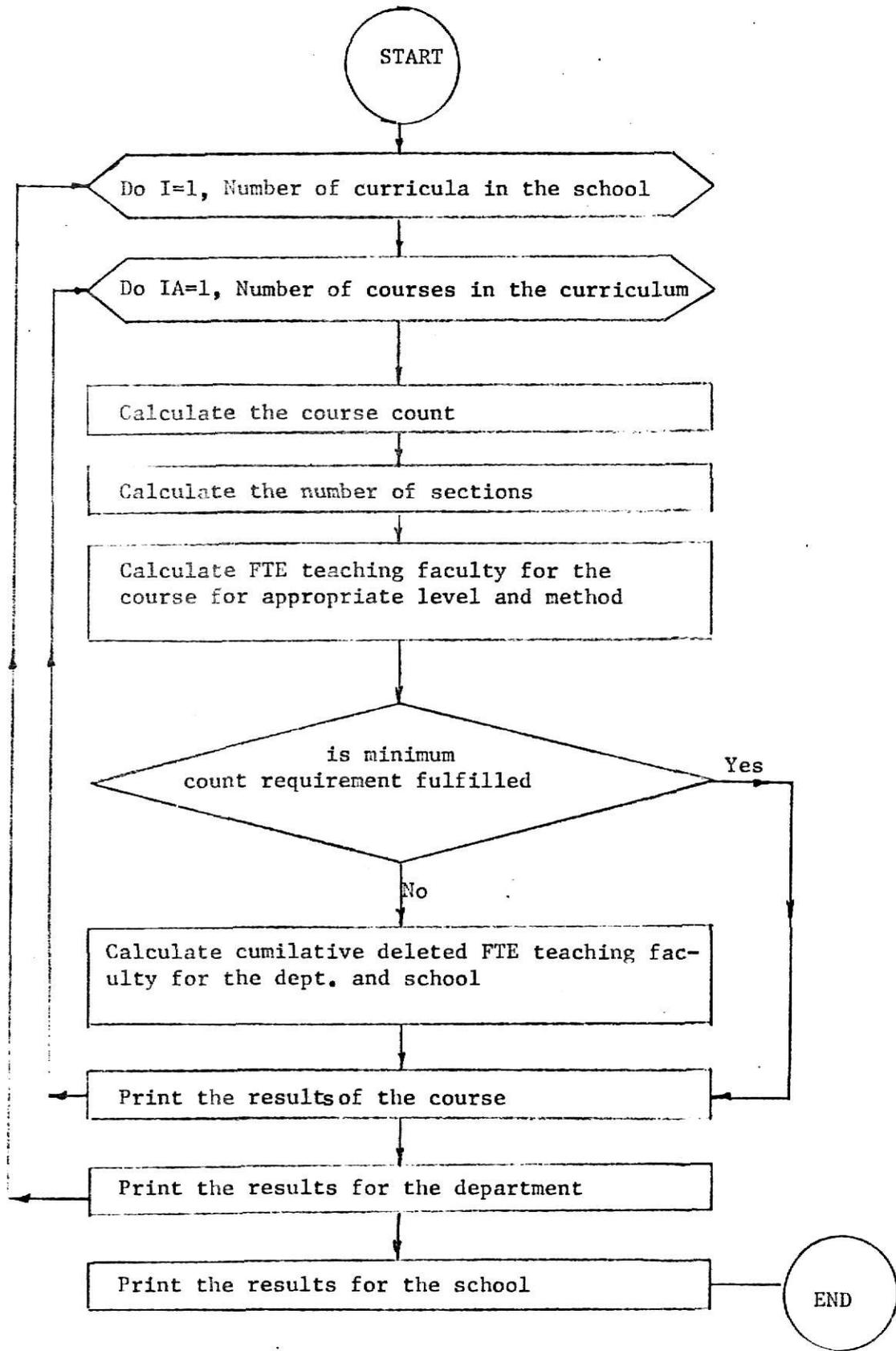


Figure 2.1 General flow chart

CHAPTER III

ESTIMATION OF COURSE PARAMETER

The four relevant course parameters used in the model were

1. Student count - Number of students enrolled in a given course.
2. Desired section size - Desired number of students to constitute one section.
3. Minimum section size - Minimum number of students required to constitute a section.
4. Number of sections

Of these parameters, the desired and minimum section size are under the jurisdiction of the administrative heads and are the input parameters in the model. The student count and the number of sections are calculated in the model.

3.1 ESTIMATION OF STUDENT COUNT

The student count is a function of the following factors

1. Curriculum specification.
2. The number of students enrolled in different curricula at different levels.
3. The relevance of the course content to other subjects, curricula, and schools.
4. Personal factors viz. compatibility of the teaching faculty, the work content etc.

The last two factors are qualitative and difficult to quantify in a regular regression model. Hence a model which includes factors based on

observation that reflect these qualitative tendencies, seems to be a reasonable answer.

A certain fraction of students from different curricula or schools constitute the student count in a particular course. We should identify the interrelationship between these fractions and the factors affecting the student count and specify these fractions so that they reflect these factors. This could be done by one or a combination of the following methods:

1. Estimation from curriculum specification.
2. Judgement of departmental heads and concerned faculty.
3. Historical data.

The first of these methods reflect the effect of curriculum specification and the other methods reflect the qualitative aspects of the problem and the electives.

This model attempts to find a method of logical estimation of these fractions, from here onwards specified as "Course curriculum fractions", based on curriculum specification and historical data. Judgement of department heads was not sought, since it was thought that historical data available will give the same information more accurately, though many times, especially when a new course is introduced, the departmental heads' judgement will be the only way. The historical data was available for the 1970 Fall and 1971 Spring semesters as part of the work in a related field, namely estimation of classroom requirement, by Mr. Tom Compton (1). At this stage it was uneconomical to get more data. A sample of these data is given in Appendix A.

The students enrolling in a particular course were considered as belonging to the following classifications:

1. Curriculum Specifying The Course As Requirement
2. Curriculum Specifying The Course As Elective
3. A School Comprising Group Of Curriculums, Subdivided Into The Following Classification

3 i Engineering school

3 ii Other schools

In the computer program the student count is calculated by summation of the product of course curriculum fraction and the corresponding enrollment figure in the curriculum. This required easy identification of curriculum enrollment. This was done by a two dimensional variable NC, the first index representing the curriculum number, the second index representing the course curriculum fraction level, and the value of the variable representing the enrollment figure. For this purpose the schools have to be classified in the same manner as curriculum for identification purposes. Hence the schools were given artificial identification numbers from 1000 onward since a 3 digit number specifies the regular curriculum numbers. They were

1000 - School of Agriculture

1001 - School of Architecture

1002 - School of Arts and Science

1003 - School of Business

1004 - College of Education

1005 - Engineering School

1006 - College of Home Economics

1007 - College of Veterinary Science

1008 - Portion of graduate school representing students enrolled for master program.

1009 - Portion of graduate school representing students enrolled for PhD program.

1010 - Portion of graduate school representing students enrolled for masters program in the Engineering curriculum.

1011 - Portion of graduate school representing students enrolled for Phd program in the Engineering curriculum.

Course curriculum fractions, except for the courses specified as curriculum requirement, was based on historical data. For the courses offered in both the semesters, the average of the fall and spring data was taken. The undergraduate students were divided into eight levels, each subsequent pair representing Freshmen, Sophomore, Junior and Senior level, the odd numbers representing the Fall semester and the even number representing the Spring semester. The graduate students were divided into two levels, the level 1 representing the Fall semester and the level 2 representing the Spring semester.

3.2 ESTIMATION OF COURSE CURRICULUM FRACTIONS FROM CURRICULUM SPECIFICATION

If the students were considered regular, adhering to the curriculum strictly, then a fraction equal to 1.0, assigned to the corresponding level would be most appropriate. But rarely does this regularity assumption hold true, and the deviation from this regularity is likely to be more marked in those cases where the course is offered in both the semesters. Examples to the effect are given in Appendix B. However it would be desirable to explore the feasibility of a method to represent the degree of irregularity. This could be represented by a fractional factor with which the result obtained by the regularity assumption has to be multiplied. This fractional factor could be specified for individual courses, for individual curricula, or for the whole school. Specifying the fraction for individual courses would have

to be based on historical data, which would result in discarding the curriculum specification. The amount of data available did not justify the fractional factor being specified for individual departments. Moreover this would mean that sensitivity analysis due to change in curriculum specification would be more elaborate. Hence this was done for the whole school. Any uniform pattern applied to the whole school is likely to create a certain amount of distortion in the estimate. Hence the following performance criteria were formulated:

1. To minimize absolute deviation from the mean.
2. To prefer overestimation to underestimation.
3. To determine the fraction to the accuracy of one decimal place.

Since the irregularity pattern is likely to be related to the number of times the course is taught in a given period, the courses were divided into the following three categories.

1. Category 1. Courses are offered in one semester only.
2. Category 2. The courses are offered in both the semesters and the curriculum specifies it as a requirement only in one semester.
3. Category 3. Courses are offered in both the semesters and the curriculum specifies it as requirement in both semesters. This category of courses are usually lecture courses viz. engineering assembly.

Appendix 'C' gives the estimated student count and the deviation from the actual student count for individual courses. The summarized results are given in Tables 1, 2 and 3. Based on the performance criteria a fraction of 0.5 was chosen for Category 1 courses and 0.8 for Category 2 courses.

Table 1

Summary of Deviation of Estimated Student Count From Actual Student Count

For Category 1 Courses

Fraction	Fall Semester		Spring Semester		Both semesters	
	Mean Deviation	Mean abs. Deviation	Mean Deviation	Mean abs. Deviation	Mean Deviation	Mean Deviation
0.7	0.2857	3.7571	0.7222	4.2778	+2.2813	4.0938
0.8	-2.2143	4.2143	-1.333	4.3333	-1.7188	4.2813
0.9	-4.3572	5.2143	-3.3889	5.0556	-3.8125	5.1250

Table 2

Summary of Deviation of Estimated Student Count From Actual Student Count

For Category 2 Courses

0.4	4.625	9.7917	3.3125	8.6250	3.8750	9.125
0.5	-6.5417	12.8750	-1.8475	8.4063	-3.8714	10.3214
0.6	-13.500	17.1667	-7.0625	10.6250	-9.8214	13.4286

Table 3

Summary of Deviation of Estimated Student Count From Actual Student Count

For Category 3 Courses

0.7	15.0000	15.0000	18.6667	18.6667	16.8333	16.8333
0.8	5.8333	5.8333	7.5000	7.5000	6.6667	6.6667
0.9	-3.6667	5.3333	-3.8333	4.0000	-3.7500	9.3333

Category 3 courses are usually held in one section irrespective of the student count and hence FTE faculty requirements are not affected appreciably due to the sectioning rule. Hoyt's method correlates the student count and FTE teaching faculty requirement linearly and to partially offset the effect due to overestimation a fraction of 0.8 was chosen, though 0.9 was more appropriate from the point of view of the chosen performance criteria.

3.3 DECISION RULE FOR SECTION SIZE

Ideally, the number of students in a class is bound by two limits, the upper limit and the lower limit. The lower limit is the minimum student count in a section, representing the tolerance limit to wasteful utilization. The upper limit is limited by any of the following factors:

1. Teaching effectiveness
2. Classroom size
3. Speciality requirement of the course
4. Course demand and faculty availability
5. Equipment available in the case of laboratory classes.

The department head usually uses his judgement in deciding the section size depending on personnel resources, conceived importance of the course, course demand and load balancing of faculty. Hence there are likely to be some exceptions to the uniform sectioning rule, whatever the logical rules for decision may be. But the uniform sectioning rule aids the administrator in overall policy decision and acts as a base criterion to examine the exceptional cases.

These exceptions are not likely to affect the final estimate of faculty requirements appreciably, as long as they remain as exceptions and not a rule.

The following decision rules may be considered logical:

- 3.3.1 Divide the number of students by the maximum section size and if there is a remainder increase the number of sections by 1.
- 3.3.2 Divide the student count by desired section size and examine the remainder if any, with a percentage of desired section size to decide whether any additional section should be created or whether the students should be redistributed in the existing number of sections.
- 3.3.3 Divide the student count by the desired section size and examine the remainder, if any, to find the number of students who will be redistributed into the existing section. If this number is greater than a specified level, then increase the number of sections by 1.

Though no method can be considered as overwhelmingly superior to others and any one method may be considered reasonable depending on the view of the administrator, it may be desirable to examine the various methods. The method 3.3.1 is rigid in the sense that it bases the decision on maximum section size and irrespective of the remaining number of students increases the section size by 1. This basically leads to the assumption that the factors influencing the decision of maximum section size are not flexible. The method 3.3.2 and 3.3.3 are similar to each other except for the specified level of students with which the remaining students are examined. In method 3.3.3 it is specified as a fixed percentage of desired section size.

The method 3.3.3 allows a certain amount of flexibility to the administrator in deciding these parameters, based on the practical situation. However it should be pointed out this method in practice will coincide with

the method 3.3.1 when the maximum section size is specified as the sum of the desired section size and the specified level with which the remainder is examined. The advantage of method 3.3.3 is in the philosophy of deciding these section sizes. If method 3.3.3 is chosen then the question arises as to what the specified comparison level should be. Again there is no one answer to such questions, but minimum section size may be considered as reasonable from the following view points:

1. Most of the decision arises as to whether the number of sections should be 1 or 2. In such cases it would not be wise to create a section less than the minimum section size.
2. This could be easily conceived by the administrator and is no longer an arbitrary level.

Hence the method 3.3.3 with minimum section size as the specified level was chosen.

CHAPTER IV

COMPUTATIONAL ALGORITHM

4.1 PROGRAM COMPONENTS

The model is comprised of

- a. One main program
- b. Five control subroutines - CALCUL, LECTUR, RECIT, LAB, AND PROB
- c. Seven functional subroutines - COUKNT, COUDET, TEADET, MINSIZ, FINRES, and INIATL

The control subroutines classify the courses according to different course levels and mode of instruction, and route them to appropriate functional subroutines for calculation. The detailed functions of these subroutines are given in Appendix D.

4.2 INPUT PARAMETERS

The main input variables that reflect the administrators judgement are:

1. Desired section size for various levels and modes of instruction.
2. Minimum section size for various levels and modes of instruction.
3. Constant and varying load weighing factors for different course levels and mode of instruction.
4. Curriculum enrollment figures.
5. Course curriculum details.

The administrator can study the effect due to changes in any of these parameters. Besides these variables there are other input variables for the identification of the semester, method of calculation and initiating the program. The details of all the parameters are given in Appendix E-1. The data and types are given in E-4.

4.3 OUTPUT VARIABLES

The main output variables are FTE teaching faculty requirements of each course, different curricula and the school, classified both by the mode of instruction and the course level. In addition to this, deleted FTE teaching faculty requirement due to minimum section size requirements, as well as generated credit hours are given for various classifications. The details are given in Appendix E-2.

4.4 GENERAL COMPUTATION PROCEDURE

The first step in the computation is to estimate the student count from enrollment figures and course curriculum fractions. The courses are then grouped as to their mode of instruction and level. The projected number of sections for each course is calculated from this estimated student count and the desired and minimum section sizes. The full time equivalent teaching faculty requirement is calculated by either method using the appropriate weighting factors. The estimated student count is compared with minimum section size for the subject modes of instruction. If this requirement is not met for at least for one mode of instruction, then the full time equivalent teaching faculty requirement for that course is not included for curricula and the school, but is cumulated separately. The courses which do not satisfy the minimum section size requirements are also identified with asteriks, in the output. This procedure is repeated for different courses in the curriculum and for all the curricula comprising the school. A general flow chart is given in Appendix F.

4.5 USE OF THE COMPUTER PROGRAM

As stated earlier the model should aid the administrator in planning the

faculty requirement, help him in recognizing wasteful utilization and optimize the utilization of the existing teaching faculty. The five input parameters under the control of the administrator help him in these objectives. The model is dependent on the estimation procedure for student count, and the input parameter which plays a vital role in this estimation is course curriculum details, and in particular course curriculum fractions. These are determined either by curriculum specification or historical data. Of these, course curriculum fractions determined by curriculum specification are of special importance, since not much could be done to the fractions based on historical data. An administrator might have recognized some of the courses as being wasteful in the sense that they have a low student count, though they may satisfy the minimum section size requirement. This may be due to either of the following reasons:

1. The courses are service courses and similar courses are offered in different curricula.
2. The courses are offered in both the semesters.

To overcome this low student count, an administrator will introduce curriculum changes and would like to study the effect on the teaching faculty requirement. These changes will involve grouping of similar courses taught in different curricula into one single course, and/or dropping a course from one of the semesters. In the case where the courses are offered only in one semester instead of two semesters, the change involves alteration of the modification factor applied to course curriculum fraction, for the appropriate curriculum. But the change which involves replacement of a group of similar service courses by a single course requires a special mention. The best procedure would be to study the behavioral pattern of the student enrollment

for these courses. The new course introduced would have the curriculum fractions, which will be a combination of the new curriculum requirements introduced and combined data available from the group of courses.

Similar procedures could be adapted for the introduction of a new course, which corresponds to an existing course. But in those cases where the courses introduced are completely new, then the course curriculum fractions must be based on curriculum requirement and judgement of the department heads.

The use of other input parameters are apparent and their data card type are given in Appendix E-4.

CHAPTER V

SENSITIVITY ANALYSIS

In the procedure for estimation of FTE teaching faculty requirement, many factors have been introduced to quantify the assessment of the qualitative factors by the administrator. Each administrator may consider an interval range of these parameters instead of a point estimate and this again may vary with the different administrators. Hence it would be necessary to consider the effect of changes with parameters on the FTE teaching faculty requirement. Since the 'Method 2' incorporates the 'Method 1' as a limiting case, only 'Method 2' will be considered for analysis. The FTE teaching faculty requirement for the j^{th} course can be written as,

$$\text{FTE}_j = (\text{FTE}_1 + \text{FTE}_2) \frac{H_{jk}}{A}, \text{ where}$$

$$\text{FTE}_1 = S_j \times P_{jk} \times F_1, \text{ representing a constant load due to preparation,}$$

and

$$\text{FTE}_2 = \frac{K_j}{B_c} \times \omega_{jk} (1-F_1), \text{ representing the varying load due to student count}$$

$$S_j = \text{Number of sections for the } j^{\text{th}} \text{ course}$$

$$P_{jk} = \text{preparation load factor for the } j^{\text{th}} \text{ course, and } k^{\text{th}} \text{ mode of instruction}$$

$$F_1 = \text{A fraction between 0 and 1 representing the proportion of the load attributed to preparation for the course.}$$

$$K_j = \text{Student count in the } j^{\text{th}} \text{ course}$$

$$B_c = \text{Base student count representing one full load due to student count.}$$

$$\omega_{jk} = \text{Varying load weighing factor for the } j^{\text{th}} \text{ course, and } k^{\text{th}} \text{ mode of instruction.}$$

H_{jk} = Number of teaching hours in the k^{th} mode of instruction for the j^{th} course.

A = Base credit hours to represent one FTE teaching faculty load.

The factor H_{jk}/A represents the ratio of the number of hours the course is taught and the base credit hour constituting 1 FTE faculty load. To simplify the analysis H_{jk} is taken as equal to A. The effect of other factors on FTE_1 , FTE_2 and FTE_j is analysed.

5.1 EFFECTS OF INPUT PARAMETER ON FTE_1

S_{jk} is a function of student count ' K_j ' and the sum of desired and minimum section size ' R ' represented by the following relationship.

$$S_{jk} = \text{Quotient of } \frac{K_j}{R} + K, \text{ where } \begin{array}{l} K=0 \text{ if } K_j \text{ MOD } R=0 \\ K=1 \text{ if } K_j \text{ MOD } R \neq 0 \end{array}$$

Hence FTE_1 is a right continuous step wise function as shown in Figure 5.1. The size of the step width is equal to ' R ' and the step height is equal to $P_{jk} \times F_1$. Hence the desired and minimum section size, in particular their sum, define the step width and the product $P_{jk} \times F_1$ affect the step height. Of the two factors, P_{jk} and F_1 , P_{jk} is an independent factor not having any effect on the other component FTE_2 . Hence for a fixed P_{jk} , the step height increases in direct proportion with F_1 .

5.2 EFFECT OF INPUT PARAMETERS ON ' FTE_2 '

FTE_2 is independent of the number of sections and hence is not affected by desired and minimum section sizes. It can be represented by a straight line with slope $\frac{\omega_{jk}}{B_c} (1-F_1)$, as shown in Figure 5.2. Here ω_{jk} , B_c and $(1-F_1)$ are all positive and hence the slope is always positive. Further since FTE_2 equals zero where student count is equal to zero, it passes through the

origin. $1-F_1$ varies between 0 and 1 and hence the slope varies between 0 and $\frac{\omega_{jk}}{B_c}$. The slope decreases with the increase in student count and the effect of ω_{jk} is to modify this student count for the j^{th} course and k^{th} mode of instruction. Hence $\frac{\omega_{jk}}{B_c}$ can be designated by a single factor $1/M_c$, where M_c is the modified base student count. $(1-F_1)$ represents the proportion of the load attributed to the student count and increase in the proportion increases the slope.

5.3 CUMULATIVE EFFECT OF INPUT PARAMETERS

Since FTE_j is the sum of FTE_1 and FTE_2 it can be represented by a combination of linear and stepwise function, the slope of the linear portion being $1/M_c (1-F_1)$, step width equal to 'R' and step height equal to $P_{jk} \times F_1$. This is illustrated in Figure 5.3. The limiting cases of this graph are, purely stepwise function with no linear increase, when F_1 equals 1 and a purely linear function without any step when ' F_1 ' equals zero. The second of these cases is the Hoyt's line for estimation of teaching faculty requirement. Figures 5.4 to 5.11 represent the behavior of the fulltime equivalent teaching faculty requirement for the data specified in Table 5.3.1.

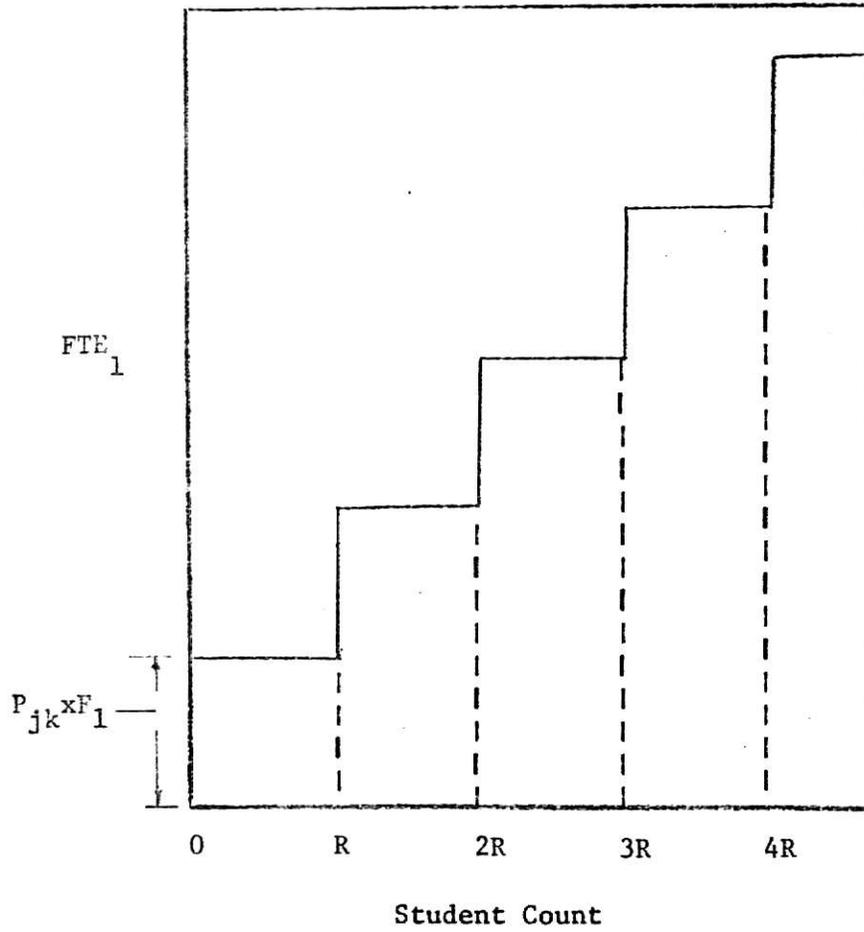


Figure 5.1. FTE_1 vs. Student Count

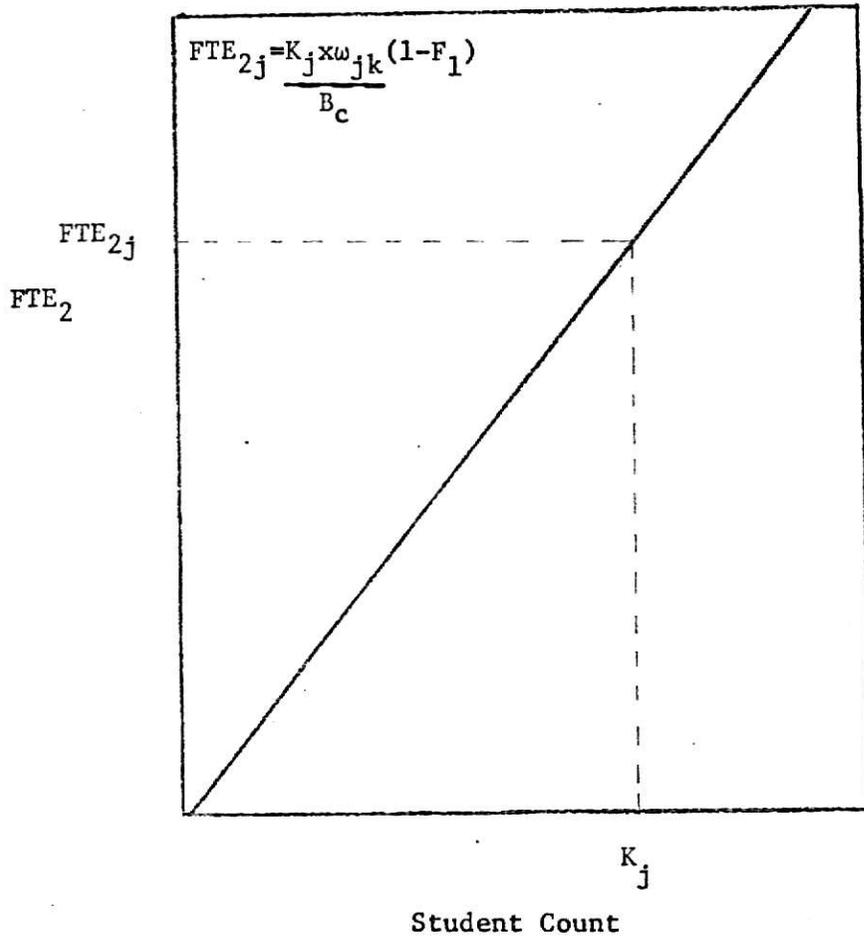


Figure 5.2. FTE₂ vs. Student Count

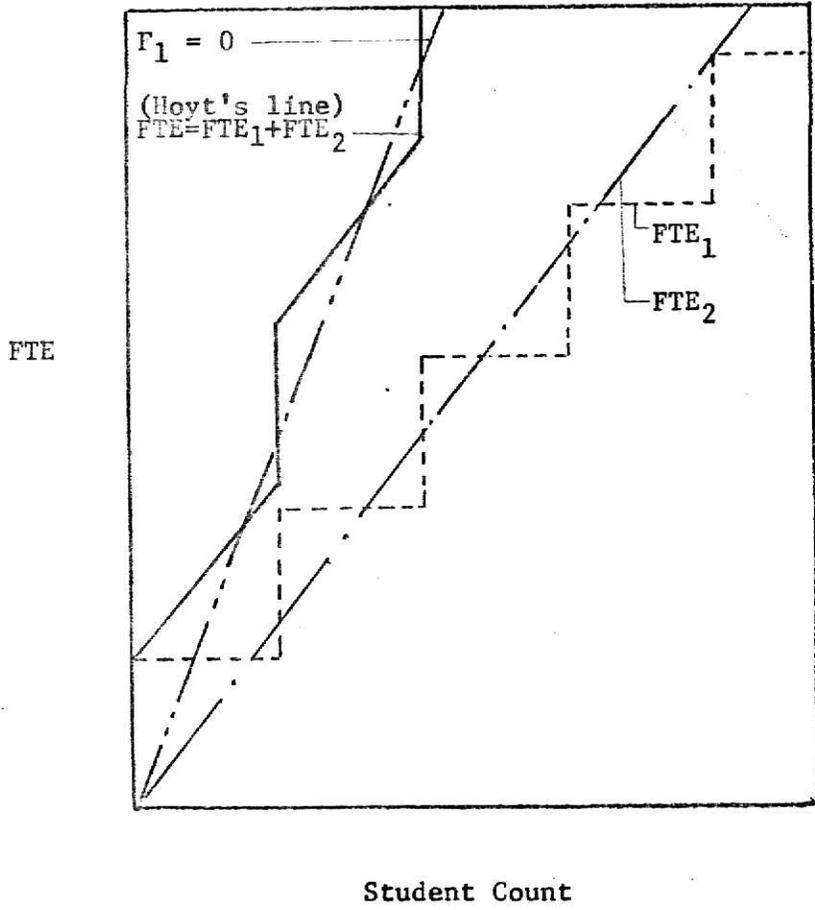


Figure 5.3. FTE vs. Student Count

Table 5.3.1 Input parameter for the sample problem

	<u>Desired Section Size</u>				<u>Minimum Section Size</u>			
	000-399	400-599	600-799	>800	000-399	400-599	600-799	>800
Lecture	1000	1000	1000	1000	20	15	10	3
Recitation	40	35	30	30	20	15	10	3
Labaratory	20	15	12	10	10	8	7	5

	<u>Weighting Factor</u>				<u>Preperation Load Factor</u>			
	000-399	400-599	600-799	>800	000-399	400-599	600-799	>800
Lecture	1.00	1.50	2.00	4.00	1.00	1.10	1.30	1.50
Recitation	1.00	1.50	2.00	4.00	1.00	1.10	1.30	1.50
Labaratory	2.00	3.00	4.00	4.00	1.00	1.10	1.30	1.50
Problems	3.00	4.50	6.00	12.00				

Base Student Count (B_c) = 32

Fractional Load For Preperation (F_1) = 0.0, 0.5, 1.0

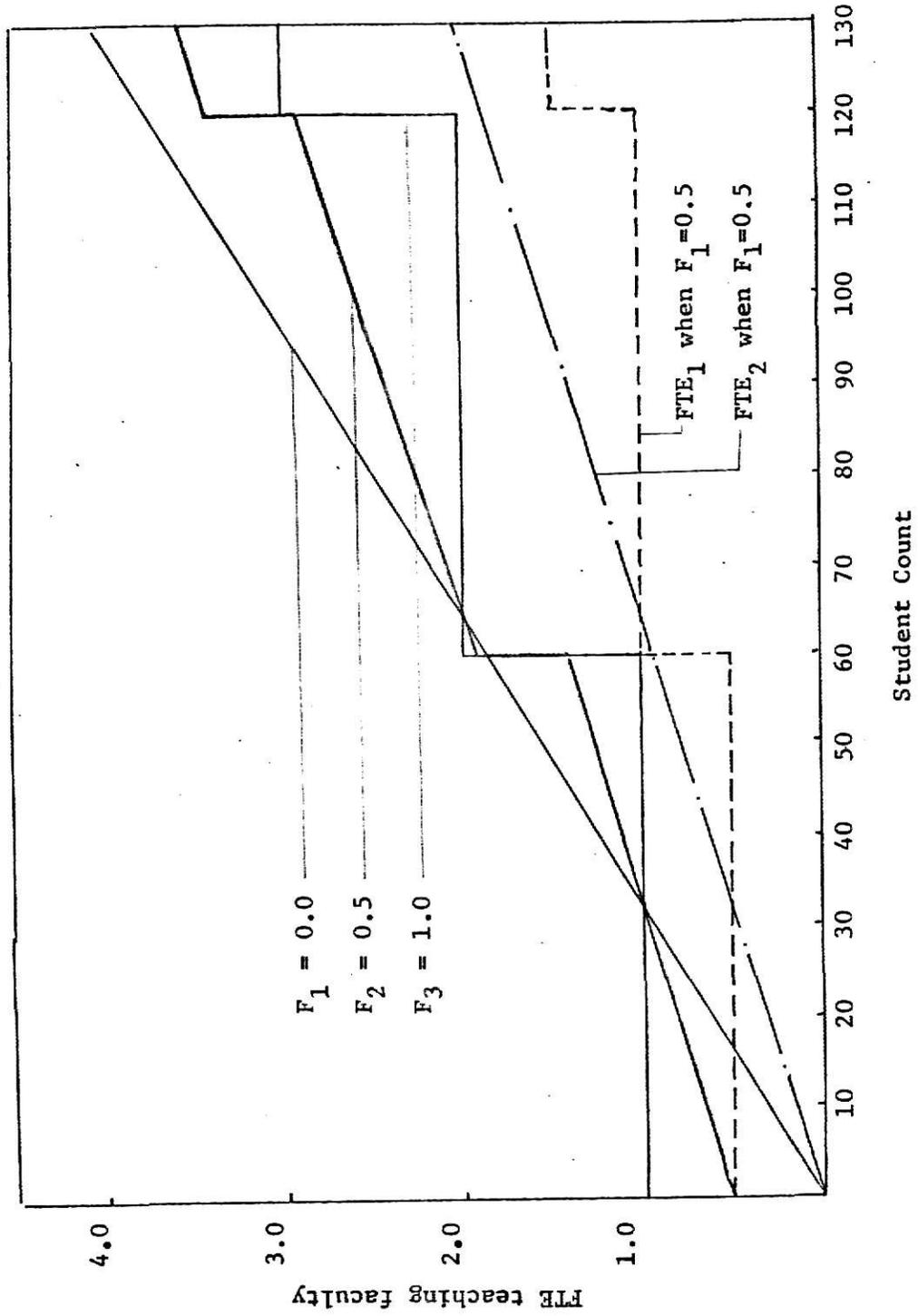


Figure 5.4. FTE teaching faculty requirement for the courses taught in recitation classes; course level 000-399

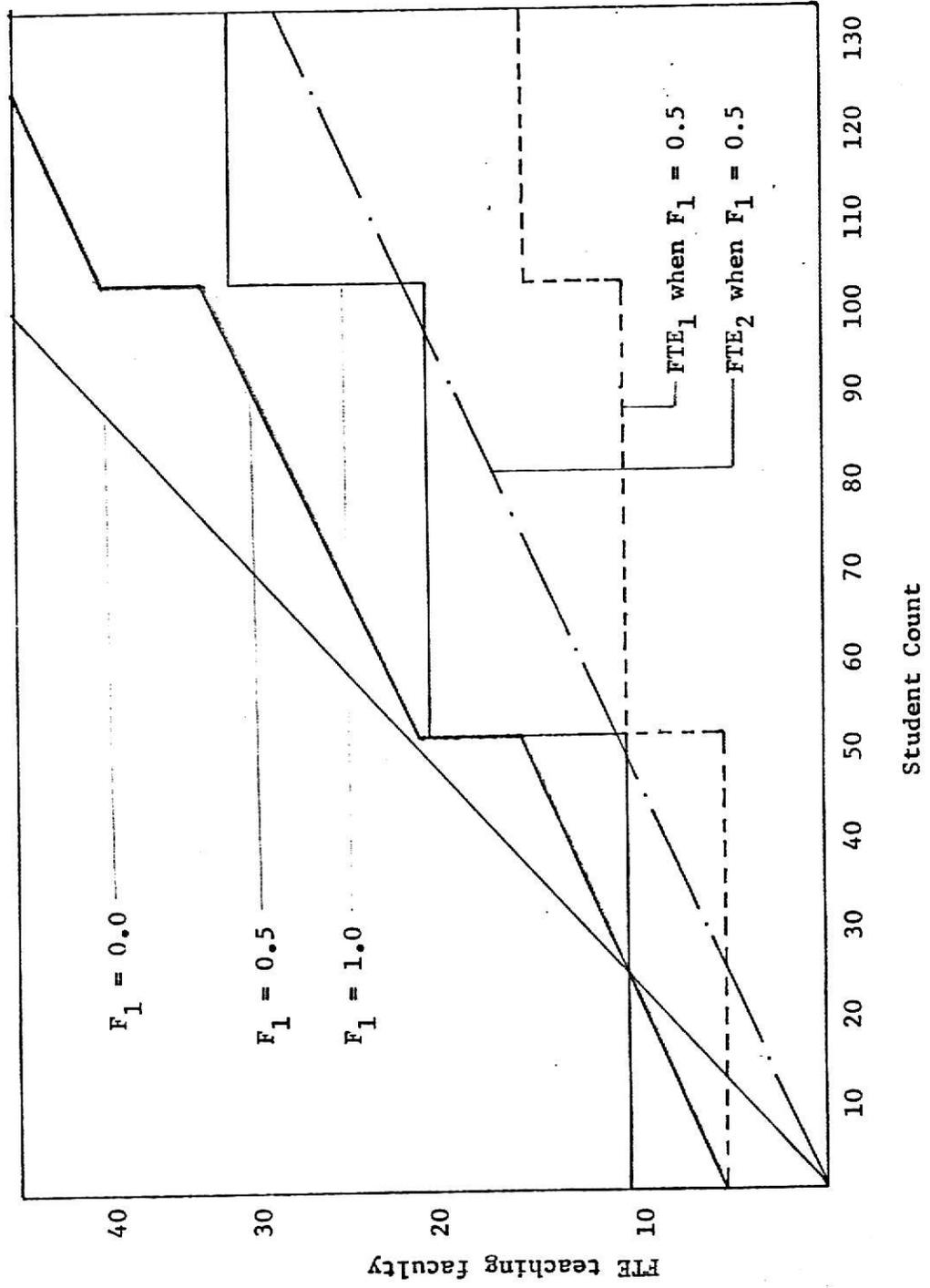


Figure 5.5. FTE teaching faculty requirement for the courses taught in recitation; course level - 400-599

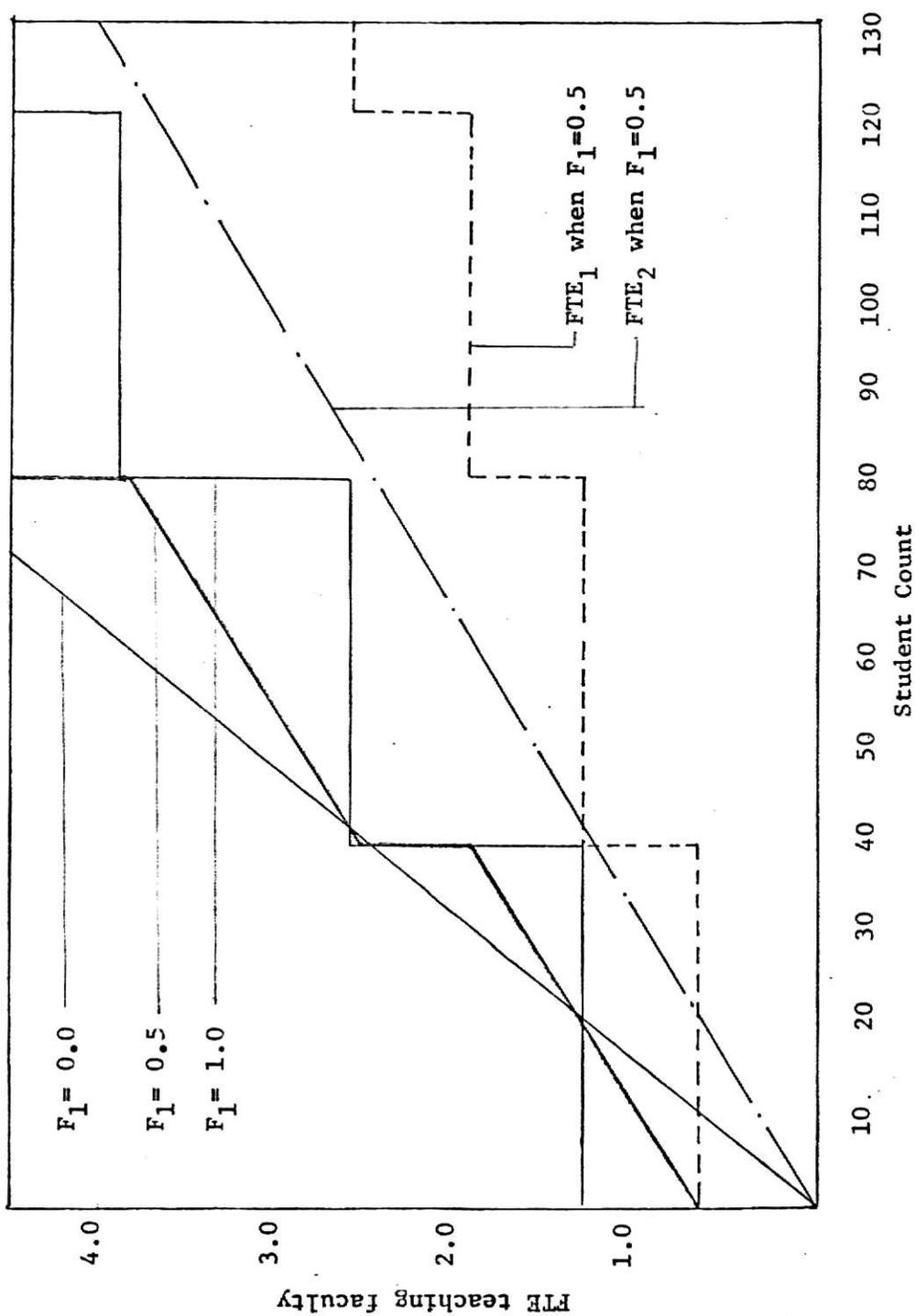


Figure 5.6. FTE teaching faculty requirement for the courses taught in recitation classes; course level - 600-799

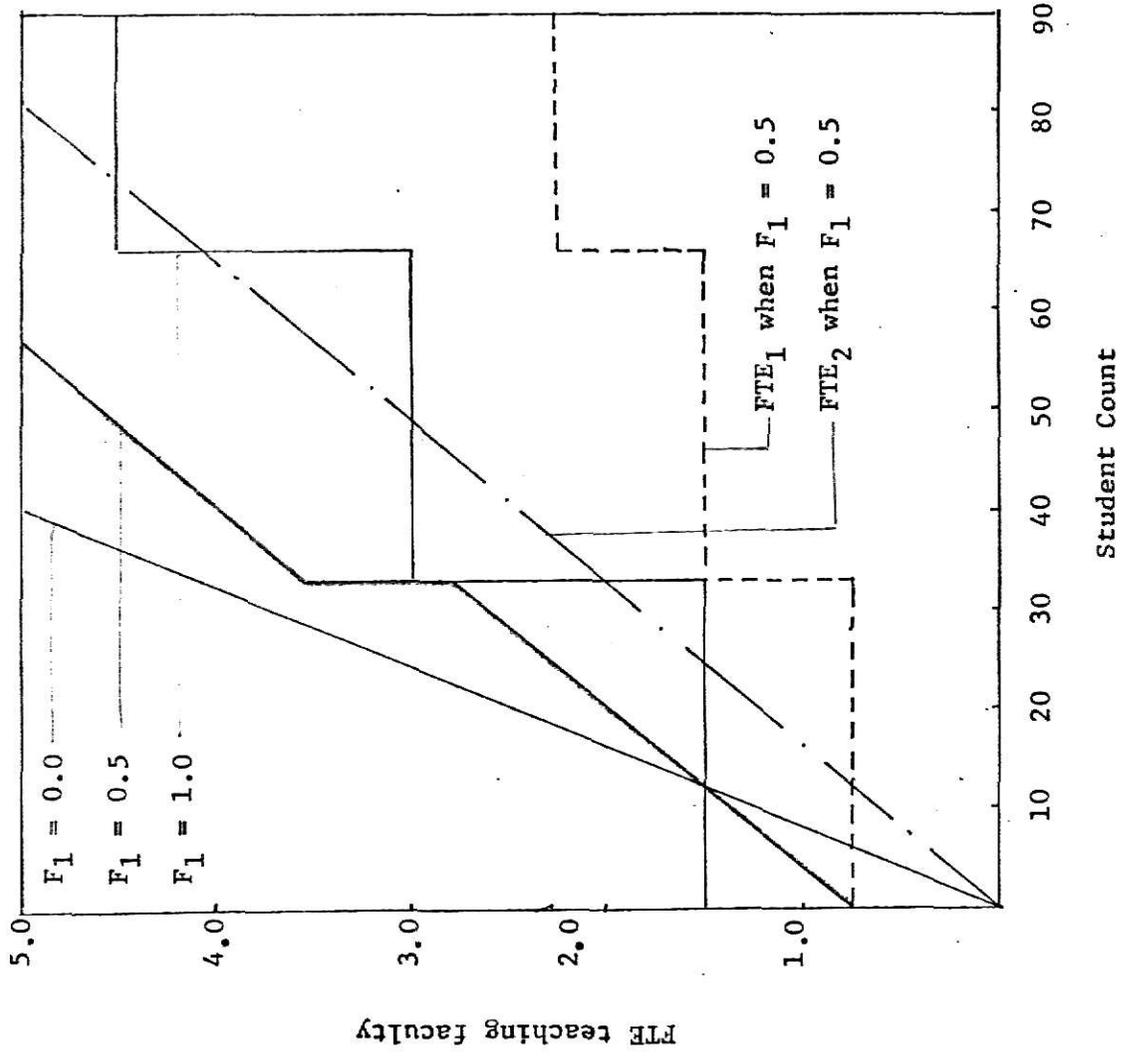


Figure 5.7. FTE teaching faculty requirement for the courses taught in recitation classes course level >800

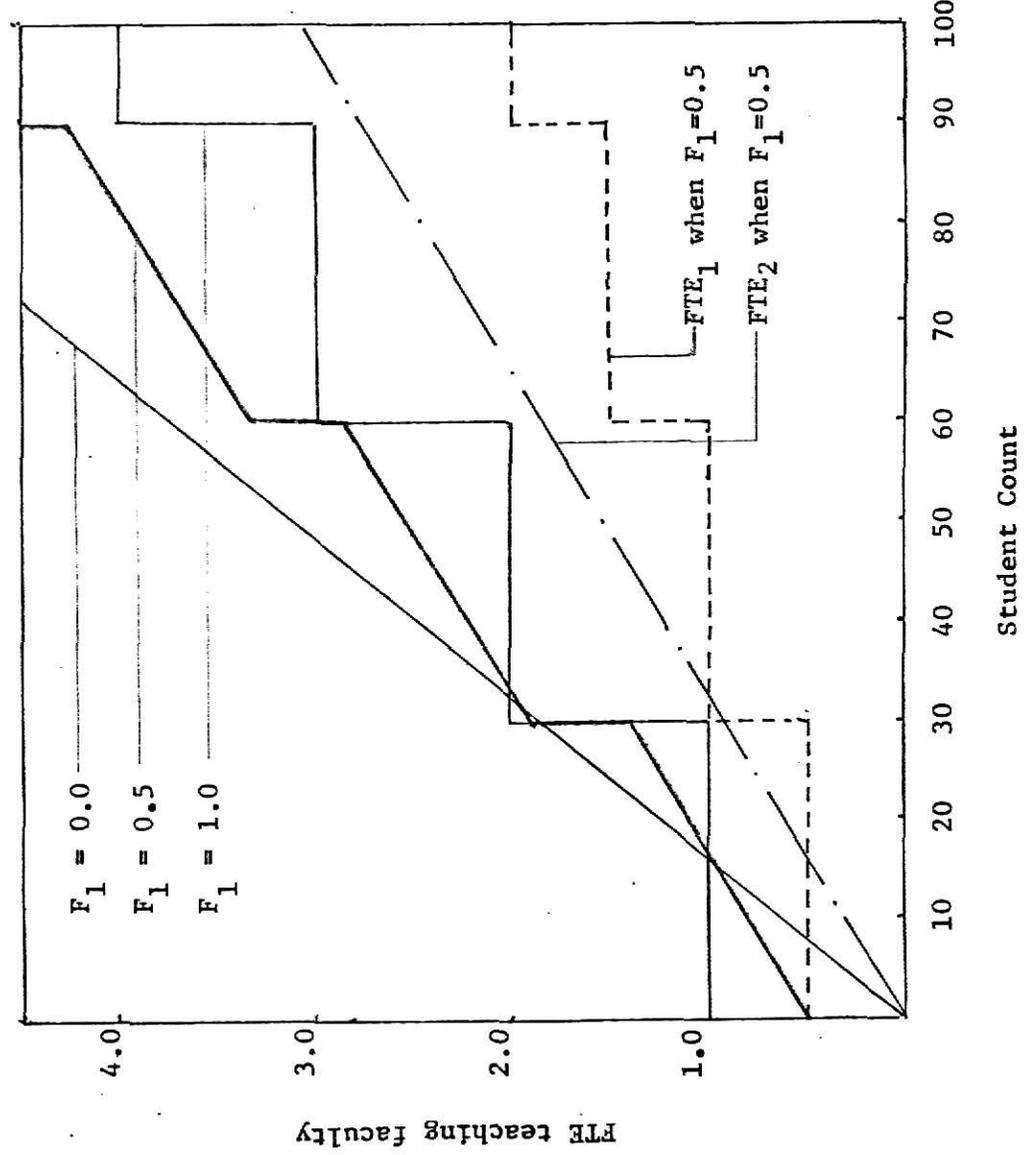


Figure 5.8. FTE teaching faculty requirement for the courses taught in laboratory classes; course level 000-399

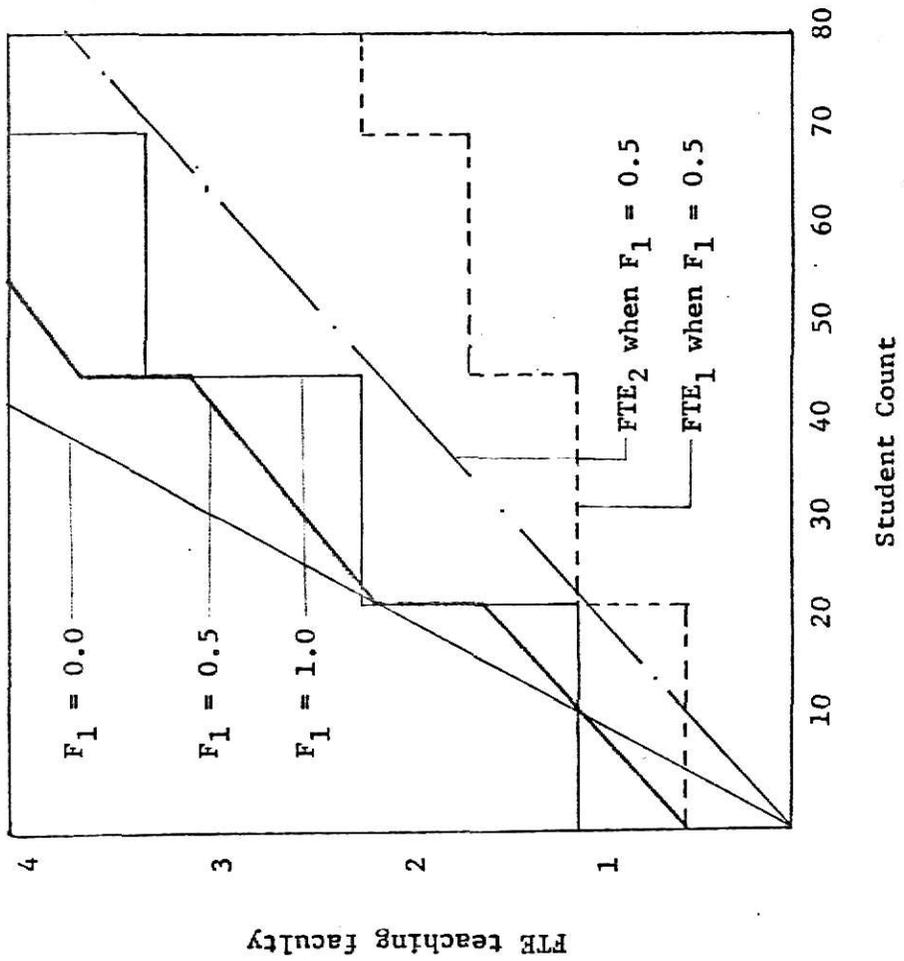


Figure 5.9. FTE teaching faculty requirement for the courses taught in laboratory classes; course level - 400-599

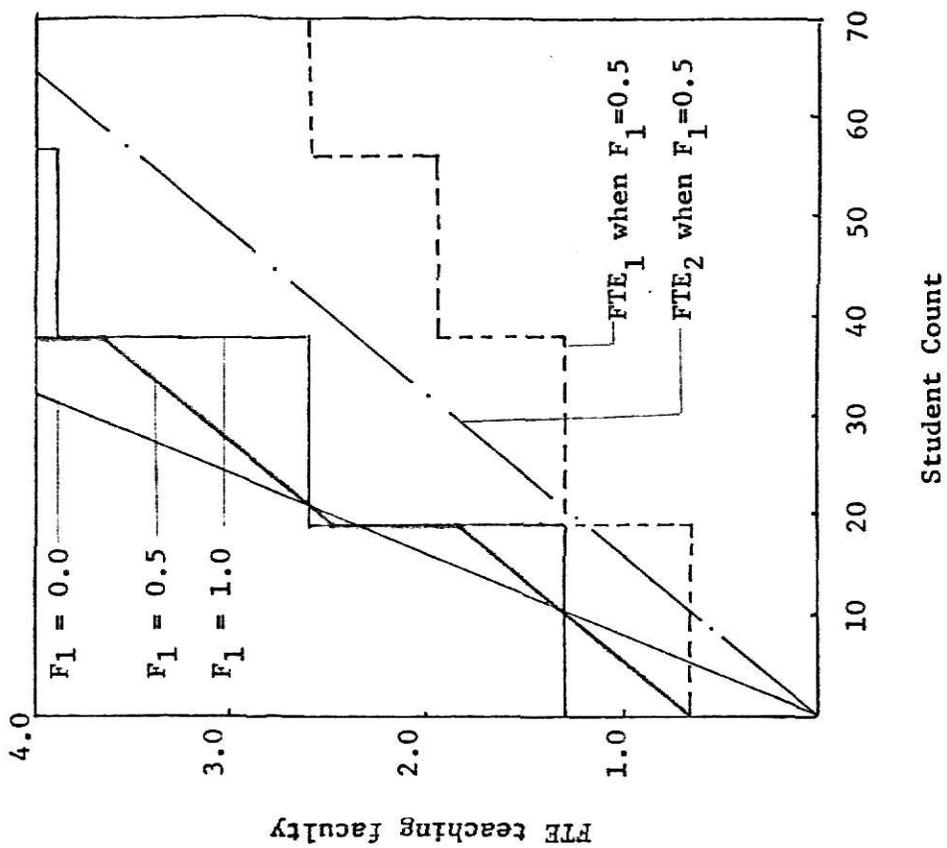


Figure 5.10. FTE teaching faculty requirement for the courses taught in laboratory classes; course level 600-799

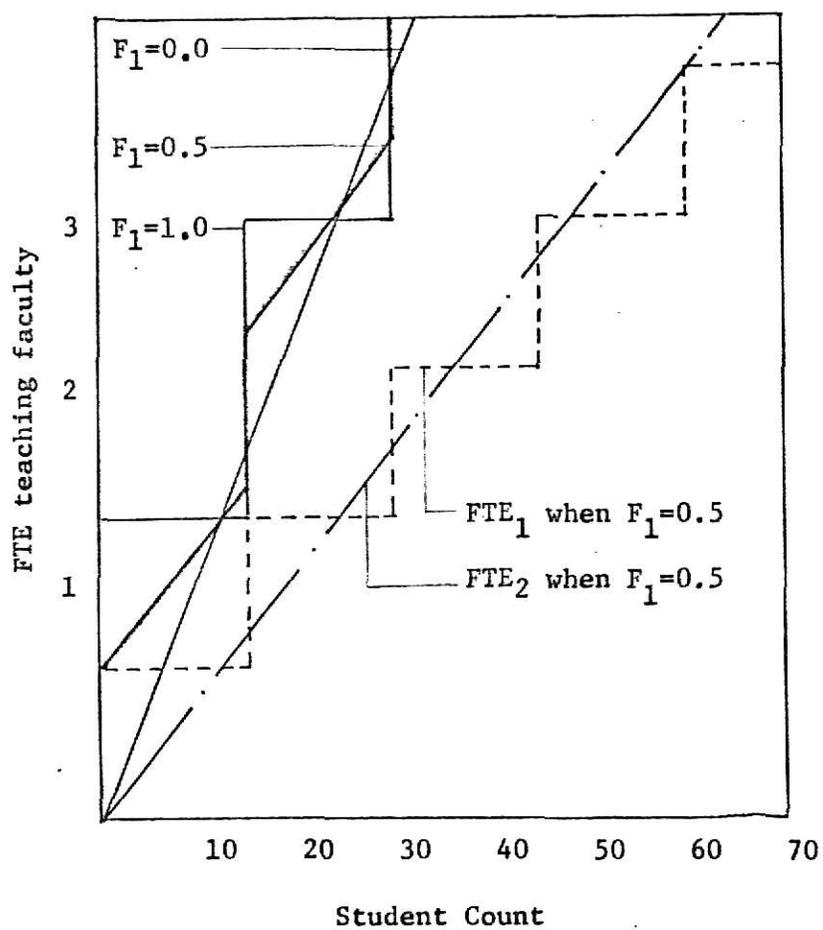


Figure 5.11. FTE teaching faculty requirement for the courses taught in laboratory classes; course level - >800

CHAPTER VI

RESULTS AND DISCUSSION

6.1 RESULTS

In a study of this nature one can not draw conclusive results, one can only evaluate the procedure at a few known reference points, mainly current practice. This has been done. Student course preference was measured from historical data and forecasts were made using a current definition of a full time work load; these projections (see Appendix G-1) agree reasonably well with the current levels. Various other explorations are possible by altering the definitions and weights. Other investigations are possible; for example the effect of curriculum revisions are easy to determine. It would be possible to specify the various curricula in such a manner as to minimize faculty requirements. Modest curriculum modification could be made to eliminate the least productive courses as identified by this program.

Results obtained in this manner would be a means of forecasting future loads. Some studies of this nature were conducted. Validation of these forecasts must wait for some future comparisons. The factor that will play a major role in the accuracy of these estimates is the choice of course curriculum fractions based on student irregularity.

6.2 DISCUSSION

This program is highly sensitive to the estimation of student count and one must keep the curriculum course factors updated for this method to be used with any reasonable accuracy. At present the cost of obtaining this

data is prohibitive and hence it would be desirable to incorporate a procedural device for updating this data. This could be done by maintaining a control chart for each of these courses and whenever there is an indication of the student count in a given course going outside the limits of control chart, it would be worthwhile to look into the details, recognize the attributes, and update the control chart.

The program gives the result only for one semester, and only one method could be used at a time. It would be useful to modify the program so that results of both the semesters could be carried out by the methods in the same run. Finally the possibility of reducing the program size, by including the weight factors and section sizes as arrays, is also worth investigating.

APPENDIXES

APPENDIX A

EXAMPLES TO SHOW THE DEVIATING ESTIMATED STUDENT COUNT FROM

ACTUAL STUDENT COUNT, WITH REGULARITY ASSUMPTION

Course No.	Actual Student Count	Student Count With Regularity Assumption	Student Count With Appropriate Modification Factors*
505 436	7	16	13
520 415	24	30	24
520 432	11	21	17
525 443	11	56	28
525 331	24	43	23
530 404	58	75	40
530 417	58	89	45
530 431	48	71	36
530 432	34	88	44
530 497	51	72	37
550 371	11	23	16
550 372	81	121	37
550 511	22	24	19
550 522	17	24	19
550 575	14	26	14
560 212	95	145	112
560 413	106	229	136
560 452	38	93	47
560 521	43	100	54
560 535	34	95	33
560 560	39	104	58
580 410	30	90	46
580 670	19	29	24
580 680	17	29	24

*Choice of modification factor is explained in Chapter 3.

APPENDIX B

EXAMPLE OF HISTORICAL DATA

Dept.	Course	Count	Frac.	Cur.	Lev.	Count	Frac.	Cur.	Lev.
1-E	550451	1	.0222	MTH 221	2	1	.0166	CS 223	1
		1	.0212	IE 550	2	8	.1481	IE 877	1
		***Course Total =				16			
	550481	1	.0666	BM 058	2	1	.0357	IE 550	1
		***Course Total =				30			
	550571	1	.0833	ARS 120	1	2	.0444	MTH 221	2
		1	.0526	GRS 818	1	1	.2500	MTH 838	1
		***Course Total =				33			
	550572	20	.4255	IE 550	2	2	.0645	CE 868	1
		***Course Total =				22			
	550575	7	.1489	IE 550	2	1	.0454	COS 805	1
		***Course Total =				23			
	550581	21	.4468	IE 550	2				
		***Course Total =				21			
	550603	11	.3928	ARI 150	2				
		***Course Total =				11			
	550625	15	.3191	IE 550	2				
		***Course Total =				15			
	550701	1	.0212	IE 550	2	2	.0370	IE 877	1
		***Course Total =				3			
	550711	1	.0526	GRS 818	1	18	.3333	IE 877	1
		***Course Total =				22			

Count = Student count

Frac = Fraction of the students from the curriculum specified who have enrolled in the course.

Cur = Curriculum from which the student enrolled in the course

Level = Students enrollment level

level = 1 Freshmen, Sophomore

level = 2 Junior and Senior

APPENDIX C

MODIFICATION FACTOR FOR COURSE CURRICULUM FRACTIONS

This appendix gives the basis for modification factor to be applied to the course curriculum fractions of the curricula which specify the course as requirement, to take care of irregularities. The estimated student count can be represented by

$$N = N_1 + N_2$$

where

N = Student count

N_1 = Student count resulting from curricula which specify the course as requirement

N_2 = Student count resulting from electives and additional enrollment from other curricula and schools

N_2 is derived from historical data

$$N_1 = \sum_{i=1}^{\text{NOC}} \sum_{j=1}^8 C_{f_{ij}} \times S_{ij}, \text{ where}$$

$C_{f_{ij}}$ = Course curriculum fraction specified for i^{th} curriculum, j^{th} level

S_{ij} = Student enrollment in i^{th} curricula for the j^{th} level

NOC = Number of curricula which specify the course as requirement

In the following tables N_2 is kept unaltered and modification factor is applied $C_{f_{ij}}$ and the resulting student count estimates are compared with the actual student count, for the three categories of courses. The level for which the curricula specify the course as requirement is enclosed within the parantheses.

C.1 ESTIMATED AND ACTUAL STUDENT COUNT FOR CATEGORY1 COURSES - COURSES

OFFERED ONLY IN ONE SEMESTER SPECIFIED AS REQUIREMENT IN THE CURRICULUM

C.1.1 Fall Semester

Course Number	Cur.	Actual Student Count	Student Count With modifying factors			Deviation of estimated student count from actual student count for modified factor		
			0.7	0.8	0.9	0.7	0.8	0.9
505 375	505(3)	11	14	14	15	-3	-3	-4
505 436	505(7)	7	11	13	15	-4	-6	-8
520 415	520(5)	24	21	24	27	3	0	-3
520 432	520(7)	11	15	17	19	-4	-6	-8
520 520	520(7)	13	15	17	19	-2	-4	-6
520 530	520(7)	12	15	17	19	-3	-5	-7
550 502	550(7)	23	17	19	21	6	4	2
550 511	550(7)	22	17	19	21	5	3	1
550 522	550(7)	17	17	19	21	0	-2	-4
550 552	550(7)	12	17	19	21	-5	-7	-9
580 110	580(1)	27	20	22	24	7	5	3
580 611	580(5)	19	15	17	19	4	2	0
580 670	580(7)	19	22	24	27	-3	-6	-8
580 680	580(7)	17	22	24	27	-5	-7	-10
Total		234	238	265	295	-4	-31	-61
Absolute deviation of actual student count from estimated student count						54	59	73

C.1.2 Spring Semester

Course Number	Cur.	Actual Student Count	Student Count With modifying factors			Deviation of estimated student count from actual student count for modified factor		
			0.7	0.8	0.9	0.7	0.8	0.9
505 160	505(2)	9	11	12	13	-2	-3	-4
505 311	505(4)	13	13	14	14	0	-1	-1
505 446	505(6)	13	15	16	17	3	2	1
505 466	505(8)	11	10	11	12	1	0	-1
505 480	505(8)	10	10	11	12	0	-1	-2
505 500	505(8)	10	10	11	12	0	-1	-2
505 581	505(8)	9	10	11	12	-1	-2	-3
520 422	520(6)	24	19	22	25	5	2	-1
520 430	520(8)	31	21	23	25	10	8	6
520 442	520(8)	11	15	17	19	-4	-6	-8
520 540	520(8)	12	15	17	19	-3	-5	-7
525 110	525(2)	35	30	34	38	5	1	-3
550 481	550(6)	30	19	21	23	11	9	7
550 572	550(8)	22	18	20	22	4	2	0
550 581	550(8)	21	16	18	20	5	3	1
560 583	560(8)	24	45	52	60	-21	-28	-36
580 606	580(8)	15	16	18	20	-1	-3	-5
580 691	580(8)	17	16	18	20	1	-1	-3
Total		322	309	346	383	13	-24	-61
Absolute deviation						77	78	91

C.2 ESTIMATED AND ACTUAL STUDENT COUNT FOR CATEGORY 2 COURSES - COURSES
OFFERED IN BOTH SEMESTERS AND SPECIFIED AS REQUIREMENT FOR ONLY ONE
SEMESTER IN THE CURRICULUM

C.2.1 Fall Semester

Course Number	Cur.	Actual Student Count	Student Count With modifying factors			Deviation of estimated student count from actual student count for modified factor		
			0.4	0.5	0.6	0.4	0.5	0.6
525 331	525(5)	24	19	23	27	5	1	-3
525 443	525(7)	11	22	28	34	-11	-17	-23
525 452	525(7)	7	22	28	34	-15	-21	-27
530 403	505(5)	18	11	12	13	7	6	5
530 404	530(5)	58	33	40	47	25	18	11
530 415	530(5)	105	79	86	93	26	19	12
530 417	530(7)	33	36	45	54	-3	-12	-21
530 431	530(5)	48	29	36	43	19	12	5
530 432	530(7)	34	35	44	53	-1	-10	-19
530 433	530(7)	18	35	44	53	-17	-26	-35
530 497	530(5)	51	30	37	44	21	14	7
530 502	530(7)	13	35	44	53	-22	-31	-40
550 221	550(3)	81	73	75	77	8	6	4
550 371	505(1)	11	14	16	18	-3	-5	-7
550 372	530(3)	81	80	87	94	1	-6	-13
550 451	550(3)	30	29	31	32	1	-1	2
550 575	550(7)	14	11	14	17	3	0	-13
560 212	505(1)	95	105	112	119	-10	-17	-24
	550(1)							
	525(3)							
560 413	560(5)	106	117	136	155	-11	-30	-49
	530(7)							
	550(5)							
	505(5)							
560 452	560(7)	38	38	47	56	0	-9	-18
560 521	560(7)	43	45	54	61	-2	-11	-20
560 535	560(7)	31	24	33	42	7	-2	-11
560 560	560(7)	39	49	58	67	-10	-19	-28
580 410	530(7)	30	37	46	55	-7	-16	-25
Total		1019	1008	1178	1343	111	-157	-324
Absolute deviation						235	309	412

C.2.2 Spring Semester

Course Number	Cur.	Actual Student Count	Student Count With modifying factors			Deviation of estimated student count from actual student count for modified factor		
			0.4	0.5	0.6	0.4	0.5	0.6
520 215	520(4)	28	23	25	27	5	3	1
525 332	525(6)	24	15	19	23	9	5	1
525 356	525(8)	21	19	24	29	2	-3	-8
525 422	525(6)	27	20	24	28	7	3	-1
525 426	525(8)	20	22	27	32	-2	-7	-12
525 444	525(8)	20	20	25	30	0	-5	-10
525 463	525(6)	27	19	23	27	8	4	0
525 465	525(8)	16	20	25	30	-4	-9	-14
525 471	525(8)	10	19	24	29	-9	-14	-19
530 391	530(4)	80	72	78	84	8	2	-4
530 403	550(6)	25	19	21	23	6	4	2
530 405	530(6)	41	31	37	43	10	4	-2
530 416	530(6)	59	38	44	50	21	15	9
530 434	530(8)	12	29	36	43	-17	-24	-31
530 498	530(6)	40	30	36	42	10	4	-2
530 501	530(6)	41	30	36	42	11	5	-1
530 520	530(8)	32	30	37	44	2	-7	-12
530 530	530(8)	22	29	36	43	-7	-14	-21
550 372	550(2)	75	57	58	59	18	17	16
550 401	550(4)	75	56	58	60	19	17	15
550 441	550(4)	84	66	68	70	18	16	14
550 571	550(6)	33	31	34	37	2	-1	-4
550 603	550(8)	11	13	16	19	-2	-5	-8
550 625	550(8)	15	9	12	15	6	3	0
560 212	560(2)	103	92	102	112	11	1	-9
	530(4)							
560 453	560(8)	22	27	33	40	-5	-11	-18
560 513	560(6)	32	21	30	39	11	2	-7
560 551	560(8)	33	26	33	40	7	0	-7
560 583	560(8)	24	45	52	59	-21	-28	-35
560 622	560(8)	22	28	35	42	-6	-13	-20
560 656	560(8)	16	27	34	41	-11	-18	-25
580 410	560(6)	30	31	37	43	-1	-6	-13
Total		1120	1014	1179	1345	106	-59	-226
Absolute deviation						276	269	340

C.3 ESTIMATED AND ACTUAL STUDENT COUNT FOR CATEGORY 3 COURSES - COURSESOFFERED IN BOTH THE SEMESTERS SPECIFIED AS CURRICULUM REQUIREMENT IN BOTHTHE SEMESTERSC.3.1 Fall Semester

Course Number	Cur.	Actual Student Count	Student Count With modifying factors			Deviation of estimated student count from actual student count for modified factor		
			0.7	0.8	0.9	0.7	0.8	0.9
505 115	505	29	25	27	29	4	2	0
520 115	520	67	58	65	73	9	2	-6
525 115	525	118	93	103	113	25	15	5
530 115	530	205	169	192	215	36	13	-10
550 115	550	63	52	59	66	11	4	-3
580 115	580	59	54	60	67	5	-1	-8
<u>Total</u>		<u>541</u>	<u>451</u>	<u>506</u>	<u>563</u>	<u>90</u>	<u>35</u>	<u>-22</u>
<u>Absolute deviation</u>						<u>90</u>	<u>37</u>	<u>32</u>

C.3.2 Spring Semester

Course Number	Cur.	Actual Student Count	Student Count With modifying factors			Deviation of estimated student count from actual student count for modified factor		
			0.7	0.8	0.9	0.7	0.8	0.9
505 115	505	42	32	36	41	10	6	11
520 115	520	86	73	83	93	13	3	-7
525 115	525	100	78	89	100	22	11	0
530 115	530	243	199	226	253	44	17	-10
550 115	550	69	55	62	69	14	7	0
580 115	580	73	64	72	80	9	1	-7
<u>Total</u>		<u>613</u>	<u>501</u>	<u>568</u>	<u>636</u>	<u>112</u>	<u>45</u>	<u>-23</u>
<u>Absolute Deviation</u>						<u>112</u>	<u>45</u>	<u>24</u>

APPENDIX-D

FUNCTIONS OF SUBROUTINES

Name of the Subroutine	Function
CALCUL	<ol style="list-style-type: none"> 1. To classify the courses as one or a combination of lecture, recitation, laboratory and problems, based on the method of teaching. 2. To code the courses according to whether they do or do not satisfy the minimum student count in a given course. 3. To calculate cumulative credit hours for the nine level of courses.
LECTUR	<p>To classify the lecture courses by course levels, and to route to appropriate subroutines for calculation of FTE teaching faculty requirement, deleted FTE faculty requirement for the subject classification of course levels.</p>
RECIT	<p>To classify the courses taught as recitation according to different course levels, and to route to subroutines COUDET and MINSIZ.</p>
LAB	<p>To classify the courses taught in laboratory classes according to different course levels, and to route to subroutines COUDET and MISIZ.</p>
PROB	<ol style="list-style-type: none"> 1. To classify the problem and research courses according to different course levels.

Name of the Subroutine	Function
PROB	2. To calculate FTE teaching faculty requirements for problem and research courses.
COUKNT	To calculate cumulative student count.
COUDET	1. To calculate number of sections. 2. To calculate FTE teaching faculty requirement for a given course by either Method 1 or Method 2.
TEADET	To calculate FTE teaching faculty requirement for the department and the school..
MINSIZ	To check whether the minimum course count requirement is fulfilled and codify accordingly.
MINTEA	To calculate cumulative deleted FTE teaching faculty requirement for the department and school.
FINRES	To summarize FTE teaching faculty requirement for the department and school and print the results for the department.
INIATL	To initialize variables pertaining to department.

APPENDIX-EGLOSSARY OF TERMSE1 INPUT VARIABLESE1-1 Desired Section Size

Method of Teaching	Course Level			
	000-399	400-599	600-799	>800
Lecture	LESIU1	LESIU2	LESIU3	LESIGR
Recitation	MRU1	MRU2	MRU3	MRGR
Labaratory	LSIZU1	LSIZU2	LSIZU3	LSIZGR

E1-2 Minimum Section Size

Method of Teaching	Course Level			
	000-399	400-599	600-799	>800
Lecture	MILEU1	MILEU2	MILEU3	MILEGR
Recitation	MINRU1	MINRU2	MINRU3	MINRGR
Labaratory	MINLU1	MINLU2	MINLU3	MINLGR

E1-3 Weighting Factor

Method of Teaching	Course Level			
	000-399	400-599	600-799	>800
Lecture	FRLEU1	FRLEU2	FRLEU3	FRLEGR
Recitation	FRCRU1	FRCRU2	FRCRU3	FRCRGR
Labaratory	FRCLU1	FRCLU2	FRCLU3	FRCLGR
Problems	FRCPU1	FRCPU2	FRCPU3	FRCPGR

E1-4 Preparation Load Factor

Method of Teaching	Course Level			
	000-399	400-599	600-799	>800
Lecture	PFLEU1	PFLEU2	PFLEU3	PFLEGR
Recitation	PFRU1	PFRU2	PFRU3	PFRGR
Labaratory	PFL1	PFL2	PFL3	PFLGR

E1-5 Load Factors To Represent Proportion Of Load Attributed To Preparation
And Student Count

1. FACLD1 - Proportional load factor attributed to the number of sections irrespective of count.
2. FACLD2 - Proportional load factor attributed to the student count only.

E1-6 Factors Representing Base Elements Of FTE Teaching Faculty Load

1. Base - Base student count (non weighted) to constitute full time varying load due to student count.
2. A - Base student credit hour constituting one FTE faculty load in Method 1 and problem courses.
3. B - Base credit hours to represent IFTE teaching faculty load in Method 2.

E1-7 Miscellaneous Factors

1. NODEP - Number of departments in the school
2. KA - First course number read.
3. METHOD - Code representing method of calculation of faculty load.

4. L - Code to represent semester, L=1 represents fall L=2 represents spring.
5. DEP(I,J) - Variable representing the names of the department.
6. (NOCURCJ) = Vector representing enrollment details
 J=1,10
 NOCUR(1) = Curriculum number
 (NOCUR(I), I=2,9) = Enrollment at (I-1)th level
 NOCUR(10) = Record mark, value of 1 denotes the end of this type of data cards.
7. (COUCUR(J),J=1,16) = Course curriculum details
 COUCUR(1) = Course numbers
 COUCUR(2) = Curriculum number from which the students will be enrolling for the course.
 (COUCUR(J),I=3,10) = Course curriculum fraction to represent the fraction of students from the curriculum specified, who are expected to enroll for the course.
 COUCUR(11)= Code to represent the method of teaching
 1. recitation 2. lecture
 3. lab and recitation 4. problems
 5. lecture 6. lecture and lab
 7. lecture and recitation
 8. lecture, recitation and problems
 COUCUR(12)= No. of hours taught in recitation classes
 COUCUR(13)= No. of hours taught in lab classes
 COUCUR(14)= No. of hours taught in problem classes
 COUCUR(15)= Credit hours
 COUCUR(16)= No. of hours taught in lecture classes

E2 OUTPUT VARIABLESE2-1 Results Pertaining to the Course

1. KA = Course number
2. K15 = Credit hours (=COUCUR(15))
3. KOUNT = Student count
4. LESIZE = Number of sections for the lecture classes
5. MDESSR = Number of sections for the recitation classes
6. LSIZE = Number of sections for the laboratory classes
7. NDROB = Number of sections for the problem classes (=KOUNT)
8. TKTLEC = FTE teaching faculty required for the lecture classes
9. TKTREC = FTE teaching faculty required for the recitation classes
10. TKTLAB = FTE teaching faculty required for the laboratory classes
11. TKTPRO = FTE teaching faculty required for the problem classes

E2-2 Classified Teaching Faculty Requirement for the Department

Classification	Variable Name	Classification	Variable Name
Lecture	TOTLEC	U.G. level 1 (000-399)	TOTUG1
Recitation	TOTREC	U.G. level 2 (400-599)	TOTUG2
Laboratory	TOTLAB	U.G. level 3 (600-799)	TOTUG3
Problems	TOTPRO	Graduate (>800)	TOTUGR
Department	TOTTEA		

E2-3 Classified Generated Student Credit Hours

Classification	Variable Name	Classification	Variable Name
U.G. level 1 (000-399)	JCRD1	Graduate (>800)	JCRD4
U.G. level 2 (400-599)	JCRD2	Department	JHRDEP
U.G. level 3 (600-799)	JCRD3		

E2-4 Classified Deleted FTE Teaching Faculty Requirement Due to NonFulfillment of Minimum Student Count Requirement

Classification	Variable Name	Classification	Variable Name
Lecture	TOTMLE	U.G. level 1	TOTMU1
Recitation	TOTMR	U.G. level 2	TOTMU2
Labaratory	TOTML	U.G. level 3	TOTMU3
Problems		Graduate	TOTMGR
Department	TOTDEL		

E2-5 Classified Deleted Generated Student Credit Hours Due to NonFulfillment of Minimum Student Count Requirement

Classification	Variable Name
U.G. level 1	MHRU1
U.G. level 2	MHRU2
U.G. level 3	MHRU3
Graduate	MHRGR
Department	MHRDEP

E2-6 Classified Teaching Faculty Requirement For The School

Classification	Variable Name	Classification	Variable Name
Lecture	FINLEC	U.G. level 1	FINUG1
Recitation	FINREC	U.G. level 2	FINUG2
Labaratory	FINLAB	U.G. level 3	FINUG3
Problems	FINPRO	Graduate	FINGRA
Department	NTOT		

E2-7 Classified Generated Student Credit Hours For The School

Classification	Variable Name
U.G. level 1	JCR(1)
U.G. level 2	JCR(2)
U.G. level 3	JCR(3)
Graduate	JCR(4)
Department	JHRSCH

E2-8 Classified Deleted Faculty Requirement Due To Non Fulfillment Of
Minimum Student Count Requirement For The School

Classification	Variable Name	Classification	Variable Name
Lecture	SCMLE	U.G. level 1	SCMU1
Recitation	SCMR	U.G. level 2	SCMU2
Labaratory	SCML	U.G. level 3	SCMU3
		Graduate	SCMGR
Department	NDEL		

E2-9 Classified Deleted Generated Credit Hours Due To Non Fulfillment Of
Minimum Student Count Requirement For The School

Classification	Variable Name
U.G. level 1	MSHRU1
U.G. level 2	MSHRU2
U.G. level 3	MSHRU3
Graduate	MSHRGR
Department	MHRSCH

E-3 SOME RELEVANT PROGRAM VARIABLES

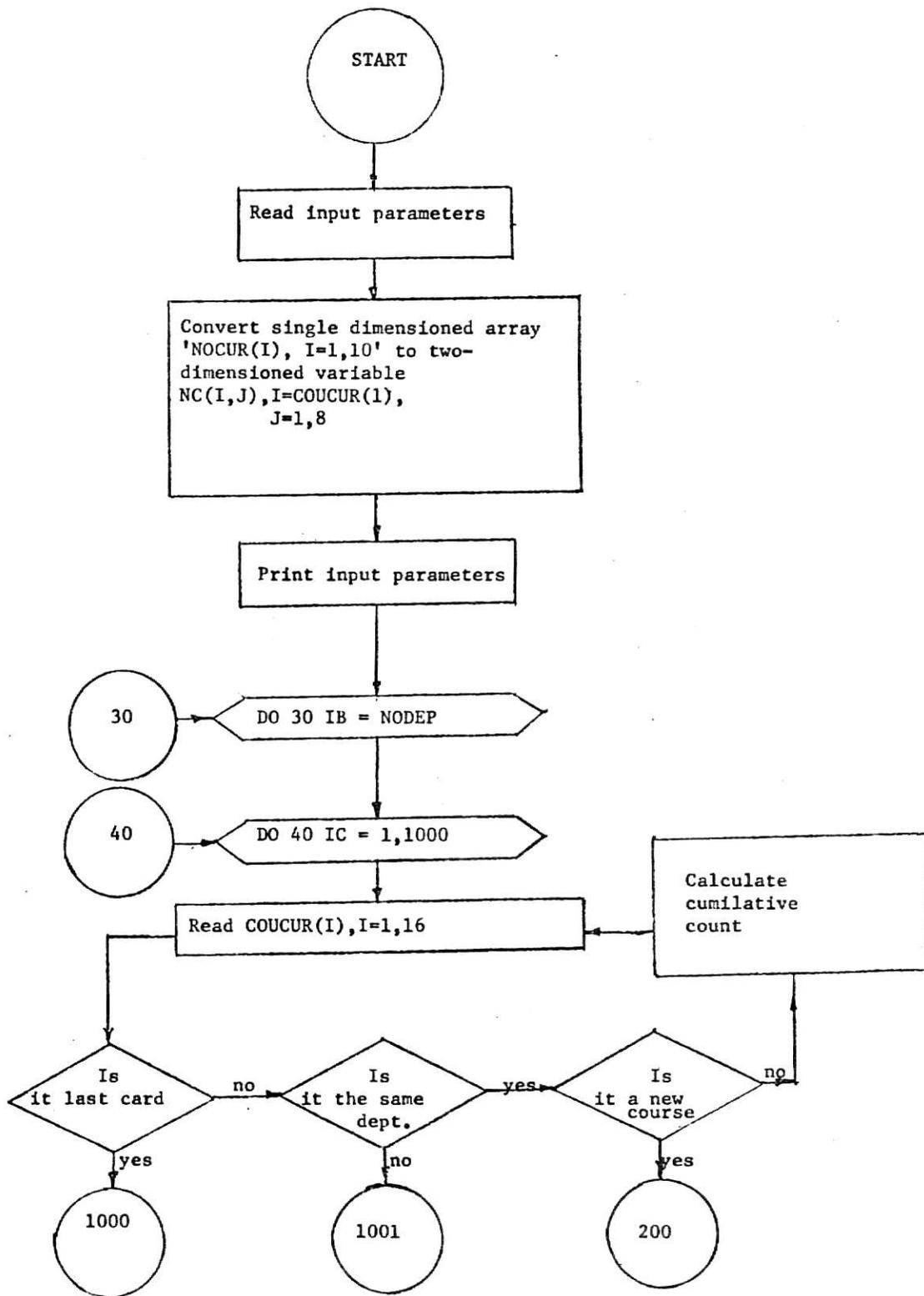
1. NC(J,J); I = 1,1012 - Two dimensional variable representing
 J = 1,8 curriculum enrollment equivalent to
 NOCUR(k), where
 I=NOCUR(1)
 J=1 to 8 represent the different levels
2. KL = same as COUCUR(11) in integer
3. K12 = same as COUCUR(12) in integer
4. K13 = same as COUCUR(13) in integer
5. K14 = same as COUCUR(14) in integer
6. K15 = same as COUCUR(15) in integer
7. K16 = same as COUCUR(16) in integer
8. KB = variable representing the present course number read
9. LB = Variable representing the curriculum to which the read
 course belongs. This is the first three digit of the course
 number.
10. LA = Same as LA except that it refers to the previous course read.
11. RKOUNT = Student count calculated in fractions.
12. KODE = Code to represent whether the minimum student count
 requirements are met.
 1. represents that the requirements are met.
 2. represents that the requirements are not met.

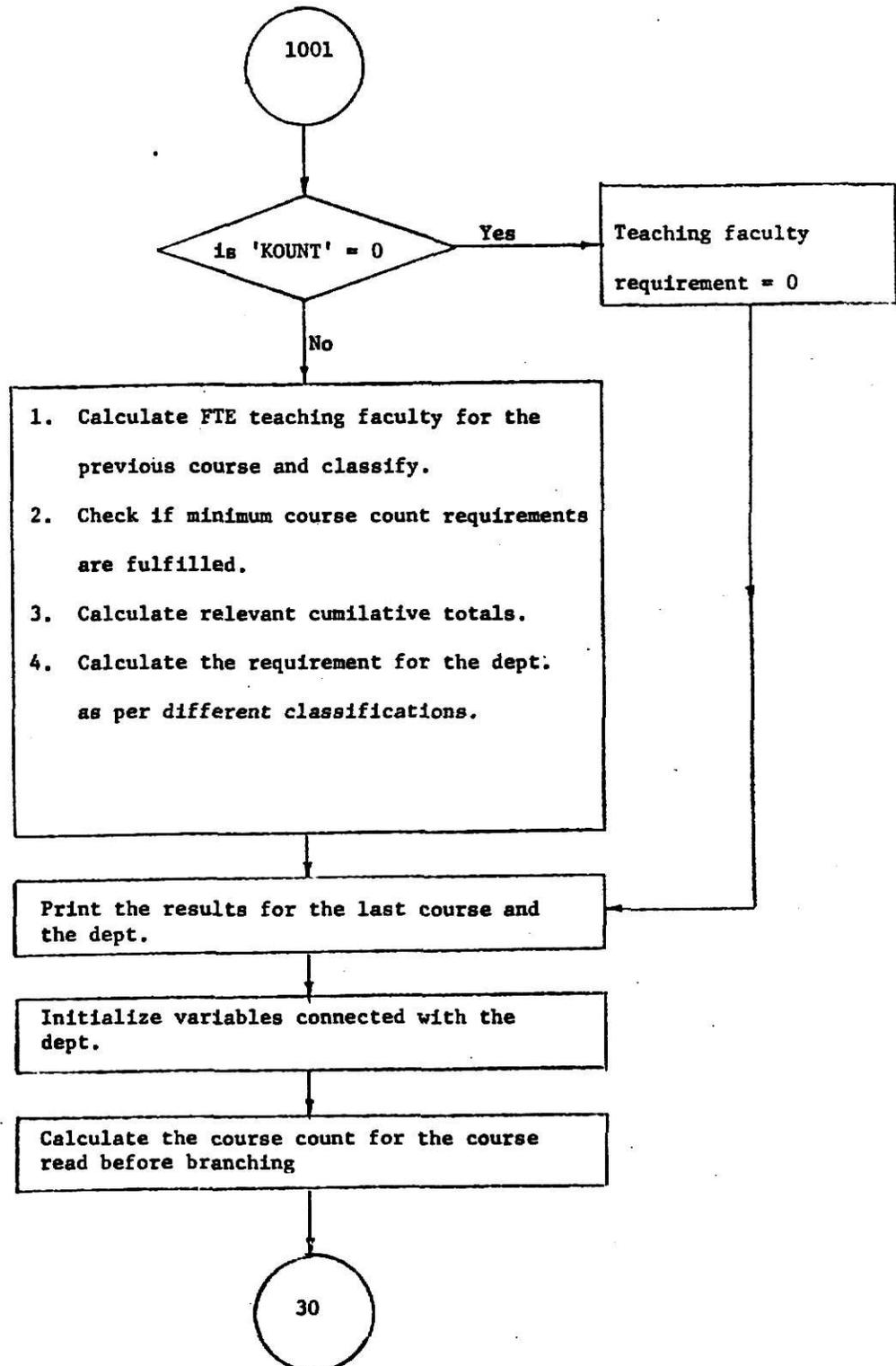
E-4 INPUT DATA CARDS AND THEIR TYPE

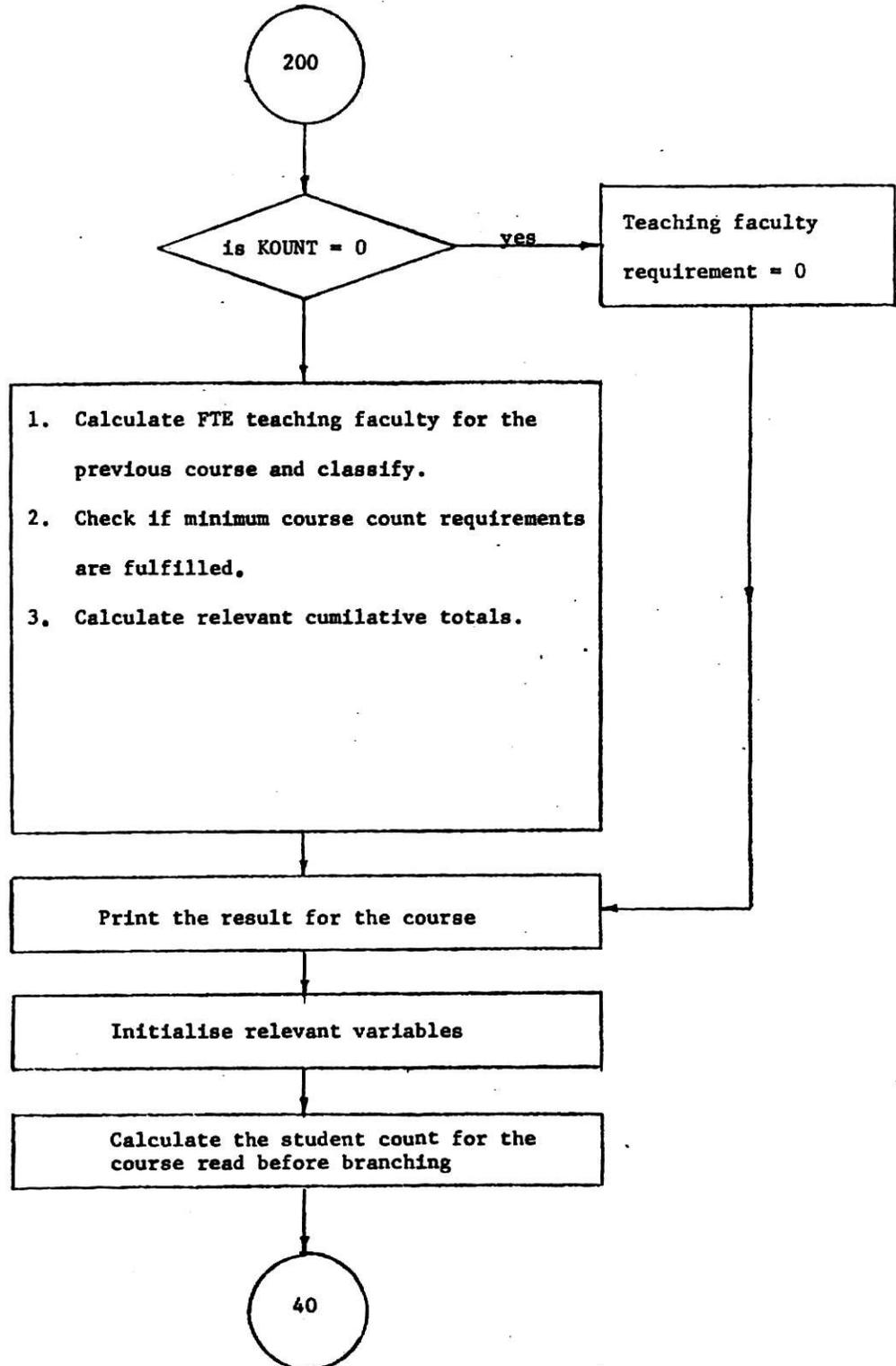
Type	Number of Cards	Format	Variables			
1	3	4I5	LESIU1	LESIU2	LESIU3	LESIGR
			MRU1	MRU2	MRU3	MRGR
			LSIZU1	LSIZU2	LSIZU3	LSIZGR
2	3	4I5	MILEU1	MILEU2	MILEU3	MILEGR
			MINUR1	MINRU2	MINRU3	MINRGR
			MINLU1	MINLU2	MINLU3	MINLGR
3	4	4F10.4	FRLEU1	FRLEU2	FRLEU3	FRLEGR
			FRCRU1	FRCRU2	FRCRU3	FRCRGR
			FRCLU1	FRCLU2	FRCLU3	FRCLGR
			FRCPU1	FRCPU2	FRCPU3	FRCPGR
4	3	4F10.4	PFLEU1	PFLEU2	PFLEU3	PFLEGR
			PFR1	PFR2	PFR3	PFRGR
			PFL1	PFL2	PFL3	PFLGR
5	1	4I10,2F5.1	NODEP	KA	METHOD	L A B
6	NODEP	4X,19A4	DEP (I,J), J=1,20, (I=1, NODEP)			
7	Unlimited to include all cur	1X,105	NOCUR(J), J=1,10			
8	Unlimited insets of two cards	F7.0, F4.0, 8F7.4 1, 4X, 8F7.4	COUCBR (I), I=1,16 J=1,10 is in the first card J=11,16 is in the second card			

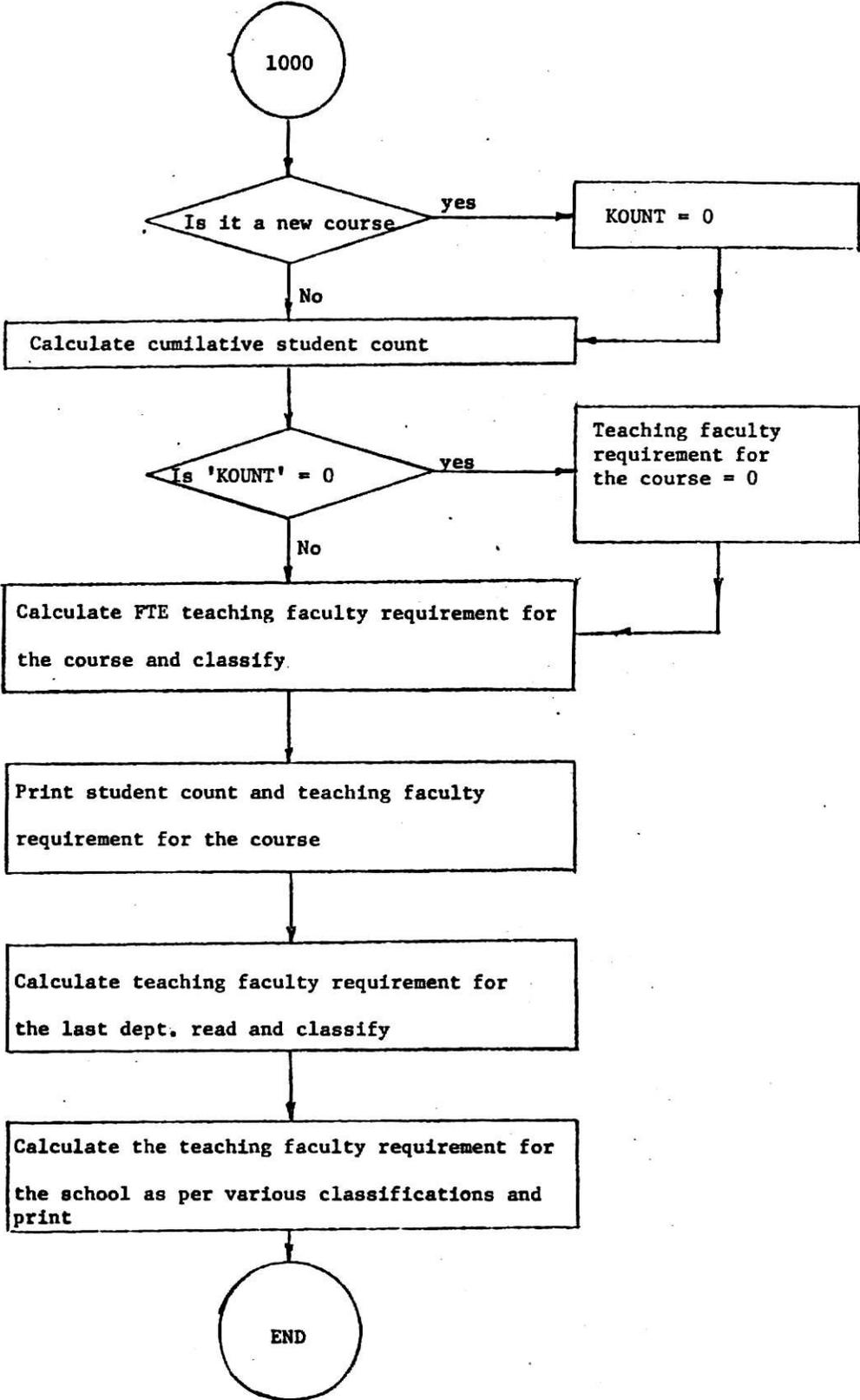
APPENDIX F

GENERAL FLOW CHART AND COMPUTER PROGRAM









```

$JOB          P.K.,TIME=10,PAGES=30,RLN=NOCHECK
1      IMPLICIT INTEGER*2(P)
2      COMMON T,AKTR,TEAKT,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
      1FRCLGR,FRCLP1,FRCLP2,FRCPGR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
      2TKPU1,TKPU2,TKPGP,TOTRU1,TOTRU2,TOTRGR,TOTLU1,TOTLU2,TOTLGR,TCTPU1
      3,TOTPL2,TOTFGR,TCTTFA,RFQUNT,MCESSR,PREM,KA,KOUNT,KC,KR,KL,K13,
      4LSIZL1,LSIZL2,LSIZGP,MINLL1,MINLL2,MINLGR,MINRU1,MINRU2,MINRGR,
      5MIPU1,MIPU2,MINPGP,LSIZE,LREM,NPRCR,NC(1C12,8),NB
3      COMMON/PK/CCUCUR(3C),TKRU3,TKLU3,TKPU3,TOTKU3,TOTLU3,TOTPU3,A,B,
      1MRU1,MRU2,MRU3,MRGR,LSIZL3,MINRU3,MINLL3,K12,K14
4      COMMON/RKS/METHOD,L
5      COMMON/RKSW/FPCRU3,FRCLU3,FRCPU3
6      COMMON/GROSH/TKTEC,FRLEU1,FRLEU2,FRLEU3,FRLEGR,TCLEU1,TCLEU2,TOLE
      1U3,TCLEGR,TKLEU1,TKLEU2,TKLEU3,TKLEGR,LESIU1,LESIU2,LESIU3,LESIGR,
      2MILEU1,MILEU2,MILEU3,MILEGR,LESIZE,LEREM
7      COMMON/SA/K14
8      COMMON/SB/TOTSCH
9      COMMON/FINAL/FINLEC,FINREC,FINLAB,FINPRO,FINUG1,FINUG2,FINUG3,
      1FINGRA,FINDEL,MSHRU1,MSHRU2,MSHRU3,MSHRGR
10     COMMON/TKTM2/FACLD1,FACLD2,BASE
11     COMMON/MET2/PFE1,PFE2,PFE3,PFEGR,PFR1,PFR2,PFR3,PFRGR,PFL1,
      1PFL2,PFL3,PFLGR
12     COMMON/SEM/JCR(10),JCRD(10),MHR(10),MHRSCH,JHRSCH,KCDE,K15
13     COMMON/MINIM/TOTMU1,TOTMU2,TCTMU3,TCTMGR,TOTML2,TOTMR,TOTML,SCMU1,
      1SCMU2,SCMU3,SCMGR,SCMLE,SCMR,SCML
14     DIMENSION NOCUR(10),DEP(10,2C)
15     FINDEL=0
16     SCML1=0
17     SCMU2=0
18     SCMU3=0
19     SCMGR=0
20     SCMLE=0
21     SCMR=0
22     SCML=0
23     MSHPL1=0
24     MSHRL2=0
25     MSHRL3=0
26     MSHRCR=0
27     FINLEC=0
28     FINREC=0
29     FINLAB=0
30     FINPRC=0
31     FINUG1=0
32     FINUG2=0
33     FINUG3=0
34     FINGRA=0
35     TOTSCH=0
36     JHRSCH=0
37     MHRSCH=0
38     FINDEL=0
39     DO 25C JA=1,4
40     25C JCR(JA)=0
41     READ 1,LESIU1,LESIU2,LESIU3,LESIGR,MRU1,MRU2,MRU3,MRGR,LSIZU1,
      1LSIZL2,LSIZU3,LSIZGP,MILEU1,MILEU2,MILEU3,MILEGR,MINRU1,MINRU2,
      2MINRU3,MINRCR,MINLL1,MINLL2,MINLL3,MINLGR,FRLEU1,FRLEU2,FRLEU3,
      3FRLEGR,FRCRU1,FRCRU2,FRCRU3,FRCRGR,FRCLU1,FRCLU2,FRCLL3,FRCLGR,
      4FRCLP1,FRCLP2,FRCLP3,FRCPGR
42     READ 2,PFE1,PFE2,PFE3,PFEGR,PFR1,PFR2,PFR3,PFRGR,PFL1,PFL2,PFL
      13,PFLGR,FACLD1,FACLD2,BASE
43     READ 3,NODEP,KA,METHOD,L,A,E

```

```

44     PRINT 15
45     PRINT 16
46     PRINT 17
47     PRINT 18,LESIU1,LESTU2,LESIL3,LESIGR,MILEL1,MILEU2,MILEU3,MILEGR,
      1MRU1,PRU2,PRU3,MRGR,MINRU1,MINRU2,MINRL3,MIARGR,LSIZL1,LSIZU2,LSIZ
      2U3,LSIZGR,MINLU1,MINLU2,MINLU3,MINLGR
48     PRINT 19
49     PRINT 16
50     PRINT 17
51     PRINT 27,FRLEU1,FRLFU2,FRLEL3,FRLEGR,PFLE1,PFLE2,PFLE3,PFLEGR,
      1FRCRU1,FRCRU2,FRCRU3,FRCRGR,PFR1,PFR2,PFR3,PFRGR,FRCLL1,FRCLU2,FRC
      2LU3,FRCLGR,PFL1,PFL2,PFL3,PFLGR,FRCPU1,FRCPU2,FRCPU3,FRCPCR
52     PRINT 21,FACLDL,FACID2,BASE
53     PRINT 22,NOCEP,KA,PFTMOD,A,B
54     PRINT 28,L
55     PRINT 26
56     DO 50 I=1,NOCEP
57     READ 3,(DEP(I,J),J=1,19)
58     50 CONTINUE
59     LA=KA-POD(KA,1000)
60     CALL IAATL
61     DC 10 IA=1,400
62     READ 2,(NOCUR(J),J=1,10)
63     PRINT 13,(NOCUR(J),J=1,10)
64     K=NCCLR(1)
65     DC 20 IA=1,8
66     NC(K,IA)=NOCUR(IA+1)
67     20 CONTINUE
68     IF(NCCLR(10).EQ.1) GO TO 100
69     10 CONTINUE
70     100 PRINT 6,(DEP(1,J),J=1,19)
71     PRINT 11
72     PRINT 12
73     PRINT 7
74     PRINT 12
75     DC 30 IB=1,NCDEP
76     DO 40 IC=1,1000
77     300 READ(5,4,END=1000)(COUCUR(I),I=1,18)
78     KB=CCLCUR(1)
79     LB=KB-POD(KB,1000)
80     IF(LB.NE.LA)GO TO 1001
81     IF(KB.NE.KA) GO TO 200
82     PRINT 24,(COLCUR(J),J=2,16)
83     KL=CCLCUR(11)
84     K12=CCLCUR(12)
85     K13=CCLCUR(13)
86     K14=CCLCUR(14)
87     K15=CCLCUR(15)
88     K16=CCLCUR(16)
89     CALL CCCKNT
90     KA=KB
91     LA=KA-POD(KA,1000)
92     GO TO 300
93     200 KOUNT=RKOUNT+0.5
94     IF(KLAT-0)504,504,505
95     504 MDESSR=0
96     LSIZE=C
97     NPRCP=C
98     TEAKTR=0
99     TEAKTL=0

```

```

100      TEAKTP=0
101      TKTLEC=0
102      KCDE=1
103      GO TO 502
104      505 CALL CALCUL
105      502 PRINT5,KA,K15,KOUNT,LESIZE,PCESSR,LSIZE,NPRCB,TKTLEC,TEAKTR,TEAKTL
          1,TEAKTP
106      IF(KCDE=50.2) PRINT 25
107      KA=KB
108      PRINT 24,(CCUCUR(J),J=2,16)
109      LA=KA-POD(KA,1000)
110      KL=CCLCUR(11)
111      K12=CCLCUR(12)
112      K13=CCLCUR(13)
113      K14=CCLCUR(14)
114      K15=CCLCUR(15)
115      K16=CCLCUR(16)
116      RKOUNT=C.0
117      CALL CCUKNT
118      40 CONTINUE
119      1001 KOUNT=RKOUNT+0.5
120      IF(KCLAT=0)514,514,515
121      514 MCESSR=0
122      LSIZE=C
123      TEAKTP=0
124      TEAKTL=0
125      TEAKTR=0
126      NPRCB=C
127      TKTLEC=0
128      KCDE=1
129      GO TO 512
130      515 CALL CALCUL
131      512 PRINT5,KA,K15,KOUNT,LESIZE,PCESSR,LSIZE,NPRCB,TKTLEC,TEAKTR,TEAKTL
          1,TEAKTP
132      IF(KCDE=50.2) PRINT 25
133      PRINT 12
134      CALL FINRES
135      KA=KB
136      LA=KA-POD(KA,1000)
137      KL=CCLCUR(11)
138      K12=CCLCUR(12)
139      K13=CCLCUR(13)
140      K14=CCLCUR(14)
141      K15=CCLCUR(15)
142      K16=CCLCUR(16)
143      K=IB+1
144      PRINT 6,(DEP(K,J),J=1,19)
145      PRINT 11
146      PRINT 12
147      PRINT 7
148      PRINT 12
149      PRINT 24,(CCUCUR(J),J=2,16)
150      CALL INIATL
151      CALL CCUKNT
152      30 CONTINUE
153      1000 KL=CCLCUR(11)
154      K12=CCLCUR(12)
155      K13=CCLCUR(13)
156      K14=CCLCUR(14)
157      K15=CCLCUR(15)

```

```

158      K16=CCLCUR(16)
159      KB=CCLCUR(1)
160      IF(KB.EQ.KA) GO TO *01
161      RKOUNT=0.0
162      501 CALL CCUKNT
163      KOUNT=RKOUNT+0.5
164      IF(KCOUNT=0)506,506,*07
165      506 MDESSR=0
166      LSIZE=C
167      NPRCB=C
168      TEAKTR=0
169      TEAKTL=0
170      TEAKTP=0
171      TKTLEC=0
172      KCDE=1
173      GO TO 503
174      507 CALL CALCUL
175      503 PRINTS,KA,K15,KOUNT,LESIZE,MDESSR,LSIZE,NPRCB,TKTLEC,TEAKTR,TEAKTL
176      1,TEAKTR
177      IF(KCDE.EQ.2) PRINT 25
178      PRINT 12
179      CALL FINRES
180      PRINT 23
181      FINDEL=SCMU1+SCMU2+SCMU3+SCMGR
182      TOTSCH=TOTSCH-FINDEL
183      NTOT=TOTSCH+0.5
184      NDEL=FINDEL+C.5
185      PRINT 31,FINLEC,SCMLE,FINLG1,SCMU1,JCR(1),MSHRU1
186      PRINT 32,FINREC,SCMP,FINUG2,SCMU2,JCR(2),MSHRU2
187      PRINT 33,FINLAB,SCPL,FINUG3,SCMU3,JCR(3),MSPRU3
188      PRINT 34,FINPRO,FINGRA,SCMGR,JCR(4),MSHRGR
189      PRINT 35,NTOT,NDEL,JHRSCH,MFRSCH
190      PRINT 14
191      1 FORMAT(4I5,/,4I5,/,4I5,/,4I5,/,4I5,/,4I5,/,4F10.4,/,4F10.4,/,4F10.
192      24,/,4F10.4)
193      2 FORMAT( 10I4)
194      3 FORMAT(4X,19A4)
195      4 FORMAT(F7.0,F4.0,8F7.4,/,4X,8F7.4)
196      5 FORMAT(4I10,2F5.1)
197      6 FORMAT('1',4X,19A4)
198      7 FORMAT(3X,'COUR.NO OR STU SECTION SIZE FTE TEACHING LOAD CUR
199      1',6X,'FRESHMEN',6X,'SOPHOMORE',6X,'JUNIOR',6X,'SENIOR',5X,'COUR.D
200      2DETAILS',/,18X,'-----',11X,4I13(1H-),1X
201      3),1X,'-----',/,18X,'LE RE LA PR LE RE LA PR
202      4 LEVEL1 LEVEL2 LEVEL3 LEVEL4 LEVEL5 LEVEL6 LEVEL7 LEVEL8 C P L
203      5 P CR L')
204      8 FORMAT(4F10.4,/,4F10.4,/,4F10.4,/,4F10.4)
205      9 FORMAT(3X,I6,2X,I2,1X,I3,4I3,1X,4F5.2)
206      11 FORMAT(24X,35(1H-))
207      12 FORMAT(3X,128(1H-))
208      13 FORMAT(1X,10I5)
209      14 FORMAT(3X,'NCTE.--COURSES WHICH DO NOT FULFIL THE MINIMUM COURSE R
210      1EQUIREMENTS AND THE RESULTING DELETIONS ARE ENCLCSED WITH-IN ASTERI
211      2KS.')
```

```

208 21 FORMAT('0',10X,'FRACTIONAL LCAD FOR PREPERATION=',F6.2,10X,'FRACTI
209 22 FORMAT('0',2X,'NUMBFR OF DEPARTMENT=',I2,' FIRST COURSE NUMBER REA
    1C=',I6,' METHOD NO=',I2,' FTE TEACHING LOAD FOR PROBLEMS=',F6.1,' F
    2TE CREDIT HCURS=',F6.1,/,13C(1H-))
210 23 FORMAT('1',3X,'FINAL FTE TEACHING LCAD FOR THE SCHOOL UNDER CONSID
    ERATION',/,3X,60(1H-))
211 24 FORMAT(55X,F5.0,1X,F7.4,1X,6F2.0)
212 25 FORMAT('+',10X,48X,10X)
213 26 FORMAT('1',/,'STUDENT ENROLLMENT DATA',/,5X,23(1H-),//,' CLR
    1 L1 L2 L3 L4 L5 L6 L7 L8 R.M',/,1X,5C(1H-))
214 27 FORMAT('0',1X,'LECTURE ',4F8.2,15X,4F8.2,/,2X,'RECITATION',4F8.2
    1,15X,4F8.2,/,2X,'LABORATORY',4F8.2,15X,4F8.2,/,2X,'PROBLEMS ',4F8
    2.2,/,51(1H-))
215 28 FORMAT(' ',2X,'SEMESTER=',I2,2X,'*1* REPRESENTS FALL AND *2* REPRE
    SENTS SPRING')
216 31 FORMAT(3X,'FTE RECITATION ',F6.2,1X,10X,F6.2,10X,3X,'FTE U.G.LE
    VEL1 ',F6.2,1X,10X,F6.2,10X,3X,'TOTAL UG1 CR+RS ',I5,1X,10X,
    2,I5,10X)
217 32 FORMAT(3X,'FTE RECITATION ',F6.2,1X,10X,F6.2,10X,3X,'FTE U.G.LE
    VEL2 ',F6.2,1X,10X,F6.2,10X,3X,'TOTAL UG2 CR+RS ',I5,1X,10X,
    2,I5,10X)
218 33 FORMAT(3X,'FTE LABORATORY ',F6.2,1X,10X,F6.2,10X,3X,'FTE U.G.LE
    VEL3 ',F6.2,1X,10X,F6.2,10X,3X,'TOTAL UG3 CR+RS ',I5,1X,10X,
    2,I5,10X)
219 34 FORMAT(3X,'FTE PROBLEMS ',F6.2,14X,'FTE GRADUATE ',F6.2,1X,
    10X,F6.2,10X,3X,'TOTAL GRAD CR+RS ',I5,1X,10X,10X,I5,10X,10X,
    2))
220 35 FORMAT(3X,'FTE SCHOOL ',C1I5,1X,10X,0I5,10X,41X,'TOTAL SCH-
    1 CR+RS ',I5,1X,10X,10X,I5,10X,10X,10X,10X,10X,10X,10X,10X,
    10X)
221 STOP
222 END

223 SUBROUTINE COUNT
224 IMPLICIT INTEGER*2(N)
225 COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
    1FRCLGR,FRCPL1,FRCPL2,FRCPL3,TKLUI,TKLU2,TKLGR,TKRUI,TKRU2,TKRGR,
    2TKPU1,TKPU2,TKPGR,TOTRU1,TOTRU2,TOTRGR,TCTLL1,TCTLU2,TCTLGR,TCTPU1
    3,TOTPL2,TOTPCR,TCTTFA,RKOUNT,MCESSR,MREM,KA,KOUNT,KC,KR,KL,K13,
    4LSIZL1,LSIZL2,LSIZGP,MINLL1,MINLU2,MINLGR,MINRU1,MINRL2,MINRGR,
    5MINPU1,MINPU2,MINPGP,LSIZE,LREM,NPRGR,NC(1C12,8),NB
226 COMMON/RK/CCUCUR(30),TKRU3,TKLU3,TKPU3,TOTRU3,TOTLU3,TOTPU3,A,B,
    1MRUI,MRU2,MRU3,MRGR,LSIZU3,MINRU3,MINLU3,K12,K14
227 COMMON/RKS/FRCRU3,FRCLU3,FRCPL3
228 COMMON/RKS/METHCC,L
229 LA=L+7
230 CO 5C IC=L,LA,2
231 KC=CCLCUR(2)
232 RKDLAT=RKDLNT+NC(KC,IC)*CCLCLR(IC+2)
233 50 CONTINUE
234 RETURN
235 END

236 SUPROUTINE CALCUL
237 IMPLICIT INTEGER*2(N)
238 COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
    1FRCLGR,FRCPL1,FRCPL2,FRCPL3,TKLUI,TKLU2,TKLGR,TKRUI,TKRU2,TKRGR,
    2TKPU1,TKPU2,TKPGR,TOTRU1,TOTRU2,TOTRGR,TCTLL1,TCTLU2,TCTLGR,TCTPU1
    3,TOTPL2,TOTPCR,TCTTFA,RKOUNT,MCESSR,MREM,KA,KOUNT,KC,KR,KL,K13,
    4LSIZL1,LSIZL2,LSIZGP,MINLL1,MINLU2,MINLGR,MINRU1,MINRU2,MINRGR,

```

```

239      SMINPL1,MINPL2,MINPGR,LSIZE,LREM,NPRCB,NC(1012,8),NB
        COMMON/RK/CCUCUR(30),TKRU3,TKLU3,TKPU3,TOTRU3,TOTLU3,TOTPU3,A,B,
        1PRU1,PRU2,PRU3,PRGR,LSIZU3,MINRU3,MINLU3,K12,K14
240      COMMON/RKSW/FRCRU3,FRCLU3,FRCPU3
241      COMMON/GROSH/TKTLEC,FRLEU1,FRLEU2,FRLEU3,FRLEGR,TOLEU1,TOLEU2,TOLE
        1U3,TCLEGR,TKLEU1,TKLEL2,TKLEL3,TKLEGR,LESIU1,LESIU2,LESIU3,LESIGR,
        2MILEU1,MILEU2,MILEU3,MILEGR,LESIU2,LEREM
242      COMMON/SA/K16
243      COMMON/SFM/JCR(10),JCRC(10),MHR(10),PHRSCH,JHRSCH,KODE,K15
244      COMMON/MINIM/TCTPU1,TCTPU2,TCTPU3,TCTMGR,TCTMLE,TCTMR,TOTML,SCMU1,
        1SCHU2,SCMU3,SCMGR,SCMLE,SCMR,SCML
245      NB=MCC(KA,1000)/100
246      JCRC(NB)=JCRC(NB)+KOUNT*K15
247      GO TC(1,1,1,2,2,3,3,4,4),NB
248      1 MLE=MILEU1
249      MR=MINRU1
250      ML=MINLU1
251      GO TC 10
252      2 MLE=MILEU2
253      MR=MINRU2
254      ML=MINLU2
255      GO TC 10
256      3 MLE=MILEU3
257      ML=MINLU3
258      MR=MINRU3
259      GO TC 10
260      4 MLE=MILEGR
261      ML=MINLGR
262      MR=MINRGR
263      10 CONTINUE
264      GO TC(101,102,103,104,105,106,107,108),KL
265      101 CALL MINSIZ(MR,KODE)
266      CALL RECIT
267      LSIZE=C
268      TEAKTL=0
269      TEAKTP=0
270      NPRCP=C
271      LESIZE=C
272      TKTLEC=0
273      IF(KCDE=NE*2) GO TC 900
274      MHR(NB)=MHR(NB)+KOUNT*K15
275      RETURN
276      102 CALL MINSIZ(ML,KCDE)
277      CALL LAB
278      MDESSR=C
279      TEAKTR=C
280      TEAKTP=0
281      NPRCB=C
282      LSIZE=C
283      TKTLEC=0
284      IF(KCDE=NE*2) GO TO 900
285      MHR(NB)=MHR(NB)+KOUNT*K15
286      RETURN
287      103 CALL MINSIZ(MR,KODE1)
288      CALL MINSIZ(ML,KCDE2)
289      KCDE=2
290      IF((KCDE1.EQ.1).OR.(KCDE2.EQ.1)) KCDE=1
291      CALL RECIT
292      CALL LAB
293      TEAKTP=0

```

```

294     NPRCP=C
295     LESIZE=0
296     TKTLEC=0
297     IF(KCDE=NE.2) GO TO 900
298     MHR(NB)=MHR(NB)+KOUNT*K15
299     RETURN
300 104 CALL PROB
301     MDESSR=C
302     LSIZE=C
303     TEAKTL=0
304     TEAKTR=0
305     NPRCP=KOUNT
306     LESIZE=0
307     TKTLEC=0
308     KOCF=1
309     RETURN
310 105 CALL MINSIZ(MLE,KOCF)
311     CALL LECTUR
312     MDESSR=0
313     LSIZE=C
314     NPRCP=C
315     TEAKTR=0
316     TEAKTL=0
317     TEAKTP=0
318     IF(KCDE=NE.2) GO TO 900
319     MHR(NB)=MHR(NB)+KOUNT*K15
320     RETURN
321 106 CALL MINSIZ(MLE,KOCF1)
322     CALL MINSIZ(ML,KCDE?)
323     KCDE=2
324     IF((KCDE1.EQ.1).OR.(KCDE2.EC.1)) KCDE=1
325     CALL LECTUR
326     CALL LAB
327     MDESSR=0
328     NPRCP=C
329     TEAKTR=0
330     TEAKTP=C
331     IF(KCDE=NE.2) GO TO 900
332     MHR(NB)=MHR(NB)+KOUNT*K15
333     RETURN
334 107 CALL MINSIZ(MLE,KOCF1)
335     CALL MINSIZ(MR,KCDE?)
336     KCDE=2
337     IF((KOCF1.EQ.1).OR.(KCDE2.EC.1)) KCDE=1
338     CALL LECTUR
339     CALL RECIT
340     LSIZE=C
341     NPRCP=C
342     TEAKTL=0
343     TEAKTP=0
344     IF(KCDE=NE.2) GO TO 900
345     MHR(NB)=MHR(NB)+KOUNT*K15
346     RETURN
347 108 CALL MINSIZ(MLE,KOCF1)
348     CALL MINSIZ(MR,KCDE?)
349     CALL MINSIZ(ML,KCDE?)
350     KCDE=2
351     IF((KOCF1.EC.1).OR.(KCDE2.EC.1).OR.(KCDE3.EC.1)) KCDE=1
352     CALL LECTUR
353     CALL RECIT

```

```

3,TOTPL2,TOTPCR,TCTTFA,RKOUNT,MDESSR,MREM,KA,KOUNT,KC,KR,KL,K13,
4LSIZL1,LSIZL2,LSIZGP,MINLL1,MINLL2,MINLGR,MINRU1,MINRU2,MINRGR,
5MINPL1,MINPL2,MINPGP,LSIZE,LREM,NPRCR,AC(1012,8),NB
400  CCMPCN/RK/CCUCUR(30),TKRU3,TKLU3,TKPL3,TCTRL3,TOTLU3,TOTPU3,A,B,
1PRU1,PRU2,PRU3,PRGR,LSIZL3,MINRU3,MINLL3,K12,K14
401  CCMPCN/RKS#/FRCL3,FRCLL3,FRCLP3
402  CCMPCN/MET2/PFLE1,PFLE2,PFLE3,PFLEGR,PFR1,PFR2,PFR3,PFRGR,PFL1,
1PFL2,PFL3,PFLGR
403  CCMPCN/SEM/JCR(10),JCRD(10),PHR(10),MHRSCHE,JHRSCHE,KCCE,K15
404  CCMPCN/MINIM/TOTMU1,TOTMU2,TCTMU3,TCTMGR,TOTMLE,TOTMR,TOTML,SCMU1,
1SCML2,SCMU3,SCMGR,SCMLE,SCMR,SCML
405  GC TC(500,500,500,600,600,650,650,700,700),NB
406  500 CALL CCUDET(MRU1,MINRU1,FRCL1,MDESSR,MREM,TKRU1,K12,PFR1)
407  CALL TEADET(TKRU1,TCTRU1)
408  TEAKTR=TKRU1
409  IF(KCCE=EQ.1) GO TC 900
410  CALL MINTEA(TEAKTR,TOTMU1,TCTMR,SCML1,SCMR)
411  GC TC 500
412  600 CALL CCUDET(MRU2,MINRU2,FRCL2,MDESSR,MREM,TKRU2,K12,PFR2)
413  CALL TEADET(TKRU2,TCTRU2)
414  TEAKTR=TKRU2
415  IF(KCCE=EQ.1) GO TC 900
416  CALL MINTEA(TEAKTR,TOTMU2,TCTMR,SCML2,SCMR)
417  GC TC 500
418  650 CALL CCUDET(MRU3,MINRU3,FRCL3,MDESSR,MREM,TKRU3,K12,PFR3)
419  CALL TEADET(TKRU3,TCTRU3)
420  TEAKTR=TKRU3
421  IF(KCCE=EQ.1) GO TC 900
422  CALL MINTEA(TEAKTR,TOTMU3,TCTMR,SCML3,SCMR)
423  GC TC 900
424  700 CALL CCUDET(MRGR,MINRGR,FRGR,MDESSR,MREM,TKRGR,K12,PFRGR)
425  CALL TEADET(TKRGR,TCTRGR)
426  TEAKTR=TKRGR
427  IF(KCCE=EQ.1) GO TC 900
428  CALL MINTEA(TEAKTR,TOTMGR,TCTMR,SCMGR,SCMR)
429  900 RETURN
430  END

431  SUBROUTINE LAB
432  IMPLICIT INTEGER*(*)
433  CCMPCN TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLL2,
1FRCLCH,FRCPUI,FRCPU2,FRCPGR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
2TKPL1,TKPU2,TKPGR,TCTRU1,TCTRU2,TCTRGR,TCTLU1,TOTLU2,TOTLGR,TOTPU1
3,TOTPL2,TOTPCR,TCTTFA,RKOUNT,MDESSR,MREM,KA,KOUNT,KC,KR,KL,K13,
4LSIZL1,LSIZL2,LSIZGP,MINLL1,MINLL2,MINLGR,MINRU1,MINRU2,MINRGR,
5MINPL1,MINPL2,MINPGP,LSIZE,LREM,NPRCB,AC(1012,8),NB
434  CCMPCN/RK/CCUCUR(30),TKRU3,TKLU3,TKPL3,TCTRL3,TCTLU3,TCTPU3,A,B,
1PRU1,PRU2,PRU3,MRGR,LSIZL3,MINRU3,MINLL3,K12,K14
435  CCMPCN/RKS#/FRCL3,FRCLL3,FRCPU3
436  CCMPCN/MET2/PFLE1,PFLE2,PFLE3,PFLEGR,PFR1,PFR2,PFR3,PFRGR,PFL1,
1PFL2,PFL3,PFLGR
437  CCMPCN/SEM/JCR(10),JCRD(10),PHR(10),MHRSCHE,JHRSCHE,KCCE,K15
438  CCMPCN/MINIM/TOTMU1,TCTMU2,TCTMU3,TCTMGR,TCTMLE,TCTMR,TOTML,SCMU1,
1SCMU2,SCMU3,SCMGR,SCMLE,SCMR,SCML
439  GC TC(500,500,500,600,600,650,650,700,700),NB
440  500 CALL CCUDET(LSIZL1,MINLL1,FRCLL1,LSIZE,LREM,TKLU1,K13,PFL1)
441  CALL TEADET(TKLU1,TCTLU1)
442  TEAKTL=TKLU1
443  IF(KCCE=EQ.1) GO TC 900
444  CALL MINTEA(TEAKTL,TOTMU1,TCTML,SCML1,SCPL1)

```

```

354      CALL LAB
355      NPRCP=C
356      IF(KCCE=NE.2) GO TC 900
357      MHR(NB)=MHR(NB)+KOLNT*K15
358 900  RETURN
359      END

360      SUBROUTINE LECTUR
361      IMPLICIT INTEGER*2(M)
362      COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
1FRCLGR,FRCPL1,FRCPL2,FRCPL3,FRCPL4,FRCPL5,FRCPL6,FRCPL7,FRCPL8,
2TKPU1,TKPU2,TKPGR,TCTRU1,TCTRU2,TCTRGR,TCTLU1,TOTLU2,TOTLGR,TOTPU1
3,TOTPL2,TOTPGR,TOTIFA,RKOLAT,MDESSR,PREM,KA,KOUNT,KC,KR,KL,K13,
4LSIZU1,LSIZU2,LSIZG,MINLU1,MINLU2,MINLGR,MINRU1,MINRL2,MINRGR,
5MINPL1,MINPL2,MINPGP,LSIZE,LEREM,NPRCE,AC(1G12,8),AB
363      COMMON/RK/CCUCUR(3C),TKRU3,TKL3,TKPU3,TCTRL3,TCTLU3,TCTPU3,A,B,
1PRU1,PRU2,PRU3,MRGR,LSIZU3,MINRU3,MINLU3,K12,K14
364      COMMON/RKS/METHCC,L
365      COMMON/RKSW/FRCRU3,FRCLU3,FRCPL3
366      COMMON/CROSH/TKLE1,FRLEU1,FRLEU2,FRLEU3,FRLEGR,TOLEU1,TCLEU2,TOLE
1U3,TKLEGR,TKLEU1,TKLEU2,TKLEU3,TKLEGR,LESIU1,LESIU2,LESIU3,LESIGR,
2MILE1,MILEU2,MILEU3,MILEGR,LESIU,LEREM
367      COMMON/SA/K16
368      COMMON/MET2/PFLE1,PFLE2,PFLE3,PFLEGR,PFR1,PFR2,PFR3,PFRGR,PFL1,
1PFL2,PFL3,PFLGR
369      COMMON/SEM/JCR(10),JCRD(10),MHR(10),MHRSCH,JHRSCH,KODE,K15
370      COMMON/MINIM/TOTMU1,TOTMU2,TOTMU3,TCTMGR,TOTMLE,TOTMR,TOTML,SCMU1,
1SCMU2,SCMU3,SCMGR,SCMLE,SCMR,SCML
371      GO TC (500,500,500,600,600,650,650,700,700),AB
372 500  CALL CCUDET(LESIU1,MILEU1,FRLEU1,LESIU,LEREM,TKLEU1,K16,PFLE1)
373      CALL TEADET(1TKLEU1,TOLE1)
374      TKLE1=TKLEU1
375      IF(KCCE=EQ.1) GO TC 900
376      CALL MINTEA(1TKLE1,TOTMU1,TCTMLE,SCMU1,SCMLE)
377      GO TC 500
378 600  CALL CCUDET(LESIU2,MILEU2,FRLEU2,LESIU,LEREM,TKLEU2,K16,PFLE2)
379      CALL TEADET(1TKLEU2,TOLE2)
380      TKLE2=TKLEU2
381      IF(KCCE=EQ.1) GO TC 900
382      CALL MINTEA(1TKLE2,TOTMU2,TCTMLE,SCMU2,SCMLE)
383      GO TC 500
384 650  CALL CCUDET(LESIU3,MILEU3,FRLEU3,LESIU,LEREM,TKLEU3,K16,PFLE3)
385      CALL TEADET(1TKLEU3,TOLE3)
386      TKLE3=TKLEU3
387      IF(KCCE=EQ.1) GO TC 900
388      CALL MINTEA(1TKLE3,TOTMU3,TCTMLE,SCMU3,SCMLE)
389      GO TC 500
390 700  CALL CCUDET(LESIU4,MILEU4,FRLEU4,LESIU,LEREM,TKLEU4,K16,PFLE4)
391      CALL TEADET(1TKLEU4,TOLE4)
392      TKLE4=TKLEU4
393      IF(KCCE=EQ.1) GO TC 900
394      CALL MINTEA(1TKLE4,TOTMU4,TCTMLE,SCMU4,SCMLE)
395 900  RETURN
396      END

397      SUBROUTINE RECIT
398      IMPLICIT INTEGER*2(M)
399      COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
1FRCLCH,FRCPL1,FRCPL2,FRCPL3,FRCPL4,FRCPL5,FRCPL6,FRCPL7,FRCPL8,
2TKPU1,TKPU2,TKPGR,TCTRU1,TCTRU2,TCTRGR,TCTLU1,TCTLU2,TCTLGR,TCTPU1

```

```

445      GC TC 900
446      600 CALL CCUDET(LSIZU2,MINLU2,FRCLU2,LSIZE,LREM,TKLU2,K13,PFL2)
447      CALL TEADET(TKLU2,TOTLU2)
448      TEAKTL=TKLU2
449      IF(KCCE.EQ.1) GC TC 900
450      CALL MINTEA(TEAKTL,TOTPU2,TCTPL,SCML2,SCML)
451      GO TO 900
452      650 CALL CCUDET(LSIZU3,MINLU3,FRCLU3,LSIZE,LREM,TKLU3,K13,PFL3)
453      CALL TEADET(TKLU3,TOTLU3)
454      TEAKTL=TKLU3
455      IF(KCCE.EQ.1) GO TO 900
456      CALL MINTEA(TEAKTL,TOTMU3,TCTML,SCML3,SCML)
457      GO TO 900
458      700 CALL CCUDET(LSIZGR,MINLGR,FRCLGR,LSIZE,LREM,TKLGR,K13,PFLGR)
459      CALL TEADET(TKLGR,TOTLGR)
460      TEAKTL=TKLGR
461      IF(KCCE.EQ.1) GO TO 900
462      CALL MINTEA(TEAKTL,TOTMGR,TCTML,SCMGR,SCML)
463      900 RETURN
464      END

465      SUBROUTINE PROB
466      IMPLICIT INTEGER*2(I)
467      COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLU1,FRCLU2,
1FRCLCR,FRCPU1,FRCPU2,FRCPCR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
2TKPL1,TKPU2,TKPGR,TOTRU1,TOTRU2,TOTRGR,TOTLU1,TOTLU2,TOTLGR,TOTPU1
3,TOTPL2,TOTPCR,TCTTFA,RKOLAT,PCESSR,MREM,KA,KCUNT,KC,KR,KL,K13,
4LSIZU1,LSIZU2,LSIZGR,MINLU1,MINLU2,MINLGR,MINRU1,MINRU2,MINRGR,
5MINPL1,MINPU2,MINPCR,LSIZE,LREM,NPRCB,NC(1012,8),NB
468      COMMON/RK/CCUCUR(30),TKRU3,TKLU3,TKPU3,TOTRU3,TOTLU3,TOTPU3,A,B,
1MRU1,MRU2,MRU3,MRGR,LSIZL3,MINRU3,MINLU3,K12,K14
469      COMMON/RKS6/FRCRU3,FRCLU3,FRCPU3
470      GC TC(500,500,500,600,600,650,650,700,700),NB
471      500 TEAKTP=FRCPU1*KCUNT*K14/A
472      KTPU1=TEAKTP
473      CALL TEADET(TEAKTP,TOTPU1)
474      GO TO 500
475      600 TEAKTP=FRCPU2*KCUNT*K14/A
476      KTPL2=TEAKTP
477      CALL TEADET(TEAKTP,TOTPU2)
478      GO TO 500
479      650 TEAKTP=FRCPU3*KCUNT*K14/A
480      KTPU3=TEAKTP
481      CALL TEADET(TEAKTP,TOTPU3)
482      GC TC 500
483      700 TEAKTP=FRCPCR*KCUNT*K14/A
484      KTPGR=TEAKTP
485      CALL TEADET(TEAKTP,TOTPCR)
486      900 RETURN
487      END

488      SUBROUTINE CCUDET(MSIZCO,MINCC,FRAC,KSIZE,KREM,TKCC,KCL,PRFRAC)
489      IMPLICIT INTEGER*2(I)
490      COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLU1,FRCLU2,
1FRCLGR,FRCPL1,FRCPL2,FRCPCR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
2TKPU1,TKPU2,TKPGR,TOTRU1,TOTRU2,TOTRGR,TOTLU1,TOTLU2,TOTLGR,TOTPU1
3,TOTPL2,TOTPCR,TCTTFA,RKOUNT,PCESSR,MREM,KA,KCUNT,KC,KR,KL,K13,
4LSIZU1,LSIZU2,LSIZGR,MINLU1,MINLU2,MINLGR,MINRU1,MINRU2,MINRGR,
5MINPL1,MINPU2,MINPCR,LSIZE,LREM,NPRCB,NC(1012,8),NB
491      COMMON/RK/CCUCUR(30),TKRU3,TKLU3,TKPU3,TOTRU3,TOTLU3,TOTPU3,A,B,

```

```

1PRU1,PRU2,PRU3,MRGR,LSIZU3,MINRU3,MINLU3,K12,K14
492 COMMON/RKS/METHOD,L
493 COMMON/TKTM2/FACLD1,FACLD2,BASE
494 IF(KCLAT.LE.KSIZCO)GOTO 201
495 KSIZE=KOUNT/KSIZCO
496 KREM=KCC(KCUNT,KSIZCO)
497 NEXTRA=KREM/KSIZE
498 IF(NEXTRA.LE.MINCO)GOTO 10
499 KSIZE=KSIZE+1
500 GO TO 10
501 201 KSIZCO=1
502 KREM=C
503 10 GO TO(20,30),METHOD
504 20 TKCC=FRAC*KCUNT*KCL/A
505 GO TO 40
506 30 TKCC=(KSIZE*PRFRAC*FACLD1+KCLNT/BASE*FRAC*FACLD2)*KCL/B
507 40 RETUFA
508 END

509 SUBROUTINE TEADET(TYCO,TKFCL)
510 IMPLICIT INTEGER*2(I*)
511 COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
1FRCLGR,FRCPU1,FRCP2,FRCPGR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
2TKPU1,TKPU2,TKPGR,TOTRU1,TOTRU2,TOTRGR,TOTLL1,TOTLU2,TOTLGR,TCTPU1
3,TOTPL2,TOTPGR,TCTTFA,RKCLNT,MCESSR,MREM,KA,KOUNT,KC,KR,KL,K13,
4LSIZU1,LSIZU2,LSIZGP,MINLU1,MINLU2,MINLGR,MINRU1,MINRU2,MINRGR,
5MINPU1,MINPU2,MINPGP,LSIZE,LREM,NPRCB,NC(1012,8),NB
512 COMMON/RK/CCUCUR(30),TKRU3,TKLU3,TKPU3,TOTRU3,TOTLU3,TOTPU3,A,B,
1PRU1,PRU2,PRU3,MRGR,LSIZU3,MINRU3,MINLU3,K12,K14
513 COMMON/SB/TOTSCH
514 TKFCL=TKFCL+TKCC
515 TOTTEA=TOTTEA+TKCC
516 TOTSCH=TOTSCH+TKCC
517 RETUPA
518 END

519 SUBROUTINE MINSI7(MIN,KODEK)
520 IMPLICIT INTEGER*2(I*)
521 COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
1FRCLGR,FRCP1,FRCP2,FRCPGR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
2TKPU1,TKPU2,TKPGR,TOTRU1,TOTRU2,TOTRGR,TOTLL1,TOTLU2,TOTLGR,TCTPU1
3,TOTPU2,TOTPGR,TCTTFA,RKOUNT,MCESSR,MREM,KA,KOUNT,KC,KR,KL,K13,
4LSIZU1,LSIZU2,LSIZGP,MINLU1,MINLU2,MINLGR,MINRU1,MINRU2,MINRGR,
5MINPU1,MINPU2,MINPGP,LSIZE,LREM,NPRCB,NC(1012,8),NB
522 COMMON/SFM/JCR(10),JCRD(10),PHR(10),MRSCH,JRSCH,KODE,K15
523 COMMON/MINIM/TOTMU1,TCTMU2,TCTMU3,TCTMGR,TOTMLE,TCTMR,TOTML,SCMU1,
1SCHU2,SCMU3,SCMGR,SCMLE,SCMF,SCML
524 IF (KCLNT.LT.MIN) GO TO 100
525 KCODEK=1
526 GO TO 200
527 100 KODEK=2
528 200 RETUPA
529 END

530 SUBROUTINE MINTEA(TFAKT,TCMCL1,TCMCL2,SCMCL1,SCMCL2)
531 IMPLICIT INTEGER*2(I*)
532 COMMON TEAKTR,TEAKTL,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLL1,FRCLU2,
1FRCLGR,FRCPU1,FRCP2,FRCPGR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
2TKPU1,TKPU2,TKPGR,TOTRU1,TOTRU2,TOTRGR,TOTLL1,TOTLU2,TOTLGR,TCTPU1
3,TOTPL2,TOTPGR,TCTTFA,RKOUNT,MCESSR,MREM,KA,KOUNT,KC,KR,KL,K13,

```



```

579      JCR(4)=JCR(4)+JCRD4
580      MHRDEF=MHRU1+MHRU2+MHRU3+MHRGR
581      JHRDEF=JCRD1+JCRD2+JCRD3+JCRD4
582      MHRSCH=MHRSCH+MHRDEP
583      JHRSCH=JHRSCH+JHRDEP
584      MSHRL1=MHRU1+MSHRU1
585      MSHRL2=MHRU2+MSHRU2
586      MSHRL3=MHRU3+MSHRU3
587      MSHRCR=MHRGR+MSHRGR
588      PRINT 1,TOTLEC,TCTMLE,TCTUG1,TCTMU1,JCRD1,MHRU1
589      PRINT 2,TOTRC,C,TCTMP,TOTUG2,TCTML2,JCRD2,MHRU2
590      PRINT 3,TOTLAB,TCTMI,TOTUG3,TOTML3,JCRD3,MHRU3
591      PRINT 4,TOTPRC,TCTGPA,TCTMGR,JCRD4,MHRGR
592      PRINT 5,TOTTEA,TCTGFL,JHRDEF,MHRDEF
593      PRINT 14
594      1 FORMAT(3X,'FTE RECITATION =',F6.2,1X,'***',F6.2,'***',3X,'FTE U-G-LE
      LEVEL1 =',F6.2,1X,'***',F6.2,'***',3X,'TOTAL UG1 CR-HRS =',I5,1X,'***
      2,I5,'***')
595      2 FORMAT(3X,'FTE RECITATION =',F6.2,1X,'***',F6.2,'***',3X,'FTE U-G-LE
      LEVEL2 =',F6.2,1X,'***',F6.2,'***',3X,'TOTAL UG2 CR-HRS =',I5,1X,'***
      2,I5,'***')
596      3 FORMAT(3X,'FTE LABATORY =',F6.2,1X,'***',F6.2,'***',3X,'FTE U-G-LE
      LEVEL3 =',F6.2,1X,'***',F6.2,'***',3X,'TOTAL UG3 CR-HRS =',I5,1X,'***
      2,I5,'***')
597      4 FORMAT(3X,'FTE PROBLEMS =',F6.2,1X,'***',F6.2,1X,'
      1***',F6.2,'***',3X,'TOTAL GRAD CR-HRS =',I5,1X,'***',I5,'***',/,130(1H-
      2))
598      5 FORMAT(3X,'FTE DEPARTMENT =',F6.2,1X,'***',F6.2,'***',39X,'TOTAL DEPT
      1 CR-HRS =',I5,1X,'***',I5,'***',/,130(1H-))
599      14 FORMAT(3X,'.ACTE.-COURSES WHICH DO NOT FULFIL THE MINIMUM COURSE R
      EQUIREMENTS AND THE RESULTING DELETIONS ARE ENCLOSED WITHIN ASTERI
      ZKS.')
600      RETURN
601      END

602      SUBROUTINE INIATL
603      IMPLICIT INTEGER*2(I)
604      COMMON TEAKTR,TEAKTI,TEAKTP,FRCRU1,FRCRU2,FRCRGR,FRCLU1,FRCLU2,
      1FRCLGR,FRCPU1,FRCPL2,FRCPGR,TKLU1,TKLU2,TKLGR,TKRU1,TKRU2,TKRGR,
      2TKPU1,TKPU2,TKPGP,TOTRU1,TOTRU2,TOTRGR,TOTLU1,TOTLU2,TOTLGR,TCTPU1
      3,TOTPL2,TOTPGR,TCTTFA,RKOUNT,PCESSR,PREM,KA,KCUNT,KC,KR,KL,K13,
      4LSIZU1,LSIZU2,LSIZGP,MINLL1,MINLL2,MINLGR,MINRU1,MINRU2,MINRCR,
      5MINPU1,MINPU2,MINPGP,LSIZE,LREM,NPROB,NC(1C12,8),NB
605      COMMON/RK/CCUCUR(30),TKRU3,TKLU3,TKPU3,TOTRU3,TOTLU3,TOTPU3,A,B,
      1MRU1,MRU2,MRU3,MRGR,LSIZU3,MINRU3,MINLL3,K12,K14
606      COMMON/GROSH/TKTEC,FRLEU1,FRLEU2,FRLEU3,FRLEGR,TOLEU1,TOLEU2,TOLE
      1U3,TCLFGR,TKLEU1,TKLEU2,TKLEU3,TKLEGR,LESIU1,LESIU2,LESIU3,LESIGR,
      2MILEU1,MILEU2,MILEU3,MILEGR,LESIZE,LEREM
607      COMMON/SFM/JCR(10),JCRD(10),MHR(10),MHRSCH,JHRSCH,KCCE,K15
608      COMMON/MINIM/TOTMU1,TOTMU2,TCTMU3,TCTMGR,TCTMLE,TOTNR,TOTML,SCMU1,
      1SCMU2,SCMU3,SCMGR,SCMLE,SCMP,SCPL
609      TOLEU1=0
610      TOLEU2=0
611      TOLEU3=0
612      TOLEGR=0
613      TCTRU1=0
614      TCTRU2=0
615      TOTRU3=C.0
616      TOTRGR=0
617      TCTLU1=0.0

```

```
618      TOTL2=0.
619      TOTL3=0.0
620      TCTLER=0.
621      TOTPL1=0.0
622      TOTPL2=0.0
623      TOTPL3=0.0
624      TCTPCR=0.0
625      TOTTEA=0.0
626      RKOUNT=C.0
627      CO 25C JA=1,10
628      MHR(JA)=0
629      25C JCRD(JA)=0
630      TOTML1=0
631      TOTML2=0
632      TOTML3=0
633      TOTMCR=0
634      TOTMLE=0
635      TOTMR=0
636      TOTML=C
637      MHRU1=C
638      MHRU2=C
639      MHRU3=0
640      MHRGR=0
641      RETURN
642      END
```

\$ENTRY

	DESIRED SECTION SIZE				MINIMUM SECTION SIZE			
	CCC-355	400-599	600-799	ABCVE800	CCC-355	400-599	600-799	ABCVE800
LECTURE	1000	1000	1000	1000	40	40	40	40
RECITATION	30	25	20	15	20	15	10	5
LABORATORY	20	10	10	5	10	8	5	3

	WEIGHTING FACTOR				PREPERATION LEAD FACTOR			
	CCC-355	400-599	600-799	ABCVE800	CCC-355	400-599	600-799	ABCVE800
LECTURE	1.00	1.50	1.50	4.00	1.00	1.10	1.30	1.50
RECITATION	1.00	1.50	1.50	4.00	1.00	1.10	1.30	1.50
LABORATORY	2.00	3.00	3.00	4.00	1.00	1.10	1.30	1.50
PROBLEMS	3.00	4.50	4.50	12.00				

FRACTIONAL LOAD FOR PREPRATION= 0.70 FRACTIONAL VARYING LOAD= 0.30 BASE STUDENT COUNT= 32.00

NUMBER OF DEPARTMENT= 9 FIRST COURSE NUMBER READ=505115 METHOD NO= 2 FTE TEACHING LOAD FOR PROBLEMS= 300.0 FTE CREDIT HOURS= 9.0

SEMESTER= 2 *1* REPRESENTS FALL AND *2* REPRESENTS SPRING

 STUDENT ENROLLMENT DATA

CUR	L1	L2	L3	L4	L5	L6	L7	L8	R-M
42	0	4	C	14	0	20	0	21	0
75	0	10	0	15	0	37	0	25	0
820	0	50	C	0	0	0	0	0	0
873	0	2	C	0	0	0	0	0	0
862	0	4	C	0	0	0	0	0	0
910	0	4	0	0	0	0	0	0	0
506	0	0	C	0	0	0	0	0	0
1000	0	285	C	301	0	331	0	255	0
1001	0	127	0	140	0	120	0	154	0
1002	0	1853	C	1402	0	927	0	578	0
1003	0	272	C	260	0	235	0	205	0
1004	0	0	C	82	0	350	0	253	0
1005	0	285	0	201	0	243	0	263	0
1006	0	393	0	354	0	335	0	210	0
1008	0	1420	C	0	0	0	0	0	0
1009	0	576	C	0	0	0	0	0	0
1010	0	181	0	0	0	0	0	0	0
1011	0	68	C	0	0	0	0	0	0
500	0	0	C	0	0	0	0	0	0
505	0	12	C	5	0	12	0	14	0
520	0	24	0	24	0	28	0	21	0
525	0	40	C	24	0	35	0	48	0
530	0	66	C	61	0	71	0	71	0
550	0	8	0	15	0	24	0	23	0
580	C	29	C	18	0	15	0	21	0
560	0	39	C	53	0	57	0	65	0
872	0	21	0	0	0	0	0	0	0
927	0	8	0	0	0	0	0	0	0
879	0	21	C	0	0	0	0	0	0
948	0	20	C	0	0	0	0	0	0
860	0	12	0	0	0	0	0	0	0
870	0	13	C	0	0	0	0	0	0
921	0	15	C	0	0	0	0	0	0
868	0	29	C	0	0	0	0	0	0
878	0	26	0	0	0	0	0	0	0
944	0	22	0	0	0	0	0	0	0
877	0	53	C	0	0	0	0	0	0
977	0	3	C	0	0	0	0	0	1

FINAL FTE TEACHING LOAD FOR THE SCHCL LACER CONSIDERATION

 FTE RECITATION = 1.86 ** 0.23** FTE L.G.LEVEL1 = 18.64 ** 1.08** TOTAL UG1 CR.HRS = 2123 ** 105**
 FTE RECITATION = 45.97 ** 9.62** FTE U.G.LEVEL2 = 45.89 ** 3.07** TOTAL UG2 CR.HRS = 5117 ** 185**
 FTE LABORATORY = 42.67 ** 1.90** FTE L.G.LEVEL3 = 13.71 ** 4.84** TOTAL UG3 CR.HRS = 1253 ** 234**
 FTE PROBLEMS = 15.62 FTE GRADUATE = 31.85 ** 2.76** TOTAL GRAC CR.HRS = 1251 ** 75**

FTE SCHCL = 110 ** 12** TOTAL SCH. CR.HRS = 9744 ** 595**

NOTE--COURSES WHICH DO NOT FULFIL THE MINIMUM COURSE REQUIREMENTS AND THE RESULTING DELETIONS ARE ENCLCSEC WITHIN ASTERISKS.

 CCRE USAGE OBJECT CODE= 42392 BYTES,ARRAY AREA= 17876 BYTES,TOTAL AREA AVAILABLE= 180352 BYTES
 DIAGNOSTICS ALPBER OF ERRORS= C, NUMBER OF WARNINGS= 0, NUMBER OF EXTENSICNS= 0
 CCNPILE TIME= 49.98 SEC, EXECUTION TIME= 170.92 SEC, NATFIV - VERSION 1 LEVEL 3 MARCH 1971 DATE= 72/122

APPENDIX G

SAMPLE COMPUTER OUTPUT

G1 Sample output 1

G1.1 In put variables

G1.1.1 Section size specification

	<u>DESIRED SECTION SIZE</u>				<u>MINIMUM SECTION SIZE</u>			
	000-399	400-599	600-799	>800	000-399	400-599	600-799	>800
Lecture	1000	1000	1000	1000	40	40	40	40
Recitation	30	25	20	15	20	15	10	5
Labaratory	20	10	10	5	10	8	5	3

G1.1.2 Weight Factors

	<u>WEIGHTING FACTOR</u>				<u>PREPARATION LOAD FACTOR</u>			
	000-399	400-599	600-799	>800	000-399	400-599	600-799	>800
Lecture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Recitation	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Labaratory	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Problems	1.00	1.00	1.00	1.00				

 Fractional load for preparation = 0.70

Fractional load varying load = 0.30

Base student count = 32.00

Method of calculation = 2

Base fulltime credit hour = 9.00

FTE student credit hour for
problems = 30.00

Semester = 2

G1.2 Teaching requirement for school

FTE Lecture = 1.86 ** 0.23** FTE U.G.Level 1 = 16.55** 1.06** Total UG1 Cr.Hrs = 2123** 105**
 FTE Recitation=35.05** 7.27** FTE U.G.Level 2 = 35.86** 2.58** Total UG2 Cr.Hrs = 5117** 185**
 FTE Labaratory=31.82** 1.52** FTE U.G.Level 3 = 12.22** 3.67** Total UG3 Cr.Hrs = 1253** 234**
 FTE Problems =18.67 FTE Graduate = 22.78** 1.71** Total GradCr.Hrs = 1251** 75**

FTE School = 87 ** 9**

Total Sch. Cr. Hrs = 9744**599**

APPENDIX G

SAMPLE COMPUTER OUTPUT

G2 Sample output 2

G2.1 Input variables

G2.1.1 Section size specification

	<u>DESIRED SECTION SIZE</u>				<u>MINIMUM SECTION SIZE</u>			
	000-399	400-599	600-799	>800	000-399	400-599	600-799	>800
Lecture	1000	1000	1000	1000	40	40	40	40
Recitation	30	25	20	15	20	15	10	5
Labaratory	20	10	10	5	10	8	5	3

G2.1.2 Weight Factors

	<u>WEIGHTING FACTOR</u>				<u>PREPARATION LOAD FACTOR</u>			
	000-399	400-599	600-799	>800	000-399	400-599	600-799	>800
Lecture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Recitation	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Labaratory	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Problems	1.00	1.00	1.00	1.00				

Fractional load for preparation = 1.00
 Fractional load varying load = 0.00
 Base student count =32.00
 Method of calculation = 2
 Base fulltime credit hour = 9.00
 FTE student credit hour for prob. =30.00
 Semester = 2

G2.2 Teaching requirement for school

FTE Lecture	=	1.22**	0.22**	FTE U.G.Level 1	=	17.33**	1.22**	Total UG1	Cr.Hrs	=	2123**	105**
FTE Recitation	=	38.89**	9.56**	FTE U.G.Level 2	=	40.93**	3.33**	Total UG2	Cr.Hrs	=	5117**	185**
FTE Labaratory	=	38.33**	2.00**	FTE U.G.Level 3	=	13.99**	4.89**	Total UG3	Cr.Hrs	=	1253**	234**
FTE Problems	=	18.67		FTE Graduate	=	24.86**	2.33**	Total Grad	Cr.Hrs	=	1251**	75**

FTE School	=	97**	12**	Total Sch.	Cr.Hrs	=	9744**	599**
------------	---	------	------	------------	--------	---	--------	-------

REFERENCES

- Compton, T. E., An inquiry with the problem of predicting university classroom requirements, A Master's Thesis, Industrial Engineering Department, Kansas State University, 1972.
- Hoyt, D. P., Staffing at Kansas State University Office of Educational Research, Research Report No. 2, Kansas State University, 1970.

ACKNOWLEDGEMENTS

The author wishes to acknowledge to Dr. L. E. Grosh and Dr. Ralph H. Nevin for their valuable assistance and guidance.

Thanks are due to Mrs. Bruce Peil for her help in the typing and composition of this report.

Finally the author wishes to acknowledge the assistance of Dr. Doris Grosh for her suggestions and corrections.

COMPUTER MODEL FOR ESTIMATION OF TEACHING FACULTY REQUIREMENT

by

KRISHNASWAMY RANGANATHAN

M. Tech (Agricultural Engineering)

Indian Institute of Technology, Kharagpur, 1963

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1972

ABSTRACT

In this report an attempt has been made to develop a working computer model for estimation of the teaching faculty requirement at Kansas State University. Historical data for 1970-71 and the Engineering school curricula for the semesters were considered in developing and validating this model.

The model incorporates a method of estimating student count in a given course, a decision rule for deciding the number of sections and two methods for estimating the teaching faculty requirements. While estimating the student count, it was noticed that the students do not strictly adhere to the curriculum. A modifying factor to take care of this deviation from regularity has been discussed with reference to a decision criterion chosen.