A METHOD FOR JUDGING AND IMPROVING SYSTEM PERFORMANCE OF A COMPUTER INSTALLATION.

A Thesis presented to the Faculty of Science of the University of Manitoba

---

In partial fulfillment of the requirements for the Degree Master of Science in Computer Science

---

by

WALTER H. DORAN

March, 1971.

© Walter H. Doran 1972

ACKNOWLEDGEMENTS

The author feels indebted for assistance and support to :

1) The Computer Center at the University of Manitoba for the use of computer facilities for experimenting.

2) Professor J.M.Wells of Computer Science for his acceptance of the topic and for his continuing encouragement.

3) Professor J.C.Muzio of Computer Science, thesis advisor, for his supervision, interest, advice and encouragement.

4) Professors W.J.Craddock and M.H.Yeh of Agricultural Economics for their review.

- A B S T R A C T -

This thesis proposes methods for assessing the performance of a computer installation and for improving the performance of that installation.

Performance is interpreted as the quotient of useful work performed, by the system potential for work. Each working hour is assessed separately, from accounting information logged for each job, for channel , CPU, storage and overall system performance. Results are combined and presented as separate three-dimensional plots of each activity for a period of any chosen time-span.

The method proposed for improving system performance operates on the input job queue. It selects a feasible dispatch pattern and then proceeds to improve on the feasible pattern. The measure of improvement is a reduction of resources waste. This method uses, as data, resource requirements estimates submitted by the program author.

A METHOD FOR JUDGING AND IMPROVING SYSTEM PERFORMANCE OF A COMPUTER INSTALLATION.

Walter Doran

University of Manitoba.

TABLE OF CONTENTS.

Chapter 1

PREAMBLE

A. Objects of the study.

B. Justification for the study.

C. Review of existing methods.

D. Choice of method.

Chapter 2

MEASURING FACILITIES

A. Variables.

B. Sensors.

C. Reliability of assessment of variables.

D. Selection of variables as parameters.

Unit cost of 6 activities.

Unit revenues.

Chapter 3

REPORTING AND ASSESSING

A. Assignment of independent variable for reporting and judging.

B. Assignment of dependent variables for reporting and judging.

C. Measurement of system activity.

D. Assessment of system activity.

E. Method of reporting and interpretation of the reports.

F. Interpretation of recent performance.

Chapter 4

SCHEDULING FOR IMPROVED PERFORMANCE

A. Definition of scheduling.

B. Discussion of a scheduling rationale.

C. Assignment of independent variable for scheduling.

D. Assignment of dependent variables for scheduling.

E. Derivation of a scheduling rationale.

F. Illustration of the family planning rationale.

G. Selection of rationale.

Chapter 5

PROPOSED EXTENSIONS TO THE STUDY

Chapter 6

SUMMARY AND CONCLUSIONS

INDEX OF ILLUSTRATIONS

ITEM	DESCRIPTION	FOLLOWING PAGE
Table A	Schematic of Revenue and Efficiency Arrays	44, 50
Table 3	Hourly activity of computer	55
Table 4a	CPU efficiencies	57
Table 4b	CPU efficiencies, frequency	58
Table 4c	CPU efficiencies, cumulative frequency	59
Table 4d	CPU efficiencies, percentage of days per decile per hour	59
Table 4e	CPU efficiency, profile	60
Table 4f	CPU efficiency, block	61
Table 4g	CPU efficiency, smoothed profile	61
Table 3a	Frequency (days/decile/hour)	62
Table 3b	Cumulative frequency (days/decile/hour)	62

Table 3c	Percentage of days/decile/hour - Plot	62
Plate 5a	Channels efficiency - Plot	69
Plate 5b	CPU efficiency - Plot	69
Plate 5c	Core efficiency - Plot	69
Plate 5d	System efficiency - Plot	69
Plate 4c-0	Three-dimensional plot of resources	78
Plate 4c-1	CPU job - resource waste	81
Plate 4c-2	CPU and channel job - resource waste	81
Plate 4c-3	Multiple tasking - resource waste	82
Table 7	Data with some interpretation	95
Table 8	Job stream in order of kbseconds	96
Table 9	Job stream in order of vector angle	96
Table 10	Job requirements in order of vector angle	96
Table 11	Family structuring	96
Plate 1	CPU, Channel, and Core interaction	99
Plate 2	CPU, Channels, and Core interaction	104
Plate 3	Skeleton model	114
Plate 4	A family mix	115
Plate 5	A family mix	116

Appendix 1	Program MKCC	develops unit costs
Appendix 2	Program RESEVAL	develops hourly revenues and efficiencies
Appendix 3	Program MOREV	develops frequency, cumulative frequency and percent of days/decile/hr for plotting equi-efficiency contours
Appendix 4	Program GRAPH	produces equi-efficiency contour plots
Appendix 5	Program WASTE	develops family groups for dispatch

Chapter 1.

P R E A M B L E .

1 A. OBJECTS OF THE STUDY.

There are four main purposes to this study:

1) To describe a computer installation in terms of variables which, when properly inter-related, describe the system's performance and reflect the effect on system performance of modifications to equipment configuration or modifications to operating procedures. The description should be

capable of testing the efficacy of modifications which might be introduced,with the intention of improving some aspect of the system's performance.

2) To identify the most significant systems variables; to select from these an independent variable which will describe system performance and reflect the effect of the remaining significant variables as dependent variables. Thus, efficiency becomes the independent variable. It represents the quotient of useful work performed by the total potential for work. In this context work is dimensionless : device activity in minutes / time period in minutes.

3) To derive a method of reporting system activity in terms of the independent variable.

4) To derive a scheduling procedure which will

select some arbitrary dispatch order and proceed to a more efficient schedule by modifying the dispatch order to reduce waste (to reduce idle equipment time).

1 B. JUSTIFICATION FOR THE STUDY.

One attractive goal in the management of a process is to improve the efficiency of that process. In some processes the immediate savings resulting from improved efficiency may be only marginal : a reduction in operating costs resulting from reduced operating hours. But,if the process load is increasing,an improvement in efficiency may lead to significant savings in capital expenditure; if an enlargement of plant can be delayed, or circumvented permanently, the full efficiency improvement may be realized in dollar savings.

Computing load is increasing throughout the whole industrial society. Since 1957, the load has increased exponentially. Perhaps the rate of increase is beginning to decline but even if it

slows to a mere linear increase, additions to plant seem inevitable. This report attempts to provide management with some simple tools for assessing the basic efficiency of plant and for assessing the effect on efficiency of variations in operating procedures. Hopefully these simple tools will help to rationalize decisions concerning plant expansion.

The attention of this treatise is directed to the in-line processing type of computer, specifically the family of IBM 360 digital computers using IBM's full operating system (OS) with multiprogramming with a variable number of tasks (MVT). The approach is general, but would need to be modified to suit the measurable activities of another machine if adaptation to another family of computers were required.

This study includes :

1) An examination of variables affecting plant throughput .

2) An examination of the variables which are measurable and an exposition of efficiency .

3) A method of assessing plant efficiency statistically and through graphical or visual display.

4) A survey of philosophies of operating procedures in general use and a critique of these procedures.

5) A proposal for deciding upon a rational operating procedure.

6) A proposal for an operating rationale which should logically improve efficiency.

The Rand Corporation reported ([8]) experiments they performed in an attempt to develop an equation for throughput as a function of work load using regression analysis and cluster analysis of data derived from job-account cards. They used as a unit of time a work day from 9.00 a.m. to 5.00 p.m.. By this choice they restricted themselves to those days when there were no significant machine problems. Their results were inconclusive, partly because the data were sparse. This author suggests that an hour would be a better unit of time; it would increase their data points 20-fold. This work is cited as evidence of the industry's continuing interest in performance evaluation.

An indication of the industry's concern about the subject is contained in an editorial in Computerworld November 25, 1970, p.4 in review of ([2]) : "SPIES ARE NEEDED. HOUSTON - Any manufacturer today can be shown in a harsh light by a non-partial user. Monitors should be used to give us feedback to the manufacturers. This was the conclusion of a paper, " Spy " at a session on Operating Systems and Schedules. The authors developed a monitor for OS/360 and, as one of the critics on the panel said, it showed that users tend to overestimate their core storage needs. While the authors did not suggest modifications, their presentation indicated some modification was needed. A major problem facing the manufacturer of large-scale computer systems today is that of measuring the performance of a computer in conjunction with the operating system which drives it. The problem is under study for operating

systems currently in the design and implementation stages. The goal is to establish reasonable criteria for measuring performance, to find means of actually making the measurements, and ways of using the information to optimize performance for a particular user's environment."

1 C. REVIEW OF EXISTING METHODS.

There exists no industry-wide standard for assessing or measuring improvement or deterioration in plant performance. Development of such a standard seems to this author to be essential to an orderly discussion and investigation of the effect of modifications to a plant. What good reconfiguring a plant for improved efficiency if the rearrangement effects a lower efficiency? But what will prevent such a deterioration unless some standard of 'goodness' is available? Hence the need for a definition and measure of partial and overall efficiencies.

Because plant performance in a computer installation is transient any measure of efficiency is valid for only a short period of time. It is difficult to assimilate a

comprehensive view of these transient variables, hence the desire for a visual display of activity.

Because operating philosophies and particularly order-of-dispatch philosophies are emotional, it is the opinion of this investigator that the industry should seek a rational dispatch theory to use as a base from which to deviate for emotional or expedient considerations. There is no presumption contained herein that the dispatch algorithm presented is optimal, but rather that it is a rational method for improving performance. This report proposes the argument that a plant manager should be free to set job priorities according to his interpretation of the needs of his clients, but that he will be able to manage his plant better if he is afforded an efficient, feasible schedule of job priorities and a dollar assessment of the loss accruing from satisfying

his clients' requests for deviations from the efficient schedule. This treatise proposes a method for measuring additional waste accruing from deviations from the computed efficient schedule.

The following list shows some of the philosophies used by various computer installations for deciding an order for dispatching jobs:

1) Some computer installations use a first-come first-served philosophy.

2) Some use a usurping or ranking procedure, wherein a job's priority is a function of the importance of its sponsor.

3) Some use a predictable turnaround procedure, wherein the completion time of each job is

predicted upon receipt, with a liberal contingency factor. This system could accommodate a resource levelling algorithm.

4) Some use a latest-completion-time argument for each job. This procedure can be necessary in a commercial shop. It may be practical, while observing some optimization of dispatch. This method presumes that someone schedules the arrival of jobs relative to their latest-completion arguments.

5) Some installations allot blocks of time to classes of users, with non-scheduled users obtaining access only if the sheduled users are leaving parts of the facility idle.

6) Some set priorities to the inverse ratio of predicted job time. Some of these upgrade

priority gradually with time in the input queue, to prevent a long job being held indefinitely.

This investigation recognizes how these methods evolved but proposes that none of them be used blindly without first comparing its effect with the effect of a dispatch theory which has logically reduced waste. For the sceptic, the investigation says : " Let us draw a picture of the results of the existing operation, and then modify the dispatch by some attractive rationale and draw a picture of the response to the new procedure. Then let us compare pictures."

1 D. CHOICE OF METHOD.

Examination of system performance each hour over a period of six months indicates that there is no cyclical pattern which would forecast what the performance will be at, say,5 p.m. on a Tuesday. One might suspect that performance would be stable and predictable when the job queue was long, for example, and indeed, contrary to intuition, there is a pattern which indicates that the system performance is poorer when the job queue is long than when it is short, under a specific scheduling scheme. The method reports performance adequately but to date has not been able to forecast system performance except as it relates to the input job stream queue.

If a cyclical pattern were apparent,assessment of a policy change would be much easier : the change

could be introduced during the hours when performance was predictable so trends could be identified quickly.

This observer concluded that if the performance could not be forecast by extrapolating long-term trends of thousands of tasks, performance certainly could not be assessed through a few jobs selected arbitrarily as bench-marks.

There was a hope, early in the study, that the value of the scheduling algorithm might be tested by scheduling a few minutes of the job stream in batch, submitting the jobs pre-scheduled by appropriate priority assignment, and comparing results with forecast results. But if the forecast is unreliable, any comparison would be unreliable.

The choice then is to observe long-range effects

of change. The methods developed and reported herein are intended to aid in the long-term pursuit.

Chapter 2.

MEASURING FACILITIES.

2 A. VARIABLES.

It appears to this observer that the reason the computer community has not derived a simple rationale for measuring, reporting, and controlling system performance is that the variables which control performance appear to be multitudinous and that their interrelationships appear difficult to determine. These variables include, among others :

1) The task function : link edit, execute, compile : Fortran, or Cobol or Assembler, or Algol, or etc. Attending each are size and time variables.

2) Makeup of input queue and size of input queue.

3) Availability of core for problem programs.

4) Channel activity for data transfers between core and disk, drum, or tape.

5) Access method for storage and DCB parameters.

6) Disk arm travel and disk contention.

7) Peripheral input and output requirements.

8) Artificial priority constraints.

9) Time of day, week, month and year.

10) Order of arrival of jobs.

11) Order of initiation to core.

12) Order of servicing in core.

13) Hardware configuration.

14) Equipment costs and relative values.

15) Equipment shortages and surpluses.

16) Task source : from interactive job stream, from a compiler monitor, from the batch or from a terminal monitor.

17) Job costs.

18) System efficiency.

19) Program efficiency : compilers, operating systems and problem programs.

Only a few of these variables can be controlled by operating decisions; some are controlled by the whims of the users while the rest are outside the control of the installation or the user. The operations manager usually does not have authority to dictate when jobs will be submitted or how programs will be written. He may recommend procedures but in an installation serving research and teaching he cannot exercise much control over the users' schedules. But he can control some of the variables and he will hopefully identify those

which are controllable and learn to identify and evaluate their effect on system performance.

Among those things which are controllable are :

1) The time a job is held in the input queue.

2) The order of passing jobs from input queue to core for processing.

3) The amount of system resources assigned to the job after it reaches the execution stage. This will be controlled by time-slicing, by disk and tape mounts, by default resource limits, by job termination with or without checkpoint and by its importance relative to other jobs with which it shares core.

4) The amount of system resource wasted by mismatching jobs. This item may not be entirely controllable but it is a contention of this report

that it is a most significant factor .

Among those variables which are not controllable but are measureable and significant in their measurement are :

1) Region requirements.

2) User's estimate of central processor and channel requirements.

3) Recorded CPU and channel use along with other job accounting statistics.

2 B. SENSORS.

The computer on which this study was conducted has an interval timer which is activated by cycling of the alternating current supply. Its resolution is therefore 1/60 th of a second. The computer's operating system makes available cumulative CPU time for each task as accurately as it can be recorded by such a gross sensor.

Swoyer and Armstrong ([1]), among others, report that " it is possible for a fortunate user to use hours of CPU time and still be clocked zero time if he computes in intervals smaller than 16.6 milliseconds and hits the inter-incrementing gaps. It is also possible to compute 1.3 microseconds and be clocked 16.6 milliseconds if he is unlucky. Experience shows that in the long run, luck smiles

on all users royally, and discrepancies are very small on the order of 2 % at most." It should be noted also that the CPU user will have clocked against him, the CPU cycles stolen for I/O activity while he is using the CPU.

There is a more sensitive clock available, but its cost has not yet been justified for most installations.

There are many measuring instruments available for logging system activity on a short term basis, but none which assesses long-term activities. IBM offers a Systems measurement Instrument Service, for a fee of $360/day, to assist the user to identify imbalances in his hardware configuration.

Allied Computer technology offers an instrument with tape output called CPMII for measuring CPU

wait time (waiting for completion of an I/O request) and channel busy time.

Computer Synectics market an instrument with tape output called SUM for measuring CPU wait and channel busy time.

Computer Programming and Analysts market a device called CPA 7700 for similar measurements, but only via manual readout.

Boole and Babbage market a software package called SMS/360 CUE for measuring direct access device activity, and storage assignment within OS/360. They claim too to be able to measure code efficiency in the problem programs.

These short term measurements might be better than the job accounting data used in this study,

but they tend to be expensive. When their cost can be justified, their results should enhance the long range evaluation proposed here.

Though the industry is conscious of the need for systematic assessment, it is still struggling with a definition of its needs. Evidence of this search for direction can be found in the few papers written on the subject and in views expressed at seminars. For example Shope Kashmarak, Ingram and Decker ([7]) report an extensive evaluation study undertaken by the University of Iowa and IBM. It stressed the need for methodology and local system performance evaluation but was unable to recommend methodology or evaluation techniques for others to follow. This is not a criticism of the work done at Iowa but rather a recognition of the immaturity of the art. Also this is not an isolated item, but

was chosen for citation because it is such a thouroughly documented report of several years of serious pioneering work by the industry's keen young men.

Western Electric Engineering Research Centre has developed a monitoring system called SPY ([2]) which displays system performance in dynamic graphical display. It measures core use, CPU activity, and disk activity. Unfortunately it is not available to users who do not have graphics equipment.

None of these devices look to long range assessment. All look at performance in a short time interval; i.e., seconds or minutes. When comparative studies are required, investigators such as Shope ([7]) and Lasser ([5]) depend on a library of a few programs to use as a bench-mark of performance before a system change and compare these bench-mark results with performance of the

revised system on the same job stream. This investigator contends that the bench-mark method is inadequate in a university computer centre, where the job stream varies so significantly . It seems irrational to use such a crutch as bench-marks for assessing an operating plant if job accounting data are available.

Most installations gather accounting statistics through Hasp, System Management Function, IBM's Operating System or some other software or hardware logging device. Most have sufficient data for rational interpretation but no indication could be found in American Federation of Information Processing, Association for Computing Machinery, Proceedings of Share, or Computing Reviews of any methodology for reporting long range trends in system performance.

2 C.RELIABILITY OF ASSESSMENT OF VARIABLES.

The literature suggests ([6]) also that the resolution of the timer is not small enough to accurately measure task data transfer time via channels, and proposes instead to keep cumulative count of the number of channel requests issued by each task and to convert this count to time by assuming the average transfer time to be .025 seconds.([6])

The region request made by the user, or the default region, is an accurate sensor of core region used because it measures the region assigned exclusively to a task.

The user's estimate of the time his job will occupy core is implied by his declaration of CPU time and channel counts. The estimates are not

accurate but are known to be an upper limit, because the job will be abandoned when the estimated time has been expended. The estimates tend to be high but an adjustment can be made, through experience, when matching jobs to available resources. If the user does not bother to supply estimates of core and channel times the operating system will provide default values. These also will tend to be high, but a downward shift can be introduced to compensate, when matching jobs to available resources.

2 D. SELECTION OF VARIABLES AS PARAMETERS.

The CPU time used and reported by job accounting can be interpreted as a reliable measure of CPU efficiency. Similarly,an approximation of channel and core efficiencies can be derived from job accounting statistics.

The accounting of peripheral activity; such as cards read and lines printed is accurate, but the controllable part of this activity has little effect on system efficiency. However the allocation of cost to this activity is significant in establishing units of work.

Six distinct activities can be measured or estimated in a computer installation. These six were chosen as the control variables in this study :

1) CPU time used.

2) Channel activity.

3) Core used.

4) Cards read.

5) Lines printed.

6) Cards punched.

Six matching units of work were defined. These might be considered constants, though they vary with changes in system configuration and with changes in the volume of work being processed. These units of work are concurrently cost and revenue. The units were derived by distributing all items of capital cost plus recurring operating

costs across the six items cited above by some sensible balance. For example the total cost of printers and their controllers and their paper stock was assigned to printing. But some of the cost of disks and channels was distributed to printing because disk storage is used for queueing printer output and channel activity is required to transfer output from core to spool and from disk to printer.

Costs were reduced to a common denominator of annual expenditure; rental charges were in that form, but purchased items were amortized over four years to get annual expenditure.

System activity reports from the past supplied annual activity figures; such as, lines printed per year, CPU hours used per year, kilo-byte hours of core used per year, and channel counts used per year. Unit costs of the six work items were

derived from total annual cost attributed to the work item divided by the annual activity of that work item. These quotients provide a relative measure of work which is common to all activities. Call the unit dollars or work units,or unit cost,or unit revenue; regardless of name,the appropriate units multiplied by a job's activities or the system's activities over an hour gives a relative measure of system efficiency. A detailed description of the method will be found in Chapter 3 item C. Units costs were derived by program MKCC, shown in Appendix 1.

Chapter 3.

REPORTING AND ASSESSING.

3 A. ASSIGNMENT OF INDEPENDENT VARIABLE FOR REPORTING AND JUDGING.

Plant efficiency was chosen as the independent variable which might best reflect the system operation. A detailed definition of efficiency will be found in paragraph 3D. In brief review efficiency might be envisioned as the quotient of work done, by the maximum capability for work. Card input, printer output, and card output were excluded from the model because :

1) Job submission at the card readers is not controllable by the operations staff. Jobs are read onto spool at the whim of the user.

2) Printing and punching times are controllable and controlled by management decision but these activities are more or less independent of the multi-tasking features of the computer and have little effect on other system activities; i.e. the order chosen for these activities has no effect on the amount of system resource wasted. Efficiencies were computed and reported for each hour of operation for CPU activity, channel activity, and core- storage utilization. In addition an overall plant efficiency was computed and reported for each hour.

3 B. ASSIGNMENT OF DEPENDENT VARIABLES FOR REPORTING AND JUDGING.

Accounting statistics which are logged for each batch job include :

1) The number of job steps.

2) A cumulative count of channel requests, called here I/O counts, one for each time a data transfer is initiated. Please refer to paragraph 2C for a discussion of the validity of this measurement.

3) The cumulative central processing time used, within the limit of recording accuracy.

4) The assigned priority.

5) The time of day the job was submitted to spool.

6) The time of day the job left the output spool.

7) The number of cards read.

8) The number of lines printed.

9) The number of cards punched.

10) The execution time; i.e., elapsed time from job- initiation to job-end, or the time the job was in storage in core.

11) The channel activity associated with job initiation.

12) The time-of-day the job entered core storage.

13) Job core usage. The accounting procedure measures for each job step the region size requested (or supplied by default) and the job step time; i.e., (CPU seconds used) plus (.0025 x channel requests) seconds. The accounting procedure accumulates the product of these two figures for each job step, and reports the sum of the products in kilobyte-seconds when the job is closed. This data logging and computation is defined by the equation :

$$\text{Job core usage (kilobyte-sec.)} = \sum_{\text{step}=1}^{n} (\text{step region}) \times (\text{step time})$$

This study takes no credit for the routine which logs and reports job accounting statistics. IBM's operating system permits accounting procedures to be attached. The accounting procedures were

installed by Computer Centre personnel at the University of Manitoba. The accounting procedure punches one accounting card for each job. The study used these cards as one source of data .

Items 2 + 11, 3, 12, and 13 were chosen as significant variables for reporting and judging system management. These items are listed again for clarity. Item 10 was used for prorating long jobs : see page 44 for the application.

Items 2 + 11 : channel requests per job, each.

Item 10 : execution time; i.e., time in central memory.

Item 3 : CPU time used per job, seconds.

Item 12 : Time of day the job entered main

storage, clock time.

Item 13 : job core usage, kilobyte-seconds.

Item 1, the number of job steps above, was considered unworthy as a significant variable for judging system performance because it cannot be correlated with the other variables at job initiation time; there is no separate estimate in the job card of resource requirements by job step.

Item 4, the assigned priority, was discarded as a governing variable, because it is assigned by an algorithm which considers the four significant variables listed above; i.e., it is a dependent function of 2, 3, 11, and 13, so can be ignored.

Items 5 and 6, the times the job reached input spool and left output spool, were not used by the reporting program because they are not a function of the job they attend; the computer views

variable 5 as random and variable 6 as dependent upon 2, 3, 11, 12, and 13 and therefore not worthy of separate reporting.

Items 7, 8, and 9, cards and lines are not controllable by operating decision, and they do not have any significant effect on internal queues, except as they govern initiation times. They should be considered in forecasting channel requirements, but the users' methods of forecasting are not sophisticated enough to warrant special attention for peripherals.

3 C. MEASUREMENT OF SYSTEM ACTIVITY.

The program RESEVAL, to be found in appendix 2, was developed to use the accounting data from each job to report for each working hour the separate revenues derived from CPU, channel, and core. RESEVAL reports this activity by a separate page for each day. For purposes of exposition , Array A, shown on Table A, represents the result of n days output . The incremental values to be added to the array for each job are computed from :

Δ REVENUE$_{I,J,K}$ = Δ ACTIVITY$_J$ x UNIT COST$_J$, where

J refers to the column selection;

K is defined by the date; and

I is the hour the activity was performed.

TABLE A.

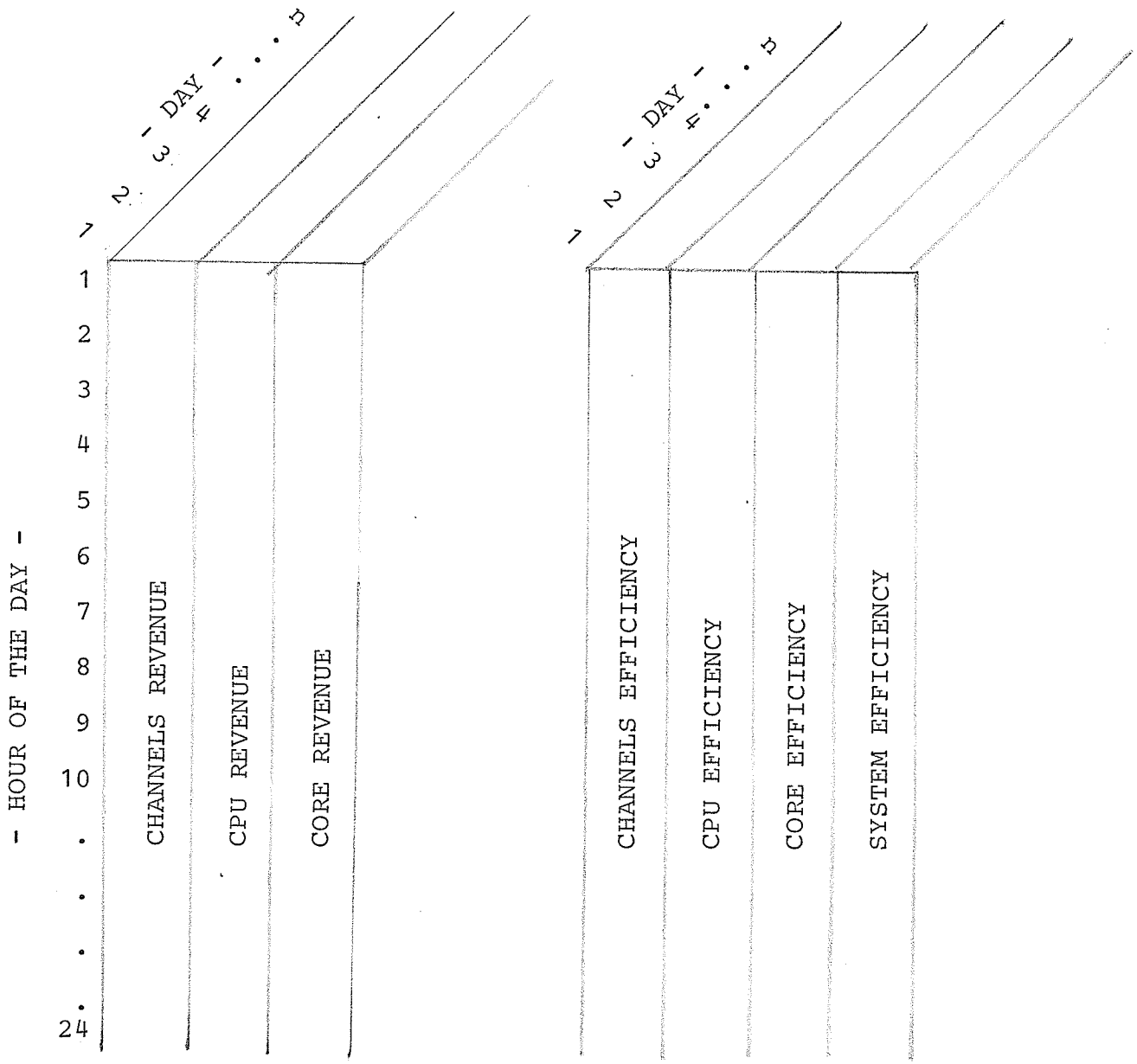
Dimension A (24,3,n) , B (24,4,n)
A (I,J,K) , B (I,J,K)

Array A

Activity revenue

Array B

Activity efficiency



(Please see Chapter 2, section 2D for derivation of unit costs). Item 12 gives start time; item 12 + item 10 gives end time. A routine was included to prorate each revenue item over the correct 'I' slots if (item 12) and (item 12 + item 10) did not fall within the same clock hour. Many jobs occupy part of two separate hours; some bridge 3, 4, 5 ... up to 24 hours.

An example may lead to easier understanding. Assuming unit costs of :

$ 2.50 / CPU-minute

$ .0015 / channel count

$ .014166 / kilobyte-minute

and job statistics for one job as :

Start time : 10.15 o'clock

Exec.time : 3.00 hrs.

CPU time : 20 minutes

Channel requests: 800

Core use : 600 kilobyte-minutes.

Compute :

End time : 10.15 + 3.00 = 13.15

CPU revenue : 20 x 2.5 = $ 50

Channel revenue : 800 x .0015 = $ 1.2000

Core revenue : 600 x .014166= $ 8.4996

These revenues must be distributed over 12 cells of array A, because the time span covers four hour- intervals. The ratios of distribution are :

10.00-11.00	11.00-12.00	12.00-13.00	13.00-14.00
3/12	4/12	4/12	1/12

Thus, increments added to array A will be,(assuming this event occured in day 1),

CELL	REVENUE
A (11,1,1)	3/12 x 1.20
A (11,2,1)	3/12 x 50.00
A (11,3,1)	3/12 x 8.50
A (12,1,1)	4/12 x 1.20
A (12,2,1)	4/12 x 50.00
A (12,3,1)	4/12 x 8.50
A (13,1,1)	4/12 x 1.20
A (13,2,1)	4/12 x 50.00
A (13,3,1)	4/12 x 8.50
A (14,1,1)	1/12 x 1.20
A (14,2,1)	1/12 x 50.00
A (14,3,1)	1/12 x 8.50

When all data have been applied for the period under study, array A will contain :

A (I,1,K) = $\sum$(Channel revenue in hour I of day K)

A (I,2,K) = $\sum$(CPU revenue in hour I of day K)

A (I,3,K) = $\sum$(Core revenue in hour I of day K)

3 D. ASSESSMENT OF SYSTEM ACTIVITIES.

When the revenue array is completed for a period, work item efficiencies are computed. These are shown in the output of program RESEVAL in appendix 2. For purposes of exposition a 3-dimensional array of n pages of output is represented by the array sketched in Array B of Table A.

CPU EFFICIENCY $= (A_{I,2,K})/(\text{UNIT-CPU-RATE})$

The unit is hours of CPU time used per hour of wall clock time.

$$\text{i.e.} \quad \eta_{CPU} = \sum_{job=1}^{\text{no. in the hour}} REV_{job} \; / \; \text{UNIT-CPU-COST}$$

TABLE A.

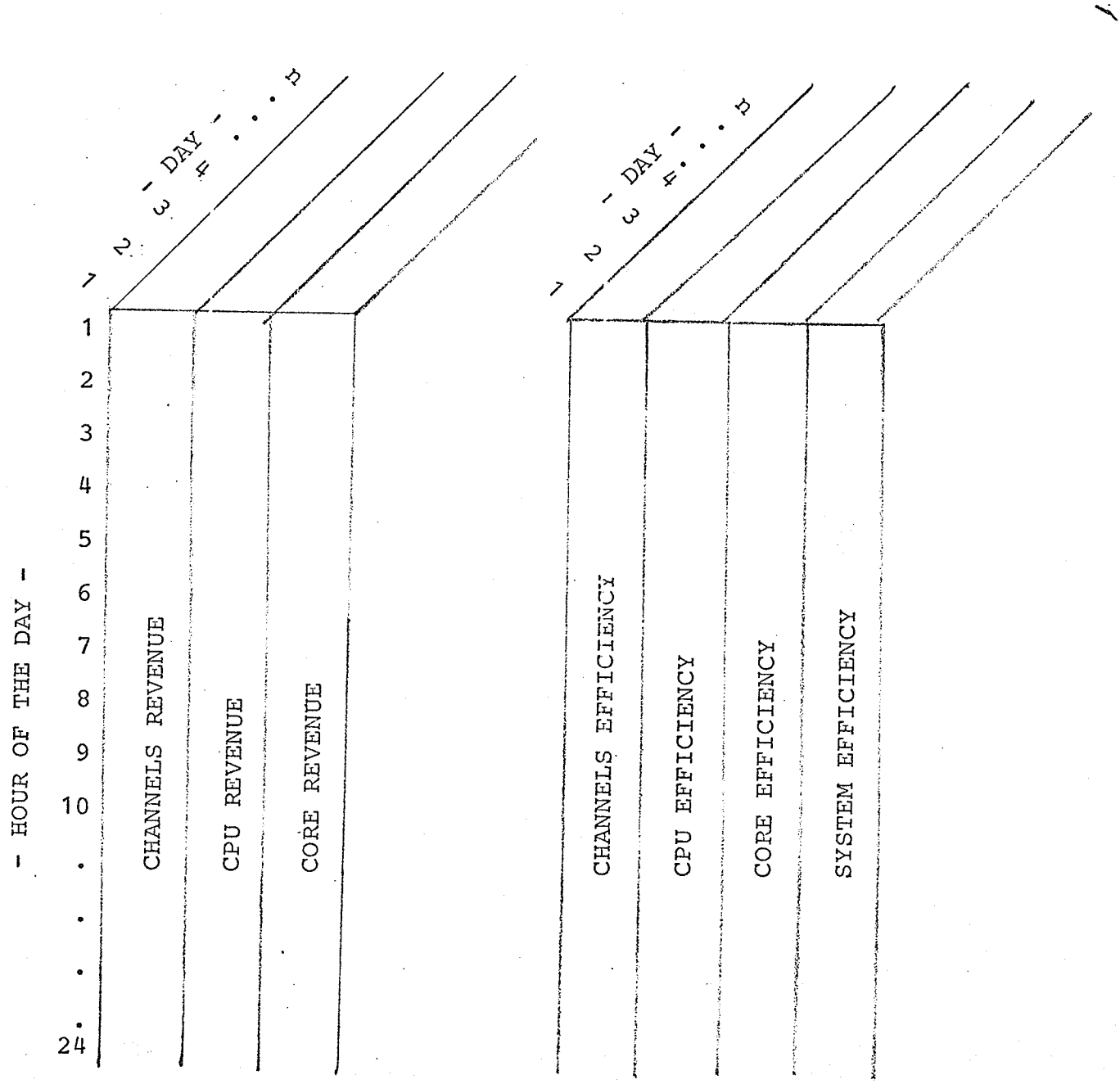
Dimension A (24,3,n) , B (24,4,n)
A (I,J,K) , B (I,J,K)

Array A

Activity revenue

Array B

Activity efficiency



$$\text{CORE EFFICIENCY}_{I,3,K} = \frac{A_{I,3,K}}{\text{UNIT COST OF CORE}} \times \frac{1}{R}$$

where R = core region available to users.

The unit is hours of user core used per hour.

Note that core region available to users may not be constant. There is a tendency to make available all core not committed to system overhead; but system overhead may change with service. For example several control programs or monitors may be resident for only part of each day. At the installation considered here there is a student monitor resident 12 hours a day, a teleprocessing monitor is resident 15 hours; an interactive

time-slicing compile and GO monitor is resident 10 to 15 hours per day and an interactive mathematics monitor is resident sometimes. So the system overhead may vary from 300 to 800 kilobytes. The corresponding region left for the users would vary from 724 to 224 kilobytes.

Array B contains, among other values, the computed value of channel efficiency in B(I,1,K) where I varies from 1 to 24 through the hours of the day, and K varies from 1 by 1 through the number of days in the period under study.

Array A contains, among other values, the cumulative revenue achieved by the channels in a given hour, where channel revenue from a job is : (I/O counts) x (Unit cost of a channel transfer). Channels revenues are stored in A(I,1,K) where I and K vary as described in the previous paragraph.

Channel efficiency is defined as the quotient of channel revenue in an hour divided by the channel revenue which would accrue if the channel were busy all the hour; or the quotient of channel counts divided by the maximum number of counts the channel could accrue in an hour. It is computed each hour as channel revenue for the hour divided by the maximum potential of channel revenue for that hour.

Thus, channel efficiency is :

B(I,1,K) = (A(I,1,K))/C

where C = channel cost for one hour.

The unit is hours of channel time used per hour.

Note that channel efficiency is not an absolute;

it is based, to a degree, on mumbo-jumbo. There is no accurate measure of transaction time for a data transfer from disk to core or core to tape. Instead an average figure of 25 milliseconds per transfer is assumed arbitrarily. It is unlikely that a single figure would be valid for 2314 disks, 2311 disks and 2400 tapes; but unfortunately nothing better than an estimate is available. Fortunately the inaccuracy of logging channel activity does not necessarily invalidate the proposed method of analysis. The approach proposed is to observe changes of efficiencies and trends of changes. Absolute values are not vital.

System efficiency for each hour was computed as :

$$\frac{\text{Channels revenue + CPU revenue + core revenue}}{\text{Total available revenue}}$$

This total available revenue is CPU rate, plus

the product of core rate multiplied by core available to users, plus per hour cost of all channels.

Thus :

$$B(I,4,K) = \frac{A(I,1,K) + A(I,2,K) + A(I,3,K)}{\text{Total available revenue}}$$

This method of transposing to dollars and back to efficiencies was chosen because an overall efficiency seems meaningless if it is a numeric combination of unmatched units.

The program for developing efficiencies is illustrated in the following listing. Its function as reported above is simple, but its implementation was fraught with pitfalls.

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
	JUNE 16,1970											
1	70167	2	26.76	34.44	34.86	96.05	13.00	0.12	0.23	41.01	20.88	
2	70167	2	199.97	112.37	139.70	452.04	20.00	0.93	0.75	164.36	0.63	
3	70167	2	184.42	68.69	81.06	334.17	18.00	0.85	0.46	95.37	27.08	
4	70167	2	89.98	29.26	50.18	169.42	5.00	0.42	0.20	59.03	0.60	
5	70167	2	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	
6	70167	2	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	
7	70167	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	
8	70167	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9	70167	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.72	
10	70167	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83	
11	70167	2	101.44	55.27	68.48	225.18	89.00	0.47	0.37	80.57	57.50	
12	70167	2	90.66	24.69	49.48	164.83	85.00	0.42	0.16	58.21	42.15	
13	70167	2	79.19	46.43	66.59	192.21	67.00	0.37	0.31	78.34	31.75	
14	70167	2	66.36	72.00	72.79	211.15	51.00	0.31	0.48	85.64	61.72	
15	70167	2	84.67	36.38	62.33	183.39	81.00	0.39	0.24	73.33	61.51	
16	70167	2	84.79	39.41	73.38	197.59	96.00	0.39	0.26	86.33	72.84	
17	70167	2	56.77	68.80	76.13	201.71	81.00	0.26	0.46	89.57	49.54	
18	70167	2	61.67	92.96	99.59	254.21	67.00	0.29	0.62	117.17	53.52	
19	70167	2	55.37	96.13	79.87	231.38	48.00	0.26	0.64	93.97	33.49	
20	70167	2	62.77	123.28	92.43	278.48	45.00	0.29	0.82	108.75	37.65	
21	70167	2	70.82	121.12	122.08	314.02	42.00	0.33	0.81	143.63	15.33	
22	70167	2	18.67	6.17	11.34	36.18	7.00	0.09	0.04	13.34	31.60	
23	70167	2	88.96	162.90	145.90	397.76	41.00	0.41	1.09	171.66	41.68	
24	70167	2	116.78	165.30	158.44	440.52	48.00	0.54	1.10	186.41	30.52	
25	70167	2	1540.07	1355.61	1484.61	4380.29	909.00				675.56	
ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25												
33	70167	2	1539.98	1355.53	1484.53	4379.54	920.00				672.10	5051.64
ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY												
26	70167	2	563.89	342.98	469.17	1376.06	550.00				377.85	1753.91
27	70167	2	250.63	433.50	393.97	1078.09	202.00				140.00	1218.09
28	70167	2	224.42	334.36	315.68	874.46	96.00				103.80	978.26

TABLE 3

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70168	3	57.98	97.54	92.52	248.04	35.00	0.27	0.65	108.85	34.41
2	70168	3	12.95	139.96	76.36	229.27	4.00	0.06	0.93	89.84	9.83
3	70168	3	4.39	152.85	55.06	212.30	0.00	0.02	1.02	64.77	9.06
4	70168	3	6.74	147.79	51.70	206.23	4.00	0.03	0.99	60.83	7.30
5	70168	3	7.65	149.09	55.61	212.35	8.00	0.04	0.99	65.43	5.11
6	70168	3	24.45	149.50	65.00	238.95	7.00	0.11	1.00	76.47	26.10
7	70168	3	8.51	158.98	108.44	275.93	3.00	0.04	1.06	127.58	2.27
8	70168	3	8.86	10.35	12.20	31.41	8.00	0.04	0.07	14.35	0.68
9	70168	3	80.36	94.25	83.27	257.88	29.00	0.37	0.63	97.97	35.95
10	70168	3	92.27	51.96	61.02	205.24	64.00	0.43	0.35	71.79	55.27
11	70168	3	106.47	40.18	71.68	218.33	57.00	0.49	0.27	84.33	42.40
12	70168	3	118.13	67.93	105.59	291.64	80.00	0.55	0.45	124.23	63.63
13	70168	3	105.17	64.98	100.23	270.37	68.00	0.49	0.43	117.92	57.92
14	70168	3	81.85	38.62	69.18	189.65	72.00	0.38	0.26	81.39	66.24
15	70168	3	59.80	21.21	32.46	113.47	64.00	0.28	0.14	38.19	37.42
16	70168	3	88.80	43.85	69.88	202.54	105.00	0.41	0.29	82.22	79.53
17	70168	3	75.33	48.08	64.60	188.01	91.00	0.35	0.32	76.01	46.82
18	70168	3	70.70	53.05	64.89	188.64	85.00	0.33	0.35	76.35	59.79
19	70168	3	80.05	105.79	106.37	292.21	61.00	0.37	0.71	125.15	27.59
20	70168	3	92.49	93.70	94.43	280.62	53.00	0.43	0.62	111.10	35.09
21	70168	3	89.82	118.38	120.79	328.99	50.00	0.42	0.79	142.12	47.41
22	70168	3	126.92	136.38	156.16	419.45	26.00	0.59	0.91	183.72	38.32
23	70168	3	76.56	79.73	109.70	265.98	44.00	0.35	0.53	129.06	26.04
24	70168	3	51.08	41.65	61.61	154.34	39.00	0.24	0.28	72.49	33.40
25	70168	3	1527.31	2105.78	1888.75	5521.83	1057.00				847.58

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70168	3	1527.21	2105.69	1888.65	5520.28	1071.00			849.65	6369.93

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70168	3	727.80	376.80	574.64	1679.25	601.00			449.23	2128.48
27	70168	3	333.06	370.91	386.49	1090.47	249.00			169.88	1260.34
28	70168	3	254.55	257.75	327.46	839.77	109.00			97.76	937.53

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70169	4	33.93	41.07	38.02	113.02	18.00	0.16	0.27	44.73	13.86	
2	70169	4	75.07	137.66	77.05	289.78	15.00	0.35	0.92	90.65	3.44	
3	70169	4	85.28	143.30	102.62	331.20	14.00	0.39	0.96	120.74	4.07	
4	70169	4	29.79	117.04	77.36	224.20	10.00	0.14	0.78	91.02	8.33	
5	70169	4	14.49	1.43	4.75	20.67	12.00	0.07	0.01	5.59	4.82	
6	70169	4	2.89	0.23	0.78	3.89	2.00	0.01	0.00	0.91	0.26	
7	70169	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70169	4	1.04	0.02	0.09	1.15	2.00	0.00	0.00	0.11	0.06	
9	70169	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
10	70169	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	
11	70169	4	113.69	30.78	55.01	199.47	94.00	0.53	0.21	64.72	56.13	
12	70169	4	97.73	69.01	94.07	260.81	105.00	0.45	0.46	110.68	53.66	
13	70169	4	113.49	99.00	113.52	326.01	87.00	0.53	0.66	133.56	48.54	
14	70169	4	60.39	125.12	102.58	288.09	29.00	0.28	0.83	120.69	52.10	
15	70169	4	109.14	38.93	65.19	213.26	91.00	0.51	0.26	76.69	47.53	
16	70169	4	81.72	29.02	45.38	156.11	80.00	0.38	0.19	53.39	43.65	
17	70169	4	81.31	59.33	79.63	220.27	86.00	0.38	0.40	93.69	50.19	
18	70169	4	114.16	79.87	82.24	276.27	81.00	0.53	0.53	96.76	41.11	
19	70169	4	101.39	59.66	83.83	244.88	55.00	0.47	0.40	98.63	39.11	
20	70169	4	65.59	148.34	164.89	378.82	49.00	0.30	0.99	193.99	34.63	
21	70169	4	35.57	76.77	73.39	185.72	34.00	0.16	0.51	86.34	33.59	
22	70169	4	103.97	64.63	114.81	283.40	55.00	0.48	0.43	135.08	37.47	
23	70169	4	90.01	58.94	95.36	244.32	33.00	0.42	0.39	112.20	35.47	
24	70169	4	62.46	17.25	41.26	120.96	31.00	0.29	0.11	48.54	25.06	
25	70169	4	1473.09	1397.38	1511.82	4382.30	983.00				633.09	

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70169	4	1478.65	1790.01	1692.14	4959.87	1010.00				633.90	5593.77

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70169	4	657.46	451.19	555.37	1664.02	572.00				351.84	2015.86
27	70169	4	316.71	364.64	404.34	1085.69	219.00				148.44	1234.13
28	70169	4	256.44	140.82	251.43	648.69	119.00				98.00	746.68

3 E. METHOD OF REPORTING AND INTERPRETATION OF THE REPORTS.

Information developed by the program described in section 3 D is factual but not easily assimilated. At best an array of efficiencies translated from array A of Table A into array B of Table A would require as many pages to report as B is deep in days. An observer tends to get lost in the surfeit of numbers. How to reduce a three-dimensional array to a two-dimensional array or to a pictorial representation ?

The method developed interprets each column of array B separately. A profile of each hour's activity through the period K days is developed by reordering the efficiencies from order by time to order by efficiency. For example, separate profiles of $B(1,1,K)\Big|_{k=1}^{n}$, $B(2,1,K)\Big|_{k=1}^{n}$, ..., $B(24,1,K)\Big|_{k=1}^{n}$ are considered.

Table 4a is an excerpt of array B for the period June 16 through June 30, 1970. Specifically it is B(I,2,K) for 'I' varying between hour 1 and hour 24 in the day and for 'K' varying from the 16th day of June to the 30th day of June; i.e., from year-day 167 through year-day 181. The range of 'K' is 1 to 15 in this example but will vary according to the period. This portion reports average CPU efficiencies for each hour of the period.

Consider the top row of Table 4a. It represents CPU efficiencies for the mid-night to 1.00 a.m. hour throughout the period. The 5th, 10th and 13th slots are empty because the computer was not working the first hour of days 171, 176, 179.

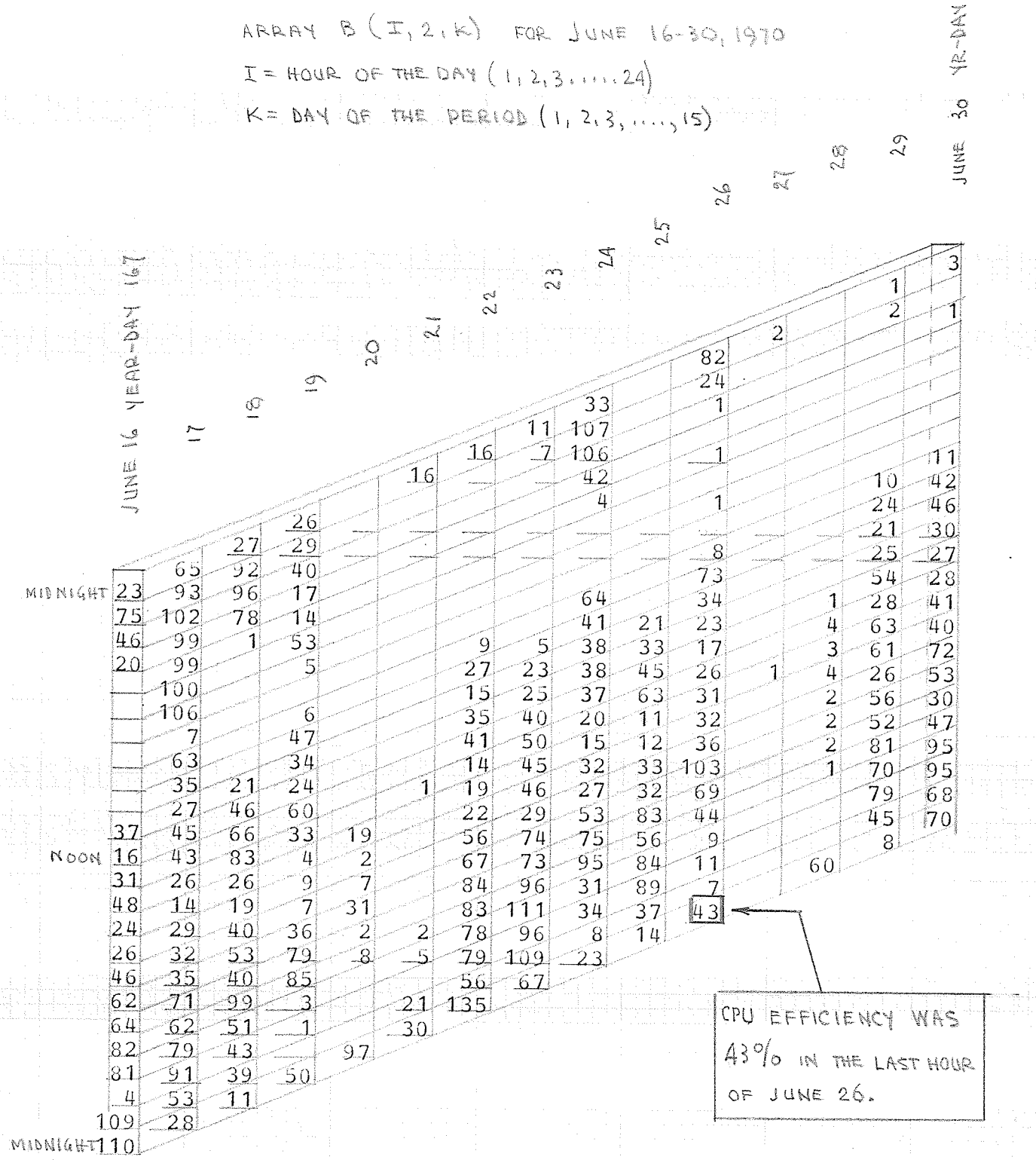
TABLE 4 a

CPU EFFICIENCIES

ARRAY B (I, 2, K) FOR JUNE 16-30, 1970

I = HOUR OF THE DAY (1, 2, 3, ... 24)

K = DAY OF THE PERIOD (1, 2, 3, ... 15)



Those hours should be ignored to bring data to a common base. There is no immediate interest in improving machine efficiency when the machine is idle.

The top row of the array shown on table 4b represents a frequency distribution of decile percentage efficiencies of the top row of table 4a :

3 items of table 4b had efficiencies between 0 and 10 %

3 had efficiencies between 10 and 20 %

3 had efficiencies between 20 and 30 %

1 had efficiency between 30 and 40 %

1 had efficiency between 60 and 70 %

1 had efficiency between 80 and 90 %.

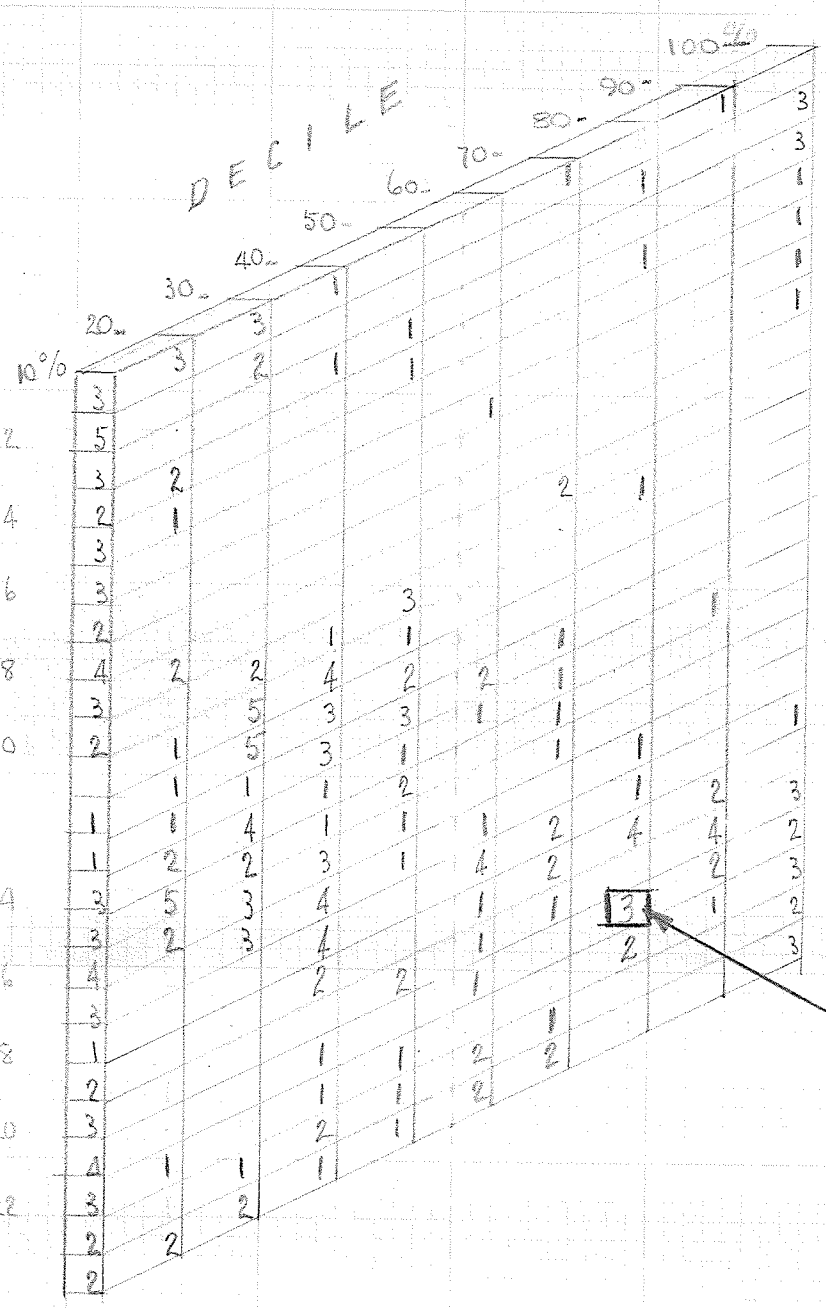
The top row of the array shown on table 4 c

TABLE 4 Q

C P U EFFICIENCIES

FREQUENCY

(DAYS PER DECILE PER HOUR)



3 DAYS IN THE PERIOD THE CPU EFFICIENCY FOR THE HOUR 8.00 TO 9.00 p.m WAS BETWEEN 70 & 80%

represents cumulative distributions of the top row of table 4b. But neither a frequency distribution nor a cumulative distribution table is a good comparative measure of efficiency profiles because the scale of the profile varies with the number of observations. So a percentage-of-total-cumulative-frequency profile was developed because its scale is constant regardless of the number of observations. With such a profile, activity of a one-week period may be compared validly to the activity of a one-year period.

The array shown on table 4a represents the information from table 4c, in a percentage frequency.

Tracing the previous examples; i.e., the top row, table 4d shows a profile that says all items were

TABLE 4C

CPU EFFICIENCIES

CUMULATIVE FREQUENCY
(DAYS PER DECILE PER HOUR)

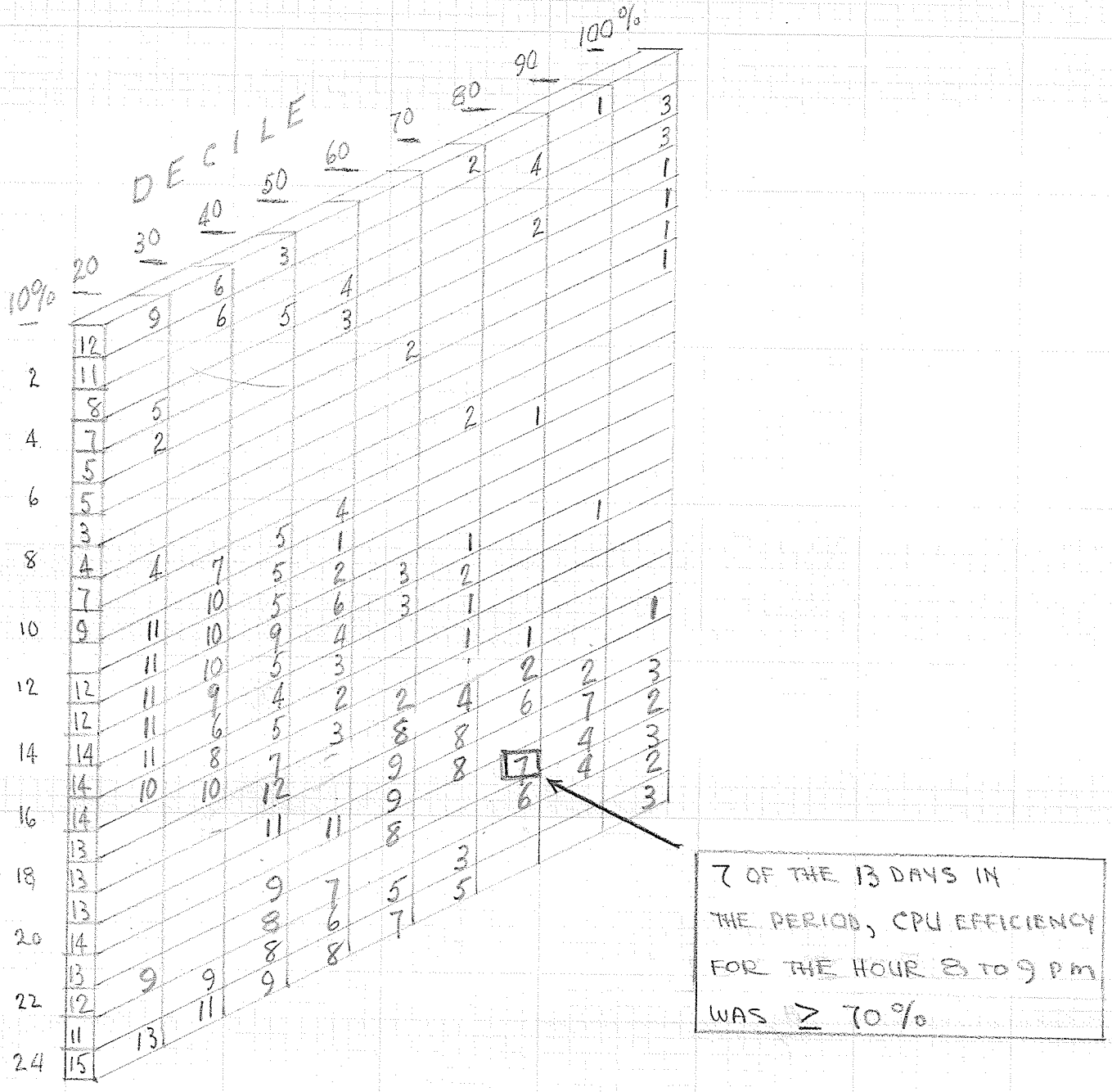
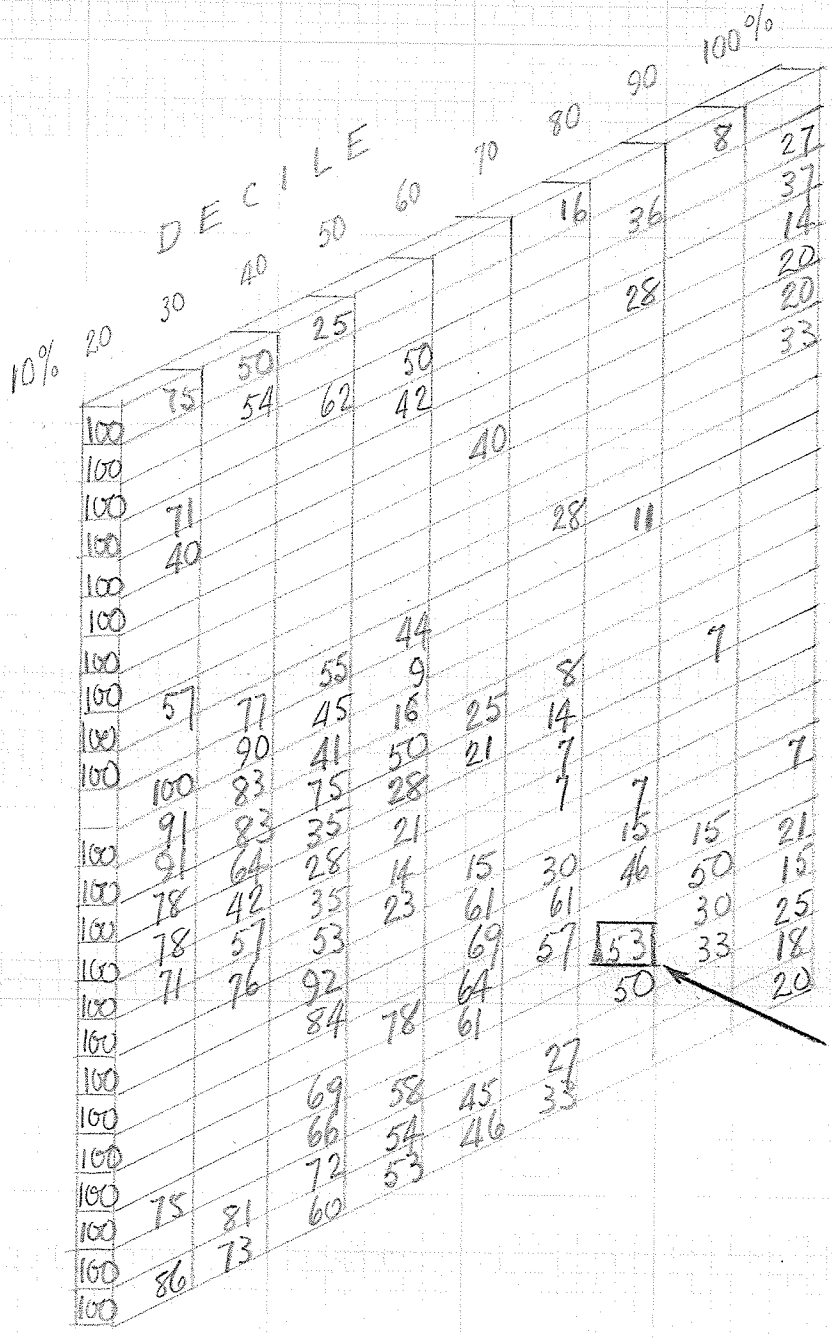


TABLE 4 a

CPU EFFICIENCIES

PERCENTAGE OF DAYS
PER DECILE PER HOUR



53% OF THE DAYS IN THE PERIOD, CPU EFFICIENCY FOR THE HOUR 8 TO 9 P.M WAS ≥ 70%

of size at least 10 %,

75 % of the items were at least size 20 %

50 % of the items were at least size 30 %

25 % of the items were at least size 40 %

16 % of the items were at least size 70 %

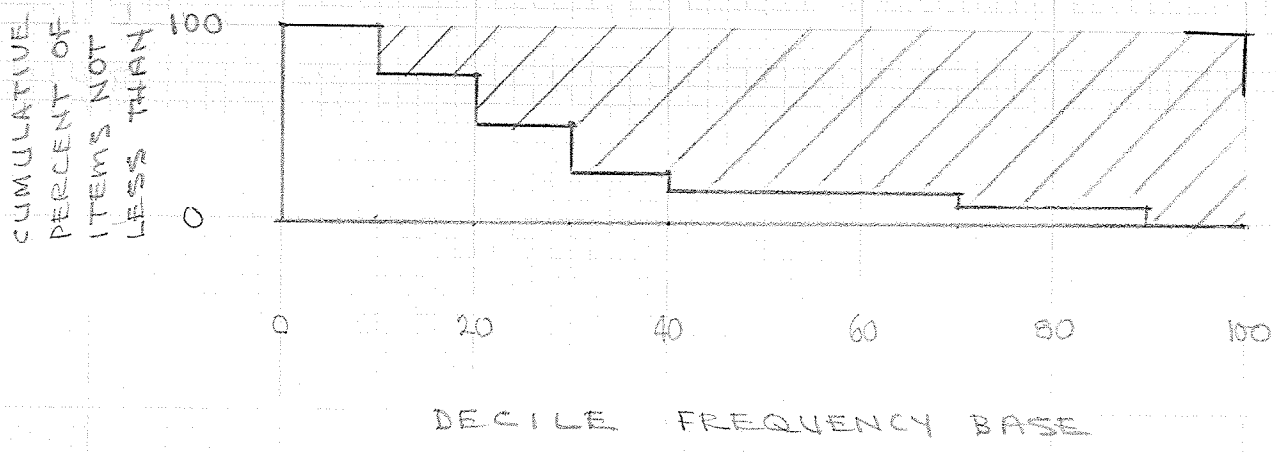
8 % of the items were at least size 90 %.

Table 4e is a to-scale plot of the top row of array of table 4d. It illustrates the efficiency and inefficiency of the use of the CPU resource the first hour after midnight. The cross-hatched area represents wasted CPU resources. The waste could not have been avoided entirely, but one is tempted to ponder on the disparity between 80 % efficiency and 10 % efficiency, and to ask if additional control might have saved some of the lost resource.

TABLE 4 e

CPU EFFICIENCY PROFILE

MIDNIGHT TO 1:00 A.m.
JUNE 16 TO JUNE 30, 1970



DECILE FREQUENCY BASE

PERCENT EFFICIENCY OF CPU

Table 4f is a perspective view of the top row of table 4d. If such a model were drawn for each row and if these models were each rotated 90 degrees clockwise, a pattern of plateaus would emerge.

This is basically the pictorial model projected by the programs described herein, with one liberty imposed to make the job a little easier and to make the model a little more readable : an assumption was introduced that there would be no sharp discontinuities in the profiles. The effect of this assumption is to say that the figure of table 4g represents the figure developed in table 4e and that the transition from row-to-row of the array is smooth. Then these data may be depicted as contours of equi-efficiency.

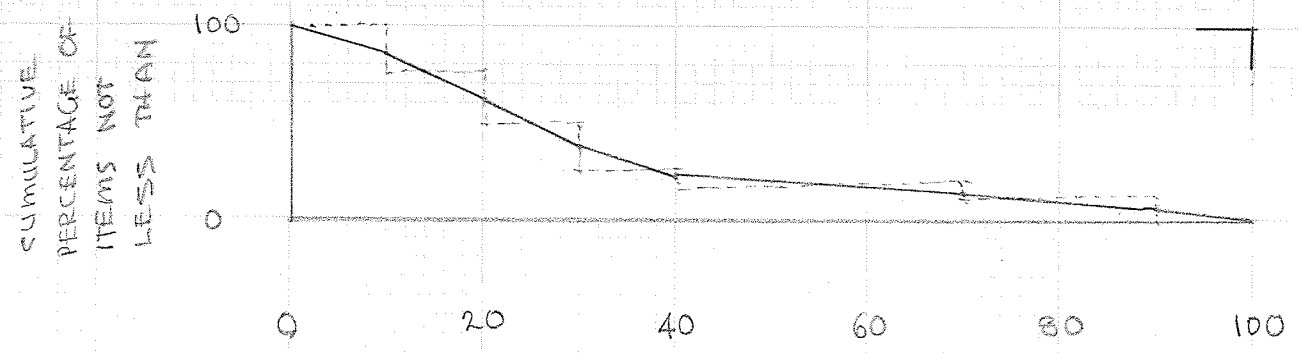
Please note that tables 4a, 4b, 4c, and 4d are

TABLE 4 f

TABLE 4g

CPU EFFICIENCY PROFILE

MIDNIGHT TO 100 a.m
JUNE 16 TO JUNE 30, 1970



PERCENT EFFICIENCY OF CPU

printed for the record on each study. Table 4a was copied directly from the enclosed results of program 'RESEVAL' (Appendix 2). Table 4b was copied from Table 3a, table 4c was copied from Table 3b and table 4d was copied from Table 3c.

The efficiency of interpretation is illustrated by these samples; 14 sheets of Table 3 are replaced by one sheet of table 4b without loss of clarity.

Table 3c shows this development for all 24 hours with decile positions printed at equal intervals. Lines of equi-efficiency can be plotted by interpolating between printed decile positions. Experience suggested that an equi-efficiency interval of 20 % gave an adequate pictorial representation. If the printer range is chosen as 100 units of length, hour 1 has 20 %-interval equi-efficiency points at distances, working from the left, of 10 units, 18 units, 26 units, 34

JUNE 16 TO 30, 1970 CORE 768K

FREQUENCY (DAYS / DECILE / HR).

HR	PERCENT USAGE OF CHANNELS											PERCENT USAGE OF CPU										PERCENT USAGE OF CORE										SYSTEM EFFICIENCY									
	0	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50						100
1		3	6	1				1	1			3	3	3	1			1		1		2	1	3	2	1		2			1	3	2	3		2	2					
2		6		1	2				1		1	5		2					1		3	4	1		1	1	2		1		1	4	2		1	1	1				2	
3		3	2	1	1					1		3			1	1					3	3			2		1	1	1			3			2		1	2				
4		4	1	1		1						2	2			1			1		1	2		1	3		1					2		2	2	1						
5		4	1									3	1								1	3		1	1							3	1		1							
6		3	1		1							3					1				1	3				1					1	3				1	1					
7		3										2									1	2						1				2				1						
8		4										4										4										4										
9			3	2	2							3	2						2				5			1	1							4	1	1	1					
10		2		1	2	4						2		2	1	3			1			2			1	3		3				2			3	3	1					
11			1		1	6	3						1	5	4	1							1	4	5	1								1	6	4						
12		1			4	3	2	2				1	1	5	3	2						1			4	3	2	1	1			1			6	3	2					
13		1		2	5	1	3					1	1	1	3	3	2	1				1			1	3	4	2	1			1			6	2	2	1				
14		3		1	6	2	1	1				3	2	4	1	1	1	1		1		3			3	3	2	1	1	1		3		1	6	2	1	1				
15		3	1	2	3	3	2					3	5	2	1	2		1				3	1	2	2	3	1	2				3	1	4	2	3	1					
16		3	1		4	5	1					4	2	3	3	1		1				3	1	1	1	2	4	2				3	1	2	4	2	2					
17		1	2	1	4	3	1	1				3		3	4	1	1		1			1	1		3	3	3		1	1		1	2		5	3	1	1				
18			1	2	3	2	5					1			4		4	2	1		1	1			1	2	4	1	1	3		1			3	2	4	2	1			
19		2	1	3	3	4						2			2		1	2	4	2		2			1	4	1	4		1		2		1	2	2	3	3				
20		2	6	3	1	2						3				2	1	1		4	3	2			2	4	3	2		1		2		1	3	3	3	2				
21		4	3	1	4	1						4			1		1		3	2	2	3		1		1	2		1	2	3	3	1		3		3	2	1			
22		4	3	1	2	1	1					3	1		1	1			2	1	3	3	1			2		2	2		2	3	1	1		1	5		1			
23		3	1	3	2	2						2		1	2	1	2	1			2	1	1		2	1	2	1	2		1	1	1	2	3	2	1		1			
24		3	3	5		1	1	1		1		2	2	2	1	1	2	2			3	1	1	2	2	1	2	1	1		4	1	2	3	4		2			1		

TABLE 3a

JUNE 16 TO 30, 1970 CORE 768K

CUMLATIVE FREQUENCY (DAYS / DECILE / HR).

	PERCENT USAGE OF CHANNELS											PERCENT USAGE OF CPU										PERCENT USAGE OF CORE										SYSTEM EFFICIENCY						
HR	0	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50			100
1		12	9	3					2	1		12	9	6	3			2		1		12	10	9	6	4		3		1		12	9	7		4	2		
2		11		5	4				2		1	11		6				4		3		11	7		6	5	4		2		1	11	7		5	4	3		2
3		8	5	3	2					1		8			5	4				3		8			5		3	2	1			8			5		3	2	
4		7	3	2		1						7	5			3		2		1		7		5	4		1					7		5	3	1			
5		5	1									5	2							1		5		2	1							5	2		1				
6		5	2		1							5					2			1		5				2				1		5				2	1		
7		3										3								1		3						1				3					1		
8		4										4										4										4							
9			7	4	2							7	4					2					7			2	1						7		3	2	1		
10		9		7	6	4						9		7	5	4			1			9			7	6		3				9			7	4	1		
11			11		10	9	3						11	10	5	1							11	10	6	1								11	10	4			
12		12			11	7	4	2				12	11	10	5	2						12			11	7	4	2	1			12			11	5	2			
13		12		11	9	4	3					12	11	10	9	6	3	1				12			11	10	7	3	1			12			11	5	3	1		
14		14		11	10	4	2	1				14	11	9	5	4	3	2		1		14			11	8	5	3	2	1		14		11	10	4	2	1		
15		14	11	10	8	5	2					14	11	6	4	3		1				14	11	10	8	6	3	2				14	11	10	6	4	1			
16		14	11		10	6	1					14	10	8	5	2		1				14	11	10	9	8	6	2				14	11	10	8	4	2			
17		13	12	10	9	5	2	1				13		10	7	3	2		1			13	12		11	8	5		2	1		13	12		10	5	2	1		
18			13	12	10	7	5					13			12		8	4	2		1	13				12	11	9	5	4	3	13			12	9	7	3	1	
19		13	11	10	7	4						13			11		9	8	6	2		13				11	10	6	5		1	13		11	10	8	6	3		
20		14	12	6	3	2						14				11	9	8		7	3	14				12	10	6	3		1	14		12	11	8	5	2		
21		13	9	6	5	1						13			9			8		7	4	2	13		10		9	8		6	5	3	13	10		9		6	3	1
22		12	8	5	4	2	1					12	9		8	7			6	4	3	12	9			8		6	4		2	12	9	8		7	6		1
23		11	8	7	4	2						11		9	8	6	5	3			2	11	10		9	7	6	4	3		1	11	10	9	7	4	2		1
24		15	12	9		4	3	2			1	15	13	11	9	8	7	5			3	15	14	13	11	9	8	6	5		4	15	14	12	9		5			3

TABLE 3 b

JUNE 16 TO 30, 1970 CORE 768K

P E R C E N T A G E O F D A Y S / D E C I L E / H R .

HR	PERCENT USAGE OF CHANNELS 0 10 20 30 40 50 60 70 80 90 100	PERCENT USAGE OF CPU 10 20 30 40 50 60 70 80 90 100	PERCENT USAGE OF CORE 10 20 30 40 50 60 70 80 90 100	SYSTEM EFFICIENCY 10 20 30 40 50 100
1	100 75 25 16 8	100 75 50 25 16 8	100 83 75 50 33 25 8	100 75 58 33 16
2	100 45 36 18 9	100 54 36 27	100 63 54 45 36 18 9	100 63 45 36 27 18
3	100 62 37 25 12	100 62 50 37	100 62 37 25 12	100 62 37 25
4	100 42 28 14	100 71 42 28 14	100 71 57 14	100 71 42 14
5	100 20	100 40 20	100 40 20	100 48 20
6	100 40 20	100 40 20	100 40 20	100 40 20
7	100	100 33	100 33	100 33
8	100	100	100	100
9	100 57 28	100 57 28	100 28 14	100 42 28 14
10	100 77 66 44	100 77 55 44 11	100 77 66 33	100 77 44 11
11	100 90 81 27	100 90 45 9	100 90 54 9	100 90 36
12	100 91 58 33 16	100 91 83 41 16	100 91 58 33 16 8	100 91 41 16
13	100 91 75 33 25	100 91 83 75 50 25 8	100 91 83 58 25 8	100 91 41 25 8
14	100 78 71 28 14 7	100 78 64 35 28 21 14 7	100 78 57 35 21 14 7	100 78 71 28 14 7
15	100 78 71 57 35 14	100 78 42 28 21 7	100 78 71 57 42 21 14	100 78 71 42 28 7
16	100 78 71 42 7	100 71 57 35 14 7	100 78 71 64 57 42 14	100 78 71 57 28 14
17	100 92 76 69 38 15 7	100 76 53 23 15 7	100 92 84 61 38 15 7	100 92 76 38 15 7
18	100 92 76 53 38	100 92 61 30 15 7	100 92 84 69 38 30 23	100 92 69 53 23 7
19	100 84 76 53 30	100 84 69 61 46 15	100 84 76 46 38 7	100 84 76 61 46 23
20	100 85 42 21 14	100 78 64 57 50 21	100 85 71 42 21 7	100 85 78 57 35 14
21	100 69 46 38 7	100 69 61 53 30 15	100 76 69 61 46 38 23	100 76 69 46 23 7
22	100 66 41 33 16 8	100 75 66 58 50 33 25	100 75 66 50 33 16	100 75 66 58 50 8
23	100 72 63 36 18	100 81 72 54 45 27 18	100 90 81 63 54 36 27 9	100 90 81 63 36 18 9
24	100 80 60 26 20 13 6	100 86 73 60 53 46 33 20	100 93 86 73 60 53 40 33 26	100 93 80 60 33 20

TABLE 3c

units, 57 units and 90 units and hour 2 has : 10 units, 19 units, 27 units, 69 units, 100 units and 100 units.

If 10 on line 1 is joined to 10 on line 2

If 18 on line 1 is joined to 19 on line 2

If 26 on line 1 is joined to 27 on line 2

If 34 on line 1 is joined to 69 on line 2

If 57 on line 1 is joined to 100 on line 2

If 90 on line 1 is joined to 100 on line 2

the line segments will represent equi-efficiency lines for CPU usage for midnight to 2.00 a.m. over the half month.

If profiles are developed for each of the 24 rows of a column of array B, rotated 90 degrees, and aligned on class at equal vertical intervals down a page, this array will form a basis for plotting contours of equal efficiencies. These contour

plots will give a good quick overall visual impression of the activity over the period for one activity. A program was developed to print separate contour bases for channel activity, CPU activity, core activity and for a combination of the three.

Table 3c shows the complete contour map for June 15 - 30 activities, sketched by approximation. Plates 5a, 5b, 5c, and 5d show the same plot by an incremental plotter.Please see page 68 for a detailed description and plates following page 70 for the plots.

These contour plots provide comparitive statistics, reduced to a common base, for periods of variable length. Thus a plot of one week's activity may be compared validly with a plot of six month's activity. If the time-span of any plot gets too small, variations in system load; i.e., in job mix, may make the plot non-representative

of the system's operating characteristics. The time-span must be long enough to span a period, if indeed system load is periodic. A day may be quite misleading; a week is better; a month is quite sufficient.

Hopefully this representation will eliminate the pervasive fear of error in predicting the characteristics of the job mix.

3 F. INTERPRETATION OF RECENT PERFORMANCE.

Table 3c shows the results of an analysis of the computer's activity June 15 - 30, 1970. This analysis considers all batch stream jobs run in that period;i.e., it includes all system activity except : 1) interactive terminal work monitored by University of Manitoba Monitor and 2) the student batch work submitted through the student terminal. The programs will handle these jobs, but to date accounting cards for those two classes of jobs are not available from HASP or OS.

Table 3c is a contour map of system activity. A brief tour will ease understanding : the figures to the right of hour 1 indicate that of the 12 midnight-to-1.00 a.m. hours in the period all fell

in the first decile; 75 % in the 2nd decile, 25 % in the 3rd, 16 % in the 7th and 8 % in the 8th decile. Thus 8 % of the days in which the computer was working between mid-night and one a.m. the channel efficiency was between 70 % and 80 %; 16 % of the days channel efficiency was 60 % or greater, etc.. Lines of equi-activity sketched to tie together the hours of the day supply a visual judgement of relative activity through night and day. Thus though the plant is often dormant around 6.00 to 7.00 a.m., the CPU and core efficiencies, between 6.00 and 7.00 a.m. on the days it is not dormant,are nearly as good as at 4.00 p.m., when the job queue is longest.

These contour plans might be vizualized as the plot of a plateau at level 100 at the left of each picture. If all resources were used to their maximum the plateau would stretch across three

inches of plot and form a vertical plane at the 100 % usage position. But the resources are not used to 100 % efficiency; the erosion from a 100 % plateau represents unused resource.

A plotting program was developed to plot the efficiency charts on a Calomp plotter. Though hand-plotting is relatively quick for a single study, machine plotting helps considerably if multiple periods are being considered in a study.

Plates 5a, 5b, 5c, and 5d show a machine plot to much the same scale as Table 3c. (Please note that the time scales are reversed on the two plots.) In the printer plot hours run from 1 to 24 down the page. The Calcomp plot runs from hour 1 through 24 up the page. Otherwise the plots match. Some features have been added to make visual interpretation easier :

1) A cross-section of each hour-group activity, except 11.00 p.m. to midnight, has been plotted to the right of each efficiency contour plot. If the activity were 100 % efficient the cross-section would be rectangular, with abscissa, or height, constant from left to right over the section. Thus the difference between the plotted area and the full rectangle represents wasted resource.

2) An asterisk has been added to each hour line and has been superimposed on each cross-section. It's abscissa value is :

100 x (cross-sectional area)/(rectangular area); i.e., it represents the mean efficiency of the hour group.

3) A vertical line has been added to the cross-sectional plots. It's abscissa value is the

sum of the hour-group means divided by 24; i.e., it represents the period mean for the activity it appends.

The first four plots immediately following this sheet replace plate 3c. Other plots are attached to Appendix 4 simply to illustrate the general pattern of activity through the months.

PLATE 5a

CHANNELS EFF

OORAN

JUN 30'70

PLOT NUMBER 32

PLOT NUMBER 33

PLOT NUMBER 34

PLOT NUMBER 35

PLOT NUMBER 35

JUN 30'70

PLOT NUMBER 37

PLOT NUMBER 38

PLOT NUMBER 39

PLATE 5b

CPU EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

JUN 30'70

PLOT NUMBER 41

PLOT NUMBER 42

PLOT NUMBER 43

PLOT NUMBER 44

PLOT NUMBER 45

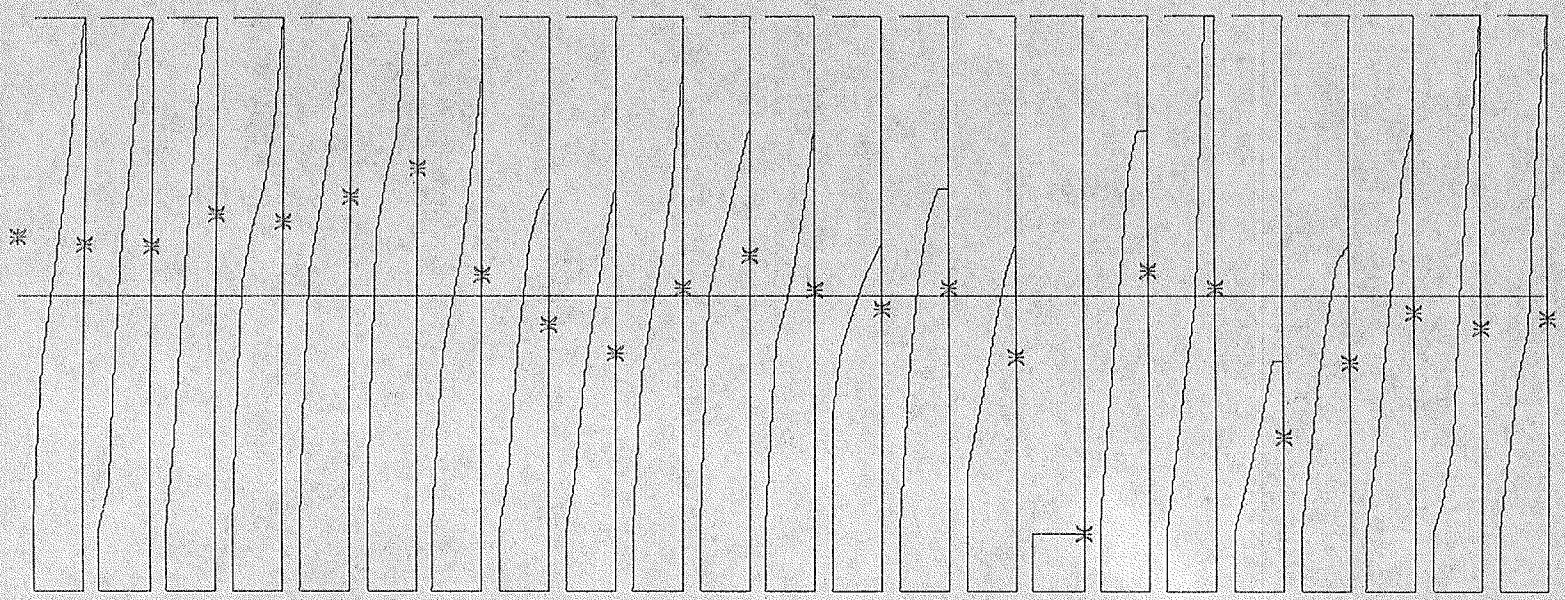
JUN 30'70

PLOT NUMBER 46

PLOT NUMBER 47

PLOT NUMBER 55

PLOT NUMBER 54



JUN 30'70

PLOT NUMBER 53

PLOT NUMBER 52

PLOT NUMBER 51

PLOT NUMBER 50

PLATE 5C

PLOT NUMBER 55

PLOT NUMBER 57

JUN 30'70

PLOT NUMBER 58

PLOT NUMBER 59

PLOT NUMBER 60

PLOT NUMBER 61

JUN 30'70

PLOT NUMBER 62

PLOT NUMBER 63

PLATE 5d

Chapter 4.

SCHEDULING FOR IMPROVED PERFORMANCE.

4 A. DEFINITION OF SCHEDULING.

Scheduling within the context of this report refers to decisions concerning the order in which the jobs in the input queue are presented to core storage and CPU for running. How the jobs are allocated system resources once they get into execution is not part of scheduling, by this definition. Much good work ([3]), ([5]) has been done to hurry the jobs through efficiently once they get into core, but the best system for resource dispatch is crippled if jobs are not

initiated in a good mix. It is clearly understood that the input queue does not contain a mixture of jobs which will assure ideal matching throughout, because the plant manager may have no control over the order in which users submit their jobs. So the aim of a good scheduling procedure is to find the best job matches available within their input queue environment, before the jobs are committed to an execution environment, because there is no facility for reconsidering and recovering from a bad decision once the jobs get into execution.

So far as this author can determine, no rational, automated ([4]), scheduling policy is in use in any computer centre to attempt to make the best use of installed resources. Brynolfson suggests a different interpretation of the resources scheduling problem. " The standard HASP job selection algorithm is to select the highest priority job in the class for which the initiator

is $DEFINED. Unfortunately, if the core required for that job is not available, the initiator will wait for it to become available, thereby degrading the multi-programming performance of the system and decreasing overall job thruput.

A similar but more impacting situation (allocation recovery) arises if all of the job's required devices are not available.

The resource scheduling extension to HASP attempts to prevent resource allocation problems by selecting for execution only those jobs whose resource requirements can be met."

The policy proposed herein solves Brynolfson's concern about degradation of the system when a job is dispatched without sufficient core to accommodate it. Further it could adapt to Brynolfson's solution to tape and disk contention, and be better for the adaptation. It is this

author's belief that the rationale presented herein is not degraded by Brynolfson's views.

4 B. DISCUSSION OF A SCHEDULING RATIONALE.

All computer installations have one common aim : to do useful work either cheaper than by non-computer methods or faster than by non-computer methods or more thoroughly than by non-computer methods or some combination of these benefits. A computer installation in a university may have an additional aim : to aid in teaching about computers. Perhaps the first objective is easier to address; it can be contemplated in terms of efficiencies and costs, and might be approached or realized by rational planning. Though the second objective is not so easily understood in terms of costs, for it contains many intangibles, the two objectives are in many ways parallel.

Much of the credit for job improvements in cost, time, and thoroughness by using a computer is due

to the analyses and problem programs which attend the computer solution. But the management of these functions does not come within the jurisdiction of computer operations. However, operational improvements may be pursued independently to improve the processing of those programs committed by the problem programmer.

A computer manager who has program development and testing as a significant proportion of his work load may be tempted to forsake system efficiency for the expediency of giving quick turnaround to development programs. A university may be tempted to forsake system efficiency for the expediency of giving special service to a particular class of jobs or to a particular type of user.

The scheme proposed here does not quarrel with

the educator who is prepared to request and pay for special privilege, though the privilege may introduce inefficiencies. The scheme's basic concept is to try to establish a schedule for transferring jobs from the input queue to core, which will waste the least plant resource. With knowledge of the waste accompanying this feasible schedule, one may then examine the additional waste that will develop if the schedule is modified to meet a user's request for special service. If the user and the plant manager can come to some agreement for introducing unnecessary waste, then the scheduling algorithm can be overridden. But if the plant approaches maximum loading the manager may choose to review the special agreements rather than install more plant.

4 C. ASSIGNMENT OF INDEPENDENT VARIABLE FOR SCHEDULING.

Plate 4c-0 is used as a template for illustrating the interactive use of channel, core, and CPU resources. The plate depicts these three resources in three dimensions :

x depicts CPU activity in units of time

y depicts channel activity in units of time

z depicts core activity in kilobytes.

The top view will be used to plot true lengths of x and y normal to each other. The centre view, immediately below the top view depicts an elevation in true lengths; the horizontal vector is a projection of CPU units, the vertical is core

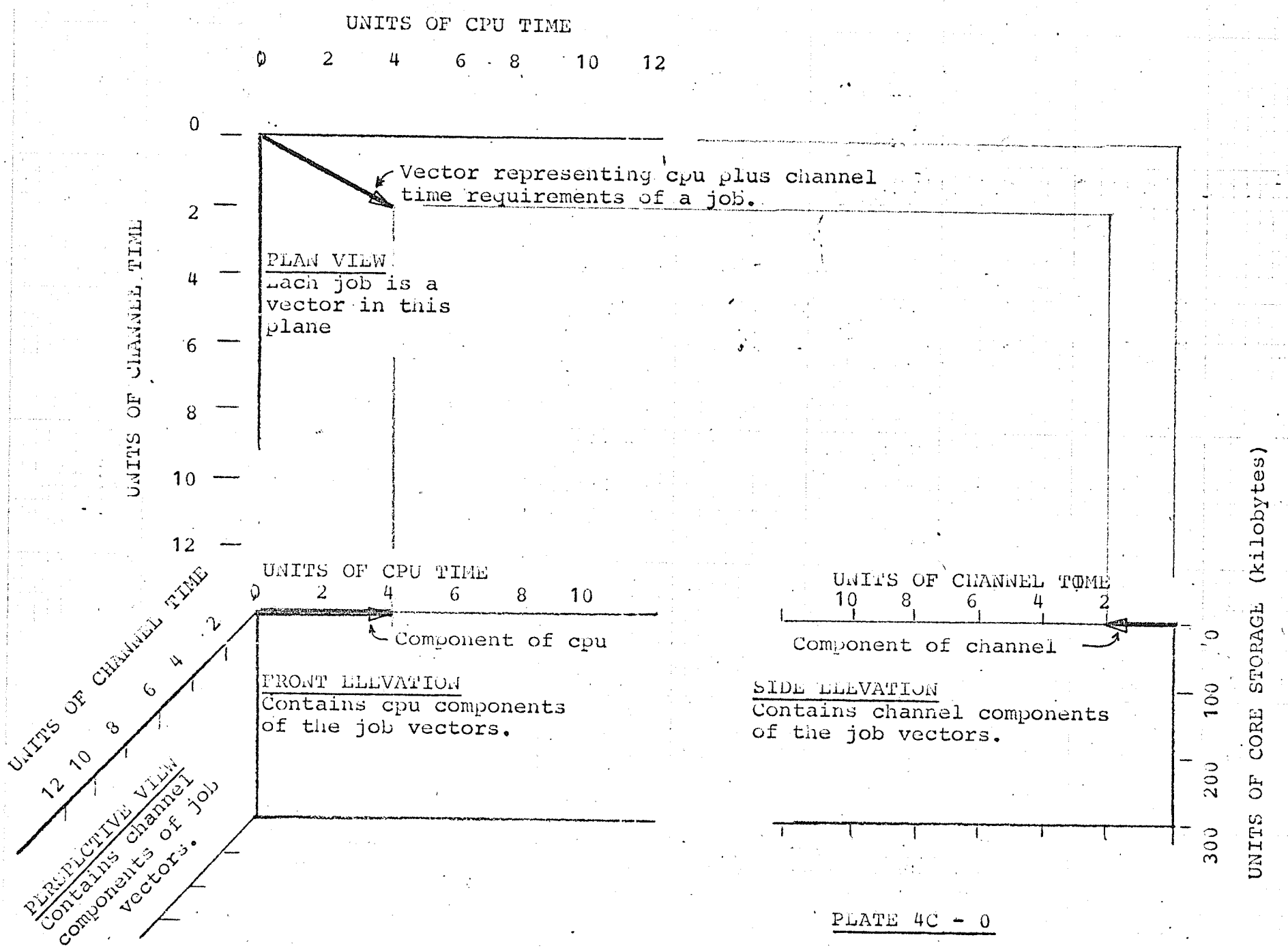

UNITS OF CPU TIME
0 2 4 6 8 10 12
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
Vector representing cpu plus channel time requirements of a job.
PLAN VIEW
Each job is a vector in this plane
UNITS OF CPU TIME
0 2 4 6 8 10
Component of cpu
FRONT ELEVATION
Contains cpu components of the job vectors.
UNITS OF CHANNEL TIME
10 8 6 4 2 0
Component of channel
SIDE ELEVATION
Contains channel components of the job vectors.
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
PERSPECTIVE VIEW
Contains channel components of job vectors.
0 100 200 300
UNITS OF CORE STORAGE (kilobytes)

PLATE 4C - 0

used. This front elevation depicts kilobyte-seconds of CPU activity as an area. The side view, lower right , is defined by a true length projection of channel units and by core used. The enclosed area is kilobyte seconds of channel activity. The lower left view is a perspective view of the lower right true projection. It is redundant,but is included to aid in visual interpretation.

Resource waste was chosen as the independent variable which will give a reasonable reflection of the adequacy of a job mix in core.

A few illustrations of waste computations for hypothetical job mixes may add to the understanding of the argument. The users' region is assumed to be 300 kilobytes. The installation

is assumed to have only one data channel.

1) Suppose a job with region size of 200 kilobytes and CPU requirement of 4 seconds is alone in core. Its useful work will be

4 x 200 = 800 kilobyte-seconds

But it commits

4 x 300 = 1200 KBS of CPU resource and

4 x 300 = 1200 KBS of channel resource.

The wasted resource will be

2400 - 1800 = 1600 KBS

Plate 4C-1 shows this 1600 KBS as the shaded area.

2) Suppose a job of region size 200 kilobytes, CPU requirements of 4 seconds and channel requirements of 4 seconds stands alone in core. Please see plate 4C-2. Its useful work will be

4 x 200 = 800 KBS of CPU and
4 x 200 = 800 KBS of channel but it will require a minimum of 4 + 4 = 8 seconds of core residence and so will commit

8 x 300 = 2400 KBS of CPU potential and
2400 KBS of channel potential.

Wasted resource will be 4800 - 1600 = 3200 KBS.

There may be some parallel processing in some

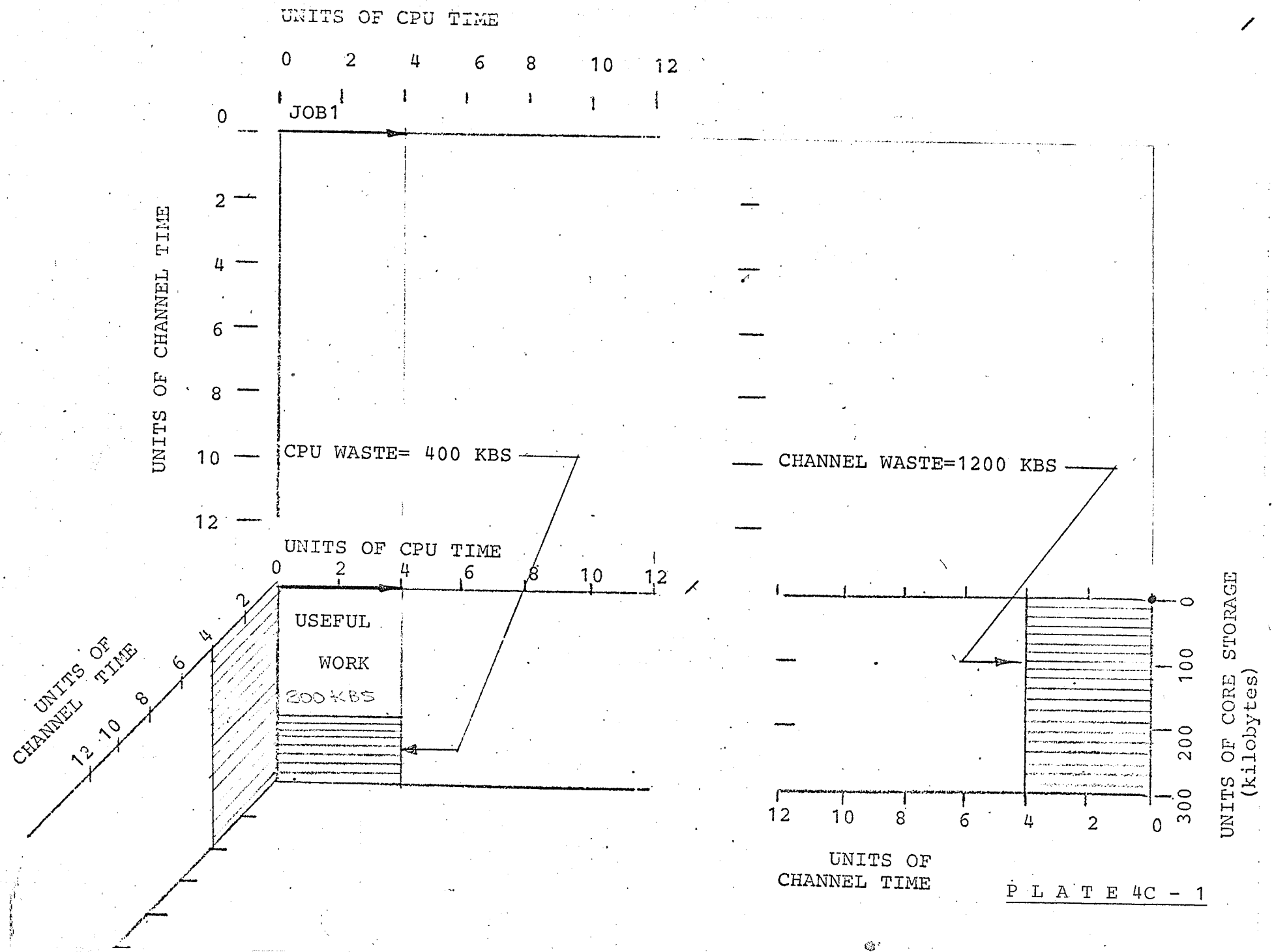

UNITS OF CPU TIME
0 2 4 6 8 10 12
JOB1
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
CPU WASTE= 400 KBS
CHANNEL WASTE=1200 KBS
UNITS OF CPU TIME
0 2 4 6 8 10 12
USEFUL WORK
200 KBS
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
12 10 8 6 4 2 0
UNITS OF CHANNEL TIME
0 100 200 300
UNITS OF CORE STORAGE (kilobytes)

PLATE 4C - 1

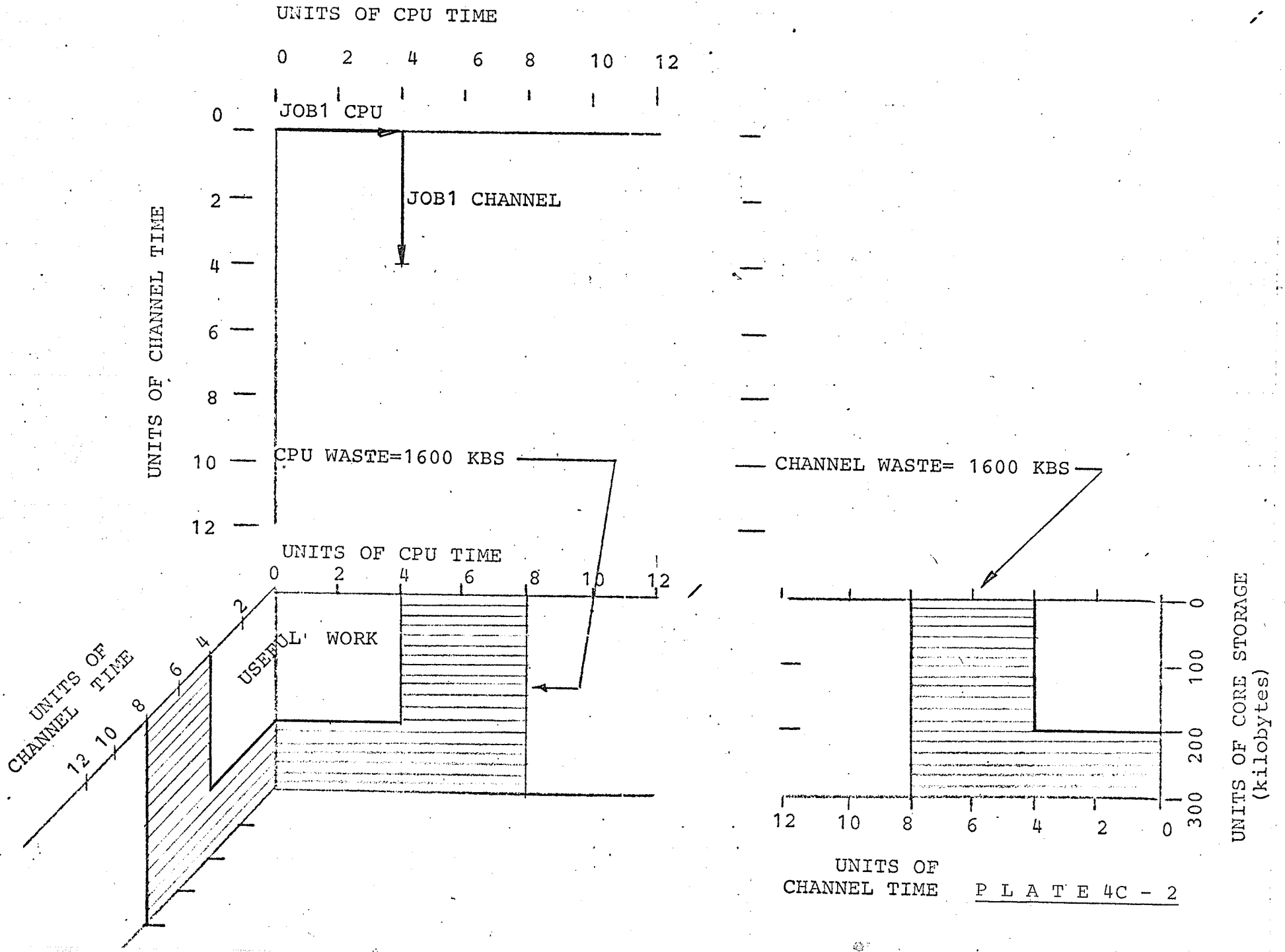

UNITS OF CPU TIME
0 2 4 6 8 10 12
JOB1 CPU
JOB1 CHANNEL
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
CPU WASTE=1600 KBS
CHANNEL WASTE= 1600 KBS
UNITS OF CPU TIME
0 2 4 6 8 10 12
USEFUL WORK
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
UNITS OF CORE STORAGE (kilobytes)
0 100 200 300
UNITS OF CHANNEL TIME

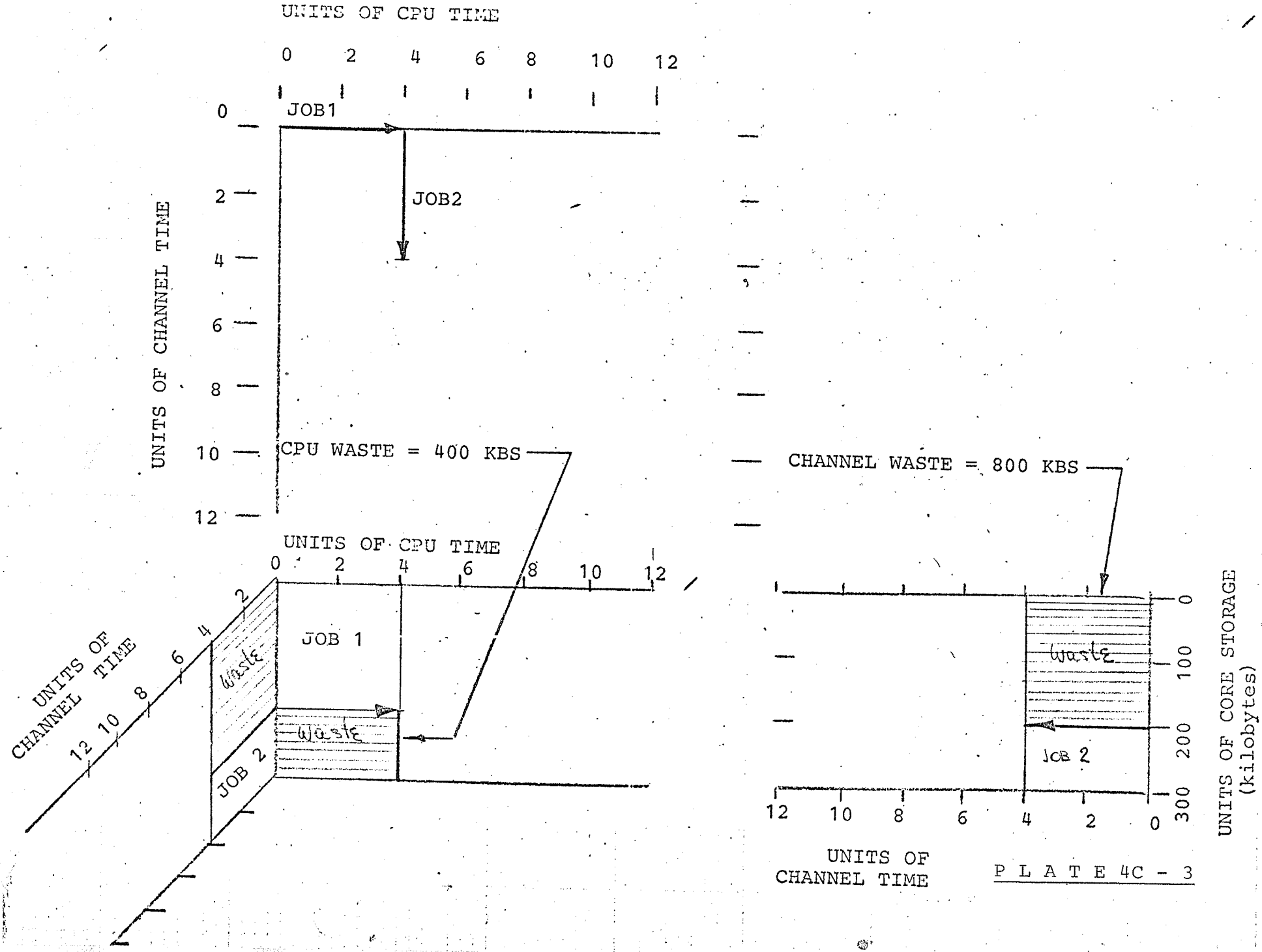
PLATE 4C - 2

jobs with CPU and channel work but the pattern of most jobs is serial; i.e., compute until data are required and then fetch. Certainly parallel processing is more likely to occur if records are blocked in good balance, but the rationale developed in this report is not dependent upon the jobs being completely serial or mainly parallel processing. The point is mentioned for the sake of clarity.

3) Suppose another job; with region size 100 kilobytes, CPU time = 0, and channel time = 4 seconds, is resident with the job described in example 1. Please see plate 4C-3.

Useful work = 800 + 400 = 1200 KBS

Committed resource = 2400 KBS


UNITS OF CPU TIME
0 2 4 6 8 10 12
JOB1
JOB2
UNITS OF CHANNEL TIME
CPU WASTE = 400 KBS
CHANNEL WASTE = 800 KBS
UNITS OF CPU TIME
JOB 1
JOB 2
waste
waste
waste
JOB 2
UNITS OF CHANNEL TIME
UNITS OF CORE STORAGE (kilobytes)
0 100 200 300

P L A T E 4C - 3

Wasted resource will be : 2400 - 1200 = 1200 KBS.

This example illustrates how 400 KBS of extra work may be completed at no additional expenditure of resources, simply by mixing jobs with complementary characteristics.

Waste, as defined here, is not all avoidable. It is a measure of job region x the sum of all unused facility times; but is most easily assessed as the difference between resource potential and useful work. Whenever more than one job is resident there is inherent waste, but the magnitude of the waste may be smaller for the presence of multiple jobs. A single job in good balance of parallel processing might attain zero waste, but such jobs are rare.

4 D.ASSIGNMENT OF DEPENDENT VARIABLES FOR SCHEDULING.

The dependent variables which govern waste are :

1) The channel requirements of each job in core.

2) The CPU requirements of each job in core.

3) The storage requirements of each job in core.

By the time the jobs reach execution stage it is too late to correct mismatching of jobs in the hope of reducing waste; so the scheduling should operate on the input queue.

The user usually can know the core, channel and CPU requirements of his job. When he first attempts compilation of a new job he may know only

the compiler's region requirements, and may need to guess CPU and channel requirements, though experience may make his guesses realistic. But the results of the first attempt will report, for his information, the resources used. He will probably ignore the report unless his job's requirements exceed the default levels set by the installation.

There is a fairly standard procedure, observed by most installations with multi-tasking facilities, for inviting the user to report, in his job card, the system resources his job will require. If he does not report his requirements, the scheduling routine will insert some arbitrary values which will serve as upper bounds for the resources. It is customary for the user to accept default values. If they are too low he will be obliged to submit his own, but there exists a disparity

between predicted and actual requirements for the fact that he may be lazy, uninformed, or cautious.Perhaps the disparity is predictable and an adjustment could be applied at the scheduling stage. At an installation that runs short jobs first, there is an incentive for the user to give accurate resource estimates; for if his guesses are low his job will be flushed before its work is finished, but if his guesses are high he will wait longer to get his job run. Unfortunately this incentive will not obtain in an environment using a scheduling procedure of the type proposed by this report, so the estimates of CPU and channel requirements will always be high or just sufficient. But the incentive will remain to be accurate in the estimate of region requirements, for the user is likely to be charged for the region he requests. If he requests more than he

needs he will pay more than he need.

4 E. DERIVATION OF A SCHEDULING RATIONALE.

The data available for scheduling are :

1) A knowledge of the system's resources; including users' core, CPU resource and data channel resource.

2) A table of job requirements for each job in the input queue; i.e., region requirement, an estimate of CPU time required and an estimate of channel requests which will be made. The channel counts will be converted to seconds of channel time in accordance with the method of section 3 B.

The aim of this section is to develop a routine which will reduce gross resource waste. The routine proposed here measures its success and thereby could supply a comparison of its

scheduling success with the results of any alternative proposed by others. The argument does not purport to be absolute but rather to lead from a feasible solution to an improved feasible solution, measured by a reduction in waste.

Core limitations restrict the number of jobs resident at any one time to a finite number. The region requirements of the average job further restrict the number to a small range. This fact suggests that a good scheduling procedure should select a single job from the input queue then search the remainder of the queue for the job or jobs which best complement the one selected, then choose another and repeat. Such a procedure requires a rationale for selecting the first job of a group and another rationale for selecting its mates. Intuition suggests that the job which uses the most kilobyte-seconds should be assigned the

resources to supply its needs efficiently because it is the job which is most likely to waste resources if held in execution unnecessarily. If other jobs can be assigned to soak up unused resources, the revenue accruing from the extra jobs will tend to be profit. Another argument which supports considering the largest job first is that if a different selection is made first, large jobs will be difficult to fit; i.e., it will be difficult to find a suitable combination to complement the large jobs. This is not to suggest that the large jobs be released to execution first, but that in arranging a schedule shake-down the large jobs be positioned first and the smaller jobs be fitted into the voids left between big jobs.

A program was developed to interpret the resource requirement of each job in the input queue. The

only job data available for interpretation are those supplied on the job card accompanying each job submission : core requirement, channel calls (sometimes called I/O counts), and job time. These figures are supplied explicitly by the user or implicitly by the default option of each. Job time is, by definition, CPU time plus I/O activity translated into time. This translation is (I/O counts x .025) seconds.

Thus the work awaiting service at any moment is known by interpretation of the input queue. The task of this program is to match jobs into family units which fit together in execution.

The essence of the argument for this approach is that it reduces the problem to exceptionally simple terms : given any philosophy of job dispatch, system resources will be used efficiently, provided only that the program be allowed to form family groups to go together. Thus

regardless of the reason, or lack of reason, behind a decision to give job number 97 a priority of 13, the decision will not degrade the system efficiency if jobs 47 and 22 (assuming these are members of 97's family) are permitted to co-habit in core with 97.

The rationale for family formation is :

1) Assess the amount of system resource wasted by allowing job 97 to stand alone. If job # 97 does not require all available core go to (2), otherwise dispatch # 97 alone.

2) Search the job queue for the job which will make best use of plant resources which job # 97 does not require.

3) Assign the job found in (2) for dispatch with host 97.

4) As in (1), assess plant resource wasted when 97 and its guest co-habit; continue the loop through the procedure until no jobs fit wasted resources. The family group so formed is in balance : the use of resources; i.e., system efficiency associated with the group will not vary with time of dispatch.

Details of the program follow. The program described and demonstrated is much more expository than it needs to be for practical application; only a small portion of the coding would need to be used in practice. The remainder of the coding is simply exposition to make the argument easier to understand.

PROGRAM TO MINIMIZE WASTED RESOURCES.

A. DATA.

The program uses as data :

1) The amount of core available for use by the batch stream,

2) Characteristics of each individual job awaiting initiation in the input queue; i.e.,

a) job number

b) CPU time estimate

c) channel-calls estimate

d) core requirement.

For illustration, consider a job queue containing 27 jobs shown on Table 7. The first four fields of the list contain job data.

B. INTERPRETATION OF DATA.

1) Core time requirements, in k-byte-seconds, for each job are computed as :

(CPU time + Channel time) x Region ... contained in Field 5 of Table 1.

2) A vector angle is computed for each job :

$$\theta = \text{Arctan}\left(\frac{\text{Chan-sec}}{\text{CPU-sec}}\right)$$

... contained in Field 6 of Table 1. (In practice the tangent

of the angle is simpler to use than the angle. The angle is used in the description for the convenience of being able to draw explanatory sketches.)

3) The job list is sorted, or flagged, on k-byte-seconds requirements, in descending order. ... see Table 8.

4) The job list is sorted, or flagged, on vector angle characteristics, in descending order. ... see Table 9.

5) The job list is arranged in family groups. Each group contains at least a host and may contain several guests, depending on resources left available for guests. A family group, by definition, is a host plus whatever unassigned jobs in the queue best complement the host. The

DATA WITH SOME INTERPRETATION

---------- DATA ----------					
JOB NO.	CPU-SEC.	CHAN-SEC.	CORE-KB.	KBSEC.	THETA (RADIANS)
28	5.00	1.00	100.00	600.00	0.20
29	0.10	2.00	60.00	126.00	1.52
30	0.10	1.00	52.00	57.20	1.47
31	0.30	0.80	65.00	71.50	1.21
32	0.50	0.60	54.00	59.40	0.88
33	0.70	0.40	62.00	68.20	0.52
34	0.90	0.20	56.00	61.60	0.22
35	1.10	0.40	75.00	112.50	0.35
36	0.90	0.60	58.00	87.00	0.59
37	0.70	0.80	85.00	127.50	0.85
38	0.50	1.00	64.00	96.00	1.11
39	0.30	0.80	95.00	1o4.50	1.21
40	0.10	0.60	66.00	46.20	1.41
41	0.30	0.40	105.00	73.50	0.93
42	0.50	0.20	68.00	47.60	0.38
43	0.70	0.20	110.00	99.00	0.28
44	0.90	0.40	70.00	91.00	0.42
45	1.10	0.60	115.00	195.50	0.50
46	0.90	0.80	72.00	122.40	0.73
47	0.70	0.20	120.00	108.00	0.28
48	0.50	0.20	74.00	51.80	0.38
49	0.30	0.20	125.00	62.50	0.59
50	0.10	0.20	76.00	22.80	1.11
51	6.00	0.10	130.00	793.00	0.02
52	0.10	0.20	78.00	23.40	1.11
53	0.30	0.10	80.00	32.00	0.32
54	0.50	0.10	135.00	81.00	0.20

(Copied from program results shown in Appendix 5)

- T A B L E 7 -

JOBS IN ORDER OF (CORE x TIME) REQUIREMENTS

kilobyte-seconds

JOB NO.	KB-SEC.
51	793.00
28	600.00
45	195.50
37	127.50
29	126.00
46	122.40
35	112.50
47	1o8.00
39	1o4.50
43	99.00
38	96.00
44	91.00
36	87.00
54	81.00
41	73.50
31	71.50
33	68.20
49	62.50
34	61.60
32	59.40
30	57.20
48	51.80
42	47.60
40	46.20
53	32.00
52	23.40
50	22.80

(Copied from program results shown in Appendix 5)

- T A B L E 8 -

JOBS IN ORDER OF VECTOR ANGLE

JOB NO.	THETA.
29	1.52
30	1.47
40	1.41
31	1.21
39	1.21
38	1.11
50	1.11
52	1.11
41	0.93
32	0.88
37	0.85
46	0.73
36	0.59
49	0.59
33	0.52
45	0.50
44	0.42
42	0.38
48	0.38
35	0.35
53	0.32
43	0.28
47	0.28
34	0.22
54	0.20
28	0.20
51	0.02

(Copied from program results shown in Appendix 5.)

- T A B L E 9 -

JOBS REQUIREMENTS IN ORDER OF VECTOR ANGLE

JOB NO.	REGION	SECONDS	RADIANS
29	60.00	2.10	1.52084
30	52.0	1.10	1.47113
40	66.0	0.70	1.40565
31	65.0	1.10	1.21202
39	95.0	1.10	1.21202
38	64.0	1.50	1.10715
50	76.0	0.30	1.10715
52	78.0	0.30	1.10715
41	1o5.0	0.70	0.92730
32	54.0	1.10	0.87606
37	85.0	1.50	0.85197
46	72.0	1.70	0.72664
36	58.0	1.50	0.58800
49	125.0	0.50	0.58800
33	62.0	1.10	0.51915
45	115.0	1.70	0.49935
44	70.0	1.30	0.41822
42	68.0	0.70	0.38051
48	74.0	0.70	0.38051
35	75.0	1.50	0.34877
53	80.0	0.40	0.32175
43	110.0	0.90	0.27830
47	120.0	0.90	0.27830
34	56.0	1.10	0.21867
54	135.0	0.60	0.19740
28	100.0	6.00	0.19740
51	130.0	6.10	0.01667

(Copied from program results shown in Appendix 5)

- T A B L E 10 -

FAMILY STRUCTURING

JOB NO.	HOST	ORDER	CUMULATIVE WASTE
28	28	6	1200.
37	28	7	1073.
39	28	8	968.
29	51	2	911.
30	51	3	884.
40	51	4	772.
51	51	1	1037.
52	51	5	762.
31	35	16	266.
35	35	15	338.
50	35	17	243.
32	36	24	304.
36	36	23	363.
33	33	25	262.
42	33	26	244.
34	34	27	268.
38	45	10	219.
45	45	9	314.
41	46	12	314.
46	46	11	388.
48	46	13	262.
49	46	14	208.
43	43	20	171.
44	44	21	299.
54	44	22	248.
47	47	18	162.
53	47	19	160.

(Copied from program results shown in Appendix 5)

- T A B L E 11 -

aim of family selection is to minimize resource waste.

6) Wasted resource is defined as the difference between the resources committed by the jobs in core and the total useful work performed.

Let A = total resource expended

B = total useful work performed

C = net wasted resource

D = minimum system time for all jobs assigned to a family

Then $C = A - B$

And $B = \sum_{i=1}^{n} r_i (a_i + b_i)$

where a_i = CPU-second of job

b_i = channel-seconds of job

r_i = core requirement of job

n = number of jobs in the family

Then D = the largest of a, b, and c below, where :

a) $\sum_{i=1}^{n} a_i$

b) $\sum_{i=1}^{n} b_i$

c) Largest $(a + b)_i$

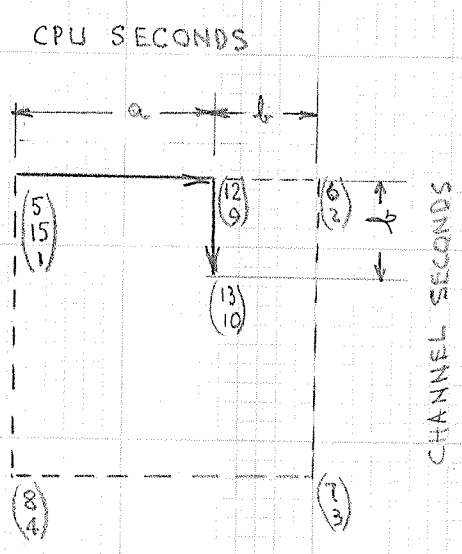
Then A = (2 x D x (batch region)) kbsec;

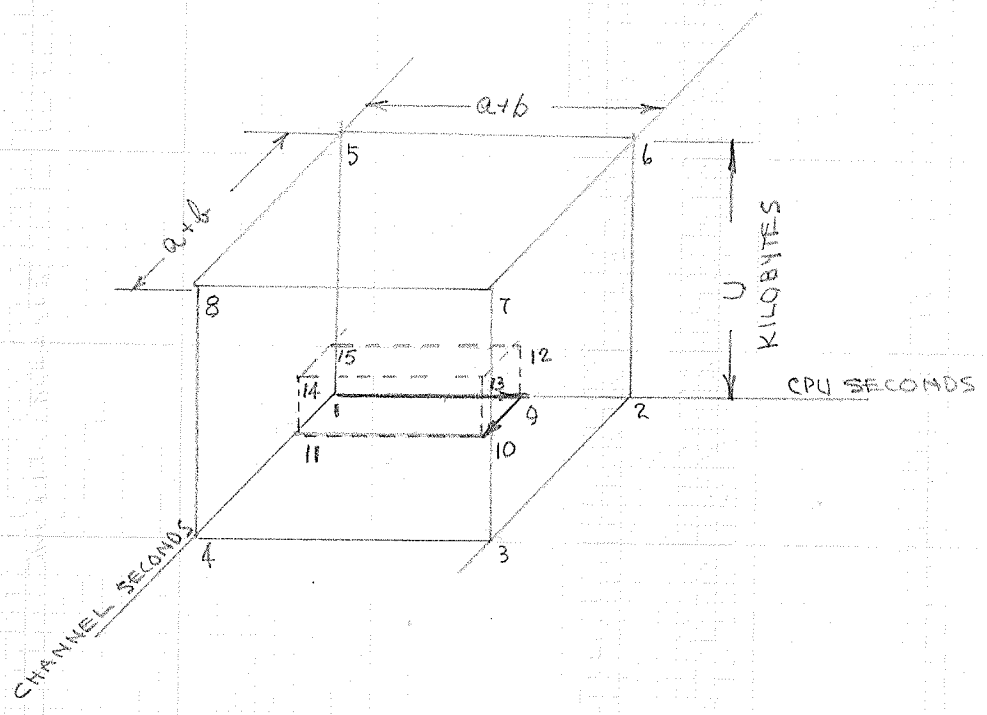
i.e., (D x batch region) kbsec of CPU resource

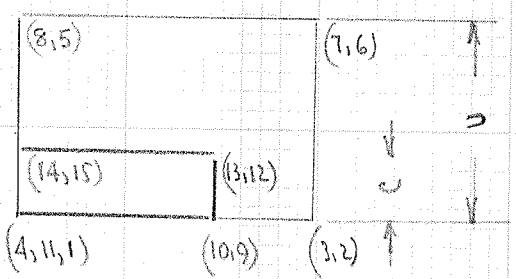
plus (D x batch region) kbsec of channel resource

A single job, standing alone, fits category c above because a single job is most unlikely to be able to use channel and CPU simultaneously on a non-parallel processing computer.

This development of A is illustrated in Plate 1 : if a job with CPU-seconds requirements of 'a', channel-seconds requirements of 'b', and kilobytes requirement of 'c', stands alone in batch region 'u', the job expends resources of 2 x (a+b)x u kilobyte-seconds but performs useful work of only (a+b) x c kilobyte-seconds. Additional jobs can be added without expending more resources provided only that the new jobs can be blocked into a route


CPU SECONDS
a
b
(5, 15, 1)
(12, 9)
(6, 2)
(13, 10)
(8, 4)
(7, 3)
c
CHANNEL SECONDS


a+b
a+b
5
6
8
7
15
12
14
13
9
11
10
2
4
3
d
KILOBYTES
CPU SECONDS
CHANNEL SECONDS


CPU SECONDS
(8,5)
(7,6)
(14,15)
(13,12)
(4,11,1)
(10,9)
(3,2)
d
c
KILOBYTES

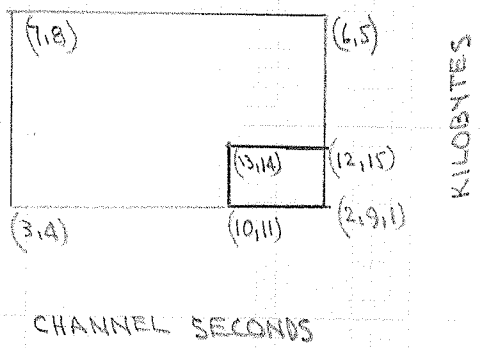

(7,8)
(6,5)
(13,14)
(12,15)
(3,4)
(10,11)
(2,9,1)
KILOBYTES
CHANNEL SECONDS

PLATE 1

between corner 13, and corner 7; i.e., a job of 'a' channel-seconds, 'b' CPU-seconds, and '(u-c)' kilobyte region would co-habit without expending more resources and would increase useful work by (a+b)(u-c) kbsec. Wasted resource would be reduced

from $2(a+b)u - (a+b)c = (a+b)u + (a+b)(u-c)$ kbsec.

to $2(a+b)u - (a+b)c - (a+b)(u-c) = (a+b)u$ kbsec.

for a net saving of (a+b)(u-c) kbsec.

This can be phrased more simply thus : any job which will fit in the unused core, and will require not more time than (a+b) seconds and will fit between $\binom{10}{13}$ and $\binom{7}{3}$ will improve the resource use.

The best job for improvement of resource use would be one whose $\frac{\text{CPU-sec}}{\text{channel-sec}}$ ratio $= \frac{b}{a}$, whose total time requirement is (a+b) and whose core requirement is (u-c).

Stated another way : the vector angle of this best match would be $(\pi/2 - \arctan(\frac{a}{b}))$ radians; i.e., a job which would nudge the vector angle of the resultant towards $(\pi/4)$ radians.

7) The analysis of paragraph 6 may not be entirely correct for a computer with multiple channels because it assumes that maximum channel activity is 1/.025 channel data transfers per second. As additional channels are installed it would seem likely that the system ultimately would be capable of servicing twice as many data transfers, or more. Unfortunately the job card

estimates do not specify which channels will be required. However in an installation with only one channel for 2314 disks and with only one channel for magnetic tapes and with most activity on disk, the analysis of paragraph 6 will be adequate.

Modifications to adapt to multiple channels, assuming that most data activity is with disk, follows, assuming that maximum attainable activity with perfect job balance is two channels working full time. Modifications for 1 1/2 channels, or 3/4 channel or 2 1/2 channels will follow the same rationale, but using suitable multipliers.

Let D = minimum system time for a family of jobs.

Then D = max(a,b,c) where

$$a = \sum_{i=1}^{n} a_i$$

$$b = \sum_{i=1}^{n} b_i$$

$$c = \max_{1 \leq i \leq n} (a+b)_i$$

Let A = total system resource expended

Then A = (3 x D x (batch region)) kbsec;

i.e., (D x batch region) of CPU resources

plus 2(D x batch region) of channel resources

Let B = total useful work performed

Then, as in paragraph 6,

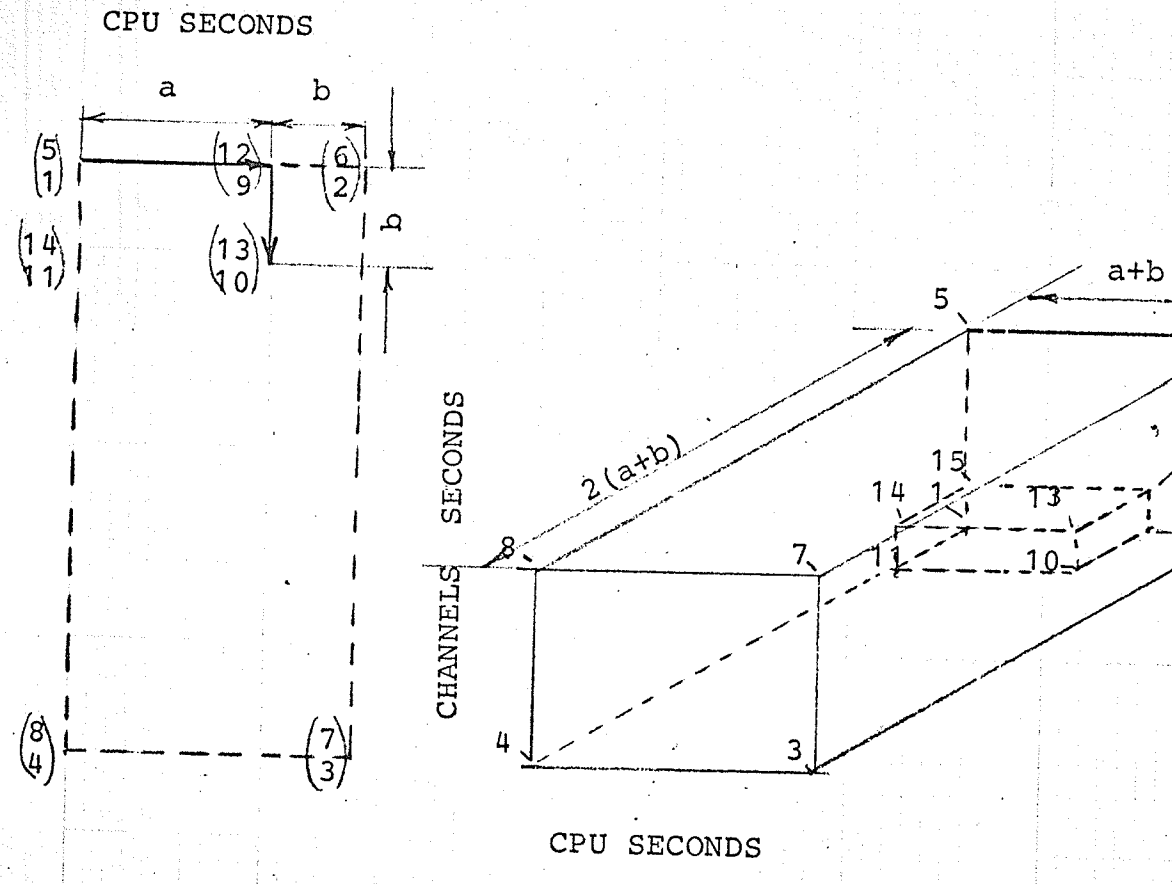
$$B = \sum_{i=1}^{n} R_i (a_i + b_i) \quad \text{kbsec}$$

where R_i = batch region

Let C = net wasted resource

Then C = (A-B) kbsec.

This development of A is illustrated in Plate 2. The logic follows the form of paragraph 6 : if a job with CPU-seconds requirements of 'a', channel-seconds requirements of 'b' and kilobytes requirements of 'c' stands alone in batch, or user, region 'u', the minimum job time is (a+b) seconds. During this time the system could perform; that is, expend system resources of, (a+b) x u kilobyte seconds of CPU work plus 2 x (a+b) x u kbsecs of channel work, if properly loaded. In fact, it cannot recover all this resource without parallel processing but this is a rational maximum limit. But the job performs only


CPU SECONDS
a
b
b
(5 1)
(12 9)
(6 2)
(14 11)
(13 10)
(8 4)
(7 3)
CHANNELS SECONDS
a+b
2(a+b)
u
5
6
8
7
15
14
1
13
12
2
11
10
4
3
CPU SECONDS

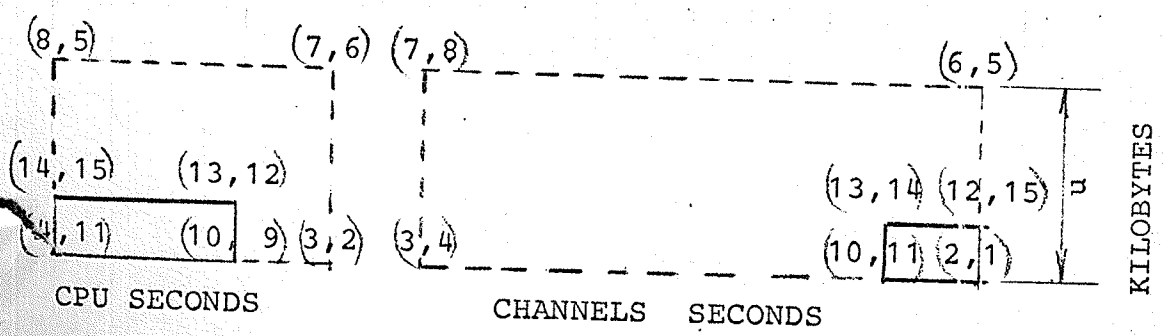

(8,5)
(7,6)
(7,8)
(6,5)
(14,15)
(13,12)
(13,14)
(12,15)
u
(4,11)
(10,9)
(3,2)
(3,4)
(10,11)
(2,1)
KILOBYTES
CPU SECONDS
CHANNELS SECONDS

PLATE 2

$(a+b)_i$ x $region_i$ kilobyte-seconds of useful work. Additional jobs can be added without expending more resources provided only that the new jobs can be blocked into a route between corner 13 and corner 7. The best single match at any stage of stacking is one with region requirement of (batch-region) - $\sum_{i=1}^{n} region_i$; and $(a_i + b_i)$ time requirements such that the vector angle, θ , of the best match would be

$$\theta = \text{Arctan}\left(\frac{2 \times Tmin - \sum_{i=1}^{n} b_i}{T\ min - \sum_{i=1}^{n} a_i}\right)$$

where Tmin is the D = minimum system time defined on the previous page; i.e., a job which would nudge the vector angle of the resultant vector

towards (π/3) radians.

8) The object of the next step is to rearrange jobs into groups such that complementary jobs are matched and, ideally, so that net wasted resource is minimized. One approach would be to compute net wasted resource for each job standing alone, sort the job list in descending order on net wasted resource, and match complementary jobs to the first job in the list; then to select the second if not already assigned and choose the best complements of remaining unassigned jobs; and so on until the list is exhausted.

This method has the disadvantage that the small-region jobs would show largest net wasted resource; i.e., the small-region jobs would head the list. Intuitively this procedure seems weak because the measure of net wasted resource is

distorted by large spare region; even the small job with vector angle of $\pi/4$ would appear inefficient, when in fact it is ideally efficient.

Another method would be to sort the jobs in descending order of region requirement and select hosts starting at the top of the list. This method seems reasonable because the large-region job will tend to waste much resource as its vector angle digresses from $\pi/4$ radians.

An alternative would be to sort the jobs in descending order of time and select hosts from the top. The disadvantage here resembles the disadvantages of the first proposal : the job is not necessarily wasteful because it is long, though any imbalance will be magnified by the time factor.

The method which was adopted is a combination of the second and third proposals : sort the jobs in descending order of kilobyte-seconds requirement and select hosts starting from the top of the list. This method has the advantage that the job at the top of the list carries both imbalances : wasted core and wasted time if the vector angle differs from $\pi/4$ and thereby offers much room for improvement.

The sorted list is illustrated in Table 8. So the largest job, job no. 51 of Table 8, is selected as first host. The unassigned members of the list should be scanned to find guests which would best complement job 51.

In the example study the users' core was restricted to 300 k-bytes, arbitrarily. If job 51 had required more than 300 k-bytes it would have

been passed over as a host, forever or until the users' core was increased sufficiently to accommodate. If job 51 had required 300 k, it would have been flagged as a host with no guests because no other job can cohabit. But reference to Table 1 shows that 51's core requirement was 130 k so it was selected as a host and flagged as such in Table 5. A search for suitable guests followed.

The guest(s) which would best complement job 51 would have characteristics :

Core requirement = 300 - 130 = 170 k-bytes

Vector angle = $\pi/2 - \theta$

= 1.5708 - .017 = 1.554 rad.

Time = Time of host = 6.10 seconds

Before selecting a guest to accompany job 51, a measure of the resources available to the guest must be established. Job 51 standing alone will waste :

Total resource expended = 300 x 2 x 6.1

= 3660 k-byte-seconds

Useful work performed = 6.1 x 130 = 793

Net wasted resource = 3660 - 793

= 2867 k-byte-seconds

So any guest who reduces this waste without extending the time is a paying guest, furthermore, the nicest guest is the one who uses the most waste.

The most significant variable of these three characteristics of the best guest is the vector angle. Compatible guests are those which swing the net vector angle towards $\pi/4$ radians, even if in tiny steps of time or core : they reduce waste by the amount of resource they use. So, using the argument $\theta = 1.554$, enter Table 9. There is no job in the list with this attribute, so job 29 should be considered, then 30, 40, 31, etc.. The questions to ask are : does the job under consideration, say no. 30, exceed the maximum time of 6.10 seconds, or exceed the maximum core of 170 bytes ? If yes, pass over it, if not compute waste with 51 and 30 cohabiting. Because there are several variables, it is well to consider a reasonably wide range of admissable guests. The program examines the ten admissable jobs closest to the Table 9 entry, then selects the one with smallest combined waste. In this example, as in

most, the job closest to the entry point turns out to be best; i.e., job 29. It reduces waste from 1037 k-byte-seconds to 911 k-byte-seconds. Job 29 is assigned, in Table 5, as a guest of job 51.

The search continues for another guest which will reduce waste further. The ideal guest has attributes :

Core requirement = 300 - 130 - 60 = 110 k-bytes

Vector angle = 1.55 radians

Time = 2.10 seconds

Job 30 turns out to be the best choice, reducing waste to 884 k-bytes.

The order of dispatch of family groups is not

significant in the context of resource economy. Any job may be given an arbitrary dispatch time without decreasing system efficiency solong as his family goes with him.

4 F. ILLUSTRATION OF THE FAMILY PLANNING RATIONALE.

Plate 3 will be used as a template for graphical illustration of the rationale used within the allocation program.

The plan view, (x, y plane), will trace the projection of CPU time and channel-time requirements of a host and his guests. The front elevation, (x, z plane), will depict the projection of (CPU-time) x (core) requirements of each job; i.e., the kilobyte second requirements imposed by the CPU-portion of the program. The side elevation will be used temporarily to trans-project the projection of (channel-time) x (core) onto the skew projection, (y, z plane). The side elevation will then be used to depict the cumulative core requirements of a family of jobs.

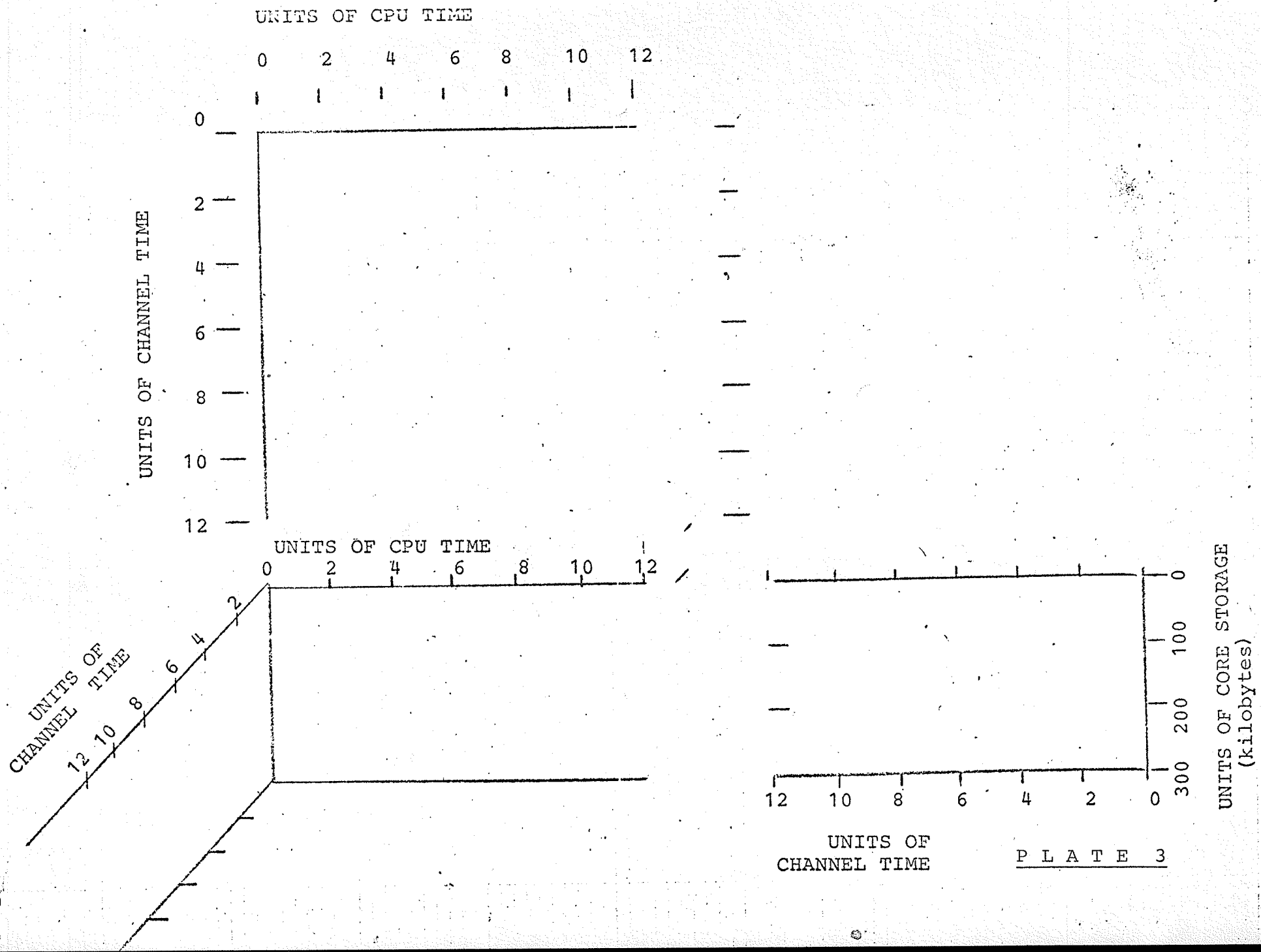
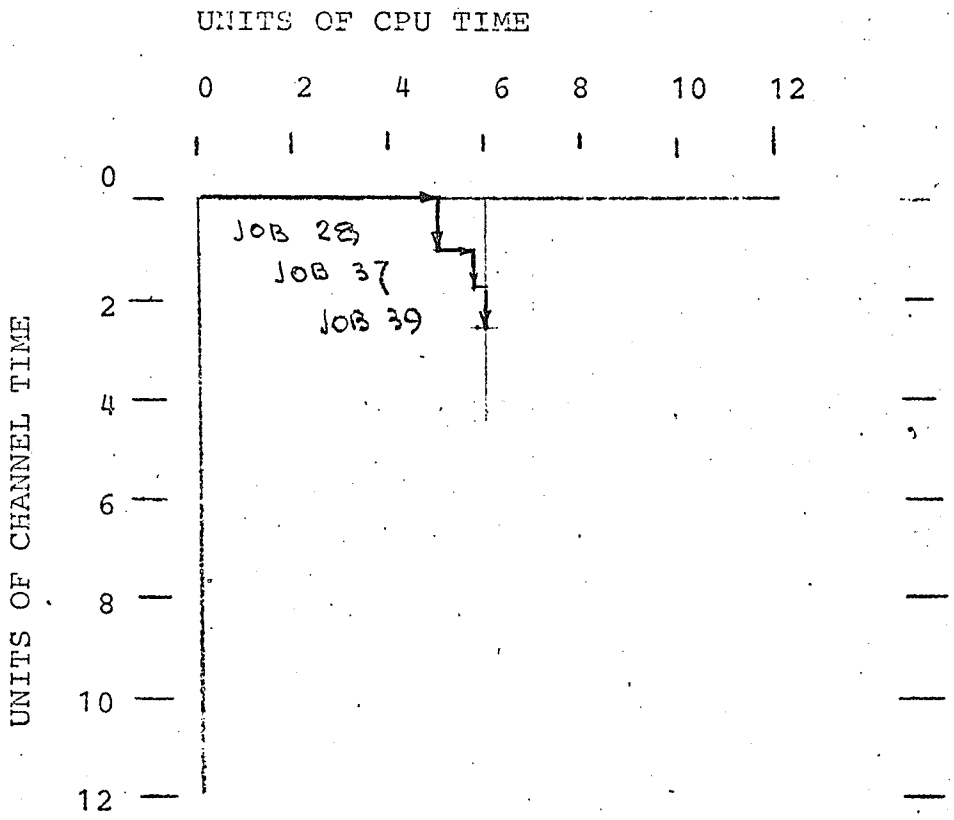

UNITS OF CPU TIME
0 2 4 6 8 10 12
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
UNITS OF CPU TIME
0 2 4 6 8 10 12
UNITS OF CHANNEL TIME
0 2 4 6 8 10 12
UNITS OF CHANNEL TIME
12 10 8 6 4 2 0
UNITS OF CORE STORAGE (kilobytes)
0 100 200 300

PLATE 3

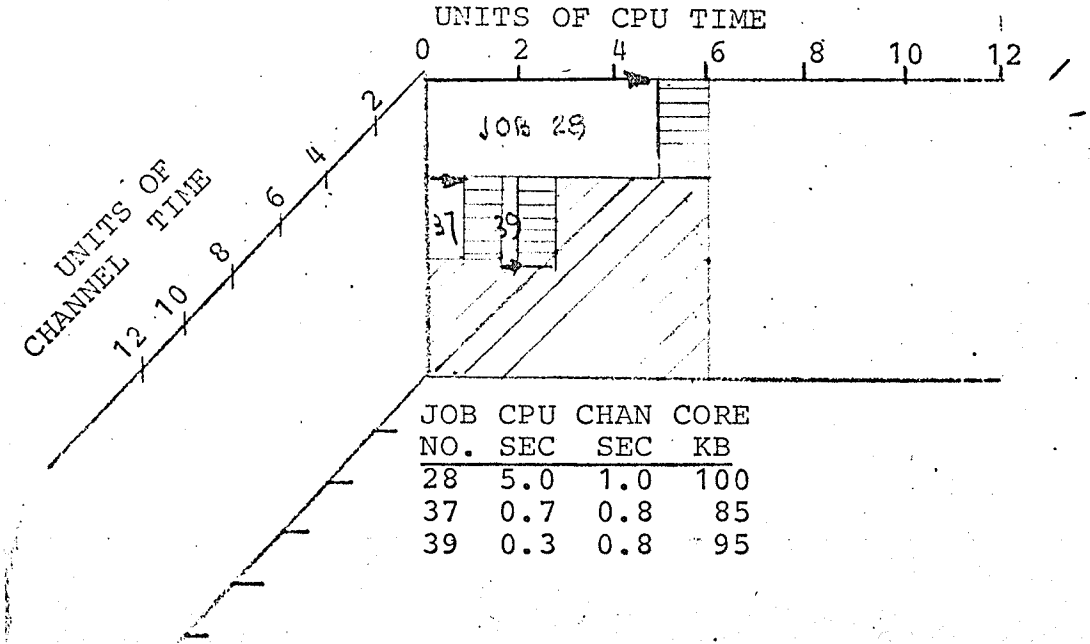
Plate 4 illustrates the dispatch of host 28 and his family. Job 28 reserves 100 k of core for 6.0 seconds. If the job stays in core longer, the extra time will be attributable to interference from other jobs; e.g., 0.7 seconds of CPU interference from job 37 and 0.3 seconds of CPU interference from job 39. The magnitude of this interference is a function of the job stream; a guest with optimal vector angle will introduce no contention, but as the vector angle deviates contention increases... However, the deviation is constrained by the rule that a job is not chosen as a guest unless it reduces the host's wasted resource.

Job 37 will occupy 85 kilobytes of core for (0.7 CPU + 0.8 channel) = 10.5 seconds. The side elevation illustrates that another job with time not greater than (6.0 - 1.5) = 4.5 seconds and



Horizontal hatching indicates commited waste.

Skew hatching indicates waste that accrued simply because the artificial job queue was tiny.



JOB NO.	CPU SEC	CHAN SEC	CORE KB
28	5.0	1.0	100
37	0.7	0.8	85
39	0.3	0.8	95

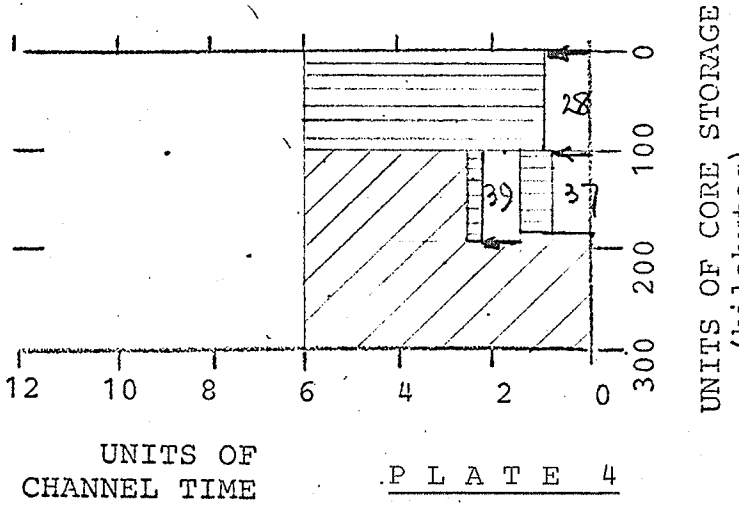
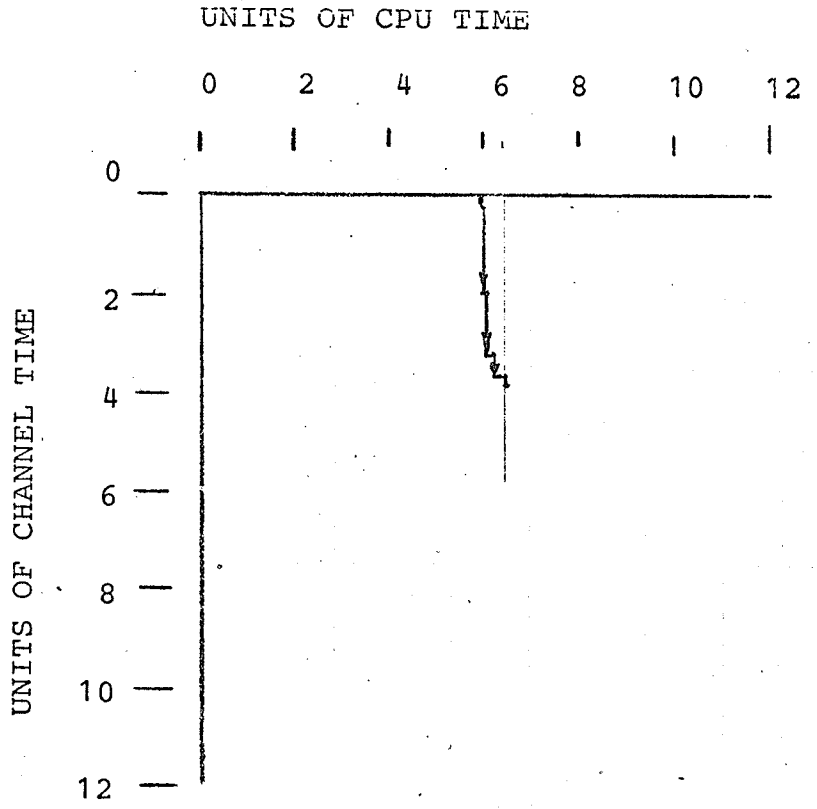


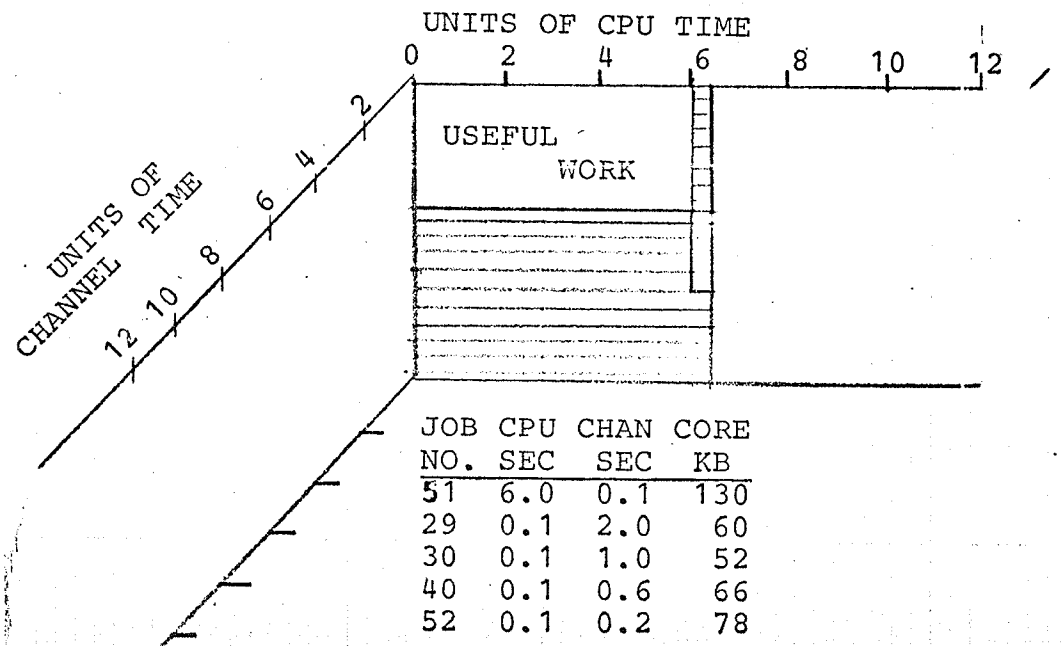
PLATE 4

core requirements not greater than 85 kb could take over 37's core assignment when 37 completed. The sample job queue does not contain an adequate co-guest.

Similarily job 39 will occupy 95 kb of core for only 1.1 seconds, leaving this 95 kb resource idle 4.9 seconds of the 6.0 reserved by the host. The program will search the job queue to select jobs to use some of this wasted core resource. The job 37 family wasted also (6.0 - 2.6) = 3.4 seconds of channel time or 3.4 (300) = 1020 kbsec. of channel resource. In practice these empty slots would be mostly filled whenever there was a reasonable large job queue.

Reference to Plate 4 will further illustrate the rationale of dispatching for multi-channels : provided that the core allocation, shown on the





JOB NO.	CPU SEC	CHAN SEC	CORE KB
51	6.0	0.1	130
29	0.1	2.0	60
30	0.1	1.0	52
40	0.1	0.6	66
52	0.1	0.2	78

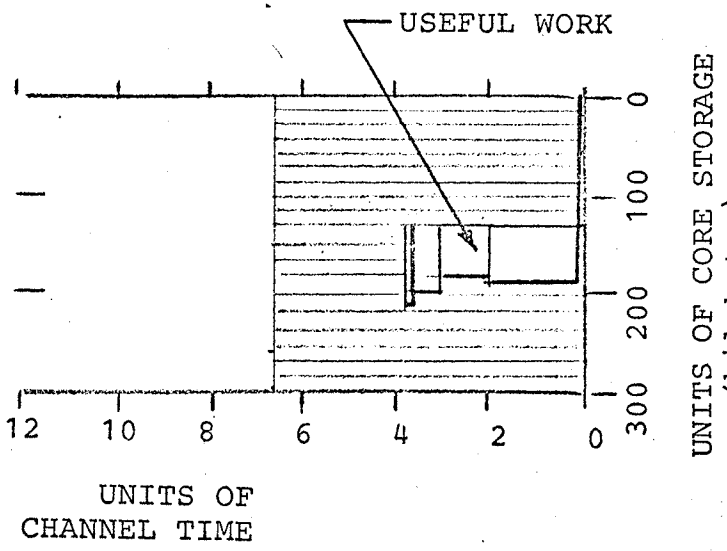


PLATE 4 a

end-elevation view, permits, the channel times need not be exclusive : they might overlap 50 % or even 100 %. For example, job 37's channel activity might slide under job 28's channel activity to the half-way point; similarily job 39's channel activity might slide under 37's.

There is no rationale offered here for the ratio of overlap; it is a function of on-line storage use and channels configuration. The ratio could be established by developing a few long jobs with high channel use and running them concurrently to measure maximum channel activity.

4 G. SELECTION OF RATIONALE.

The techniques developed here take the form they do because there are so many unpredictable factors in the work load. If a method were developed and demonstrated on a set of bench-mark tests, as is often done, the evidence of validity would be lost unless the bench-mark programs were representative of the real job stream. But no set of bench-marks can remain valid in an environment that fluctuates as wildly as a university's computer job stream.

In developing a measure of goodness, the investigator hoped to find some recognizable pattern in the use of plant resources operating under a constant dispatch philosophy and so chose to test actual operating conditions in preference to a sample of the plant activity. He hoped to find, for example, that plant efficiencies were

constant from 5.00 to 6.00 p.m. each day. This hope was not realized, though interesting patterns have developed. If conclusive patterns had developed, effects of system modifications might have been measured by applying the modifications only during a stable period; but the disadvantage of having to measure a longer period is no more significant than having to process more data : efficiency changes, positive or negative, will show up on the plots.

Hopefully the validity of the rationale for formation of family groups can be judged on its logic. Implementation of the technique would be satisfying, but permission to implement it might be elusive. It is the belief of the investigator that HASP could be modified rea dily to accommodate the coding.

Chapter 5.

PROPOSED EXTENSIONS TO THE STUDY.

This investigator believes that the rationales and programs developed for this thesis are useful management tools. The following lines would be interesting and valuable pursuits :

1) Install the scheduling proposal, along with HASP, for an on-line test over a period of a few weeks.

2) Develop criteria for judging when and how to compromise where minimum-waste scheduling upsets arbitrary or necessary priorities.

3) Develop a procedure to forecast the resources required by a program on the basis of past performance and a knowledge of the compiler involved. That is, to predict a job profile which is more informative than CPU time, channel time, and region requirement, as specified by the user.

4) Examine, via the reporting routines, the effect of reconfiguring the system one phase at a time; e.g., by changing the way data channels are shared between devices ..., to gain an understanding of whether or not improvement might accrue by rearranging.

5) Establish an understanding of what factors contribute most significantly to low efficiencies. Intuitively one would suspect that when the system is working hard on CPU jobs, with little else in execution that the efficiency would be low because

(channels x region) waste is high. Perhaps it is low, but it is even lower when the high CPU jobs are not in execution. This would suggest the possibility of a tremendous increase in throughput of the plant with minimum-waste scheduling.

Chapter 6.

SUMMARY AND CONCLUSIONS.

The investment in computing plant is tremendous and increasing . Guidelines for sane management of these plants should be established by computer science practitioners.

Simple criteria for judging system performance have not been established, heretofore. The author hopes to have isolated the essence of an overall understanding of the problems and to have developed an interpretation of events which will be informative to a wide range of observers. The

manager with a weak vocabulary in mathematical terms should derive a good understanding of what happened by even a cursory examination of the pictorial performance report. This understanding may encourage him to examine the results more closely for a deeper understanding. But the manager with a strong background in mathematical symbolism may also glean the information he wants from the same reports . Both will probably reach identical understanding; if they do the report has more than fulfilled its function.

Techniques are improving for measuring and reporting job-step use of resources. These improvements and refinements will enhance the reporting method described herein; the programs can be modified to adapt to more or better measurements.

The author suggests that this method of reporting might be attractive to computer installations because it is easy to interpret, yet it loses none of its generality through being simple. He further suggests that any method that will tell a computer manager that from one date to another his plant's efficiency increased (or decreased) is a marked improvement over any method available to the manager today. If this technique can perform that desired result it may be a useful addition to the science.

Given a satisfactory and simple method for measuring the effect on system efficiency of a change in operating procedure or in equipment configuration, one should be able to methodically examine those variables he considers to be significant and order those variables on their relative significance. He may thus gain a better

understanding of his plant and its operation. This is the scientific method. A method for measuring the effect of changes is presented herein.

It is the author's firm belief that much of the system resources wasted today in most computer installations could be reduced significantly by adopting a family planning rationale for assessing the order of dispatch of jobs from the input queue to the central processor . The dispatching rules derived by the family planning rationale may cause concern for some users, but not necessarily so. Existing rules for special jobs may be consistent with family planning rules; the family planning rationale accepts arbitrary imposition of dispatch rules, provided that they apply to only selected jobs.

The only supra-ruling imposed by family planning

is that when a family member goes to core from the input queue, the remainder of his family accompanies him.

The author believes that the low efficiencies experienced, between 2.00 and 5.00 p.m. daily, at the test site result from high resource waste brought on by a selection procedure which assures an imbalanced load when the job queue is long.

One of the variables which seem to warrant study is the order of presentation of problem programs from the input queue to the central processor. A rationale for efficient ordering is presented for consideration.

REFERENCES

1. Vincent H. Swoyer and Michael F. Armstrong, " Timing and Charging Computer Use on the IBM 360 model 65 in a Multiprogrammed Environment", Technical Report 2.1.1969, University of Rochester, p.9, February, 1969.
2. R. Sedgewick, R. Stone, J.W. McDonald, Spy -"A Program to Monitor OS/360", AFIPS Conference Proceedings, 1970 Fall, Joint Computer Conference, volume 37, pp.119-128.
3. Bruce S. Marshall,"Dynamic Calculation of Dispatching Priorities under OS/360 MVT", May 9, 1969, Lear Siegler, Inc.
4. Bruce Brynolfson,"Resource Scheduling Extension to the Houston Automatic Spooling and Priority System", Grumman Data Systems Corporation.(Undated, preliminary copy, received August, 1970).
5. Daniel Lasser,"Productivity of Multiprogrammed Computers - Progress in Developing an Analytic Prediction Method", Communications of ACM, December, 1969, pp.678-684.
6. Ralph Lawrence, July 2, 1968, SHARE SSD C5018, pp.1-12.
7. Shope, Kashmarak, Ingram and Decker,"Systems Performance Study", Proceedings of SHARE XXXIV, March, 1970, pp.569-629.
8. Richard Watson,"Using Accounting Data as an Alternative to Performance Monitors", Proceedings of SHARE XXXVI, March 1971, Los Angeles. (Not yet published)

APPENDIX 1

Program MKCC

PROGRAM MKCC

FUNCTION.

To establish unit costs of the six separate computer operations which can be and are logged: CPU time; CHANNEL time; CORE size and time, cards read, cards punched, and lines printed

DATA

1. Inventory file of manufacturers' charges.
2. Past records of activity in the six items cited above.

REFERENCE

A description of the method is given in section 2D of the text, pp 32-35.

APPENDIX 1.

```
KCC    JOB '00065,XX,01,T=10,K=130K',PROGRAMMER,WALTER DORAN,MSGLEVEL=1
 EXEC WATFIV,SIZE=130K                                                  00010001
ATFIV  PROC  SIZE=120K                                                  00020001
0    EXEC  PGM=WATFIV,REGION=&SIZE
553I SUBSTITUTION JCL - PGM=WATFIV,REGION=130K
T05F001  DD  DDNAME=SYSIN                                               00030001
T06F001  DD  SYSOUT=A,DCB=(RECFM=FA,BLKSIZE=133)                        00040001
T07F001  DD  SYSOUT=B                                                   00050001
IBUSAGE DD DSN=SYS2.WATFIV.LIBUSE,DISP=SHR                              00060001
TEPLIB  DD  DSN=SYS1.WATFIV,DISP=SHR                                    00070001
YSPRINT  DD  SYSOUT=A                                                   00080001
ATLIB DD DSN=SYS1.WAT5FUNC,DISP=SHR                                     00090001
   DD  DSN=SYS2.WATSSP,DISP=SHR                                         00100001
D.SYSIN  DD  *

373I STEP /GO      / START 71054.1756
374I STEP /GO      / STOP  71054.1758 CPU  00MIN 03.37SEC MAIN 130K LCS    0K
        104.83 SEC EXEC TIME     3.37 SEC CPU TIME     155 I/O COUNTS 130K REGION 130K USED

375I  JOB /MKCC    / START 71054.1756
376I  JOB /MKCC    / STOP  71054.1758 CPU  00MIN 03.37SEC
        104.83 SEC EXEC TIME     3.37 SEC CPU TIME       DATE 71.054 UNIVERSITY OF MANITOBA



P-II JOB STATISTICS --    303 CARDS READ --      684 LINES PRINTED --       0 CARDS PUNCHED --    1.74 MINUTES EXECUTION TIME
VERSITY OF MANITOBA --    171 I/O COUNTS --  1,724 K BYTE-SECONDS -     1.45 UNITS          --    0.05 MINUTES CPU TIME
INE ACCOUNTING INFO --    577 JOB(S) RUN --                          $$7216.19 REMAINING
```

```
      C  TO ESTABLISH UNIT RATES FOR COMPUTER USAGE JUNE 1969 WH.D.
     1      DIMENSION T(100,21),A(22),UP(21)
     2      DATA  A/22*0./
     3      DATA  T/2100*0./
     4      DATA  UP/21*0./
     5      INTEGER ITEM (900)
      C  REGISTER ANNUAL TOTALS OF:
      C         A(1)      CHANNEL COUNTS
      C              JULY & AUG AVERAGE GIVES          39,865,000 SVC COUNTS/YR
      C              (NOV'69 * 12MOS) SHOWS 129,540,168 SVC COUNTS/YR
      C  JUNE '70: ANNUAL CHANNEL COUNT EXTRAPOLATED FROM JAN THRU APRIL'70
      C            CHANNEL COUNTS*(MAY THRU APRIL CPU/ JAN-APR CPU) FOR EACH
      C            OF FACULTY, COMPUTER CENTRE, & CUSTOM WORK
      C            91705 *(34,968,117/25763) + 34384,081 *(4624/1310)+
      C            10,199,101*(17555/4426)=(159.3+121.2+40.5) MILLION
      C            =  321 MILLION COUNTS /YR
      C         A(2)      CORE IN K * HRS
      C              MAY THRU AUG'69 EXPERIENCE SHOWS      837,000 K-BYTE-HRS
      C              NOVEMBER USE  EXTRAPOLATED GIVES 253,076 K-BYE-HRS/YR
      C  JUNE'70  ANNUAL KBSCC COUNT EXTRAPOLATED BY SAME CPU RATIOS USED FOR
      C            CHANNEL COUNTS TO GIVE:(1310+225+265)MILLION KB-SCC
      C            OR 500000 KB-HRS./YR.
      C         A(3)      CPU HRS
      C              MAY THRU AUG'69 EXPERIENCE SHOWS        2,912 HRS/YR
      C              NOVEMBER '69 EXTRAPOLATED GIVES 2169 HRS/YR CPU USE
      C  JUNE'70: ANNUAL CPU HRS LOGGED FROM MAY'69 THRU APRIL'70 = 2328
      C         A(4)      PUNCHING HOURS
      C  JUNE'70: ANNUAL PUNCHING HRS DERIVED FROM MAY'69 THRU APRIL'70 FOR
      C            ALL JOBS LESS AECL,LESS MED COLL = 2084099 AT 150 CPM
      C            OR 900 CPHR = 231.5 HRS/YR
      C         A(5)      PRINTING HOURS
      C              MAY THRU AUG'69 EXPERIENCE GIVES        2,328 HRS/YR
      C  JUNE'70: ANNUAL PRINTING HRS DERIVED FROM MAY'69 THRU APRIL'70 FOR
      C            ALL JOBS,LESS ARCL,LESS MED COLL=145594715 AT 900 LPM
      C            OR 54000 LINES / HR = 2695.3 HRS/YR
      C         A(6)      READING HOURS
      C              MAY THRU AUG'69 EXPERIENCE GIVES          819 HRS/YR
      C  JUNE'70 ANNUAL READ DERIVED FROM MAY'69 THRU APRIL'70 FOR
      C            ALL JOBS,LESS AECL,LESS MED COLL = 465,590,773 AT
      C            1000 CPM OR 60 000 CARDS/HR = 776.5 HRS/YR
      C ***     A(7)      TOTAL COMPUTER-CENTRE ANNUAL EXPENDITURE  TEMPORARY
      C ***     A(22)     PROFIT RATIO    *** ARBITRARY ***
      C  CHKSUM WILL CONTAIN THE REVENUE ACCRUING FROM EXTENSION OF
      C  DERIVED UNIT PRICES
     6      CHKSUM = 0.
     7      A(1) = 76000000.                                          *******
     8      A(1) = 39865000.                                          *******
     9      A(1) = 1.25E8
    10      A(1)=1.295E8
    11      A(1) = 3.21E8
    12      A(2) = 361000.                                            *******
    13      A(2) = 837000.                                            *******
    14      A(2)= 253076.
    15      A(2) = 500000.
    16      A(3) = 1559.                                              *******
    17      A(3) = 2912.                                              *******
    18      A(3)= 2169.
    19      A(3) = 2328.
    20      A(4) = 553.                                               *******
    21      A(4) = 231.5
```

```
23          A(5) = 2312.                                                    *******
24          A(5) = 2695.
25          A(6) = 1443.                                                    *******
26          A(6) =  819.                                                    *******
27          A(6) = 776.5
28          A(22)=1.25                                                      *******
29          A(22)=1.75                                                      *******
30          A(22)=2.00                                                      *******
31          A(22) = 1.00                                                     ******
32          A(22)=1.50                                                      *******
      C EXTINT = 0. IF EDUCATIONAL ALLOWANCES ARE NOT DEDUCTED
33          EXTINT = 0.
      C  EXTINT = 1. IF EDUCATIONAL ALLOWANCES ARE DEDUCTED
34          EXTINT = 1.
35          N=1
36          LINES = 1
37          NN=9*N -8
38          MM=NN +8
39          WRITE (6,115) (A(I), I = 1,6)
40          WRITE (6,110)
41          WRITE (6,111)
42          WRITE (6,106)
43        1 READ (5,101,END=77)(ITEM(I),I=NN,MM),(T(N,J),J=1,13)
44          WRITE (6,112)  (ITEM(I), I = NN,MM)
45          WRITE (6,113)  (ITEM(I), I = NN,MM)
46          WRITE (6,114)  (T(N,J), J = 1,13)
47          IF  (LINES - 10) 10,  9, 10
48        9 LINES = 1
49          WRITE (6,110)
50          WRITE (6,111)
51          WRITE  (6,106)
      C    T(N,1) IS GROSS PURCHASE PRICE OF ITEM
      C    T(N,2)  IS FEDERAL SALES TAX RATE
      C    T(N,3)  IS EDUCATIONAL DISCOUNT
      C   T(N,4) IS CARRYING CHARGE FOR FINANCING   ONE-TIME CHARGE
      C    T(N,5) IS PROVINCIAL SALES TAX RATE
      C    T(N,6) IS EQYIPMENT WRITEOFF PERIOD IN MONTHS
      C    T(N,7) IS ANNUAL RENT OR RECURRINGG ANNUAL COST
      C    T(N,8) IS PORTION OF COST DISTRIBUTED TO CHANNEL
      C    T(N,9) IS PORTION OF COST DISTRIBUTED TO CORE
      C    T(N,10) US PORTION OF ITEM COST DISTRIBUTED TO CPU
      C    T(N,11) US PORTION OF ITEM COST DISTRIBUTED TO PUNCH
      C    T(N,12) US PORTION OF ITEM COST DISTRIBUTED TO PRINT
      C    T(N,13) US PORTION OF ITEM COST DISTRIBUTED TO READ
      C   T(N,14) IS TOTAL ANNUAL COST OF ITEM
      C   T(N,15) IS ANNUAL COST ALLOCATED TO CORE USAGE
      C   T(N,17) IS COST TO PUNCH
      C   T(N,18) IS COST TO PRINTING
      C   T(N,19) IS COST TO READING
      C   T(N,21) IS COST NOT DISTRIBUTED TO T (I,15 THRU 20)
      C   T(N,20) IS COST TO CHANNELS
52        10 IF (T(N,6)) 7,6,7
53         6 T(N,6) = 12
54         7 AA= T(N,1) * T(N,4)
55           B = T(N,1) * T(N,2)
56           C = (T(N,1)-B) * T(N,3) *  EXTINT
57           D = (T(N,1) - B - C) * T(N,5)
58           T(N,14) = (((T(N,1) + AA- B - C + D) / T(N,6))*12.) + T(N,7)
59             DO  2    K=8,13
60                J= K+7
```

```
      2 CONTINUE
        T (N,21) = (1.0000-(T(N,8)+T(N,9)+T(N,10)+T(N,11)+T(N,12)
     1 +T(N,13))) * T(N,14)
        WRITE (6,109)                    (T(N,J),J=14,21)
        N= N+1
        LINES  =  LINES + 1
        GO TO 1
     77 NMAX = N
C COMPUTE  UNIT  COSTS:
C          A(15)     TOTAL COST DISTRIBUTED TO CHANNELS SUM T (I,8)
C          A(16)     TOTAL COST TO CORE                 SUM T (I,9)
C          A(17)     TOTAL COST TO CPU                  SUM T (I,10)
C          A(18)     TOTAL COST TO PUNCHING             SUM T (I,11)
C          A(19)     TOTAL COST TO PRINTING             SUM T (I,12)
C          A(20)     TOTAL COST TO READING              SUM T (I,13)
C          A(14)     TOTAL COST     , A(8 THRU 13)
C          A(16)     $/CHANNEL CALL=((A(15)* A(8))/A(14))/A(1)
C          A(17)     $/K-HR OF CORE = ((A(15)* A(9))/A(14))/A(2)
C          A(18)     $/CPU-HR=((A(15)*A(10))/A(14))/A(3)
C          A(19)     $/PUNCH-HR=((A(15)*A(11))/ A(14))/A(4)
C          A(20)     $/PRINT-HR=((A(15)*A(12)/A(14))/A(5)
C          A(21)     $/READ-HR=((A(15)*(13)/A(14))/A(6)
        DO 4  J=14,21
        DO 4  I = 1,N
          A(J) = A(J)+ T(I,J)
      4 CONTINUE
         WRITE(6,105)  (A(I),  I = 14,21 )
        TOHXM = (A(14)* A(22) ) / ( A(14) - A(21) )
         DO 3 I=1,6
            K=I+14
        UP(K) = ( A(K) * TOHXM ) / A(I)
      3 CONTINUE
    101 FORMAT (9A4,    F6.0,  F3.3,F2.2,F3.3,  F2.2,F3.0,F6.0,1X,6F3.2)
    102 FORMAT (1H ,9A4//F9.0,2P F4.1, F4.0, F4.0, F4.0, 0PF4.0,F7.0,1X,
     X  2P 6F5.0, 0P 8F7.0)
    103 FORMAT( 1H1,T5,'$/CHANNEL-COUNT',T34,'$/KB-HR',T49,'$/CPU-HR',
     1 T69,'$/PUNCH-HR',T89,'$/PRINT-HR',T109,'$/READ-HR')
    104 FORMAT(1H0, 6F20.10)
    105 FORMAT (11H0SUB-TOTALS,  55X, F8.0, 7F8.0)
    106 FORMAT(1H0,T11,'FED',T78,'CHAN',T87,'CORE',T95,'CPU',T102,'PUNCH',
     1 T110,'PRINT',T119,'READ',T125,'UNA/C',
     2T15,'ED',T18,'INT',T22,'PROV'        ,T37,'CHAN',T42,'CORE',
     3T47,'CPU',T51,'PUNCH',T57,'PRINT',T63,'READ'//)
    107 FORMAT (1H-, 20HUNIT RATES A/C FOR $, F10.0,
     1 25H  OF ANNUAL EXPENDITURE:$, F10.0)
    108 FORMAT (1H1, 21H EXPECTED REVENUE @  , F6.2, 10H MARKUP :$, F10.0)
    109 FORMAT  (1H , 65X, F8.0,  7F8.0)
    110 FORMAT (1H1,T3,'GROSS',T25,'LIFE' ,T33,'YR',T42,
     1'0/0 DISTRIBUTION TO',T70,'EQUIV')
    111 FORMAT (1H ,T82,'DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS',
     1 T2,'PURCHASE',T11,'ADJUSTMENTS',T27,'MO',T32,'RENT',
     2T43,'MEASURABLE ITEMS',T69,'YR-COST')
    112 FORMAT  (1H0,9A4)
    113 FORMAT  (1H+,9A4)
    114 FORMAT (1H0,     F9.0,2P F4.1, F4.0, F4.0, F4.0, 0PF4.0,F7.0,1X,
     X  2P 6F5.0, 0P 8F7.0)
    115 FORMAT (1H ,6E14.5)
    116 FORMAT( 1H0, T33,'$/KB-SEC', T48,'$/CPU-MIN',T69,'$/PUNCH-CD',
     1 T89,'$/LINE', T109,'$/READ-CD')
    117 FORMAT (1H0, 20X, 5F20.10)
```

```
 97          WRITE (6,104) UP(15), UP(16), UP(17), UP(18), UP(19), UP(20)
 98          UKBSEC = UP(16) / 3600.
 99          UCPUM = UP(17) / 60.
100          UPUNCH = UP(18) / 900.
101          UPRINT = UP(19) / 54000.
102          UREAD = UP(20) / 60000.
103          WRITE (6,116)
104          WRITE (6,117) UKBSEC, UCPUM, UPUNCH, UPRINT, UREAD
105          DO 8  I = 1,6
106          J = I + 14
107          CHKSUM = ( A(I) * UP(J) ) + CHKSUM
108        8 CONTINUE
109          CHKSUM = CHKSUM / A(22)
110          WRITE (6,107)  CHKSUM, A(14)
111          CHKSUM = CHKSUM * A(22)
112          WRITE (6,108)  A(22),  CHKSUM
113           CALL EXIT
114          END
ARNING**  END STATEMENT NOT PRECEDED BY A TRANSFER
RNING**  FORMAT STATEMENT    102 IS UNREFERENCED

     $ENTRY
.32100E 09   0.50000E 06   0.23280E 04   0.23150E 03   0.26950E 04   0.77650E 03
```

CHASE ADJUSTMENTS	MO	RENT	MEASURABLE ITEMS						YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED ED INT PROV			CHAN	CORE	CPU	PUNCH	PRINT	READ		CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
311 MOD 001 DISK DRIVES																
17880. 6.8 20. 11. 00. 48.	0.	44.	00.	00.	6.	35.	15.		25214.	11094.	0.	0.	1513.	8825.	3782.	0.
316 MOD 001 DISK PACKS																
5085. 6.8 20. 11. 00. 48.	0.	00.	00.	00.	00.	00.	00.		1088.	0.	0.	0.	0.	0.	0.	1088.
316 MOD 001 DISK PACKS																
5085. 6.8 20. 11. 00. 48.	0.	100.	00.	00.	00.	00.	00.		1088.	1088.	0.	0.	0.	0.	0.	0.
065 MOD 965 SER60059																
61980. 6.8 40. 11. 00. 48.	0.	11.	00.	81.	1.	5.	2.		110749.	12182.	0.	89707.	1107.	5537.	2215.	0.
365 MOD 001 SER30259 CORE																
19300. 6.8 40. 11. 00. 48.	0.	6.	60.	20.	00.	10.	4.		53419.	3205.	32051.	10684.	0.	5342.	2137.	0.
052 MOD 007 50807																
3190. 6.8 20. 11. 00. 48.	0.	00.	00.	100.	00.	00.	4.		682.	0.	0.	682.	0.	0.	27.	-27.
365 MOD 002 SER30539 CORE																
00040. 6.8 40. 11. 5. 48.	0.	6.	60.	20.	00.	10.	4.		24373.	1462.	14624.	4875.	0.	2437.	975.	0.
65 MOD H-I CORE																
2240.00.0 30. 00. 00. 48.	0.	6.	60.	20.	00.	10.	4.		392.	24.	235.	78.	0.	39.	16.	0.
65 CORE																
2981.00.0 00. 00. 00. 48.	87318.	6.	60.	20.	00.	10.	4.		88063.	5284.	52838.	17613.	0.	8806.	3523.	0.
03 MOD ON1 SER31611 PRINTER																
8110. 6.8 20. 11. 00. 48.	0.	00.	00.	00.	00.	100.	00.									

CHASE ADJUSTMENTS	MO	RENT	MEASURABLE ITEMS						YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED ED INT PROV			CHAN	CORE	CPU	PUNCH	PRINT	READ		CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
									10291.	0.	0.	0.	0.	10291.	0.	0.
403 NEW PRINTER																
48110. 6.8 20. 11. 00. 48.	0.		00.	00.	00.	00.	100.	00.	10291.	0.	0.	0.	0.	10291.	0.	0.
416 MOD 001 SER14249 TYPECHAIN																
3375. 6.8 20. 11. 00. 48.	0.		00.	00.	00.	00.	100.	00.	722.	0.	0.	0.	0.	722.	0.	0.
442 MOD 0N2 SER50142 PUNCH																
21655. 6.8 20. 11. 00. 48.	0.		00.	00.	00.	100.	00.	00.	4632.	0.	0.	0.	4632.	0.	0.	0.
401 MOD 002 SER21400 TAPEUNIT																
26740. 6.8 20. 11. 00. 48.	0.		100.	00.	00.	00.	00.	00.	5720.	5720.	0.	0.	0.	0.	0.	0.
404 MOD 002 SER12314 TAPE&CONTROL																
76460. 6.8 20. 11. 00. 48.	0.		80.	00.	20.	00.	00.	00.	16355.	13084.	0.	3271.	0.	0.	0.	0.
501 MOD 0B2 SER20072 CARDREADER																
7640. 6.8 20. 11. 00. 48.	0.		00.	00.	00.	00.	00.	100.	3773.	0.	0.	0.	0.	0.	3773.	0.
21 MOD 002 SER20395 PRINT CNTRL																
6460. 6.8 20. 11. 00. 48.	0.		00.	00.	00.	00.	100.	00.	7799.	0.	0.	0.	0.	7799.	0.	0.
41 MOD 001 SER30393 2311 FILE C																
0000.00.0 00. 00. 00. 0.	0.		44.	00.	00.	6.	35.	15.	30000.	13200.	0.	0.	1800.	10500.	4500.	0.
60 MOD 001 SER60123 SELECTOR																
6520. 6.8 20. 11. 00. 48.	0.		75.	00.	25.	00.	00.	00.								

OSS CHASE ADJUSTMENTS				LIFE MO	YR RENT	0/0 DISTRIBUTION TO MEASURABLE ITEMS							EQUIV YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
	FED	ED	INT	PROV		CHAN	CORE	CPU	PUNCH	PRINT	READ			CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
													24924.	18693.	0.	6231.	0.	0.	0.	0.
360 MOD 003 SER60123 SELECTOR																				
99175.	6.8	20.	13.	5.	48.	0.	75.	00.	25.	00.	00.	00.	22609.	16957.	0.	5652.	0.	0.	0.	0.
3314	001		DISK DRIVE																	
0.00.0	00.	00.	00.	0.	-16000.	80.	00.	20.	00.	00.	00.		-16000.	-12800.	0.	-3200.	0.	0.	0.	0.
3314	001		DISK DRIVE																	
0.00.0	00.	00.	00.	0.	64000.	80.	00.	20.	00.	00.	00.		64000.	51200.	0.	12800.	0.	0.	0.	0.
3870 MOD 001 SER60084 MULTIPLEXOR																				
56580.	6.8	20.	11.	00.	48.	0.	50.	00.	00.	10.	20.	20.	33492.	16746.	0.	0.	3349.	6698.	6698.	0.
3816																				
1680.	00.0	00.	00.	5.	48.	0.	68.	00.	32.	00.	00.	00.	441.	300.	0.	141.	0.	0.	0.	0.
3816																				
3913.	00.0	00.	00.	5.	48.	0.	100.	00.	00.	00.	00.	00.	1027.	1027.	0.	0.	0.	0.	0.	0.
				PLOTTER																
29000.	00.0	00.	00.	5.	48.	0.	00.	00.	00.	00.	00.	00.	7613.	0.	0.	0.	0.	0.	0.	7613.
				TELETYPE																
0.00.0	00.	00.	00.	0.	1416.	00.	00.	00.	00.	00.	00.		1416.	0.	0.	0.	0.	0.	0.	1416.
50/65 PURCHASE EQUP MAINTENANCE																				
0.00.0	00.	00.	00.	0.	30000.	00.	00.	00.	00.	00.	00.									

CHASE ADJUSTMENTS	MO RENT	MEASURABLE ITEMS	YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED FD INT PROV	CHAN CORE CPU PUNCH PRINT READ			CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
			30000.	0.	0.	0.	0.	0.	0.	30000.
620 MAINTENANCE										
0.00.0 00. 00. 00. 0. 4500.	00. 00. 00. 00. 00. 00.		4500.	0.	0.	0.	0.	0.	0.	4500.
620										
0.00.0 00. 00. 00. 48. 0.	00. 00. 00. 00. 00. 00.		0.	0.	0.	0.	0.	0.	0.	0.
ADMINISTRATION OVERHEAD										
0.00.0 00. 00. 00. 0. 25000.	00. 00. 00. 00. 00. 00.		25000.	0.	0.	0.	0.	0.	0.	25000.
TAPES										
12000.00.0 00. 00. 00. 48. 0.	00. 00. 00. 00. 00. 00.		3000.	0.	0.	0.	0.	0.	0.	3000.
ENT 11760 SQ FT @ 25/SQ FT										
99000.00.0 00. 00. 00. 96. 0.	00. 00. 00. 00. 00. 00.		37375.	0.	0.	0.	0.	0.	0.	37375.
INSURANCE										
0.00.0 00. 00. 00. 0. 350.	00. 00. 00. 00. 00. 00.		350.	0.	0.	0.	0.	0.	0.	350.
UPPLIES										
0.00.0 00. 00. 00. 0. 34513.	00. 00. 00. 00. 90. 10.		34513.	0.	0.	0.	0.	31062.	3451.	0.
THER										
0.00.0 00. 00. 00. 0. 4500.	00. 00. 00. 00. 00. 00.		4500.	0.	0.	0.	0.	0.	0.	4500.
LARIES										
0.00.0 00. 00. 00. 0.167496.	00. 00. 00. 00. 00. 00.									

CHASE ADJUSTMENTS	MO RENT	MEASURABLE ITEMS	YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED ED INT PROV	CHAN CORE CPU PUNCH PRINT READ			CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
			167496.	0.	0.	0.	0.	0.	0.	167496.
026 001 KEYPUNCH										
0.00.0 00. 00. 00. 0. 680.	00. 00. 00. 00. 00. 00.		680.	0.	0.	0.	0.	0.	0.	680.
029 A2 KEYPUNCH										
0.00.0 00. 00. 00. 0. 5486.	00. 00. 00. 00. 00. 00.		5486.	0.	0.	0.	0.	0.	0.	5486.
029 A2 KEYPUNCH										
0.00.0 00. 00. 00. 0. 3360.	00. 00. 00. 00. 00. 00.		3360.	0.	0.	0.	0.	0.	0.	3360.
029 C2 KEYPUNCH										
0.00.0 00. 00. 00. 0. 1220.	00. 00. 00. 00. 00. 00.		1220.	0.	0.	0.	0.	0.	0.	1220.
029 B22 KEYPUNCH										
0.00.0 00. 00. 00. 0. 1055.	00. 00. 00. 00. 00. 00.		1055.	0.	0.	0.	0.	0.	0.	1055.
029 B22 KEYPUNCH										
0.00.0 00. 00. 00. 0. 937.	00. 00. 00. 00. 00. 00.		937.	0.	0.	0.	0.	0.	0.	937.
029 A22 KEYPUNCH										
0.00.0 00. 00. 00. 0. 883.	00. 00. 00. 00. 00. 00.		883.	0.	0.	0.	0.	0.	0.	883.
029 A2 KEYPUNCH										
0.00.0 00. 00. 00. 0. 865.	00. 00. 00. 00. 00. 00.		865.	0.	0.	0.	0.	0.	0.	865.
082 001 SORTER										
0.00.0 00. 00. 00. 0. 529.	00. 00. 00. 00. 00. 100.									

CHASE ADJUSTMENTS		MO RENT	MEASURABLE ITEMS						YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED	ED	INT	PROV		CHAN	CORE	CPU	PUNCH	PRINT	READ		CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
											529.	0.	0.	0.	0.	0.	529.	0.
082	001		SORTER															
	0.00.0	00.	00.	00.	0.	-529.	00.	00.	00.	00.	00.	100.	-529.	0.	0.	0.	0.	0.	-529.	0.
316			DISK PACKS															
	0.00.0	00.	00.	00.	0.	1088.	00.	00.	00.	00.	00.	00.	1088.	0.	0.	0.	0.	0.	0.	1088.
260	002		CRT															
	0.00.0	00.	00.	00.	0.	2458.	30.	40.	30.	00.	00.	00.	2458.	737.	983.	737.	0.	0.	0.	0.
			TAPES															
	0.00.0	00.	00.	00.	0.	0.	00.	00.	00.	00.	00.	00.	0.	0.	0.	0.	0.	0.	0.	0.
402	002		TAPE DRIVE															
	0.00.0	00.	00.	00.	0.	11340.	100.	00.	00.	00.	00.	00.	11340.	11340.	0.	0.	0.	0.	0.	0.
701	001		DATA ADAPTER															
	0.00.0	00.	00.	00.	0.	5628.	100.	00.	00.	00.	00.	00.	5628.	5628.	0.	0.	0.	0.	0.	0.
701	001		DATA ADAPTER															
	0.00.0	00.	00.	00.	0.	-5628.	100.	00.	00.	00.	00.	00.	-5628.	-5628.	0.	0.	0.	0.	0.	0.
701	001		DATA ADAPTER															
	0.00.0	00.	00.	00.	0.	5628.	100.	00.	00.	00.	00.	00.	5628.	5628.	0.	0.	0.	0.	0.	0.
701	001		DATA ADAPTER															
	0.00.0	00.	00.	00.	0.	-5628.	100.	00.	00.	00.	00.	00.								

CHASE ADJUSTMENTS		LIFE MO	YR RENT	070 DISTRIBUTION TO MEASURABLE ITEMS								YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
	FED	ED	INT	PROV	CHAN	CORE	CPU	PUNCH	PRINT	READ			CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
												-5628.	-5628.	0.	0.	0.	0.	0.	0.
701	001		DATA ADAPTER																
	0.00.0	00.	00.	00.	0.	7560.	100.	00.	00.	00.	00.	00.	7560.	7560.	0.	0.	0.	0.	0.	0.
701	001		DATA ADAPTER																
	0.00.0	00.	00.	00.	0.	-7560.	100.	00.	00.	00.	00.	00.	-7560.	-7560.	0.	0.	0.	0.	0.	0.
740	001		COMM.TERM.																
	0.00.0	00.	00.	00.	0.	1441.	100.	00.	00.	00.	00.	00.	1441.	1441.	0.	0.	0.	0.	0.	0.
740	001		COMM.TERM.																
	0.00.0	00.	00.	00.	0.	-1441.	100.	00.	00.	00.	00.	00.	-1441.	-1441.	0.	0.	0.	0.	0.	0.
740	001		COMM. TERM.																
	0.00.0	00.	00.	00.	0.	4642.	100.	00.	00.	00.	00.	00.	4642.	4642.	0.	0.	0.	0.	0.	0.
740	001		COMM. TERM.																
	0.00.0	00.	00.	00.	0.	-4642.	100.	00.	00.	00.	00.	00.	-4642.	-4642.	0.	0.	0.	0.	0.	0.
848	002		DISPLAY CONTROL																
	0.00.0	00.	00.	00.	0.	6190.	00.	00.	00.	00.	00.	00.	6190.	0.	0.	0.	0.	0.	0.	6190.
416	001		PRINT TRAIN																
	0.00.0	00.	00.	00.	0.	1134.	00.	00.	00.	00.	100.	00.	1134.	0.	0.	0.	0.	1134.	0.	0.
702	001		TRANSMISSION CNT																
	0.00.0	00.	00.	00.	0.	12342.	100.	00.	00.	00.	00.	00.								

CHASE ADJUSTMENTS	MO	RENT	MEASURABLE ITEMS						YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED ED INT PROV			CHAN	CORE	CPU	PUNCH	PRINT	READ		CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
									12342.	12342.	0.	0.	0.	0.	0.	0.
702 001 TRANSMISSION CNT																
0.00.0 00. 00. 00. 0.	-12342.	100.	00.	00.	00.	00.	00.		-12342.	-12342.	0.	0.	0.	0.	0.	0.
702 ADDITIONS																
0.00.0 00. 00. 00. 0.	874.	100.	00.	00.	00.	00.	00.		874.	874.	0.	0.	0.	0.	0.	0.
702 ADDITIONS																
0.00.0 00. 00. 00. 0.	-874.	100.	00.	00.	00.	00.	00.		-874.	-874.	0.	0.	0.	0.	0.	0.
365 001 CORE																
0.00.0 00. 00. 00. 0.	91684.	00.	50.	50.	00.	00.	00.		91684.	0.	45842.	45842.	0.	0.	0.	0.
365 001 NEXT CORE BOX																
0.00.0 00. 00. 00. 0.	91684.	00.	100.	00.	00.	00.	00.		91684.	0.	91684.	0.	0.	0.	0.	0.
PHONES COMP																
0.00.0 00. 00. 00. 0.	1000.	00.	00.	00.	00.	00.	00.		1000.	0.	0.	0.	0.	0.	0.	1000.
PHONES OFFIC																
0.00.0 00. 00. 00. 0.	800.	00.	00.	00.	00.	00.	00.		800.	0.	0.	0.	0.	0.	0.	800.
LIBRARY																
0.00.0 00. 00. 00. 0.	900.	00.	00.	00.	00.	00.	00.		900.	0.	0.	0.	0.	0.	0.	900.
FURNITURE																
0000.00.0 00. 00. 00. 48.	0.	00.	00.	00.	00.	00.	00.									

CHASE ADJUSTMENTS	MO RENT	MEASURABLE ITEMS						YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED ED INT PROV		CHAN	CORE	CPU	PUNCH	PRINT	READ		CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
								2500.	0.	0.	0.	0.	0.	0.	2500.
SUBSCRIPTION															
0.00.0 00. 00. 00.	0.	200.	00.	00.	00.	00.	00.	00.							
								200.	0.	0.	0.	0.	0.	0.	200.
PLANT MAINT															
8800.00.0 00. 00. 00.	0.	0.	00.	00.	00.	00.	00.	00.							
								8800.	0.	0.	0.	0.	0.	0.	8800.
STANDBYPOWER															
40000.00.0 00. 00. 00.	48.	0.	00.	00.	00.	00.	00.	00.							
								10000.	0.	0.	0.	0.	0.	0.	10000.
D3A DATA SET															
0.00.0 00. 00. 00.	0.	902.	100.	00.	00.	00.	00.	00.							
								902.	902.	0.	0.	0.	0.	0.	0.
D3A DATA SET															
0.00.0 00. 00. 00.	0.	-902.	100.	00.	00.	00.	00.	00.							
								-902.	-902.	0.	0.	0.	0.	0.	0.
201B DATA SET															
0.00.0 00. 00. 00.	0.	1380.	100.	00.	00.	00.	00.	00.							
								1380.	1380.	0.	0.	0.	0.	0.	0.
201B DATA SET															
0.00.0 00. 00. 00.	0.	-1380.	100.	00.	00.	00.	00.	00.							
								-1380.	-1380.	0.	0.	0.	0.	0.	0.
6C DATA SET															
0.00.0 00. 00. 00.	0.	1380.	100.	00.	00.	00.	00.	00.							
								1380.	1380.	0.	0.	0.	0.	0.	0.
6C DATA SET															
0.00.0 00. 00. 00.	0.	-1380.	100.	00.	00.	00.	00.	00.							

CHASE ADJUSTMENTS	LIFE MO	RENT	MEASURABLE ITEMS						YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
FED ED INT PROV			CHAN	CORE	CPU	PUNCH	PRINT	READ		CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
									-1380.	-1380.	0.	0.	0.	0.	0.	0.
801 NEW DRUM																
0.00.0 00. 00. 00.	0.	48366.	00.	00.	100.	00.	00.	00.	48366.	0.	0.	48366.	0.	0.	0.	0.
814 NEW																
0.00.0 00. 00. 00.	0.	40596.	00.	00.	00.	00.	00.	00.	40596.	0.	0.	0.	0.	0.	0.	40596.
LAB PRINTER																
0.00.0 00. 00. 00.	0.	2808.	00.	00.	00.	00.	100.	00.	2808.	0.	0.	0.	0.	2808.	0.	0.
LAB READER																
0.00.0 00. 00. 00.	0.	8028.	00.	00.	00.	00.	00.	100.	8028.	0.	0.	0.	0.	0.	8028.	0.
I-J																
2430.00.0 00. 00. 00.	48.	0.	75.	25.	00.	00.	00.	00.	8108.	6081.	2027.	0.	0.	0.	0.	0.
865 001 CORE																
0.00.0 00. 00. 00.	0.	-91684.	00.	50.	50.	00.	00.	00.	-91684.	0.	-45842.	-45842.	0.	0.	0.	0.
865 001 NEXT CORE BOX																
0.00.0 00. 00. 00.	0.	-91684.	00.	100.	00.	00.	00.	00.	-91684.	0.	-91684.	0.	0.	0.	0.	0.
801 NEW DRUM																
0.00.0 00. 00. 00.	0.	-48366.	00.	00.	100.	00.	00.	00.	-48366.	0.	0.	-48366.	0.	0.	0.	0.
814 NEW																
0.00.0 00. 00. 00.	0.	-40596.	00.	00.	00.	00.	00.	00.								

CHASE ADJUSTMENTS				MO	RENT	MEASURABLE ITEMS						YR-COST	DISTRIBUTION OF YR-COST TO MEASURABLE ITEMS						
	FED	ED	INT	PROV		CHAN	CORE	CPU	PUNCH	PRINT	READ		CHAN	CORE	CPU	PUNCH	PRINT	READ	UNA/C
												-40596.	0.	0.	0.	0.	0.	0.	-40596.
LAB PRINTER																			
0.	00.	0	00.	00.	00.	0.	-2808.	00.	00.	00.	00.	100.	00.	-2808.	0.	0.	0.	0.	-2808.	0.	0.
LAB READER																			
0.	00.	0	00.	00.	00.	0.	-8028.	00.	00.	00.	00.	00.	100.	-8028.	0.	0.	0.	0.	0.	-8028.	0.
I-J																			
32430.	00.	0	00.	00.	00.	48.	0.	75.	25.	00.	00.	00.	00.	-8108.	-6081.	-2027.	0.	0.	0.	0.	0.
-TOTALS													901797.	170543.	100731.	149271.	12402.	109483.	31097.	328274.

					$/READ-HR
0.0012530740	0.4751627000	151.2308000000	126.3500000000	95.8160700000	94.4559600000
	$/KB-SEC	$/CPU-MIN	$/PUNCH-CD	$/LINE	$/READ-CD
	0.0001319896	2.5205140000	0.1403889000	0.0017743710	0.0015742660

T RATES A/C FOR $ 901801. OF ANNUAL EXPENDITURE:$ 901797.

USAGE OBJECT CODE= 5552 BYTES,ARRAY AREA= 12172 BYTES,TOTAL AREA AVAILABLE= 32864 BYTES

PILE TIME= 0.59 SEC,EXECUTION TIME= 2.52 SEC,DATE= 71054

A P P E N D I X 2

PROGRAM R E S E V A L

```
WATFIV    DORAN,PAGES=470,TIME=510
PORTS UNITS OF WORK PER HOUR PERFORMED BY THE COMPUTER INSTALLATION
NPUT FORMAT

     IDENTIFICATION          COLUMNS     TYPE

     PROGRAMMER NAME         01 - 14     ALPHA
     PRINT DESTINATION       15          BINARY
     PUNCH DESTINATION       16          BINARY
     NUMBER OF STEPS         17          BINARY
     I/O COUNTS              18 - 20     BINARY
     ACCOUNT NUMBER          21 - 28     ALPHA
     CPU TIME (.01 SEC)      29 - 31     BINARY
     PRIORITY                32          BINARY
     IN TIME OF DAY          33 - 35     BINARY
     OUT TIME OF DAY         36 - 38     BINARY
     CARDS READ              39 - 40     BINARY
     LINES PRINTED           41 - 43     BINARY
     CARDS PUNCHED           44 - 45     BINARY
     EXECUTION TIME(.01)     46 - 48     BINARY
     JOB INIT I/O            49 - 50     BINARY
     NOT USED                51
     JOBNAME                 52 - 59     ALPHA
     TIME EXECUTED           60 - 62     BINARY
     K BYTE-SEC              63 - 66     BINARY
     NOT USED                67 - 69
     INPUT DEVICE            70          ALPHA
     COPIES OF OUTPUT        71          BINARY
     ID X'FF'                72          BINARY
     DATE (YYDDD)            73 - 77     ALPHA
     JOB NUMBER              78 - 80     ALPHA

UMED UNIT COSTS
PU                    $150/HR                  $.041666/SEC
HANNEL-COUNTS                                  $.0015 EACH
ORE                   $0.85/KBHR               $.00023611/KBSEC
ARDS READ                                      $.0006 EACH
INES PRINTED                                   $.0007 EACH
ARDS PUNCHED                                   $.0025 EACH
ND CONTAINS YRDAY FOR EACH MONTH-END
INTEGER  MDEND(12) / 31,59,90,120,151,181,212,243,273,304,334,365/

AY A CONTAINS 6 COLUMNS OF CUMULATIVE REVENUE &
ROWS IDENTIFIED THUSLY:
    1-24 HOUR-OF-THE-DAY
      25 24-HOUR TOTALS
ROWS 26 THROUGH 28 GROUP THE 24-HOUR TOTALS:
      26 901 A.M. TO 500 P.M.
      27 501 P.M. TO 900 P.M.
      28 901 P.M. TO MIDNIGHT
ROWS 29 THROUGH 32 CONTAIN CUMULATIVE MONTH TOTALS OF
26 THROUGH 28:
      29 24-HOUR TOTALS FOR THE MONTH
      30 900 TO 500 TOTALS FOR THE MONTH
      31 500-900 TOTALS FOR THE MONTH
      32 900-MIDNIGHT TOTALS FOR THE MONTH
      33  AUDIT TRAIL
      34 (TOTAL OF ROWS 1 THRU 24 WITH REV > 200)/ NO OF ROWS COUNTED
```

Program RESEVAL , resource evaluation

APPENDIX 2 -- Program RESEVAL.

FUNCTION.

To compute and accumulate for each hour of any arbitrary period, such as a month, the revenue earned by CPU, channels, core, and external I/O and to compute for each hour the efficiency attained by CPU, channels, core, and the combination of the three. That is, to report plant performance by the hour.

DATA.

1) One job accounting card image for each job submitted to the computer during the period (one month or so).

2) Unit costs derived from program MKCC.

REFERENCE.

The logic of program RESEVAL is described in sections 3C and 3D of the text, pp 44 - 55.

```
//JOB WATFIV    DORAN,PAGES=470,TIME=510
C  REPORTS UNITS OF WORK PER HOUR PERFORMED BY THE COMPUTER INSTALLATION
C   INPUT FORMAT
```

IDENTIFICATION	COLUMNS	TYPE
PROGRAMMER NAME	01 - 14	ALPHA
PRINT DESTINATION	15	BINARY
PUNCH DESTINATION	16	BINARY
NUMBER OF STEPS	17	BINARY
I/O COUNTS	18 - 20	BINARY
ACCOUNT NUMBER	21 - 28	ALPHA
CPU TIME (.01 SEC)	29 - 31	BINARY
PRIORITY	32	BINARY
IN TIME OF DAY	33 - 35	BINARY
OUT TIME OF DAY	36 - 38	BINARY
CARDS READ	39 - 40	BINARY
LINES PRINTED	41 - 43	BINARY
CARDS PUNCHED	44 - 45	BINARY
EXECUTION TIME(.01)	46 - 48	BINARY
JOB INIT I/O	49 - 50	BINARY
NOT USED	51	
JOBNAME	52 - 59	ALPHA
TIME EXECUTED	60 - 62	BINARY
< BYTE-SEC	63 - 66	BINARY
NOT USED	67 - 69	
INPUT DEVICE	70	ALPHA
COPIES OF OUTPUT	71	BINARY
ID X'FF'	72	BINARY
DATE (YYDDD)	73 - 77	ALPHA
JOB NUMBER	78 - 80	ALPHA

```
ASSUMED UNIT COSTS
  CPU                    $150/HR              $.041666/SEC
  CHANNEL-COUNTS                              $.0015 EACH
  CORE                   $0.85/KBHR           $.00023611/KBSEC
  CARDS READ                                  $.0006 EACH
  LINES PRINTED                               $.0007 EACH
  CARDS PUNCHED                               $.0025 EACH
MOEND CONTAINS YRDAY FOR EACH MONTH-END
   INTEGER  MOEND(12) / 31,59,90,120,151,181,212,243,273,304,334,365/

ARRAY A CONTAINS 6 COLUMNS OF CUMULATIVE REVENUE &
34 ROWS IDENTIFIED THUSLY:
      1-24 HOUR-OF-THE-DAY
        25 24-HOUR TOTALS
   ROWS 26 THROUGH 28 GROUP THE 24-HOUR TOTALS:
        26 901 A.M. TO 500 P.M.
        27 501 P.M. TO 900 P.M.
        28 901 P.M. TO MIDNIGHT
   ROWS 29 THROUGH 32 CONTAIN CUMULATIVE MONTH TOTALS OF
   26 THROUGH 28:
        29 24-HOUR TOTALS FOR THE MONTH
        30 900 TO 500 TOTALS FOR THE MONTH
        31 500-900 TOTALS FOR THE MONTH
        32 900-MIDNIGHT TOTALS FOR THE MONTH
        33  AUDIT TRAIL
        34 (TOTAL OF ROWS 1 THRU 24 WITH REV > 200)/ NO OF ROWS COUNTED
```

```
      REAL  UIO/.0015/,UCPU/.041666/,UKBS/.0002361/,KBHR
      REAL A(34,6)/204*0./
      REAL                    UR/.0006/,UP/.0007/,UPU/.0025/
      INTEGER IA(7)/7*0/, LDEST,CDEST,STEPS,PRIOR,INHR,OUTHR, T(3),
     1  PRINTS, ACT(2)
*****  JN IS THE JOB NUMBER OF A BAD CARD IMAGE WHICH SHOULD BE DELETED *
      INTEGER  JN/'930 '/
      INTEGER*2 JIIO, READS, PCHS
      INTEGER SW1
      INTEGER IO,CPU,XEQ,STEX,KBS,ENDSW/0/, PREVJN/0/, DATE, DAY,TRDAY
      REAL*8       JNAM, PNM(2)
      INTEGER SP(5)/1722,1741,1741,1741,1741/, SW3
      INTEGER SP(5)/1681,1683,1685,1687,1689/, SW3
      INTEGER EXAC(58) /8010,8020,8030,8040,8050,8070,8080,8090,8100,
     18130,8150,8160,8170,8180,8190,8200,8210,8220,8221,8222,8223,8224,
     2 8225,8226,8227,8228,8229,8230,8240,8250,8310,8320,8330,8350,8370,
     38390,8400,8410,8440,8450,8460,8480,8490,8500,8520,8530,8540,8550,
     48610,8620,8630,8640,8650,8660,8670,8680,8700,8710/
      INTEGER SW2/0/
      LOGICAL    MASK/Z00FFFFFF/, VALUE(3)
      EQUIVALENCE (T(1), VALUE(1))
      DAY=0.
      MAG IS TRACE HOUR
      MAG = 14
      MAG = 4
      MAG = 25
      MAG = 23
      TRDAY = 71020
******    TRDAY IS THE YRDAY OF THE DAY ONE MIGHT TRACE ******
********** SW4=0 ONLY TO TRACE ACTIVITY IN A SPECIFIC HOUR**********
      SW4 = 1
      SW4 = 0
      DATA MAC/'42  '/
      DATA MAC2/'21  '/
      STOT = 0.
      PIO=0.
      MNSW = 0
      SW1=0
      WHC = 0.
      JCNT = 0
      INVCNT = 0
111   FORMAT (A4)
      SW3=0
      SW3=1

                                                      READ  A/C RECORD
  1   READ(5,102,END=11)PNM,LDEST,CDEST,STEPS,IO,ACT,CPU,
      READ INPUT TAPE
  1   READ(9,102,END=11)PNM,LDEST,CDEST,STEPS,IO,ACT,CPU,
     1PRIOR,T(1),T(2), READS,PRINTS,PCHS,
     2XEQ,JIIO,JNAM,T(3),KBS,IDEV,DATE,JOBNO
102   FORMAT ( A8, A6, 3A1, A3, I4, A4,
     1                      A3,A1,2A3,A2,A3,A2,A3,A2,
     11X,A8,A3,A4,3X,A1,2X,I5,A3)
      OLDCP = A(MAG,2)
      MUG = MAG + 1
      OLDCP = A(MUG,2)
      WRITE (6,112)IO,ACT,CPU,XEQ,JIIO,JNAM,STEX,KBS,DATE,JOBNO
```

```
**************************SW2=0 BY-PASSES ALL UNIVERSITY INTERNAL JOBS*****
   SW2=0
    SW2=1
2 FORMAT (' ', 17X, A3, I4, A4,
 1                          A3,14X,A3,A2,1X,A8,A3,A4,6X,I5,A3)
-PASS MULTIPLE JOB NUMBER BECAUSE SOME STUTTERING OCCURS IN ACCOUNTING
   IF  (JOBNO.EQ.PREVJN)  GO TO 1
*******SW3=0 BYPASSES ALL JOBS EXCEPT THOSE WITH ACCOUNT NUMBERS LISTED
****** IN ARRAY SP.  SP CONTAINS ONLY 5 SLOTS BUT MAY BE EXPANDED AS
******  REQUIRED
    IF  (JOBNO .EQ.  JN )  GO TO 213
4 CONTINUE
   IF (SW3.EQ.0)  GO TO 210
   IF (SW2.EQ.1) GO TO 150
   DO 148 K=1,58
   IF(ACT(1).EQ.EXAC(K)) GO TO 150
8 CONTINUE
9 GO TO 1
3 IF  (DATE.EQ.  71008)  GO TO 1
   PRINT,  JOBNO,  DAY
   GO TO 214
0 DO 209  K=1,5
**** PROGRAM GETS HERE IFF SW3=0; IF ONLY A SELECTED DEPT IS TO BE EXAMINED
   IF (ACT(1).EQ.SP(K)) GO TO 150
   CONTINUE
   GO TO 1
0 CONTINUE
   PREVJN = JOBNO
    WRITE (6,131)  READS
   PRINTS = PRINTS/256
   PCHS = PCHS / 256


VELOP REVENUE FROM CARDS READ
   RSPIN = READS * UR
   WRITE (6,130)   READS, UR, RSPIN, PCHS, PRINTS, RSPOUT
0 FORMAT (1H ,'READS = ',A3, 'UR = ', F10.5, 'RSPIN = ',F10.5,
 1 'PCHS = ', A3, 'PRINTS = ',A3, 'RSPOUT = ',F10.5)
   FORMAT (1H , 'READS = ', A4)


'ELOP REVENUE FROM PRINTING AND PUNCHING
  RSPOUT = PRINTS* UP + PCHS     * UPU
  IF (LDEST.EQ.20987968) RSPOUT = RSPOUT * PIO
  IF (IDEV.EQ.-247447488) RSPIN=RSPIN*RIO
  WRITE(6,123)T(1)
  FORMAT(1H ,'INHR=',I8)
  ISTEX = INHR/256
  VALUE = VALUE.AND.MASK
  T(1) = T(1) / 256
  VALUE(1) = VALUE(1).AND.MASK
  WRITE(6,124) T(1)
  FORMAT(1H ,'ISTEX CUM INHR = ',I8)
ERMINE HOUR OF READING
  INHR = T(1) /360000
  WRITE(6,123) INHR
  WRITE(6,125) T(2)
  FORMAT(1H ,'OUTHR RAW =',I8)
  ISTEX = OUTHR/256
  T(2) = T(2) / 256
```

```
      WRITE(6,126) T(2)
  126 FORMAT(1H ,'ISTEX CUM OUTHR= ',I8)
      VALUE = VALUE.AND.MASK
      VALJE(2) = VALJE(2).AND.MASK
      OUTHR = ISTEX/360000
C DEVELOP HOUR OF PRINTING AND PUNCHING
      OUTHR = T(2) / 360000
      WRITE(6,127)OUTHR
  127 FORMAT(1H ,'OUTHR TO INTEGER =',I8)
      I=INHR + 1
      IF (I.GT.24) I = 24
      A(I,6)=A(I,6)+RSPIN
      WRITE(6,128)RSPIN,I,A(I,6)
  128 FORMAT(1H , F10.5, I8, F10.2)
      I = OUTHR + 1
      IF (I.GT.24) I = 24
      A (I,6)=A(I,6)+RSPOUT
      WRITE(6,129)RSPOUT,I,A(I,6)
  129 FORMAT(1H ,F10.5     ,    'I=',I8,'A(I,6)=',F10.2)
      ISTEX=STEX/256
      T(3) = T(3) / 256
      VALUE(3) = VALUE(3).AND.MASK
      ISTEX = T(3)
      FSTEX = ISTEX
      TSTEX = ISTEX
      TSTEX = TSTEX / 360000.
      TEXEC = XEQ
      TEXEC = TEXEC / 256. / 360000.
      TENDEX = TSTEX + TEXEC
      IF (JCNT . GT. 15) GO TO 19
      WRITE (6,109) JOBNO, TSTEX, TEXEC, TENDEX
  109 FORMAT (1HO, A8, 3F12.2)
   19 HR=ISTEX/360000
      ISTEX = ISTEX / 360000
      ENDEX=FSTEX+XEQ/256/360000
      EXH=XEQ/256/360000
      FSTEX = FSTEX / 360000.
      IENDEX = TENDEX
      ISTEX = TSTEX
      WRITE (6,113)FSTEX,IENDEX, ISTEX, HR
  113 FORMAT (' ' ,F6.2,5X, I10,A10, F10.6)
      IF(DAY.EQ.DATE)GO TO 2
      IF(DAY.NE.0.) GO TO 3
      IF(DAY.EQ.0.) GO TO 21
C                                     START OF NEW MONTH
   32 DO 33 I=29,34
      DO 34 J=1,6
   34 A(I,J)=0.
   33 CONTINUE
      WHC = 0.
C
C                                     START  NEW DAY
   24 DO 4 I=1,28
      DO 5 J=1,6
    5 A(I,J)=0.
    4 CONTINUE
      OLDCP = 0.
      STOT = 0.
      DO 42 J = 1,6
```

```
 42   A(33, J) = 0.
      IF (MNSW .EQ. 0) GO TO 2
      MNSW = 0
      PRINT,A
21    CONTINUE
      WRITE(6,101)
 101  FORMAT(1H1, T7,'NO',T11,'DATE',T20,'JIIO',T30,'IO', T41,'CPU',
     1 T49,'EXEC',T59,'KBS',T68,'RIO',T78,'RCPU',T88,'RKBS',T98,'START')
2     DAY=DATE
  NEXT CARD NO MIGHT BE WRONG
 17   FXEQ = XEQ / 256 / 100.
      FCPU = CPU/256/100.
      JIIO = JIIO/256
      IO = IO/256
      CPU = CPU/256/100.
      KBS = KBS/256
      RIO = (IO+JIIO) * UIO
      RCPU =FCPU * UCPU
      RKBS = KBS * UKBS
      JCNT = JCNT + 1


 AUDIT TRAIL
      TTOT = RIO + RCPU + RKBS + RSPIN + RSPOUT
      STOT = STOT + TTOT
       WRITE (6,135) ACT, IO, CPU, KBS, RIO, RCPU, RKBS, TTOT,
     1 STOT, DATE, JOBNO
 135  FORMAT(1H , 'ACT #  ', I4, A4, '   IO',I8, ' CPU', I8,
     1    '  KBS', I8, '   RIO', F8.2, ' RCPU', F8.2,' RKBS', F8.2,
     2 'TOT', F8.2, 'CUM', F8.2, I6, ' JOBNO ', A3)
      A(33, 1) = A(33, 1) + RIO
      A(33,2) = A(33,2) + RCPU
      A(33,3) = A(33,3) + RKBS
      A(33,4) = A(33,4) + RIO + RCPU + RKBS
      A(33,5) = A(33,5) + 1.
      A(33,6) = A(33,6) + RSPIN + RSPOUT



  NEXT CARD IS A TEMPORARY PRINTING BY-PASS
      IF (ISTEX.NE. MAG) GO TO 12
      IF  (DAY  .NE.  TRDAY)  GO TO 12
      IF  (SW4  .EQ.  1)  GO TO 12
      WRITE (6,152)
 152  FORMAT(1HO, T7,'NO',T11,'DATE',T20,'JIIO',T30,'IO', T41,'CPU',
     1 T49,'EXEC',T59,'KBS',T68,'RIO',T78,'RCPU',T88,'RKBS',T98,'START')
      WRITE (6,114) JOBNO, DAY , JIIO, IO, FCPU, FXEQ, KBS, RIO,RCPU,
     1  RKBS, FSTEX
 114  FORMAT  (1HO, A8,  I7, 2I8,  2F10.2, I8,  4F10.2)
      WRITE (6,118) ISTEX
 118  FORMAT ( 1HO,'ISTEX @ STMNT 12 =  ', I10)
 12   IF (ISTEX.EQ.0 ) GO TO 14
 5    IF (ISTEX .NE. IENDEX) GO TO 16
      I     = ISTEX + 1
      WRITE (6,119) ISTEX, I
 119  FORMAT (1HO,'ISTEX @ STMNT 7 =  ',I10, 'I=  ',I10)
      FAC=1.
   7  A(I,1)=A(I,1)+RIO*FAC
      A(I,2)=A(I,2)+RCPU*FAC
      A(I,3)=A(I,3)+RKBS*FAC
```

```
      A(I,4)=A(I,4)+(RIO+RCPU+RKBS)*FAC
      A(I,5) = A(I,5) + 1.
      IF  (SW4 . EQ.  1)  GO TO 20
      MUG = MAG + 1
      IF ( I . NE . MUG)  GO TO 20
      IF (ISTEX.NE.MAG) GO TO 20
      IF  (DAY.NE.81027) GO TO 20
      IF  (DAY.NE.TRDAY)   GO TO 20
      PRINT,(A(I,J), J = 1,5)
      WRITE  (6,151)   (A(I,J),  J = 1,5),  JOBNO
  151 FORMAT  (1H0,  5F8.2,  A8)
      GO TO 20
   14 IF(FSTEX.LT..0001) GO TO 25
      GO TO 15
   25 INVCNT = INVCNT + 1
      GO TO 1
   15 IENDEX = TENDEX
      ISTEX = TSTEX
      IF (IENDEX. GT. ISTEX) GO TO 6
      IENDEX = IENDEX + 24
    6 NSTG=IENDEX-ISTEX+1
      IF (SW1.EQ.0) GO TO 138
      WRITE (6,120) JOBNO, NSTG, TSTEX, TEXEC, TENDEX
  138  CONTINUE
  120 FORMAT (1H0, ' JOB', A8, ' HAS', I4, '  STAGES STARTING @ ', F7.2,
     1 '  EXEC ', F7.2, '  ENDING @ ', F7.2)
      PRINT,NSTG,IENDEX,ISTEX,ENDEX,FSTEX
      DELTA=FLOAT(IFIX(FSTEX)+1) -FSTEX
      IF(NSTG.GT.5) GO TO 206
      IF (TEXEC .LT. .0001) GO TO 25
      EXH = TEXEC
      FRAC=(FLOAT(NSTG)-2.+DELTA)/TEXEC
      IF(NSTG.EQ.5) GO TO 205
      IF(NSTG.EQ.4) GO TO 204
      IF(NSTG.EQ.3) GO TO 203
      IF(NSTG.EQ.2) GO TO 202
  205 FAC5=1.0-FRAC
      FAC4=1.0/EXH
      FAC3=1.0/EXH
      FAC2=1.0/EXH
      FAC1=DELTA/EXH
      GO TO 208
  204 FAC5=0.
      FAC4=1.0-FRAC
      FAC3=1.0/EXH
      FAC2=1.0/EXH
      FAC1=DELTA/EXH
      GO TO 208
  203 FAC5=0.
      FAC4=0.
      FAC3=1.0-FRAC
      FAC2=1.0/EXH
      FAC1=DELTA/EXH
      GO TO 208
  202 FAC5=0.
      FAC4=0.
      FAC3=0.
      FAC2=1.0-FRAC
      FAC1=DELTA/EXH
      GO TO 208
```

```
205   WRITE (6,207) NSTG
207   FORMAT(' PROGRAM DESIGNED TO HANDLE MAX OF 5 STAGES',I6,'STAGES
     2ENCOUNTERED')
      WRITE(6,110) JOBNO, FSTEX, FXEQ, ENDEX
 110  FORMAT (1HO,'JOB', A8, 'FAILED; START', F5.2,'EXEC',F12.2,
     1 'END', F12.2)
      WRITE (6,110)  JOBNO, TSTEX, TEXEC, TENDEX
      GO TO 20
208   I=ISTEX+1
      IF (I .GT. 24) I = 24
      A(I,1)=A(I,1)+RIO*FAC1
      A(I,2)=A(I,2)+RCPU*FAC1
      A(I,3)=A(I,3)+RKBS*FAC1
      A(I,4)=A(I,4)+(RIO+RCPU+RKBS)*FAC1
      I=I+1
      IF (I .GT. 24) I = 24
      A(I,1)=A(I,1)+RIO*FAC2
      A(I,2)=A(I,2)+RCPU*FAC2
      A(I,3)=A(I,3)+RKBS*FAC2
      A(I,4)=A(I,4)+(RIO+RCPU+RKBS)*FAC2
      I=I+1
      IF (I .GT. 24) I = 24
      A(I,1)=A(I,1)+RIO*FAC3
      A(I,2)=A(I,2)+RCPU*FAC3
      A(I,3)=A(I,3)+RKBS*FAC3
      A(I,4)=A(I,4)+(RIO+RCPU+RKBS)*FAC3
      I=I+1
      IF (I .GT. 24) I = 24
      A(I,1)=A(I,1)+RIO*FAC4
      A(I,2)=A(I,2)+RCPU*FAC4
      A(I,3)=A(I,3)+RKBS*FAC4
      A(I,4)=A(I,4)+(RIO+RCPU+RKBS)*FAC4
      I=I+1
      IF (I .GT. 24) I = 24
      A(I,1)=A(I,1)+RIO*FAC5
      A(I,2)=A(I,2)+RCPU*FAC5
      A(I,3)=A(I,3)+RKBS*FAC5
      A(I,4)=A(I,4)+(RIO+RCPU+RKBS)*FAC5
      A(I,5)=A(I,5)+1.
      GO TO 20
  20  MUG = MAG + 1
      IF   (SW4 . EQ. 1)  GO TO 1
      IF  (DAY.  NE  . TRDAY)  GO TO 1
 ****     TRDAY IS THE YRDAY ONE MIGHT WANT TO TRACE    ****
      IF (A(MUG,2) .EQ. OLDCP) GO TO 1
      REV = A(MUG,2) - OLDCP
      HREQ = REV / UCPU / 3600.
      CPUHR = A(MJG,2) / UCPU / 3600.
      WRITE (6,201) JOBNO, REV, HREQ, CPUHR, ACT
 201  FORMAT (1H , 'JOB ', A8, ' HAS $ ', F7.2, ' CPU, EQUIVALENT TO ',
     | F8.4, ' HRS CPU FOR TOTAL OF ', F5.2,'A/C',  I4, A4)
      GO TO 1
 211  IDC = IDC - 72000 + 5
      IDAY = DAY - 72000
      GO TO 22
 212  IDC = IDC - 71000 + 4
      IDAY = DAY - 71000
      GO TO 22
  11  ENDSW=1
```

```
                                                    END  OF  DAY
   3 WRITE (6,108)  JCNT, INVCNT, JOBNO, PREVJN
 108 FORMAT (1HO, 'DAY END TRIGGERED BY JOB', 2I10,2A8)
     WRITE (6,117)
 117 FORMAT (1H1)
     WRITE (6,104)
     WRITE (6,104)
     WRITE(6,103)
 103 FORMAT(1HO,T4,'HR',T22,'CHAN-UNITS',T33,'CPU-UNITS',
    1  T101, 'IN/OUT-UNITS',T114, 'REV',
    1 T43,'CORE-UNITS',T55, 'TOTAL', T10, 'DATE', T65, 'JOBS',
    2 T74, 'CHAN-HR', T84,'CPU-HR', T94,'KB-HR',T16, 'DAY')
     WRITE(6,104)
 104 FORMAT('0')

   COMPUTE DAY-OF-THE-WEEK  MON = 1, TUES = 2, ETC

     IDC = DAY
     IF (DAY.GT.72000)  GO TO 211
     IF (DAY.GT. 71000) GO TO 212
     IDAY = DAY - 70000
     IDC = IDC - 70000 + 3
  22 IDC=IDC-7
     IF (IDC .GT. 7) GO TO 22
     IWKDAY = IDC


                                        CONVERT  REVENUE  TO EFFICIENCY
     DO 8 I=1,24
     KBHR = A(I,3) / UKBS / 3600.
     CPUHR = A(I,2) / UCPU / 3600.
     CHANHR = (A(I,1) / UIO) * .025 / 3600.


                              CONVERT REAL ARRAY TO INTEGER  CARDS
     DO 26 J=1,6
  26 IA(J)=(A(I,J)*100.)
     ICHHR   = CHANHR * 100.
     ICPUHR = CPUHR * 100.
     IKBHR = KBHR * 100.


                                        PRINT  DAY-END  REPORT

                                                    AUDIT TRAIL
     WRITE(7,115) I,DAY,(IA(J),J=1,5),ICHHR  ,ICPUHR,IKBHR, IA(6),
    1IWKDAY
 115 FORMAT (I4, 10I7, 5X, I1)
     WRITE(6,105) I, DAY,IWKDAY,(A(I,J),J=1,5),CHANHR,CPUHR,
    1KBHR,A(I,6)
 105 FORMAT(1H ,I4,2X,I7,I4,9F10.2)
   8 CONTINUE
     DO 45 I = 1,24
     IF (A(I,4) + A(I,6) .LT. 200.) GO TO 45
     DO 43 J = 1,6
  43 A(34,J) = A(34,J) + A(I,J)
     WHC = WHC + 1.
  45 CONTINUE
     DO 9 J=1,6
```

```
      DO 10 I=1,24
      A(25,J)=A(25,J)+A(I,J)
   10 CONTINUE
    9 CONTINUE
      DO 27 J=1,6
   27 IA(J)= A(25,J) * 100.
      I = 25
      WRITE (7,116) I,DAY,(IA(J), J = 1, 6), IWKDAY
  116 FORMAT (I4, 6I7, 21X, I7, 5X, I1)
  106 FORMAT (1HO, 6X, A7, I7, 5I10)
      WRITE(6,104)
      WRITE (6,121) I, DAY, IWKDAY, (A(25,J), J= 1,6)
      I = 33
      WRITE(6,139)
      WRITE (6,121) I, DAY, IWKDAY, (A(33,J) , J= 1, 6),A(33,4)+A(33,6)
*  OTHER COMPILERS MAY NOT ALLOW EXPRESSIONS IN OUTPUT LISTS
      WRITE (6,104)
      DO 28 I=10,17
           DO 28 J=1,6
                A(26,J) = A(26,J)+ A(I,J)
   28 CONTINUE
      DO 29 I=18,21
           DO 29 J=1,6
                A(27,J)= A(27,J)+ A(I,J)
   29 CONTINUE
      DO 30 I=22,24
           DO 30 J=1,5
                A(28,J)=A(28,J)+ A(I,J)
   30 CONTINUE
  132 FORMAT (1HO, 6X, A7, 6F10.2)
      WRITE(6,140)
      DO 31      I = 26,28
      DO 39 J = 1,6
   39  IA(J) = A(I, J) * 100.
      WRITE(7, 115) I, DAY, (IA(J), J = 1, 6), IWKDAY
   31 WRITE(6,121) I,DAY,IWKDAY,(A(I,J),J=1,6),A(I,4) + A(I,6)
*  OTHER COMPILERS MAY NOT ALLOW EXPRESSIONS IN OUTPUT LISTS
      WRITE (6, 117)
  121 FORMAT(1HO,I4,2X,I7,I4,5F10.2,30X,2F10.2)
  122 FORMAT(A3,A8,6A8,16X,A4)
      JCNT = 0
      INVCNT = 0
      DO 40 L = 29, 32
      DO 40 J = 1, 6
      I = L - 4
   40 A(L, J) = A (L, J) + A(I, J)
      IF (ENDSW.EQ.1) GO TO  35
      ENDSW = 0
*****  IDAY WAS DETERMINED BACK AT CARD 245, 249, OR 265 **********
      DO 38 K=1,12
      PRINT, IDAY,  MOEND(K)
      IF (IDAY .EQ. MOEND(K)) GO TO 35
   38 CONTINUE
      GO TO 24

                                                          MONTH END
 PRINT MONTH-END SUMMARY
   35 WRITE(6,117)
      WRITE(6,104)
      WRITE(6,104)
      WRITE(6,134)
```

```
      WRITE(6,104)
      WRITE(6,103)
      WRITE(6,104)
      DO 146 K=1,12
      IF (IDAY.EQ.MOEND(K)) GO TO 147
  146 CONTINUE
  147 CONTINUE
  134 FORMAT(1HO,T30,'MONTH-END SUMMARY')
      DO 44 J = 1,6
       IF (WHC .LT. 1.) GO TO 44
         A(34,J) = A(34,J) / WHC
   44 CONTINUE
      IWKDAY=0
      WRITE(6,141)
      WRITE(6,142)
      WRITE(6,144)
      DO 36 I=29,34
      IF (I.EQ.33) GO TO 36
      DO 37 J=1,6
   37 IA(J)=(A(I,J)*100.)
      WRITE(6,121)I,K,IWKDAY,(A(I,J),J=1,6), A(I,4) + A(I,6)
      WRITE(7,116)I,K,(IA(J),J=1,6),IWKDAY
    OTHER COMPILERS MAY NOT ALLOW EXPRESSIONS IN OUTPUT LISTS
   36 CONTINUE
      WRITE(6,145)
      PRINT, WHC
      WRITE (6, 117)
      IF (ENDSW .EQ. 1) GO TO 99
      GO TO 32
  139 FORMAT(1HO,'ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25')
  140 FORMAT(1HO,'ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MI
     1DNIGHT RESPECTIVELY')
  141 FORMAT(1HO,'ITEM 29 CONTAINS 24-HR TOTALS FOR THE MONTH')
  142 FORMAT(1HO,'ITEMS 30,31,32 CONTAIN MONTH-TOTALS OF ACTIVITY 9AM TO
     15PM,5 TO 9, 9 TO MIDNIGHT RESPECTIVELY')
  143 FORMAT(1HO,'ITEM 33 IS AN AUDIT CHECK OF 24-HR TOTALS FOR MONTH')
  144 FORMAT(1HO,'ITEM 34 REPORTS TOTALS OF: EACH HR WITH REV>200/NUMBER
     1 OF HRS WITH REVENUE >200')
  145 FORMAT(1HO,'NEXT ITEM REPORTS NO OF HRS IN MONTH WITH REV >200')
   99 STOP
      END

NTRY
```

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
	JUNE 06, 1970										
1	70157	6	72.80	74.96	120.16	267.91	6.00	0.34	0.50	141.37	5.90
2	70157	6	37.39	129.99	119.30	286.67	6.00	0.17	0.87	140.36	23.63
3	70157	6	53.96	28.86	47.35	130.17	2.00	0.25	0.19	55.71	12.31
4	70157	6	146.99	49.89	109.32	306.21	3.00	0.68	0.33	128.62	3.83
5	70157	6	19.79	23.23	20.07	63.09	6.00	0.09	0.15	23.61	5.54
6	70157	6	118.63	54.36	91.93	264.93	5.00	0.55	0.36	108.16	2.91
7	70157	6	30.83	20.74	23.89	75.47	1.00	0.14	0.14	28.11	0.66
8	70157	6	6.66	78.74	126.53	211.93	4.00	0.03	0.52	148.87	10.91
9	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	70157	6	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.70
11	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.78
12	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11
13	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
15	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.45
16	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	70157	6	1.63	0.61	1.49	3.72	1.00	0.01	0.00	1.75	0.37
19	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	70157	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	70157	6	6.21	15.75	24.16	46.12	6.00	0.03	0.10	28.43	2.94
22	70157	6	21.31	64.92	94.02	180.25	18.00	0.10	0.43	110.61	24.02
23	70157	6	19.07	23.84	34.01	76.93	19.00	0.09	0.16	40.02	5.25
24	70157	6	135.65	181.60	151.11	468.36	25.00	0.63	1.21	177.78	12.32
25	70157	6	670.93	747.48	963.35	2381.76	103.00			115.64	
ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25											
33	70157	6	670.93	747.48	963.34	2381.74	104.00			114.81	2496.55
ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY											
26	70157	6	0.00	0.00	0.00	0.00	1.00			5.05	5.05
27	70157	6	7.84	16.36	25.65	49.84	7.00			3.31	53.15
28	70157	6	176.04	270.37	279.14	725.54	62.00			41.59	767.13

TABLE 3

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70158	7	4.96	2.14	2.50	9.60	3.00	0.02	0.01	2.94	2.48
2	70158	7	4.03	1.02	1.74	6.80	2.00	0.02	0.01	2.05	2.63
3	70158	7	1.89	0.35	1.03	3.27	1.00	0.01	0.00	1.21	0.60
4	70158	7	6.74	1.23	2.02	9.99	4.00	0.03	0.01	2.38	1.52
5	70158	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	70158	7	1.49	1.34	1.70	4.54	1.00	0.01	0.01	2.00	1.80
7	70158	7	51.05	40.63	40.40	132.09	3.00	0.24	0.27	47.53	0.91
8	70158	7	62.47	47.78	51.49	161.74	2.00	0.29	0.32	60.58	2.54
9	70158	7	13.80	7.01	12.09	32.89	1.00	0.06	0.05	14.22	1.78
10	70158	7	37.91	41.39	38.01	117.31	3.00	0.18	0.28	44.72	1.35
11	70158	7	7.57	28.36	37.32	73.25	6.00	0.04	0.19	43.90	12.03
12	70158	7	13.88	131.61	104.96	250.45	4.00	0.06	0.88	123.49	18.60
13	70158	7	16.27	107.93	82.99	207.20	6.00	0.08	0.72	97.64	18.04
14	70158	7	61.67	90.09	70.85	222.61	20.00	0.29	0.60	83.36	9.82
15	70158	7	30.60	95.08	73.30	198.99	6.00	0.14	0.63	86.24	10.31
16	70158	7	14.65	41.36	18.75	74.76	8.00	0.07	0.28	22.06	7.49
17	70158	7	18.86	91.97	58.91	169.75	8.00	0.09	0.61	69.31	3.80
18	70158	7	67.13	125.89	89.50	282.51	15.00	0.31	0.84	105.29	7.76
19	70158	7	39.89	47.74	48.56	136.19	6.00	0.18	0.32	57.13	5.15
20	70158	7	77.87	73.16	70.54	221.58	10.00	0.36	0.49	83.00	3.97
21	70158	7	58.08	98.67	88.52	245.27	5.00	0.27	0.66	104.15	4.77
22	70158	7	62.24	139.46	139.56	341.26	11.00	0.29	0.93	164.19	18.66
23	70158	7	77.28	109.81	118.29	305.38	5.00	0.36	0.73	139.17	7.13
24	70158	7	28.87	14.39	24.50	67.77	19.00	0.13	0.10	28.83	25.03
25	70158	7	759.22	1338.44	1177.53	3275.19	149.00			168.19	

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70158	7	759.21	1338.43	1177.52	3275.15	154.00			168.58	3443.73

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70158	7	201.42	627.80	485.09	1314.31	61.00			81.46	1395.77
27	70158	7	242.97	345.46	297.12	885.55	36.00			21.65	907.19
28	70158	7	168.40	263.66	282.34	714.41	35.00			50.83	765.24

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70159	1	88.15	50.49	48.76	187.40	8.00	0.41	0.34	57.36	3.97	
2	70159	1	161.31	65.33	109.95	336.58	5.00	0.75	0.44	129.36	4.69	
3	70159	1	116.26	87.51	74.28	278.04	11.00	0.54	0.58	87.39	5.12	
4	70159	1	201.65	126.07	127.31	455.03	6.00	0.93	0.84	149.78	8.67	
5	70159	1	129.31	113.63	93.30	336.24	9.00	0.60	0.76	109.77	4.09	
6	70159	1	118.37	69.44	59.75	247.55	14.00	0.55	0.46	70.29	4.29	
7	70159	1	85.41	66.45	79.55	231.41	4.00	0.40	0.44	93.60	3.87	
8	70159	1	217.59	72.66	66.67	356.93	4.00	1.01	0.48	78.44	0.22	
9	70159	1	59.96	22.55	21.31	103.82	4.00	0.28	0.15	25.08	1.70	
10	70159	1	73.38	21.49	36.81	131.68	56.00	0.34	0.14	43.31	33.42	
11	70159	1	96.96	32.45	57.53	186.94	82.00	0.45	0.22	67.69	45.44	
12	70159	1	70.09	24.92	41.81	136.82	68.00	0.32	0.17	49.19	60.13	
13	70159	1	81.03	40.87	63.78	185.67	65.00	0.38	0.27	75.04	41.80	
14	70159	1	71.97	32.86	55.24	160.07	64.00	0.33	0.22	64.99	51.42	
15	70159	1	85.03	56.20	61.95	203.18	88.00	0.39	0.37	72.89	40.78	
16	70159	1	81.25	31.23	52.88	165.35	62.00	0.38	0.21	62.21	38.17	
17	70159	1	59.51	27.76	48.73	136.01	59.00	0.28	0.19	57.34	43.04	
18	70159	1	72.08	30.22	52.99	155.29	49.00	0.33	0.20	62.35	48.78	
19	70159	1	45.35	52.96	49.08	147.39	47.00	0.21	0.35	57.74	23.84	
20	70159	1	56.73	59.68	79.24	195.65	69.00	0.26	0.40	93.23	33.84	
21	70159	1	43.42	93.06	97.50	233.99	44.00	0.20	0.62	114.72	35.56	
22	70159	1	41.98	82.18	80.96	205.12	42.00	0.19	0.55	95.26	33.95	
23	70159	1	51.73	45.90	55.98	153.61	30.00	0.24	0.31	65.86	11.21	
24	70159	1	168.71	131.87	137.21	437.79	44.00	0.78	0.88	161.43	34.01	
25	70159	1	2277.23	1437.77	1652.58	5367.57	934.00				612.02	
TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25												
33	70159	1	2277.14	1437.70	1652.50	5365.58	955.00				611.86	5977.45
TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY												
26	70159	1	619.22	267.77	418.73	1305.73	544.00				354.21	1659.94
27	70159	1	217.58	235.92	278.82	732.33	209.00				142.02	874.35
28	70159	1	262.43	259.95	274.15	796.52	116.00				79.17	875.69

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70160	2	24.47	36.60	56.33	117.41	8.00	0.11	0.24	66.28	24.62	
2	70160	2	34.25	144.86	105.93	285.03	2.00	0.16	0.97	124.63	26.54	
3	70160	2	21.33	43.85	72.56	137.74	3.00	0.10	0.29	85.37	36.16	
4	70160	2	2.44	126.80	201.60	330.84	0.00	0.01	0.85	237.19	0.00	
5	70160	2	13.98	43.52	52.37	109.88	7.00	0.06	0.29	61.62	17.83	
6	70160	2	8.16	25.90	15.77	49.83	2.00	0.04	0.17	18.55	0.00	
7	70160	2	8.16	25.90	15.77	49.83	1.00	0.04	0.17	18.55	0.00	
8	70160	2	1.18	3.74	2.28	7.19	0.00	0.01	0.02	2.68	0.75	
9	70160	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	2.42	
10	70160	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.18	
11	70160	2	56.22	22.28	35.50	114.00	55.00	0.26	0.15	41.76	38.65	
12	70160	2	94.86	25.55	49.60	170.01	46.00	0.44	0.17	58.36	34.74	
13	70160	2	118.36	48.17	65.41	231.95	71.00	0.55	0.32	76.96	37.90	
14	70160	2	92.58	34.69	56.51	183.78	60.00	0.43	0.23	66.48	39.08	
15	70160	2	96.99	40.94	59.15	197.08	68.00	0.45	0.27	69.60	44.89	
16	70160	2	92.32	45.73	65.41	203.46	74.00	0.43	0.30	76.96	45.86	
17	70160	2	83.68	31.89	52.13	167.70	73.00	0.39	0.21	61.33	45.54	
18	70160	2	66.82	26.75	48.01	141.58	68.00	0.31	0.18	56.49	35.21	
19	70160	2	45.56	42.97	45.71	134.24	48.00	0.21	0.29	53.78	26.74	
20	70160	2	78.69	85.65	78.18	242.52	48.00	0.36	0.57	91.98	30.88	
21	70160	2	64.21	75.33	73.28	212.82	26.00	0.30	0.50	86.22	14.63	
22	70160	2	43.76	59.89	65.97	169.62	31.00	0.20	0.40	77.62	37.50	
23	70160	2	26.69	30.65	32.75	90.09	31.00	0.12	0.20	38.53	25.18	
24	70160	2	47.82	50.81	69.87	168.50	35.00	0.22	0.34	82.20	25.22	
25	70160	2	1122.54	1072.47	1320.09	3515.11	758.00				597.53	

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70160	2	1122.46	1072.42	1320.03	3514.87	786.00				597.52	4112.39

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70160	2	635.03	249.24	383.72	1267.99	447.00				293.85	1561.84
27	70160	2	255.28	230.71	245.19	731.17	190.00				107.46	838.64
28	70160	2	118.26	141.35	168.59	428.21	97.00				87.90	516.11

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70161	3	56.62	56.22	77.53	190.37	28.00	0.26	0.37	91.22	63.81	
2	70161	3	18.30	137.43	87.93	243.65	3.00	0.08	0.92	103.45	5.54	
3	70161	3	16.47	42.20	36.72	95.39	1.00	0.08	0.28	43.20	4.55	
4	70161	3	7.45	10.57	9.81	27.82	3.00	0.03	0.07	11.54	1.84	
5	70161	3	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	
6	70161	3	45.05	77.27	116.26	238.58	29.00	0.21	0.52	136.78	14.06	
7	70161	3	10.08	44.37	36.63	91.09	2.00	0.05	0.30	43.09	0.05	
8	70161	3	11.86	39.55	24.29	75.69	0.00	0.05	0.26	28.58	0.00	
9	70161	3	22.19	29.55	21.68	73.42	7.00	0.10	0.20	25.51	8.78	
10	70161	3	99.15	62.23	82.69	244.07	59.00	0.46	0.41	97.29	39.11	
11	70161	3	119.95	53.86	94.62	268.43	66.00	0.56	0.36	111.32	42.24	
12	70161	3	98.35	43.47	71.03	212.85	44.00	0.46	0.29	83.56	47.56	
13	70161	3	84.19	37.88	63.71	185.79	63.00	0.39	0.25	74.96	31.94	
14	70161	3	74.57	47.05	67.23	188.86	83.00	0.35	0.31	79.10	58.29	
15	70161	3	58.46	24.87	47.11	130.44	64.00	0.27	0.17	55.43	45.62	
16	70161	3	79.02	39.77	64.13	182.93	91.00	0.37	0.27	75.46	42.55	
17	70161	3	69.42	33.60	45.69	148.70	57.00	0.32	0.22	53.75	45.95	
18	70161	3	40.31	31.29	31.13	102.73	37.00	0.19	0.21	36.62	24.81	
19	70161	3	37.74	32.80	31.36	101.90	41.00	0.17	0.22	36.90	25.37	
20	70161	3	32.21	30.71	36.89	99.81	51.00	0.15	0.20	43.40	24.72	
21	70161	3	32.17	49.82	43.26	125.24	49.00	0.15	0.33	50.89	26.01	
22	70161	3	48.81	43.00	48.49	140.30	45.00	0.23	0.29	57.05	37.71	
23	70161	3	41.85	61.54	65.32	168.70	35.00	0.19	0.41	76.85	24.86	
24	70161	3	60.76	145.88	125.95	332.59	48.00	0.28	0.97	148.18	25.34	
25	70161	3	1164.95	1174.95	1329.44	3669.36	908.00				640.71	
ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25												
33	70161	3	1208.78	1344.22	1430.99	3983.94	944.00				640.51	4624.45
ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY												
26	70161	3	683.10	342.75	536.21	1562.07	527.00				353.25	1915.33
27	70161	3	142.42	144.63	142.63	429.68	178.00				100.91	530.59
28	70161	3	151.42	250.42	239.76	641.59	128.00				87.91	729.50

TABLE 3 CONTINUED.

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70162	4	20.96	54.71	52.54	128.22	12.00	0.10	0.36	61.82	13.24	
2	70162	4	33.16	33.67	43.05	109.89	18.00	0.15	0.22	50.65	18.17	
3	70162	4	25.54	79.30	69.84	174.68	6.00	0.12	0.53	82.17	12.84	
4	70162	4	36.15	104.47	94.31	234.93	4.00	0.17	0.70	110.96	0.41	
5	70162	4	27.30	48.52	39.89	115.71	1.00	0.13	0.32	46.93	1.73	
6	70162	4	18.80	74.09	50.25	143.13	5.00	0.09	0.49	59.12	2.17	
7	70162	4	1.41	12.39	7.48	21.28	1.00	0.01	0.08	8.80	4.12	
8	70162	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	
9	70162	4	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	3.01	
10	70162	4	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.37	
11	70162	4	72.34	32.91	57.17	162.41	70.00	0.33	0.22	67.26	42.90	
12	70162	4	111.38	46.82	75.18	233.37	58.00	0.52	0.31	88.45	54.20	
13	70162	4	130.02	33.80	77.13	240.95	56.00	0.60	0.23	90.74	31.28	
14	70162	4	127.38	27.14	56.59	211.11	62.00	0.59	0.18	66.58	55.17	
15	70162	4	129.53	30.17	60.54	220.24	61.00	0.60	0.20	71.23	44.93	
16	70162	4	86.37	62.91	69.15	218.42	49.00	0.40	0.42	81.36	37.78	
17	70162	4	29.15	15.97	20.82	65.93	24.00	0.13	0.11	24.49	17.31	
18	70162	4	54.30	32.63	40.29	127.22	59.00	0.25	0.22	47.40	36.89	
19	70162	4	59.23	40.75	63.28	163.26	74.00	0.27	0.27	74.45	33.94	
20	70162	4	91.24	62.86	97.11	251.21	60.00	0.42	0.42	114.26	45.87	
21	70162	4	78.48	92.49	121.33	292.30	32.00	0.36	0.62	142.75	27.52	
22	70162	4	72.41	82.66	118.81	273.88	16.00	0.34	0.55	139.78	30.63	
23	70162	4	78.90	103.26	97.75	279.92	17.00	0.37	0.69	115.01	14.46	
24	70162	4	63.77	74.41	80.10	218.28	31.00	0.30	0.50	94.23	7.10	
25	70162	4	1347.78	1145.95	1392.60	3886.34	721.00				536.05	

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70162	4	1411.78	1396.74	1543.08	4351.41	744.00				529.25	4880.66

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70162	4	686.15	249.72	416.56	1352.43	382.00				283.95	1636.38
27	70162	4	283.24	228.73	322.02	834.00	225.00				144.23	978.23
28	70162	4	215.08	260.34	296.66	772.07	64.00				52.19	824.27

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70163	5	103.23	116.44	113.11	332.78	8.00	0.48	0.78	133.08	18.07	
2	70163	5	68.01	133.41	96.65	298.07	2.00	0.31	0.89	113.71	7.21	
3	70163	5	75.80	84.68	136.82	297.30	7.00	0.35	0.56	160.97	6.00	
4	70163	5	73.48	94.64	133.39	301.52	10.00	0.34	0.63	156.94	23.92	
5	70163	5	102.65	111.18	98.12	311.95	8.00	0.48	0.74	115.44	2.34	
6	70163	5	181.81	123.75	135.92	441.48	3.00	0.84	0.83	159.91	13.49	
7	70163	5	131.36	50.14	86.53	268.03	6.00	0.61	0.33	101.81	5.69	
8	70163	5	93.89	51.59	115.66	261.13	4.00	0.43	0.34	136.07	0.13	
9	70163	5	7.37	95.76	153.20	256.34	3.00	0.03	0.64	180.24	34.70	
10	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
11	70163	5	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.06	
12	70163	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.33	
13	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
14	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.47	
15	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.79	
16	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
17	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
18	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	
19	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.07	
20	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
21	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	
22	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.73	
23	70163	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	
24	70163	5	0.28	25.00	22.76	48.03	1.00	0.00	0.17	26.77	1.36	
25	70163	5	837.89	886.59	1092.15	2816.63	55.00				119.58	

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70163	5	837.88	886.58	1092.15	2816.61	57.00				126.68	2943.30

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70163	5	0.00	0.00	0.00	0.00	3.00				3.66	3.66
27	70163	5	0.00	0.00	0.00	0.00	0.00				2.29	2.29
28	70163	5	0.28	25.00	22.76	48.03	1.00				2.10	50.14

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
2	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
3	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
4	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
5	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
6	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
7	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
10	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
11	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
12	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
13	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
14	70164	6	17.69	1.71	8.60	28.00	7.00	0.08	0.01	10.11	2.67	
15	70164	6	4.64	0.12	1.92	6.69	4.00	0.02	0.00	2.26	0.52	
16	70164	6	14.95	1.80	4.00	20.75	16.00	0.07	0.01	4.71	2.26	
17	70164	6	46.71	8.47	16.96	72.14	30.00	0.22	0.06	19.95	6.33	
18	70164	6	85.51	15.01	33.74	134.26	53.00	0.40	0.10	39.69	19.36	
19	70164	6	88.45	13.64	34.28	136.37	48.00	0.41	0.09	40.33	13.83	
20	70164	6	103.40	12.72	38.85	154.96	20.00	0.48	0.08	45.70	1.38	
21	70164	6	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	
22	70164	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	
23	70164	6	1.12	0.05	0.22	1.39	1.00	0.01	0.00	0.26	4.85	
24	70164	6	2.96	0.15	0.67	3.78	2.00	0.01	0.00	0.78	0.15	
25	70164	6	365.43	53.68	139.23	558.34	186.00				51.35	

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70164	6	365.42	53.68	139.23	558.29	201.00				46.58	604.87

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70164	6	83.99	12.10	31.48	127.57	57.00				11.77	139.35
27	70164	6	277.36	41.37	106.87	425.59	126.00				34.57	460.16
28	70164	6	4.08	0.21	0.89	5.18	3.00				5.01	10.19

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
15	70165	7	4.12	0.15	0.69	4.96	8.00	0.02	0.00	0.81	0.32
16	70165	7	15.13	4.82	11.49	31.45	8.00	0.07	0.03	13.52	6.58
17	70165	7	41.75	1.09	6.81	49.65	27.00	0.19	0.01	8.01	4.86
18	70165	7	30.50	0.04	4.14	34.68	18.00	0.14	0.00	4.87	2.89
19	70165	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	70165	7	5.96	0.80	1.60	8.36	3.00	0.03	0.01	1.88	1.54
21	70165	7	18.20	4.17	7.37	29.75	16.00	0.08	0.03	8.67	7.78
22	70165	7	11.36	4.49	10.52	26.37	4.00	0.05	0.03	12.38	1.80
23	70165	7	73.73	118.08	102.48	294.29	12.00	0.34	0.79	120.57	6.26
24	70165	7	41.71	60.71	47.55	149.97	39.00	0.19	0.40	55.94	11.73
25	70165	7	242.46	194.36	192.65	629.47	135.00				43.79

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70165	7	242.46	194.36	192.65	629.47	135.00			48.29	677.76

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70165	7	61.00	6.06	18.99	86.05	43.00			11.78	97.84
27	70165	7	54.67	5.02	13.11	72.79	37.00			12.21	85.01
28	70165	7	126.80	183.28	160.55	470.62	55.00			19.79	490.41

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70166	1	33.01	16.03	29.21	78.25	17.00	0.15	0.11	34.37	4.09	
2	70166	1	25.26	6.54	18.03	49.83	23.00	0.12	0.04	21.21	6.41	
3	70166	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	
4	70166	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
5	70166	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
6	70166	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
7	70166	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70166	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9	70166	1	50.47	71.37	74.85	196.69	34.00	0.23	0.48	88.07	9.53	
10	70166	1	57.05	36.79	49.70	143.55	48.00	0.26	0.25	58.48	65.32	
11	70166	1	116.57	124.47	99.18	340.21	51.00	0.54	0.83	116.68	38.29	
12	70166	1	63.94	79.07	88.08	231.08	67.00	0.30	0.53	103.62	39.82	
13	70166	1	58.34	107.60	76.04	241.97	29.00	0.27	0.72	89.46	49.05	
14	70166	1	91.95	57.68	76.12	225.75	76.00	0.43	0.38	89.56	60.54	
15	70166	1	93.08	27.88	51.01	171.97	74.00	0.43	0.19	60.02	51.22	
16	70166	1	71.93	23.04	38.83	133.80	72.00	0.33	0.15	45.68	35.72	
17	70166	1	91.03	41.94	65.79	198.76	97.00	0.42	0.28	77.40	43.72	
18	70166	1	101.75	39.31	51.34	192.40	48.00	0.47	0.26	60.41	41.38	
19	70166	1	66.99	78.64	65.00	210.63	40.00	0.31	0.52	76.48	27.34	
20	70166	1	78.18	79.34	105.46	262.98	59.00	0.36	0.53	124.07	46.59	
21	70166	1	60.67	131.60	136.19	328.46	38.00	0.28	0.88	160.23	33.75	
22	70166	1	35.88	97.27	84.66	217.82	31.00	0.17	0.65	99.61	31.19	
23	70166	1	16.21	45.11	36.25	97.57	11.00	0.08	0.30	42.64	12.40	
24	70166	1	29.70	56.39	70.66	156.76	38.00	0.14	0.38	83.13	24.25	
25	70166	1	1142.00	1120.08	1216.40	3478.48	853.00				620.72	

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70166	1	1257.17	1126.81	1241.27	3625.22	872.00				620.83	4246.05

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70166	1	643.88	498.47	544.74	1687.09	514.00				383.68	2070.77
27	70166	1	307.60	328.88	357.99	994.47	185.00				149.06	1143.53
28	70166	1	81.80	198.78	191.57	472.15	80.00				67.83	539.99

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70170	5	37.65	38.58	48.12	124.36	13.00	0.17	0.26	56.62	19.71
2	70170	5	79.88	42.77	64.22	186.87	11.00	0.37	0.29	75.56	7.38
3	70170	5	54.91	59.93	49.70	164.55	9.00	0.25	0.40	58.47	3.20
4	70170	5	46.37	25.64	42.05	114.06	7.00	0.21	0.17	49.47	5.90
5	70170	5	28.80	21.09	38.05	87.94	5.00	0.13	0.14	44.76	1.79
6	70170	5	82.70	79.59	130.40	292.69	5.00	0.38	0.53	153.42	3.41
7	70170	5	5.91	7.42	11.09	24.42	1.00	0.03	0.05	13.04	0.84
8	70170	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
9	70170	5	41.67	9.37	17.50	68.54	23.00	0.19	0.06	20.59	8.54
10	70170	5	99.24	70.46	88.45	258.15	79.00	0.46	0.47	104.06	41.01
11	70170	5	112.17	51.71	68.70	232.57	74.00	0.52	0.34	80.82	53.49
12	70170	5	140.19	36.53	77.86	254.58	100.00	0.65	0.24	91.61	46.62
13	70170	5	63.89	89.61	86.52	240.02	52.00	0.30	0.60	101.79	43.06
14	70170	5	87.21	49.34	77.71	214.26	58.00	0.40	0.33	91.43	52.53
15	70170	5	13.73	6.19	12.08	32.00	9.00	0.06	0.04	14.21	17.07
16	70170	5	23.74	12.85	20.48	57.08	21.00	0.11	0.09	24.10	13.55
17	70170	5	42.58	10.29	20.01	72.88	43.00	0.20	0.07	23.54	27.02
18	70170	5	75.60	53.44	79.40	208.44	60.00	0.35	0.36	93.42	53.03
19	70170	5	66.27	118.80	103.42	288.49	25.00	0.31	0.79	121.67	17.53
20	70170	5	97.34	126.87	110.34	334.54	64.00	0.45	0.85	129.82	48.74
21	70170	5	18.26	4.63	11.68	34.56	15.00	0.08	0.03	13.74	9.21
22	70170	5	7.68	1.87	3.74	13.30	12.00	0.04	0.01	4.40	4.47
23	70170	5	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00
24	70170	5	55.18	75.51	57.00	187.69	2.00	0.26	0.50	67.06	0.35
25	70170	5	1280.98	992.50	1218.51	3491.99	691.00	478.46			

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70170	5	1280.91	992.44	1218.44	3491.78	700.00	477.55	3969.32		

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70170	5	582.76	326.99	451.80	1361.55	436.00	294.36	1655.91		
27	70170	5	257.46	303.74	304.83	866.04	164.00	128.50	994.54		
28	70170	5	62.86	77.39	60.74	200.99	16.00	4.82	205.81		

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.19	
2	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
3	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
4	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
5	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
6	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
7	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
10	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
11	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
12	70171	6	1.85	0.10	0.53	2.48	1.00	0.01	0.00	0.62	0.08	
13	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
14	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
15	70171	6	14.29	29.04	28.06	71.40	5.00	0.07	0.19	33.02	5.32	
16	70171	6	5.18	2.87	4.55	12.60	7.00	0.02	0.02	5.35	4.58	
17	70171	6	30.46	10.32	53.34	94.12	8.00	0.14	0.07	62.76	4.26	
18	70171	6	75.63	47.15	187.87	310.64	1.00	0.35	0.31	221.03	0.87	
19	70171	6	6.91	2.94	13.22	23.07	4.00	0.03	0.02	15.55	3.28	
20	70171	6	41.34	11.89	75.53	128.76	2.00	0.19	0.08	88.86	0.88	
21	70171	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
22	70171	6	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.06	
23	70171	6	0.17	0.01	0.03	0.21	1.00	0.00	0.00	0.04	0.06	
24	70171	6	234.74	145.43	574.28	954.45	2.00	1.09	0.97	675.65	0.45	
25	70171	6	410.58	249.74	937.41	1597.73	32.00				26.03	

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70171	6	410.57	249.74	937.41	1597.73	33.00				27.06	1624.78

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70171	6	51.79	42.32	86.48	180.59	21.00				14.23	194.83
27	70171	6	123.88	61.98	276.61	462.47	7.00				5.03	467.51
28	70171	6	234.91	145.44	574.31	954.66	4.00				0.56	955.22

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70172	7	12.83	23.32	32.53	68.68	8.00	0.06	0.16	38.27	13.51
2	70172	7	8.40	0.70	2.26	11.36	2.00	0.04	0.00	2.66	1.12
3	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
11	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	70172	7	6.30	1.25	10.26	17.81	1.00	0.03	0.01	12.07	1.73
15	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.57
16	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	70172	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	70172	7	3.39	3.48	4.98	11.86	3.00	0.02	0.02	5.86	3.14
21	70172	7	12.27	7.05	10.41	29.73	2.00	0.06	0.05	12.24	9.68
22	70172	7	0.89	0.02	0.17	1.08	1.00	0.00	0.00	0.20	0.62
23	70172	7	44.14	30.85	51.58	126.57	11.00	0.20	0.21	60.68	13.16
24	70172	7	60.81	45.69	191.46	297.97	16.00	0.28	0.30	225.26	68.54
25	70172	7	149.03	112.38	303.65	565.06	44.00			113.08	

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70172	7	154.89	114.62	315.39	584.89	46.00			110.28	695.17

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70172	7	6.30	1.25	10.26	17.81	1.00			3.31	21.12
27	70172	7	15.67	10.54	15.39	41.59	5.00			12.82	54.41
28	70172	7	105.84	76.57	243.21	425.61	28.00			82.32	507.93

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70173	1	40.98	23.28	63.19	127.45	12.00	0.19	0.16	74.35	34.89
2	70173	1	1.29	0.19	0.80	2.28	3.00	0.01	0.00	0.94	1.29
3	70173	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	70173	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	70173	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	70173	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	70173	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	70173	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	70173	1	35.41	13.87	25.07	74.34	28.00	0.16	0.09	29.49	21.59
10	70173	1	63.65	40.64	67.47	171.76	68.00	0.29	0.27	79.38	40.01
11	70173	1	68.30	22.84	35.08	126.23	71.00	0.32	0.15	41.28	47.86
12	70173	1	78.20	52.73	55.31	186.23	79.00	0.36	0.35	65.07	45.82
13	70173	1	72.68	61.59	81.18	215.46	77.00	0.34	0.41	95.51	38.99
14	70173	1	67.38	21.05	47.47	135.90	59.00	0.31	0.14	55.86	54.91
15	70173	1	64.23	28.94	51.61	144.78	56.00	0.30	0.19	60.72	112.02
16	70173	1	90.11	33.53	63.03	186.67	86.00	0.42	0.22	74.15	50.60
17	70173	1	140.35	84.53	125.13	350.01	101.00	0.65	0.56	147.22	63.66
18	70173	1	119.37	100.41	123.95	343.73	56.00	0.55	0.67	145.83	68.22
19	70173	1	106.13	126.09	105.90	338.13	25.00	0.49	0.84	124.60	35.02
20	70173	1	40.29	124.71	102.18	267.18	38.00	0.19	0.83	120.22	61.18
21	70173	1	80.43	116.51	111.90	308.84	45.00	0.37	0.78	131.65	40.07
22	70173	1	81.64	118.29	70.44	270.37	24.00	0.38	0.79	82.87	51.39
23	70173	1	54.45	83.35	62.31	200.11	15.00	0.25	0.56	73.31	24.14
24	70173	1	144.04	202.30	192.53	538.87	39.00	0.67	1.35	226.52	24.32
25	70173	1	1348.91	1254.85	1384.57	3988.33	882.00			815.96	

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70173	1	1348.83	1254.78	1384.49	3988.06	906.00			817.90	4805.96

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70173	1	644.89	345.86	526.29	1517.04	597.00			453.85	1970.89
27	70173	1	346.22	467.72	443.94	1257.88	164.00			204.48	1462.36
28	70173	1	280.13	403.94	325.28	1009.35	78.00			99.85	1109.21

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70174	2	157.27	16.51	48.73	222.51	18.00	0.73	0.11	57.33	7.45
2	70174	2	45.61	10.09	26.64	82.34	17.00	0.21	0.07	31.35	9.96
3	70174	2	24.49	0.64	2.34	27.47	3.00	0.11	0.00	2.75	0.98
4	70174	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	70174	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
6	70174	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	70174	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	70174	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	70174	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.42
10	70174	2	19.56	7.48	12.57	39.61	11.00	0.09	0.05	14.79	12.39
11	70174	2	104.27	35.14	57.41	196.82	72.00	0.48	0.23	67.55	80.00
12	70174	2	98.50	36.80	71.24	206.54	89.00	0.46	0.25	83.81	54.28
13	70174	2	72.12	59.67	81.74	213.53	57.00	0.33	0.40	96.17	27.42
14	70174	2	78.89	75.66	97.23	251.78	65.00	0.37	0.50	114.40	81.29
15	70174	2	93.23	67.94	86.44	247.62	107.00	0.43	0.45	101.70	48.03
16	70174	2	101.30	68.74	101.08	271.12	85.00	0.47	0.46	118.92	66.18
17	70174	2	84.10	43.66	62.15	189.91	79.00	0.39	0.29	73.12	41.02
18	70174	2	114.67	110.95	143.35	368.97	74.00	0.53	0.74	168.65	68.59
19	70174	2	71.70	110.14	98.90	280.75	38.00	0.33	0.73	116.36	32.84
20	70174	2	52.49	144.39	97.33	294.21	46.00	0.24	0.96	114.52	11.18
21	70174	2	82.24	166.19	168.69	417.12	54.00	0.38	1.11	198.47	100.34
22	70174	2	56.00	143.31	91.35	290.66	39.00	0.26	0.96	107.48	63.09
23	70174	2	46.64	163.61	82.33	292.58	25.00	0.22	1.09	96.86	44.04
24	70174	2	89.45	100.56	81.60	271.61	39.00	0.41	0.67	96.00	49.66
25	70174	2	1392.54	1361.47	1411.13	4165.15	919.00				799.14
TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25											
33	70174	2	1446.85	1365.27	1424.38	4236.23	932.00				797.16 5033.39
TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY											
26	70174	2	651.97	395.08	569.87	1616.93	565.00				410.60 2027.53
27	70174	2	321.11	531.67	508.27	1361.05	212.00				212.95 1574.00
28	70174	2	192.09	407.48	255.28	854.85	103.00				156.79 1011.64

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70175	3	139.57	48.98	89.76	278.30	46.00	0.65	0.33	105.60	45.50	
2	70175	3	154.13	160.86	111.20	426.18	25.00	0.71	1.07	130.83	47.25	
3	70175	3	35.66	159.16	89.97	284.79	5.00	0.17	1.06	105.85	25.65	
4	70175	3	12.52	63.17	46.87	122.56	2.00	0.06	0.42	55.14	21.54	
5	70175	3	8.17	6.43	14.24	28.84	6.00	0.04	0.04	16.75	19.54	
6	70175	3	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	
7	70175	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70175	3	1.60	0.03	0.22	1.85	4.00	0.01	0.00	0.26	0.11	
9	70175	3	37.75	95.60	64.73	198.08	20.00	0.17	0.64	76.16	9.44	
10	70175	3	93.72	62.04	92.79	248.56	65.00	0.43	0.41	109.17	37.71	
11	70175	3	41.35	56.79	79.79	177.94	44.00	0.19	0.38	93.87	63.00	
12	70175	3	80.34	56.81	67.18	204.34	80.00	0.37	0.38	79.04	61.86	
13	70175	3	51.17	55.81	68.62	175.59	43.00	0.24	0.37	80.73	35.75	
14	70175	3	81.93	29.61	55.19	166.73	72.00	0.38	0.20	64.93	37.18	
15	70175	3	65.00	21.80	43.92	130.72	40.00	0.30	0.15	51.67	35.09	
16	70175	3	83.88	48.71	73.44	206.02	108.00	0.39	0.32	86.40	55.73	
17	70175	3	82.02	40.33	55.89	178.24	66.00	0.38	0.27	65.75	47.36	
18	70175	3	49.08	79.75	87.08	215.92	34.00	0.23	0.53	102.45	33.66	
19	70175	3	107.72	112.24	108.68	328.63	29.00	0.50	0.75	127.86	32.49	
20	70175	3	45.49	142.88	76.79	265.17	22.00	0.21	0.95	90.34	17.30	
21	70175	3	80.38	46.56	62.92	189.86	36.00	0.37	0.31	74.03	40.97	
22	70175	3	22.77	50.64	63.51	136.91	30.00	0.11	0.34	74.72	6.17	
23	70175	3	16.38	11.72	12.28	40.37	14.00	0.08	0.08	14.45	34.36	
24	70175	3	32.87	34.28	43.71	110.86	26.00	0.15	0.23	51.42	24.42	
25	70175	3	1323.51	1384.20	1408.75	4116.45	820.00			732.10		

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70175	3	1323.43	1384.12	1408.67	4116.13	844.00			734.30	4850.43	

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70175	3	579.41	371.90	536.81	1488.13	518.00			373.68	1861.81	
27	70175	3	282.68	381.44	335.46	999.58	121.00			124.42	1124.00	
28	70175	3	72.01	96.63	119.49	288.14	70.00			64.95	353.09	

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70176	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	
2	70176	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	
3	70176	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
4	70176	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
5	70176	4	0.76	0.00	0.08	0.84	1.00	0.00	0.00	0.09	0.04	
6	70176	4	1.76	0.08	0.28	2.12	1.00	0.01	0.00	0.33	0.09	
7	70176	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70176	4	1.10	0.06	0.19	1.36	1.00	0.01	0.00	0.22	0.60	
9	70176	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
10	70176	4	4.81	0.28	0.90	5.99	1.00	0.02	0.00	1.06	1.13	
11	70176	4	104.75	32.12	52.15	189.01	78.00	0.48	0.21	61.35	46.77	
12	70176	4	139.50	49.81	80.67	269.98	99.00	0.65	0.33	94.91	46.34	
13	70176	4	125.72	66.90	86.61	279.23	56.00	0.58	0.45	101.90	31.79	
14	70176	4	133.20	94.54	121.15	348.90	71.00	0.62	0.63	142.54	40.83	
15	70176	4	82.81	17.02	36.34	136.17	52.00	0.38	0.11	42.76	63.42	
16	70176	4	77.82	17.34	36.07	131.23	53.00	0.36	0.12	42.43	26.50	
17	70176	4	105.88	50.03	64.61	220.52	134.00	0.49	0.33	76.01	49.47	
18	70176	4	122.73	47.26	88.18	258.16	76.00	0.57	0.32	103.74	33.59	
19	70176	4	103.81	125.08	141.75	370.63	58.00	0.48	0.83	166.77	33.82	
20	70176	4	37.39	84.28	67.97	189.64	24.00	0.17	0.56	79.96	27.97	
21	70176	4	58.51	126.01	144.74	329.26	40.00	0.27	0.84	170.29	43.95	
22	70176	4	71.04	133.54	115.42	320.00	29.00	0.33	0.89	135.79	20.89	
23	70176	4	76.84	56.21	79.07	212.12	38.00	0.36	0.37	93.03	26.64	
24	70176	4	33.26	21.63	30.74	85.63	33.00	0.15	0.14	36.17	16.51	
25	70176	4	1281.70	922.19	1146.91	3350.80	845.00				510.45	

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70176	4	1281.63	922.10	1146.82	3350.52	860.00				510.51	3861.03

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70176	4	774.49	328.04	478.50	1581.03	544.00				306.25	1887.28
27	70176	4	322.44	382.63	442.63	1147.70	198.00				139.33	1287.03
28	70176	4	181.14	211.38	225.23	617.75	100.00				64.04	681.79

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70177	5	27.52	122.87	156.79	307.18	23.00	0.13	0.82	184.46	29.81	
2	70177	5	15.71	36.00	43.77	95.48	18.00	0.07	0.24	51.50	6.76	
3	70177	5	8.84	1.40	3.17	13.42	11.00	0.04	0.01	3.73	5.32	
4	70177	5	2.99	0.07	0.31	3.37	4.00	0.01	0.00	0.36	0.12	
5	70177	5	11.11	0.97	3.05	15.13	12.00	0.05	0.01	3.59	1.31	
6	70177	5	7.44	0.30	1.59	9.33	3.00	0.03	0.00	1.87	0.09	
7	70177	5	6.14	1.48	2.57	10.18	3.00	0.03	0.01	3.02	2.38	
8	70177	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9	70177	5	58.83	12.60	23.39	94.82	27.00	0.27	0.08	27.51	15.03	
10	70177	5	78.46	109.94	93.21	281.60	37.00	0.36	0.73	109.66	27.83	
11	70177	5	87.61	51.35	58.32	197.28	66.00	0.41	0.34	68.61	30.16	
12	70177	5	126.93	35.06	67.05	229.05	81.00	0.59	0.23	78.89	40.05	
13	70177	5	114.52	25.77	47.70	187.99	75.00	0.53	0.17	56.12	28.41	
14	70177	5	122.38	38.67	69.73	230.78	81.00	0.57	0.26	82.04	48.87	
15	70177	5	104.00	46.75	69.58	220.33	75.00	0.48	0.31	81.86	42.30	
16	70177	5	112.00	47.92	74.63	234.55	74.00	0.52	0.32	87.80	39.29	
17	70177	5	125.64	54.73	85.39	265.76	78.00	0.58	0.36	100.46	46.78	
18	70177	5	90.93	154.07	149.77	394.77	57.00	0.42	1.03	176.21	40.64	
19	70177	5	23.96	104.21	78.24	206.41	17.00	0.11	0.69	92.05	9.20	
20	70177	5	35.34	65.31	74.38	175.03	44.00	0.16	0.44	87.51	30.97	
21	70177	5	36.37	13.33	32.09	81.79	22.00	0.17	0.09	37.76	22.76	
22	70177	5	40.92	16.01	28.76	85.70	22.00	0.19	0.11	33.84	24.86	
23	70177	5	22.43	10.10	22.27	54.81	9.00	0.10	0.07	26.21	17.54	
24	70177	5	43.08	65.15	79.02	187.24	22.00	0.20	0.43	92.97	17.83	
25	70177	5	1303.15	1014.06	1264.76	3581.98	861.00				528.32	
TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25												
33	70177	5	1303.07	1013.99	1264.67	3581.71	874.00				528.31	4110.02
TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY												
26	70177	5	871.54	410.19	565.59	1847.33	567.00				303.69	2151.02
27	70177	5	186.61	336.91	334.48	858.00	140.00				103.57	961.57
28	70177	5	106.43	91.26	130.06	327.75	53.00				60.22	387.97

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70178	6	27.74	3.23	17.75	48.72	4.00	0.13	0.02	20.89	4.60	
2	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
3	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
4	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
5	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
6	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
7	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
10	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
11	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
12	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
13	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
14	70178	6	1.60	0.03	0.13	1.77	2.00	0.01	0.00	0.15	0.01	
15	70178	6	37.73	2.01	8.60	48.33	17.00	0.17	0.01	10.11	5.01	
16	70178	6	11.25	0.50	2.08	13.83	11.00	0.05	0.00	2.44	3.03	
17	70178	6	0.72	0.04	0.17	0.93	0.00	0.00	0.00	0.20	0.00	
18	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
19	70178	6	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.23	
20	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
21	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
22	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
23	70178	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
24	70178	6	1.48	0.25	1.03	2.76	1.00	0.01	0.00	1.21	0.51	
25	70178	6	80.52	6.07	29.75	116.35	36.00				13.40	

ITEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70178	6	80.52	6.07	29.75	116.35	36.00				13.20	129.55

ITEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70178	6	51.30	2.59	10.97	64.86	30.00				8.06	72.92
27	70178	6	0.00	0.00	0.00	0.00	1.00				0.23	0.23
28	70178	6	1.48	0.25	1.03	2.76	1.00				0.51	3.27

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV
1	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
2	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
3	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
4	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
5	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
6	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
7	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
8	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
10	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
11	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
12	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
13	70179	7	3.20	1.06	1.94	6.21	2.00	0.01	0.01	2.29	1.61	
14	70179	7	8.41	6.29	7.79	22.48	8.00	0.04	0.04	9.16	3.77	
15	70179	7	20.86	5.15	9.45	35.47	28.00	0.10	0.03	11.12	3.29	
16	70179	7	20.03	5.62	9.41	35.06	22.00	0.09	0.04	11.07	1.81	
17	70179	7	18.00	3.08	7.05	28.13	22.00	0.08	0.02	8.30	23.44	
18	70179	7	21.61	3.26	7.59	32.47	23.00	0.10	0.02	8.94	5.66	
19	70179	7	11.88	2.87	5.10	19.85	10.00	0.05	0.02	6.00	4.30	
20	70179	7	9.14	2.21	3.76	15.11	6.00	0.04	0.01	4.42	2.06	
21	70179	7	1.17	0.00	0.05	1.23	1.00	0.01	0.00	0.06	0.04	
22	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
23	70179	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
24	70179	7	6.41	89.30	103.34	199.06	8.00	0.03	0.60	121.59	5.31	
25	70179	7	120.72	118.85	155.50	395.07	130.00				51.29	

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70179	7	153.52	121.21	163.49	438.19	131.00			50.85	489.04

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70179	7	70.50	21.20	35.65	127.35	82.00			33.92	161.27
27	70179	7	43.80	8.35	16.50	68.66	40.00			12.06	80.72
28	70179	7	6.41	89.30	103.34	199.06	8.00			5.31	204.37

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70180	1	5.74	1.66	3.61	11.01	7.00	0.03	0.01	4.25	3.87
2	70180	1	18.94	2.41	9.36	30.71	2.00	0.09	0.02	11.01	4.34
3	70180	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	70180	1	4.29	0.56	2.58	7.43	1.00	0.02	0.00	3.04	6.24
5	70180	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	70180	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	70180	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	70180	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	70180	1	44.88	15.41	22.02	82.31	24.00	0.21	0.10	25.90	12.04
10	70180	1	76.35	35.54	52.58	164.47	43.00	0.35	0.24	61.86	35.18
11	70180	1	90.59	31.73	45.69	168.00	71.00	0.42	0.21	53.75	31.79
12	70180	1	84.80	37.83	55.55	178.18	68.00	0.39	0.25	65.36	42.90
13	70180	1	78.92	80.45	98.58	257.95	50.00	0.37	0.54	115.98	10.22
14	70180	1	82.76	41.98	52.68	177.42	67.00	0.38	0.28	61.98	19.08
15	70180	1	128.42	94.38	89.34	312.14	71.00	0.59	0.63	105.11	46.03
16	70180	1	96.76	92.14	101.01	289.92	94.00	0.45	0.61	118.85	89.11
17	70180	1	90.39	38.78	57.11	186.27	94.00	0.42	0.26	67.19	64.93
18	70180	1	98.81	83.28	113.82	295.91	41.00	0.46	0.56	133.91	29.57
19	70180	1	45.36	77.30	78.75	201.41	42.00	0.21	0.52	92.65	30.44
20	70180	1	39.70	122.05	66.02	227.77	38.00	0.18	0.81	77.67	29.60
21	70180	1	28.07	105.69	77.53	211.29	31.00	0.13	0.70	91.22	36.22
22	70180	1	26.04	118.65	96.17	240.86	32.00	0.12	0.79	113.14	42.87
23	70180	1	18.50	67.20	51.91	137.61	19.00	0.09	0.45	61.07	19.49
24	70180	1	45.99	12.41	28.46	86.86	30.00	0.21	0.08	33.48	9.96
25	70180	1	1105.31	1059.44	1102.75	3267.51	825.00				563.88
TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25											
33	70180	1	1147.83	1061.94	1112.79	3322.51	838.00				564.06 3886.57
TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY											
26	70180	1	728.99	452.83	552.53	1734.35	558.00				339.24 2073.59
27	70180	1	211.93	388.32	336.12	936.37	152.00				125.83 1062.20
28	70180	1	90.53	198.26	176.54	465.33	81.00				72.32 537.65

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS REV
1	70181	2	8.66	5.19	5.31	19.16	10.00	0.04	0.03	6.24	3.34
2	70181	2	4.85	0.67	2.59	8.12	6.00	0.02	0.00	3.05	3.99
3	70181	2	3.43	2.08	5.29	10.80	7.00	0.02	0.01	6.22	5.51
4	70181	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	70181	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
6	70181	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	70181	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	70181	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	70181	2	75.27	15.81	22.41	113.50	34.00	0.35	0.11	26.37	16.36
10	70181	2	93.23	63.65	61.62	218.49	58.00	0.43	0.42	72.49	37.56
11	70181	2	113.40	69.15	69.57	252.12	76.00	0.52	0.46	81.86	42.59
12	70181	2	71.59	44.69	54.64	170.92	52.00	0.33	0.30	64.28	30.02
13	70181	2	78.07	39.89	63.02	180.98	63.00	0.36	0.27	74.14	35.89
14	70181	2	101.31	42.18	63.08	206.56	83.00	0.47	0.28	74.21	44.09
15	70181	2	107.68	61.33	98.00	267.01	72.00	0.50	0.41	115.30	45.61
16	70181	2	99.95	59.98	84.08	244.01	77.00	0.46	0.40	98.92	48.17
17	70181	2	88.98	107.71	112.95	309.63	51.00	0.41	0.72	132.89	39.26
18	70181	2	123.93	79.39	96.87	300.19	39.00	0.57	0.53	113.97	22.44
19	70181	2	50.53	45.30	65.60	161.43	24.00	0.23	0.30	77.18	31.32
20	70181	2	31.48	70.77	78.38	180.63	20.00	0.15	0.47	92.22	12.15
21	70181	2	7.39	142.52	162.06	311.97	15.00	0.03	0.95	190.67	12.73
22	70181	2	3.85	142.32	161.25	307.43	5.00	0.02	0.95	189.72	1.46
23	70181	2	7.17	101.61	111.28	220.05	9.00	0.03	0.68	130.92	2.61
24	70181	2	20.18	104.36	92.33	216.87	14.00	0.09	0.70	108.63	12.31
25	70181	2	1090.95	1198.59	1410.32	3699.86	716.00			447.42	

TEM 33 IS AN AUDIT CHECK OF DAILY TOTALS IN ITEM 25

33	70181	2	1090.88	1198.52	1410.25	3699.63	722.00			447.36	4146.99

TEMS 26,27,28 ARE TOTALS FOR 9AM TO 5PM, 5TO9, 9TO MIDNIGHT RESECTIVELY

26	70181	2	754.19	488.57	606.95	1849.72	532.00			323.20	2172.91
27	70181	2	213.33	337.97	402.91	954.21	98.00			78.65	1032.86
28	70181	2	31.20	348.29	364.86	744.35	28.00			16.37	760.73

MONTH-END SUMMARY

HR	DATE	DAY	CHAN-UNITS	CPU-UNITS	CORE-UNITS	TOTAL	JOBS	CHAN-HR	CPU-HR	KB-HR	IN/OUT-UNITS	REV

ITEM 29 CONTAINS 24-HR TOTALS FOR THE MONTH

ITEMS 30,31,32 CONTAIN MONTH-TOTALS OF ACTIVITY 9AM TO5PM,5 TO 9, 9 TO MIDNIGHT RESPECTIVELY

ITEM 34 REPORTS TOTALS OF: EACH HR WITH REV>200/NUMBER OF HRS WITH REVENUE >200

29	6	0	32879.46	28881.89	33853.38	95614.56	18458.00				13535.25	109149.80
30	6	0	14051.73	7672.00	10656.87	32380.73	10852.00				7261.98	39642.71
31	6	0	6900.35	6849.86	7881.96	21632.19	4162.00				2982.04	24614.23
32	6	0	4490.09	5932.11	6914.88	17337.07	2015.00				1812.87	19149.94
34	6	0	83.86	78.55	90.56	252.97	44.74				33.72	286.69

NEXT ITEM REPORTS NO OF HRS IN MONTH WITH REV >200
0.3010000E 03

A P P E N D I X 3

Program M O R E V

APPENDIX 3 -- Program MOREV.

FUNCTION.

To reduce hourly efficiency figures to a percentage cumulative frequency per decile base for plotting contours of equi-efficiency. That is, to pictorially report plant performance for any selected period.

DATA.

Hourly efficiency figures from program RESEVAL.

REFERENCE.

The logic of the program MOREV is described in section 3E of the text, pp 56 - 57.

```
 1

001          IDENTIFICATION DIVISION.
002          PROGRAM-ID. 'RESOURCE USE TABULATION'.
003          ENVIRONMENT DIVISION.
004          INPUT-OUTPUT SECTION.
005          FILE-CONTROL.
006              SELECT CARD-IN ASSIGN TO UR-2501-S-SYSIN.
007              SELECT PRINTER ASSIGN TO UR-1403-S-SYSC02.
008              SELECT CARD-OUT ASSIGN TO UR-1443-S-SYS005.
009              SELECT CAB      ASSIGN TO UT-2314-S-SORTIN.
010              SELECT ABCD     ASSIGN TO UT-2314-S-SORTOUT.
011              SELECT MASK      ASSIGN TO UT-2314-S-DUMM.
012              SELECT TAPE-OUT ASSIGN TO UT-2400-S-SYS003.
013          DATA DIVISION.
014          FILE SECTION.
015          FD  CARD-IN  LABEL RECORDS ARE OMITTED.
016         *    DATA RECORDS ARE MAST-CD, DOWN-CD,CORE-CD,QUEUE-CD.
017          01  SUB-CARD-IN.
018              02  ALIAS-HR                                    PIC X(2).
019              02  FILLER                                      PIC X(9).
020              02  ADJUSTMENTS                                 PIC X(68).
021              02  CD-TYPE                                     PIC X.
022          FD  PRINTER LABEL RECORDS ARE CMITTED.
023              01  SUB-PRINT.
024                  02  SUB-P-CC                               PIC X.
025                  02  SUB-PRINT-1                            PIC X(80).
026                  02  SUB-PRINT-2.
027                      03  FILLER                         PIC X(48).
028                      03 COMP-DAY                        PIC X(3).
029      FD  CARD-OUT LABEL RECORDS ARE OMITTED.
030      01  SUB-CARD-OUT                                       PIC X(80).
031      FD  CAB  LABEL RECORDS ARE STANDARD BLOCK CONTAINS
032          90  RECORDS.
033      01  SUB-CAB                                            PIC X(80).
034      FD  ABCD  RECORDING MODE IS F LABEL RECORDS ARE STANDARD
035          BLOCK CONTAINS 90 RECORDS.
036      01  SUB-ABCD                                           PIC X(80).
037      SD  MASK RECORDING MODE IS F.
038      01  SUB-MASK.
039          02  MHR-DAY-YR.
040              03  MHR                                    PIC 9(2).
041              03  MDAY                                   PIC 9(3).
042              03  MYR                                    PIC 9(2).
043          02  MWK-DAY                                    PIC 9.
044          02  MIO-REV                                    PIC 9(4).
045          02  MCPU-REV                                   PIC 9(4).
046          02  MCORE-REV                                  PIC 9(4).
047          02  MTOTAL-REV                                 PIC 9(4).
048          02  MJOBS                                      PIC 9(4).
049          02  MPC-CHAN                                   PIC 9(3)V99.
050          02  MPC-CPU                                    PIC 9(3)V99.
051          02  MKB-HR                                     PIC 9(3)V99.
052          02  MSYSEFF                                    PIC 9(3)V99.
053          02  MBCR                                       PIC X(32).
```

MOREV

```
 WORKING-STORAGE SECTION.
 77  DPERKBH    VALUE 0.85                          PIC 99V999.
 77  I        VALUE 1                                PIC 9(4).
 77  ADD-JOB VALUE ZERO                             PIC 9(4)V.
 77  IO-REV       VALUE ZERO                        PIC 9(5)V99.
 77  CPU-REV      VALUE ZERO                        PIC 9(5)V99.
 77  CORE-REV     VALUE ZERO                        PIC 9(5)V99.
 77  TOTAL-REV    VALUE ZERO                        PIC 9(5)V99.
 77  JOBS         VALUE ZERO                        PIC 9(5)V99.
 77  PC-CHAN      VALUE ZERO                        PIC 9(5)V99.
 77  PC-CPU       VALUE ZERO                        PIC 9(5)V99.
 77  KB-HR        VALUE ZERO                        PIC 9(5)V99.
 77  WK-DAY      VALUE ZERO                       PIC 9(3).
 77  AV-CORE  VALUE ZERO                           PIC 9(3).
 77  MAX-CORE VALUE 270                            PIC 9(3).
 77  MAX-IO VALUE 216                              PIC 9(3)V99.
 77  MAX-CP VALUE 150                              PIC 9(3)V99.
 77  PC-USE  VALUE ZERO                            PIC 9(3)V99.
 77  SWS        VALUE ZERO                         PIC 9.
 77  SWT    VALUE 1                                PIC 9.
 77  SWP      VALUE 1                              PIC 9.
*  NOTE THAT 2 IN COLUMN 80 OF DATA CARD WILL TRIGGER END OF SUB-SET
 77  SWNS    VALUE 0                               PIC 9.
*  NOTE THAT IF SWITCH-NO-SORT = 0 THE SORT WILL BE BY-PASSED
 77  USER-C     VALUE 300                          PIC 9(3).
 77  RATIO      VALUE 0                            PIC 9V9(4).
 77  NOPWM    VALUE 0                              PIC 9.
 77  MAXI     VALUE 0                              PIC 9(4).
 77  SUMEFF   VALUE 0                              PIC 9(3)V99.
 77  M            VALUE ZERO                       PIC 9(3).
 77  N            VALUE ZERO                       PIC 9(3).
 77  K            VALUE ZERO                       PIC 9(3).
 77  CNTR    VALUE 0                               PIC 999.
 77  M1                                            PIC 9(3).
 77  M2                                            PIC 9(3).
 77  M3                                            PIC 9(3).
 77  M4                                            PIC 9(3).
 77  KK       VALUE ZERO                           PIC 9(3).
 77  JJ       VALUE ZERO                           PIC 9(3).
 77  L        VALUE ZERO                           PIC 9(3).
 77  ARG      VALUE ZERO                           PIC 9(8).
 77  S        VALUE ZERO                           PIC 9(3).
 01  MASTABC.
     02  ABC    OCCURS 75 TIMES                    PIC X(80).
 01  CD1.
     02  YR-DAY-HR.
         03  FILLER                                PIC X(2).
         03  HR                                    PIC 9(2).
         03  FILLER                                PIC X(2).
         03  YR                                    PIC X(2).
         03  YR-DAY                                PIC 9(3).
     02  IOF                                       PIC 9(5)V.
     02  IOS                                       PIC V99.
     02  CPUF                                      PIC 9(5)V.
     02  CPUS                                      PIC V99.
     02  KBF                                       PIC 9(5)V.
     02  KBS                                       PIC V99.
```

```
     02  TOTF                                    PIC 9(5)V.
     02  TOTS                                    PIC V99.
     02  JOF                                     PIC 9(5)V.
     02  JOS                                     PIC V99.
     02  PCCF                                    PIC 9(5)V.
     02  PCCS                                    PIC V99.
     02  PCCPUF                                  PIC 9(5)V.
     02  PCCPUS                                  PIC V99.
     02  KBHF                                    PIC 9(5)V.
     02  KBHS                                    PIC V99.
     02  RLP                                     PIC 9(5)V99.
* RLP IS REVENUE  FROM CARDS IN  + OUT AND PRINTING
     02 FILLER                                  PIC X(4).
     02  C-T                                     PIC X.
     02  WDAY                                    PIC 9.
*
 01  TRANSACTION.
     02  TIME.
             03  T-HR                            PIC 9(2).
             03  FILLER                          PIC X.
             03  T-YR                            PIC X(2).
             03  T-YR-DAY                        PIC 9(3).
     02  DOWN-TIME                               PIC 9(2)V.
     02  USER-CORE                               PIC 9(3)V.
     02  ADD-IO-R                                PIC 9(3)V.
     02  ADD-CPU-R                               PIC 9(3)V.
     02  ADD-CORE-R                              PIC 9(3)V.
     02  ADD-JOBS                                PIC 9(3)V.
     02  FILLER                                  PIC X(55).
 01  SUB-BUILD-CARD.
     02  HR-DAY-YR.
         03  OHR                                 PIC 9(2).
         03  ODAY                                PIC 9(3).
         03  OYR                                 PIC 9(2).
     02  OWK-DAY                                 PIC 9.
     02  OIO-REV                                 PIC 9(4).
     02  OCPU-REV                                PIC 9(4).
     02  OCORE-REV                               PIC 9(4).
     02  OTOTAL-REV                              PIC 9(4).
     02  OJOBS                                   PIC 9(4).
     02  OPC-CHAN                                PIC 9(3)V99.
     02  OPC-CPU                                 PIC 9(3)V99.
     02  OKB-HR                                  PIC 9(3)V99.
     02  OSYSEFF                                 PIC 9(3)V99.
     02  PEFF                                    PIC 9(3)V99.
     02  SBCR                                    PIC X(32).


 01  EDIT-1.
     02  FILLER      VALUE SPACE                 PIC X.
     02  H-D-Y.
         03  H                                   PIC 9(2).
         03  FILLER     VALUE SPACE              PIC X.
         03  Y-D                                 PIC X(3).
         03  FILLER     VALUE SPACE              PIC X.
```

```
           03  Y                                          PIC 9(2).
       02  FILLER          VALUE SPACES                   PIC X(2).
       02  W-D                                            PIC 9.
       02  FILLER          VALUE SPACES                   PIC X(2).
       02  I-R                                            PIC ZZZ9.
       02  FILLER          VALUE SPACES                   PIC X(2).
       02  C-R                                            PIC ZZZ9.
       02  FILLER          VALUE SPACES                   PIC X(2).
       02  KB-R                                           PIC ZZZ9.
       02  FILLER          VALUE SPACES                   PIC X(4).
       02  T-R                                            PIC ZZZ9.
       02  FILLER          VALUE SPACES                   PIC X(4).
       02  J                                              PIC ZZZ9.
       02  FILLER          VALUE SPACES                   PIC X(4).
       02  PC-CH                                          PIC ZZ9.9.
       02  FILLER          VALUE SPACES                   PIC X(2).
       02  PC-CP                                          PIC ZZ9.9.
       02  FILLER          VALUE SPACES                   PIC X(3).
       02  PC-KB                                          PIC ZZ9.9.
       02  FILLER          VALUE SPACES                   PIC X(5).
       02  EFF                                            PIC ZZ9.9.
       02  FILLER          VALUE SPACES                   PIC X(5).
       02  PER                                            PIC 9.
       02  PIEFF                                          PIC ZZ9.9.
       02 FILLER     VALUE SPACES                        PIC X(2).
       02  1CNT                                           PIC 9(3).
       02  FILLER          VALUE SPACES                   PIC X(2).
       02  2CNT                                           PIC 9(3).
       02  FILLER          VALUE SPACES                   PIC X(30).
01  EDIT-LIST.
       02  FILLER   VALUE SPACE                           PIC X.
       02  L-HR                                           PIC Z9.
       02  L-SUB-L-H          VALUE SPACES                PIC X(129).
01  LIST-HISTO.
       02  SUB-L-H OCCURS 44 TIMES                        PIC ZZZ.
01  CHAN-HISTO.
       02  SUB1-C-H OCCURS 24 TIMES.
           03  SUB2-C-H OCCURS 10 TIMES                   PIC 9(3).
01  CPU-HISTO.
       02  SUB1-CPU-H OCCURS 24 TIMES.
           03  SUB2-CPU-H OCCURS 10 TIMES                 PIC 9(3).
01  CORE-HISTO.
       02 SUB1-CORE-H OCCURS 24 TIMES.
           03  SUB2-CORE-H OCCURS 10 TIMES                PIC 9(3).
01  PC-HISTO.
       02  SUB1-PC-H OCCURS 24 TIMES.
           03  SUB2-PC-H OCCURS 10 TIMES                  PIC 9(3).
01  ID-HEAD.
       02  ID-H                                           PIC X(79).
       02  FILLER                                         PIC X.
01  TENARG.
       02  SUBTARG  OCCURS 10 TIMES                       PIC 9(3).
01  PLOT.
       02  SUB1PLOT  OCCURS 96 TIMES.
           03  SUB2PLOT  OCCURS  6 TIMES                  PIC 9(3).
01  MASB.
       02  B     OCCURS  20 TIMES                         PIC 9(7).
```

```
01  CDIM.
    02  CDIM2   OCCURS 24 TIMES                PIC 9(3).


PROCEDURE DIVISION.
    OPEN INPUT CARD-IN  OUTPUT PRINTER.
    OPEN OUTPUT CAB, CARD-OUT.
SUB-RD.
    PERFORM B7 VARYING M FROM 1 BY 1 UNTIL M > 40.
    PERFORM B3 VARYING M FROM 1 BY 1 UNTIL M > 24.
    GO TO B5.
B3.
    PERFORM B4 VARYING N FROM 1 BY 1 UNTIL N > 10.
B4.
    MOVE 0 TO SUB2-CORE-H (M,N).
    MOVE 0 TO SUB2-C-H (M,N).
    MOVE 0 TO SUB2-PC-H (M,N).
    MOVE 0 TO SUB2-CPU-H (M,N).
B7.
    MOVE ZERO TO SUB-L-H (M).
B5.
*   COMPUTE DPERKBH = DPERKBH / 100.
*   EXHIBIT NAMED MAX-CORE, MAX-IO, MAX-CP, DPERKBH.
*   READY TRACE.


RD1.
        READ CARD-IN INTO CD1 AT END GO TO SORT-LIST.
*   EXHIBIT NAMED SUB-CARD-IN.
    ADD 1 TO CNTR.
    IF CNTR > 10 RESET TRACE.
*   NOTE THAT SWT SHOULD = 1 IF UPDATE TRANSACTIONS ARE INCLUDED
*        BUT SWT SHOULD = 0 FOR BY-PASS.
    IF SWT = 0 GO TO BY-PASS.
    IF C-T = 3 MOVE SUB-CARD-IN TO ID-HEAD GO TO RD1.
    IF C-T NOT = 1 GO TO HR-CARD.
    MOVE SUB-CARD-IN TO ABC (I).  ADD 1 TO I.  GO TO RD1.
HR-CARD.
    EXAMINE SUB-CARD-IN REPLACING ALL SPACES BY ZERO.
    MOVE SUB-CARD-IN  TO CD1.
*   IF HR = 32 MOVE 0 TO SWP  GO TO SORT-LIST.
    IF C-T = 2 MOVE 0 TO SWP    GO TO SORT-LIST.
*   IF ALIAS-HR = SPACES MOVE 0 TO SWP GO TO SORT-LIST.
    IF HR > 24
    GO TO RD1.
    IF TOTF = 0 GO TO RD1.
    IF I = 1 GO TO BY-PASS.  IF NOPWM = 0 GO TO  INITI.


M-UNM.
    MOVE ABC (I) TO TRANSACTION.
```

```
*    EXHIBIT NAMED YR-DAY-HR, TIME, I, ABC (I).
     IF YR-DAY-HR > TIME     GO TO BY-PASS.
     IF YR-DAY-HR = TIME     GO TO UP-CATE.
     ADD 1 TO I.  IF I > MAXI GO TO BY-PASS. GO TO M-UNM.
 INITI.
     SUBTRACT 1 FROM I. MOVE I TO MAXI. MOVE 1 TO I.  GO TO M-UNM.


 UP-DATE.
     PERFORM BY-PASS THRU BY-P.  MOVE ABC (I) TO TRANSACTION.
*    PERFORM UP-D.  GO TO BY-P2.


 UP-D.
          EXAMINE TRANSACTION REPLACING ALL SPACE BY ZERO.
          COMPUTE ADD-JOB = ADD-IO-R +ADD-CPU-R + ADD-CORE-R.
          IF ADD-JOB = ZERO GO TO STEP2.
     IF ADD-IO-R NOT = ZERO ADD ADD-IO-R TO IO-REV.
     IF ADD-CPU-R NOT = ZERO ADD ADD-CPU-R TO CPU-REV.
     IF ADD-CORE-R NOT = ZERO ADD ADD-CORE-R TO CORE-REV.
     ADD ADD-JOB TO TOTAL-REV.
     IF ADD-JOBS NOT = 0 ADD ADD-JOBS TO JOBS.


 STEP2.
     IF DOWN-TIME NOT = ZERO PERFORM STEP3.
     IF ADD-CORE-R NOT = ZERO MOVE ADD-CORE-R TO AV-CORE, ELSE
          MOVE USER-C TO AV-CORE.
     GO TO BY-P2.
 STEP3.
     MOVE SUB-CARD-IN TO SUB-PRINT-1 MOVE SPACES TO SUB-PRINT-2.
     WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
     COMPUTE RATIO = 60  / DOWN-TIME.
*    MULTIPLY RATIO BY IO-REV.
*    MULTIPLY RATIO BY CPU-REV.
*    MULTIPLY RATIO BY CORE-REV.
*    MULTIPLY RATIO BY TOTAL-REV.
*    MULTIPLY RATIO BY JOBS.
     MULTIPLY RATIO BY PC-CHAN.
     MULTIPLY RATIO BY PC-CPU.
     MULTIPLY RATIO BY KB-HR.



 BY-PASS.
     EXAMINE SUB-CARD-IN REPLACING ALL SPACES BY ZERO.
```

```
           MOVE SUB-CARD-IN  TO CD1.
           COMPUTE IO-REV = IOF + IOS.
           COMPUTE CPU-REV = CPUF + CPUS.
           COMPUTE CORE-REV = KBF + KBS.
           COMPUTE TOTAL-REV = TOTF + TOTS.
           COMPUTE JOBS = JOF + JOS.
           COMPUTE PC-CHAN = PCCF + PCCS.
           COMPUTE PC-CPU = PCCPUF + PCCPUS.
           COMPUTE KB-HR = KBHF + KBHS.
       *   COMPUTE WK-DAY = YR-DAY + 3.
           IF YR-DAY > 166 MOVE 170 TO AV-CORE, ELSE
           MOVE 230 TO AV-CORE.
           COMPUTE PC-USE = ((IO-REV + CPU-REV + CORE-REV * DPERKBH)
            / (MAX-IO + MAX-CP + AV-CORE * DPERKBH))  * 100.
           COMPUTE SUMEFF = (PC-CHAN * PC-CPU * KB-HR / AV-CORE) * 100.
       *   EXHIBIT NAMED PC-CHAN, PC-CPU, KB-HR, AV-CORE, SUMEFF.
           COMPUTE M = PC-CHAN * 10 + 1.
       *   EXHIBIT NAMED M, PC-CHAN.
       *   EXHIBIT NAMED SUB-CARD-IN.
       *   EXHIBIT NAMED IOF, IOS, CPUF, CPUS, KBF, KBS, TOTF, TOTS,
       *   JOF, JOS, PCCF, PCCS, PCCPUF, PCCPUS, KBHF, KBHS, YR-DAY
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                                ADD 1 TO SUB2-C-H (HR,M).
           COMPUTE M = PC-CPU * 10 + 1.
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                               ADD 1 TO SUB2-CPU-H (HR,M).
           COMPUTE M = (KB-HR / AV-CORE) * 10  + 1.
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                                   ADD 1 TC SUB2-CORE-H (HR,M).
           COMPUTE M = PC-USE / 10 + 1.
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                                  ADD 1 TO SUB2-PC-H (HR,M).
       BY-P.
           MOVE SPACES TO SUB-PRINT-2.
       *   COMPUTE WK-DAY = WK-DAY - 7.
       *   IF WK-DAY > 7 GO TO BY-P.
       *   MOVE WK-DAY TO WDAY.
           MOVE WDAY  TO COMP-DAY
       *   MOVE WK-DAY TO COMP-DAY.
           MOVE CD1 TO SUB-PRINT-1.


       BY-P2.
********** A TEST LIST OF INPUT DATA  **********************
      *    WRITE SUB-PRINT AFTER POSITIONING 1 LINES.


       BUILD-CARD.
           MOVE HR TO OHR                  MOVE YR-DAY TO ODAY
           MOVE YR TO OYR                  MOVE  WDAY TO OWK-DAY
           MOVE IO-REV TO OIO-REV          MOVE CPU-REV TO OCPU-REV
           MOVE CORE-REV TO OCORE-REV      MOVE TOTAL-REV TO  OTOTAL-REV
           MOVE JOBS TO OJOBS
```

```
           COMPUTE OPC-CHAN = PC-CHAN * 100.
           MULTIPLY 100 BY PC-CPU GIVING OPC-CPU.
           MOVE KB-HR TO OKB-HR.
           MOVE PC-USE TO OSYSEFF.
           MOVE SPACES TO SBCR.
           MOVE SUMEFF TO PEFF.



      *WRITE SUB-BUILD CARD ONTO CARD, PRINTER, + TAPE
      *AS + IF REQUIRED.
           MOVE SUB-BUILD-CARD TO SUB-PRINT-1.
           WRITE SUB-PRINT AFTER POSITIONING 1 LINES.
           WRITE SUB-CAB FROM SUB-BUILD-CARD
           GO TO RD1.
       HEAD1.
           MOVE LOW-VALUES TO SUB-PRINT.
           WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
           MOVE SPACES TO SUB-PRINT-1.
           WRITE SUB-PRINT AFTER POSITIONING 0 LINES.
           MOVE    '* PERIOD    *        REVENUE          *     NUMBER * PE
      -    'RCENTAGE USE OF EQUIPMENT  *'
           TO SUB-PRINT-1.
           WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
           MOVE SPACES TO SUB-PRINT-1.
           MOVE    'HR DAY YR      CHAN  CPU  CORE   TOTAL  OF JOBS  ( CH
      -    'ANS (CPU  AV.CORE NET EFF *'
           TO SUB-PRINT-1  WRITE SUB-PRINT AFTER POSITIONING 1 LINES.
           MOVE LOW-VALUES TO SUB-PRINT.
           WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
      SORT-LIST.
           IF SWNS = 0 GO TO REP-HISTO.
           DISPLAY 'ENTERING SORT-LIST'.
           PERFORM HEAD1.
           CLOSE CAB.
           SORT MASK ON  DESCENDING KEY MSYSEFF USING CAB GIVING ABCD.
           OPEN INPUT ABCD.
           MOVE LOW-VALUES TO EDIT-1.
           MOVE ZERO TO 1CNT.
           MOVE ZERO TO 2CNT.
      RD2.
           READ ABCD INTO SUB-BUILD-CARD, AT END GO TO EOF.
               MOVE  OHR          TO  H
               MOVE  ODAY         TO  Y-D
               MOVE  OYR          TO  Y
               MOVE  OWK-DAY      TO  W-D
               MOVE  OIO-REV      TO  I-R
               MOVE  OCPU-REV     TO  C-R
               MOVE  OCORE-REV    TO  KB-R
               MOVE  OTOTAL-REV   TO  T-R
               MOVE  OJOBS        TO  J
               MOVE  OPC-CHAN     TO  PC-CH
               MOVE  OPC-CPU          TO  PC-CP
               MOVE  OKB-HR           TO  PC-KB
               MOVE  OSYSEFF          TO  EFF.
           MOVE PEFF TO PIEFF.
```

```
           IF ODAY > 166 ADD 1 TO 2CNT, ELSE ADD 1 TO 1CNT.
           MOVE LOW-VALUES TO SUB-PRINT.
           WRITE SUB-PRINT FROM EDIT-1 AFTER POSITIONING 1 LINES.
           GO TO RD2.
       B9.
           MOVE SUB2-C-H (I, 1) TO M1.
           MOVE SUB2-C-H (I, 10) TO M1.
           MOVE SUB2-CPU-H (I, 1) TO M2.
           MOVE SUB2-CPU-H (I, 10) TO M2.
           MOVE SUB2-CORE-H (I, 1) TO M3.
           MOVE SUB2-CORE-H (I, 10) TO M3.
           MOVE SUB2-PC-H (I, 1) TO M4.
           MOVE SUB2-PC-H (I, 10) TO M4.
      *    PERFORM B8 VARYING K FROM 1 BY 1 UNTIL K > 9.
           PERFORM B8 VARYING K FROM 10 BY -1 UNTIL K < 2.
       B8.
           COMPUTE N = K + 1
           COMPUTE N = K - 1.
           ADD SUB2-C-H (I, N) TO M1.
           IF SUB2-C-H (I, N) > 0 MOVE M1 TO SUB2-C-H (I, N).
           ADD SUB2-CPU-H (I, N) TO M2.
           IF SUB2-CPU-H (I, N) > 0 MOVE M2 TO SUB2-CPU-H (I, N).
           ADD SUB2-CORE-H (I, N) TO M3.
           IF SUB2-CORE-H (I, N)   > 0 MOVE M3 TO SUB2-CORE-H (I, N).
           ADD SUB2-PC-H (I, N) TO M4.
           IF SUB2-PC-H (I, N) > 0 MOVE M4 TO SUB2-PC-H (I, N).
       B10.
           IF SUB2-C-H (I, N) > 0 MOVE SUB2-C-H (I, N) TO M.
       B13.
           PERFORM B10 VARYING N FROM 10 BY -1 UNTIL N < 1.
           PERFORM B12 VARYING K FROM 1 BY 1 UNTIL K > 10.
       B12.
           COMPUTE SUB2-C-H (I, K) = SUB2-C-H (I, K) * 100 / M.
           COMPUTE SUB2-CPU-H (I, K) = SUB2-CPU-H (I, K) * 100 / M.
           COMPUTE SUB2-CORE-H (I, K) = SUB2-CORE-H (I, K) * 100 / M.
           COMPUTE SUB2-PC-H (I, K) = SUB2-PC-H (I, K) * 100 / M.
       EOF.
           CLOSE ABCD.
       REP-HISTO.
           MOVE LOW-VALUES TO SUB-PRINT.
           WRITE SUB-PRINT AFTER POSITIONING 0 LINES.
       B18.
           PERFORM B20.
           PERFORM B15.
       B19.
           WRITE   SUB-PRINT AFTER POSITIONING 2 LINES.
      *    WRITE   SUB-PRINT AFTER POSITIONING + LINES.
           MOVE                              '    PERCENT USAGE OF CHANNELS
      -                                      '      PERCENT USAGE OF CPU
      -                                      '      PERCENT USAGE OF CORE
      -                                    '  SYSTEM EFFICIENCY'
           TO  SUB-PRINT.
           WRITE   SUB-PRINT AFTER POSITIONING 2 LINES.
           MOVE ' HR  0 10 20 30 40 50 60 70 80 90 100   10 20 30 40 50
      -    '60 70 80 90 100   10 20 30 40 50 60 70 80 90 100   10 20 30
      -    '40 50'  TO SUB-PRINT.
           MOVE '100' TO COMP-DAY.
```

```
      WRITE   SUB-PRINT AFTER POSITIONING 2 LINES.
      PERFORM B1 VARYING M FROM 1 BY 1 UNTIL M > 24.
 B14.
      GO TO B6.
 B20.
      MOVE ID-H TO SUB-PRINT WRITE SUB-PRINT AFTER POSITIONING 2
      LINES.
 B15.
      MOVE                 '                                        F
     -                     'R E Q U E N C Y   ( D A Y S  /  D E C I L E
     -                     '  /  H R ) .        '  TO SUB-PRINT.
 B16.
      MOVE                 '                          C U M L A T I V E    F
     -                     'R E Q U E N C Y   ( D A Y S  /  D E C I L E
     -                     '  /  H R ) .        '  TO SUB-PRINT.
 B17.
      MOVE                 '                                              P E R
     -                      'C E N T A G E   O F  D A Y S  /  D E C I L E
     -                     '  /  H R  .         '  TO SUB-PRINT.
 B1.
      PERFORM B2 VARYING N FROM 1 BY 1 UNTIL N > 10.
      MOVE LIST-HISTO TO L-SUB-L-H.
      MOVE M TO L-HR.
      WRITE SUB-PRINT FROM EDIT-LIST AFTER POSITIONING 2 LINES.
*     WRITE SUB-PRINT FROM LIST-HISTO AFTER POSITIONING 2 LINES.
 B2.
      COMPUTE K = N + 1.
      MOVE SUB2-C-H (M,N) TO SUB-L-H (K).
      COMPUTE K = N + 12
      MOVE SUB2-CPU-H (M,N) TO SUB-L-H (K).
      COMPUTE K = N + 23
      MOVE SUB2-CORE-H (M,N) TO SUB-L-H (K).
      COMPUTE K = N + 34
      MOVE SUB2-PC-H (M,N) TO SUB-L-H (K).
 B6.
      PERFORM B9 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM REP-HISTO.
      PERFORM B20.
      PERFORM B16.  PERFORM B19.
 B11.
*     PERFORM B10 VARYING N FROM 1 BY 1 UNTIL N > 10.
      PERFORM B10  VARYING N FROM 10 BY -1 UNTIL N < 1.
      PERFORM B13 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM  REP-HISTO.
      PERFORM B20.
      PERFORM B17.  PERFORM B19.
      PERFORM TRANS VARYING KK FROM 1 BY 1 UNTIL  KK > 24.
      PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
      PERFORM TRANS2 VARYING KK FROM 25 BY 1 UNTIL KK > 48.
      MOVE 24 TO JJ.
      PERFORM P13 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
      PERFORM TRANS3 VARYING KK FROM 49 BY 1 UNTIL KK > 72.
      MOVE 48 TO JJ.
      PERFORM P13 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
      PERFORM TRANS4 VARYING KK FROM 73 BY 1 UNTIL KK > 96.
```

```
    MOVE 72 TO JJ.
    PERFORM P13 VARYING I FROM 1 BY 1 UNTIL I > 24.
    PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
    GO TO BCH.
P13.
    ADD 1 TO JJ.  MOVE SUB1PLOT (JJ) TO SUB1PLOT (I).
TRANS.
    COMPUTE  L = KK
    MOVE SUB1-C-H (L) TO TENARG.
    MOVE 0 TO JJ.
    PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
    EXHIBIT NAMED MASB .
    PERFORM INT THRU P8     VARYING S FROM 1 BY 1 UNTIL S > 6.
    EXHIBIT NAMED SUB1PLOT (KK).
TRANS2.
    COMPUTE  L = KK - 24.
    MOVE SUB1-CPU-H (L) TO TENARG.
    MOVE  0 TO JJ.
    PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
    EXHIBIT NAMED MASB .
    PERFORM INT THRU P8     VARYING S FROM 1 BY 1 UNTIL S > 6.
    EXHIBIT NAMED SUB1PLOT (KK).
TRANS3.
    COMPUTE L = KK - 48.
    MOVE SUB1-CORE-H (L) TO TENARG.
    MOVE 0 TO JJ.
    PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
    PERFORM INT THRU P8     VARYING S FROM 1 BY 1 UNTIL S > 6.
                             EXHIBIT NAMED SUB1PLOT (KK).
TRANS4.
    COMPUTE L = KK - 72.
    MOVE SUB1-PC-H (L) TO TENARG.
    MOVE 0 TO JJ.
    PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
    PERFORM INT THRU P8     VARYING S FROM 1 BY 1 UNTIL S > 6.
                             EXHIBIT NAMED SUB1PLOT (KK).
TENTOSIX.
    IF SUBTARG (I) = 0  GO TO P7.
    ADD 2 TO JJ.
    COMPUTE M = JJ - 1.
    MOVE SUBTARG (I) TO B(M)
    COMPUTE B (JJ) = 10 * I.
    MOVE JJ TO K.
P7.
    EXIT.
INT.
    COMPUTE ARG = (6 - S) * 20
    PERFORM INT2 THRU P9   VARYING JJ FROM 2 BY 2 UNTIL JJ > K.
INT2.
    COMPUTE M = JJ - 1.
*   EXHIBIT NAMED K, S, ARG, JJ, M, B (M).
    IF B (M) < ARG GO TO P9.
    IF B (M) = ARG GO TO P10.
    IF JJ = K GO TO P12.
    COMPUTE M1 = JJ + 1.
    COMPUTE M2 = JJ + 2.
    IF B (M1) > ARG GO TO P9.
```

```
      COMPUTE SUB2PLOT (KK, S) = (B (M) - ARG)
       * (B (M2) - B (JJ))/
      (B (M)      -   B (M1))    + B (JJ).
*     EXHIBIT NAMED M, M1, M2, JJ, B (M), B (M1), B (M2), B (JJ),
*      SUB2PLOT (KK, S).
      GO TO P11.
 P12.
      MOVE B (K) TO SUB2PLOT (KK,S)
      GO TO P11.
 P10.
      MOVE B (JJ) TO SUB2PLOT (KK,S).
 P11.
      GO TO P8.
 P9.
      EXIT.
 P8.
      EXIT.
 XSWING.
      MOVE SUB2PLOT (I, JJ) TO CDIM2 (I).
 YSWING.
      PERFORM  XSWING VARYING  I  FROM 1 BY 1 UNTIL I > 24.
                                                EXHIBIT NAMED CDIM.
      MOVE SPACES  TO SUB-CARD-OUT.
      WRITE SUB-CARD-OUT FROM CDIM.
 BCH.
      IF SWP = 0 CLOSE CAB OPEN OUTPUT CAB MOVE 1 TO SWP
      MOVE LOW-VALUES TO SUB-PRINT  WRITE SUB-PRINT AFTER
      POSITIONING 0 LINES
      MOVE 0 TO I  GO TO SUB-RD.
      CLOSE  CARD-IN, PRINTER.
      CLOSE CARD-OUT, ABCD.
      STOP RUN.
```

702CC2600340034C0960015012000220004101017700006 3
702C19901120139045200240920007400164350844006581
702C184006800810334002308500045000953606300021 45
702CC8900290050016900100410001900059030317000270
7020101005500680225009504600036000805604200007 84
702C09C0024004901640096041000160005821030800022 4
702CC790046006601920085036000300007834035600049 7
702C0066007100720211007303000048000856403920007 25
702CC8400360062018300970390002400073330340000403
702CC8400390073019701130390026C0086330365000514
702C056006800760201009702600045000895603720006 16
702CC61C092009902540084028C0061001171604680011 77
702CC55009600790231006802500064C009397042900884
702C062012300920278006502900082001087405180015 21
702CC7C012101220314005903200080001436205790021 62
702CC18000600110036001900800004000133300670000 02
702CC8E016201450397005604100108001716507360044 70
7020116016501580440089054001100018641081600651 3
7030057009700920248004002600065001C88504580010 82
703CC120139007602290013005000930008984042600024 5
703C004015200550212001100200101000647703990000 76
703C00601470051020600200030009800060820388000105
703CC07014900550212002000300099000654203990001 14
703C024014900640238001901100099000764604480004 89
703C008015801080275001300300105001275705080002 36
703CC0800100012003100130040006000143500570000 02
703C0800094008302570039037000620009796048000132 1
703C0092005100610205007104200034000717803840006 02
7030106004000710218006604900026000843304060006 31
703011800670105029100950540004500124230540001775
703010500640100027000830480004300117920500001431
703CC810038006901890091037000250CC81390351000442
703CC590021003201130086027000140003818021200008 4
703CC8E004300690202012604100029000822103760005 74
703007500480064C1880109034000320CC760003490C486
703CC7000530064018801000320003500076350350000503
703C0800105010602920071037000700012515054100190 6
703C092009300940280006004200062001110905210017 01
703C089011801200328005904100078001421106080026 73
703012601360156041900380580C090001837207750056 41
703CC760079010902650057035000530012906048800140 8
703CC5100410061015400810230002700072480284000264
704003300410038011300210150002700044730210000106
704CC7501370077028900180340009100090650544001649
704C0850143010203310017039000950012073061800263 1
704002901170077022400130130CC78C0C9101041600054 2
704001400010004002000120060000000005580039000000
704C0020000000000300020010000000000910007000000
704C0010000000C0000100020000000CCCCCC0100002000000
704011300300055019900990520002000064720374000395
704CC970069C0940260011004500C46001106704830013 47
704011300990113032600970520006600133550600500269 6
704C06001250102028800420270008300120680534001590
704010900380065021301020500002500076690398000563
704CC810029004501560096037000190005338029200022 0
704CC810059007902200102037000390009936804080007 95
704011400790082C27601000520005300C96750516001568
704C0101005900830244007504600039000986204550010 40
704CC650148016403780070030000980019399069300335 4
704C035007600730185005101600051000863403420004 14
704C1030064011402830067048000430013507052100163 9
704CC9C005800950244004104100039001121904500010 55
704CC620017004101200034028000110004854022400008 7
7050C37003800480124001601700025000566102290001 41

```
705C0460025004201140012021000170004947021 1000103
705C0280021003BC087C0C701300014000447601610000047
705CC8200790130029200C80380005300153410534001817
705CC0500C700110024000300200004000130400440000000
705CC4100090017006800300190000600020590129000013
705CC9900700088025800870450004600104060479001267
7050112005100680232008705100034000808204350008240
7050140003600770254011806400024000916004750CC827
705CC6300890086024000690290005900101780444001024
705CC8700490077021400750400C0320CC914203960006880
705CC1300060012003200210060000400014200059000002
7050023001200200057002801000C8000241001C5000011
705CC4200100020C072005001900006000235301360000015
705CC7500530079020BC07003400035000934103840006530
7050066011801030288003703000079001216705340016960
705CC9701260110033400760450C840012981062200288 6
7050018000400110034002200800030000137300640000001
7050007000100030013001400300001000044000240C0000
705CC5500750057018700100250005000067060350000493
706CC0100000000002000100000C0C000C6100040C0000
7060014002900280071000600600019000330101310000022
706CC0500020004012000800200001C000053500230000000
7060C3000100053009400901400C6000627601680000031
7060C7500470187031000030350003100221020553001410
706CC0600020013002300060030C0010001555004100000 0
706CC4100110075012800040190000700088860229000069
7060234014505740954000910800096006756517000412 6
707CC1200230032006800080050001500038260124000016
7070008000000020011000200300000000026600210000000
707CC0600010010001700010020000000012070031000000
707C0030003004001100090010000200C586C021000000
707CC1200070010002900080050000400012240055000001
707CC0CC0C0C0000000100070000CCCC0000190002000000
707CC4400300051012600180200C020CCC06C680232000142
7070060004501910297001902800030002252505270011 13
701CC4C0023006301270012018000150007434023100011 8
7010C0100000000002000300000C0CCCCC0940004000000
7010035001300250074002801600C900029490138000024
7010063004000670171007202900027000793703160003 65
7010C6800220035012600800310001500041270236000112
7010C7800520055018600930360003500065070348000482
701CC7200610081021500950330004100095510398000760
7010C6700210047013500780310001400055850252000142
701CC6400280051014400730290001900060710268000196
7010C9000330063018601030410002200074150347000393
7010140008401250350011906400056001472206480031 03
7010119010001230343007405500C66001458306360031 13
7010106012601050338004304900084001245906310030 16
701CC4001240102026700530180008300120210493001056
7010C8011601110308005703700077001316505720022 06
7010C8101180070027000320370007800C828705C8001406
701CC5400830062020000230250005500073310373000592
7010144020201920538006C066001340022651C998011783
7020157001600480222001907200011000573304210002 67
702CC4500100026C0820018021000060003134015300002 3
702CC2400000002002700040110CCCC00C0275C053CC0000
7020C1900070012003900150090000400014790073000003
7020104003500570196C0820480002300C67540368000438
7020C9800360071020601050450002400C83810383000532
702CC7200590081021300800330003900096160394000727
7020078007500970251C0910360005000114390464001211
702CC9300670086024701330430004500 1C17C0459001157
7020101006801010271010704600C4500118920501001448
702CC8400430062018901000380002900073110353000473
```

702CC52014400970294006602400C960011451054700155 1
702CC82016601680417007203800110001984607670004879 1
702CC55014300910290005102500095001074805420015 01 1
702CC46016300820292C0360210C1C9CCC96860548001304 1
702CC89010000810271005604100067000960005080015 51 1
703C139004800890278C0500640003200 1C559C5180012 72 1
703015401600111042600320710010700 13C82C802005846 1
703CC3501590089028400120160010600105840531001055 1
703C012006300460122000900500042000551402260C0068 1
703000800060014002800090030000400016750052000001 1
703CC0100000000000100040000C0C0000002600030000000 1
703C037009500640198002601700063000761503680004 79 1
703C093006200920248C0770430004100109170459001132 1
703CC4100560079017700560190003700C9387032500388 1
703CC8000560067020400990370003700079040380C00636 1
703C05100550068017500610230003700C8C720323000404 1
703CC8100290055016600900370001900C64920310000268 1
703C0640021004301300062030000140005167024300012 7 1
703CC8300480073020601280380003200086400381000618 1
703CC8200400055017800860370002600065750332000372 1
703CC490079008702150055022000530010245039700070 2 1
703010701120108032800500490007400127850611002726 1
703CC4501420076026500410210009500090340496001060 1
703CC8000460062018900500370003100074020353000499 1
703002200500063013600390100003300C7471C249000145 1
703C01600110012004000200070000700014440075000004 1
703C03200340043011000330150002200051420204000099 1
704000100000000000200010000C0C0000003200040C0000 1
704CC0100000000000100010000000000000220002000000 1
7040C040000000C0C0050002002000000000105001100000 0
704010400320052018900850480002100061350354000363 1
704013900490080026901110640003300094910505001179 1
704012500660086027900740580004400101900521001529 1
704013300940121034800940610006300142530647003222 1
704CC8200170036013600750380001100042750256000105 1
704CC7700170036013100770360001100042430246000098 1
704010500500064022001590490003300076010412000722 1
704012200470088025801010560003100103740479001059 1
704010301250141037000820480008300166770684003908 1
704CC3700840067018900450170005600079960351000447 1
704CC5801260144032900580270008400170290602002271 1
704007101330115032000440320008900135790592002274 1
704CC7600560079021200510350003700093030392000708 1
704CC3300210030008500520150001400036160158000044 1
705002701220156030700240120008100184460555001054 1
705CC1500360043009501900700024000514901740000050 1
705C008000100030013001200400000000037300250000000 1
705C00200000000000300050010000000000360006000000 1
705001100C00003001500120050000000003580028000000 1
705CC0700000001000900030030C0C0000C187C017C00000 1
705000600010002001000030020000000003010019000000 1
705CC5800120023009400270270000800027510178000034 1
705C078010900930281004203600073001096500524001695 1
705008700510058019700780400003400068610369000548 1
705012600350067022900950580002300078890428000619 1
705011400250047018700930530001700056110354000297 1
705012200380069023001010560002500082030431000675 1
705010300460069022000930480003100081850411000716 1
705011200470074023400970510003100087800437000816 1
705012500540085026501030580003600100450495001233 1
705C09015401490394007904200102001762007290044 40 1
705023010400780206003701100069000920403810004 10 1
705C035006500740175005701600043000875103200003 54 1
705C036001300320081002901600C800037750150000028 1
705C040001600280085002701800010000338401590000 35 1

```
705C043006500790187003001900C430CC92960343000446
706C027000300170048000401200002000208BC090000002
706C0001000CC0000001000200000CCC0000015000300000
706CC3700020008004800180170000100010110C92000001
706CC1100000002C013001200500000000024400260C0000
706CC0100000001000200010000C0C0000012100050000000
707000300010001000600020010000C0000022800110CC000
7C7CC0800060007002200080030000400009160041000000
707C02C000500090035C028009000C3000111200660CCC01
7C70C2C000500090035002200900003000110700650000001
7070C180003C007002800230080000200C829C053000000
7C7CC2100030007003200240100000200008930061000001
7C70C1100020005C019001100500001000059900370000000
707CC090002000300150C08004CC001CC0044100280CCCC0
707000100000000000100020000000000000060C020000000
7C7C006008901030199CC0800200C590012158035900008 4
701CC0500010003001100C700200001000C424002C0C0000
701CC1800020009003000020080000100011000057000000
701CC040C000C02000700010010CCCCCCCC03030C13000000
701004400150022008200270200001000C2590C154000030
701CC760035005201640052035000230006186030600029 2
701C0900031004501680085041000210005374031500027 2
701C0840037005501780086039000250006535033200037 4
701CC7800800098025700700360005300115970476001301
701CC8200410052017700820380002700C619703320003 74
701012800940089031200860590006200105100585002261
701C0960092010102890110044000610C1188405380018 76
701009000380057018601100410002500C67180348000405
701CC9800830113029500620450005500133910546001949
701CC4500770078020100590210005100092650371000583
701003901220066022700500180008100C77670426000666
701C028010500770211003801200C70000912203910004 50
701C026011800960240C03501200C790011314044300063 0
701CC1800670051013700220080004400061070254000126
701C04500120028008600410210000800033480161000033
702C008000500050019001100400003000062400350000 00
702CC040000000020008000700200000000030500150000 00
702CC0300020005001000080010000100006210019000000
702C07500150022011300360340001000026370215000052
702C0930063006102180061043000420007249040900077 0
702011300690069025200850520004600081850473001151
7020C710044005401700067033000290006428031800036 1
702CC7800390063018000770360002600074140335000408
702C10100420063020600970460002800074210386000562
702010700610097026700860490004000115290494001329
702CC9900590084024400900460003900098920453001043
702CC8801070112030900690410007100132890573002275
702012300790096030000600570005200113960559001986
702CC5000450065016100400230003000077170296000313
702003100700078018000320140004700092210330000356
702C0070142016203110023003000950019067056300031 9
702C0301420161030700090010009400189710554000104
702CC0701010111022000120030006700130920398000154
702002001040092021600200090006900108620397000396
```

A P P E N D I X 3

Program M O R E V

APPENDIX 3 -- Program MOREV.

FUNCTION.

To reduce hourly efficiency figures to a percentage cumulative frequency per decile base for plotting contours of equi-efficiency. That is, to pictorially report plant performance for any selected period.

DATA.

Hourly efficiency figures from program RESEVAL.

REFERENCE.

The logic of the program MOREV is described in section 3E of the text, pp 56 - 57.

1

```
001         IDENTIFICATION DIVISION.
002         PROGRAM-ID. 'RESOURCE USE TABULATION'.
003         ENVIRONMENT DIVISION.
004         INPUT-OUTPUT SECTION.
005         FILE-CONTROL.
006             SELECT CARD-IN ASSIGN TO UR-2501-S-SYSIN.
007             SELECT PRINTER ASSIGN TO UR-1403-S-SYSC02.
008             SELECT CARD-OUT ASSIGN TO UR-1443-S-SYS005.
009             SELECT CAB     ASSIGN TO UT-2314-S-SORTIN.
010             SELECT ABCD    ASSIGN TO UT-2314-S-SORTOUT.
011             SELECT MASK     ASSIGN TO UT-2314-S-DUMM.
012             SELECT TAPE-OUT ASSIGN TO UT-2400-S-SYS003.
013         DATA DIVISION.
014         FILE SECTION.
015         FD  CARD-IN  LABEL RECORDS ARE OMITTED.
016        *    DATA RECORDS ARE MAST-CD, DOWN-CD,CORE-CD,QUEUE-CD.
017         01  SUB-CARD-IN.
018             02  ALIAS-HR                                  PIC X(2).
019             02  FILLER                                    PIC X(9).
020             02  ADJUSTMENTS                               PIC X(68).
021             02  CD-TYPE                                   PIC X.
022         FD  PRINTER LABEL RECORDS ARE CMITTED.
023             01  SUB-PRINT.
024                 02  SUB-P-CC                              PIC X.
025                 02  SUB-PRINT-1                           PIC X(80).
026                 02  SUB-PRINT-2.
027                     03  FILLER                          PIC X(48).
028                     03 COMP-DAY                         PIC X(3).
029         FD  CARD-OUT LABEL RECORDS ARE OMITTED.
030         01  SUB-CARD-OUT                                  PIC X(80).
031         FD  CAB  LABEL RECORDS ARE STANDARD BLOCK CONTAINS
032             90  RECORDS.
033         01  SUB-CAB                                       PIC X(80).
034         FD  ABCD  RECORDING MODE IS F LABEL RECORDS ARE STANDARD
035             BLOCK CONTAINS 90 RECORDS.
036         01  SUB-ABCD                                      PIC X(80).
037         SD  MASK RECORDING MODE IS F.
038         01  SUB-MASK.
039             02  MHR-DAY-YR.
040                 03  MHR                                 PIC 9(2).
041                 03  MDAY                                PIC 9(3).
042                 03  MYR                                 PIC 9(2).
043             02  MWK-DAY                                 PIC 9.
044             02  MIO-REV                                 PIC 9(4).
045             02  MCPU-REV                                PIC 9(4).
046             02  MCORE-REV                               PIC 9(4).
047             02  MTOTAL-REV                              PIC 9(4).
048             02  MJOBS                                   PIC 9(4).
049             02  MPC-CHAN                                PIC 9(3)V99.
050             02  MPC-CPU                                 PIC 9(3)V99.
051             02  MKB-HR                                  PIC 9(3)V99.
052             02  MSYSEFF                                 PIC 9(3)V99.
053             02  MBCR                                    PIC X(32).
```

MOREV

```
 WORKING-STORAGE SECTION.
 77  DPERKBH    VALUE 0.85                          PIC 99V999.
 77  I        VALUE 1                               PIC 9(4).
 77  ADD-JOB VALUE ZERO                            PIC 9(4)V.
 77  IO-REV       VALUE ZERO                       PIC 9(5)V99.
 77  CPU-REV      VALUE ZERO                       PIC 9(5)V99.
 77  CORE-REV     VALUE ZERO                       PIC 9(5)V99.
 77  TOTAL-REV    VALUE ZERO                       PIC 9(5)V99.
 77  JOBS         VALUE ZERO                       PIC 9(5)V99.
 77  PC-CHAN      VALUE ZERO                       PIC 9(5)V99.
 77  PC-CPU       VALUE ZERO                       PIC 9(5)V99.
 77  KB-HR        VALUE ZERO                       PIC 9(5)V99.
 77  WK-DAY      VALUE ZERO                      PIC 9(3).
 77  AV-CORE  VALUE ZERO                          PIC 9(3).
 77  MAX-CORE VALUE 270                           PIC 9(3).
 77  MAX-IO VALUE 216                             PIC 9(3)V99.
 77  MAX-CP VALUE 150                             PIC 9(3)V99.
 77  PC-USE  VALUE ZERO                           PIC 9(3)V99.
 77  SWS        VALUE ZERO                        PIC 9.
 77  SWT   VALUE 1                                PIC 9.
 77  SWP      VALUE 1                             PIC 9.
*  NOTE THAT 2 IN COLUMN 80 OF DATA CARD WILL TRIGGER END OF SUB-SET
 77  SWNS   VALUE 0                               PIC 9.
*  NOTE THAT IF SWITCH-NO-SORT = 0 THE SORT WILL BE BY-PASSED
 77  USER-C    VALUE 300                          PIC 9(3).
 77  RATIO     VALUE 0                            PIC 9V9(4).
 77  NOPWM   VALUE 0                              PIC 9.
 77  MAXI    VALUE 0                              PIC 9(4).
 77  SUMEFF  VALUE 0                              PIC 9(3)V99.
 77  M             VALUE ZERO                      PIC 9(3).
 77  N             VALUE ZERO                      PIC 9(3).
 77  K             VALUE ZERO                      PIC 9(3).
 77  CNTR   VALUE 0                               PIC 999.
 77  M1                                           PIC 9(3).
 77  M2                                           PIC 9(3).
 77  M3                                           PIC 9(3).
 77  M4                                           PIC 9(3).
 77  KK       VALUE ZERO                          PIC 9(3).
 77  JJ       VALUE ZERO                          PIC 9(3).
 77  L        VALUE ZERO                          PIC 9(3).
 77  ARG      VALUE ZERO                          PIC 9(8).
 77  S        VALUE ZERO                          PIC 9(3).
 01  MASTABC.
     02  ABC   OCCURS 75 TIMES                    PIC X(80).
 01  CD1.
     02  YR-DAY-HR.
         03  FILLER                               PIC X(2).
         03  HR                                   PIC 9(2).
         03  FILLER                               PIC X(2).
         03  YR                                   PIC X(2).
         03  YR-DAY                               PIC 9(3).
     02  IOF                                      PIC 9(5)V.
     02  IOS                                      PIC V99.
     02  CPUF                                     PIC 9(5)V.
     02  CPUS                                     PIC V99.
     02  KBF                                      PIC 9(5)V.
     02  KBS                                      PIC V99.
```

```
     02  TOTF                                    PIC 9(5)V.
     02  TOTS                                    PIC V99.
     02  JOF                                     PIC 9(5)V.
     02  JOS                                     PIC V99.
     02  PCCF                                    PIC 9(5)V.
     02  PCCS                                    PIC V99.
     02  PCCPUF                                  PIC 9(5)V.
     02  PCCPUS                                  PIC V99.
     02  KBHF                                    PIC 9(5)V.
     02  KBHS                                    PIC V99.
     02  RLP                                     PIC 9(5)V99.
* RLP IS REVENUE  FROM CARDS IN  + OUT AND PRINTING
     02 FILLER                                  PIC X(4).
     02  C-T                                     PIC X.
     02  WDAY                                    PIC 9.
*
 01  TRANSACTION.
     02  TIME.
             03  T-HR                            PIC 9(2).
             03  FILLER                          PIC X.
             03  T-YR                            PIC X(2).
             03  T-YR-DAY                        PIC 9(3).
     02  DOWN-TIME                               PIC 9(2)V.
     02  USER-CORE                               PIC 9(3)V.
     02  ADD-IO-R                                PIC 9(3)V.
     02  ADD-CPU-R                               PIC 9(3)V.
     02  ADD-CORE-R                              PIC 9(3)V.
     02  ADD-JOBS                                PIC 9(3)V.
     02  FILLER                                  PIC X(55).
 01  SUB-BUILD-CARD.
     02  HR-DAY-YR.
         03  OHR                                 PIC 9(2).
         03  ODAY                                PIC 9(3).
         03  OYR                                 PIC 9(2).
     02  OWK-DAY                                 PIC 9.
     02  OIO-REV                                 PIC 9(4).
     02  OCPU-REV                                PIC 9(4).
     02  OCORE-REV                               PIC 9(4).
     02  OTOTAL-REV                              PIC 9(4).
     02  OJOBS                                   PIC 9(4).
     02  OPC-CHAN                                PIC 9(3)V99.
     02  OPC-CPU                                 PIC 9(3)V99.
     02  OKB-HR                                  PIC 9(3)V99.
     02  OSYSEFF                                 PIC 9(3)V99.
     02  PEFF                                    PIC 9(3)V99.
     02  SBCR                                    PIC X(32).


 01  EDIT-1.
     02  FILLER      VALUE SPACE                 PIC X.
     02  H-D-Y.
         03  H                                   PIC 9(2).
         03  FILLER     VALUE SPACE              PIC X.
         03  Y-D                                 PIC X(3).
         03  FILLER     VALUE SPACE              PIC X.
```

```
        03  Y                                      PIC 9(2).
    02  FILLER          VALUE SPACES               PIC X(2).
    02  W-D                                        PIC 9.
    02  FILLER          VALUE SPACES               PIC X(2).
    02  I-R                                        PIC ZZZ9.
    02  FILLER          VALUE SPACES               PIC X(2).
    02  C-R                                        PIC ZZZ9.
    02  FILLER          VALUE SPACES               PIC X(2).
    02  KB-R                                       PIC ZZZ9.
    02  FILLER          VALUE SPACES               PIC X(4).
    02  T-R                                        PIC ZZZ9.
    02  FILLER          VALUE SPACES               PIC X(4).
    02  J                                          PIC ZZZ9.
    02  FILLER          VALUE SPACES               PIC X(4).
    02  PC-CH                                      PIC ZZ9.9.
    02  FILLER          VALUE SPACES               PIC X(2).
    02  PC-CP                                      PIC ZZ9.9.
    02  FILLER          VALUE SPACES               PIC X(3).
    02  PC-KB                                      PIC ZZ9.9.
    02  FILLER          VALUE SPACES               PIC X(5).
    02  EFF                                        PIC ZZ9.9.
    02  FILLER          VALUE SPACES               PIC X(5).
    02  PER                                        PIC 9.
    02  PIEFF                                      PIC ZZ9.9.
    02 FILLER     VALUE SPACES                     PIC X(2).
    02  1CNT                                       PIC 9(3).
    02  FILLER          VALUE SPACES               PIC X(2).
    02  2CNT                                       PIC 9(3).
    02  FILLER          VALUE SPACES               PIC X(30).
01  EDIT-LIST.
    02  FILLER    VALUE SPACE                      PIC X.
    02  L-HR                                       PIC Z9.
    02  L-SUB-L-H         VALUE SPACES             PIC X(129).
01  LIST-HISTO.
    02  SUB-L-H OCCURS 44 TIMES                    PIC ZZZ.
01  CHAN-HISTO.
    02  SUB1-C-H OCCURS 24 TIMES.
        03  SUB2-C-H OCCURS 10 TIMES               PIC 9(3).
01  CPU-HISTO.
    02  SUB1-CPU-H OCCURS 24 TIMES.
        03  SUB2-CPU-H OCCURS 10 TIMES             PIC 9(3).
01  CORE-HISTO.
    02 SUB1-CORE-H OCCURS 24 TIMES.
        03  SUB2-CORE-H OCCURS 10 TIMES            PIC 9(3).
01  PC-HISTO.
    02  SUB1-PC-H OCCURS 24 TIMES.
        03  SUB2-PC-H OCCURS 10 TIMES              PIC 9(3).
01  ID-HEAD.
    02  ID-H                                       PIC X(79).
    02  FILLER                                     PIC X.
01  TENARG.
    02  SUBTARG  OCCURS 10 TIMES                   PIC 9(3).
01  PLOT.
    02  SUB1PLOT  OCCURS 96 TIMES.
        03  SUB2PLOT  OCCURS  6 TIMES              PIC 9(3).
01  MASB.
    02  B    OCCURS  20 TIMES                      PIC 9(7).
```

```
 01  CDIM.
     02  CDIM2    OCCURS 24 TIMES                PIC 9(3).



 PROCEDURE DIVISION.
     OPEN INPUT CARD-IN  OUTPUT PRINTER.
     OPEN OUTPUT CAB, CARD-OUT.
 SUB-RD.
     PERFORM B7 VARYING M FROM 1 BY 1 UNTIL M > 40.
     PERFORM B3 VARYING M FROM 1 BY 1 UNTIL M > 24.
     GO TO B5.
 B3.
     PERFORM B4 VARYING N FROM 1 BY 1 UNTIL N > 10.
 B4.
     MOVE 0 TO SUB2-CORE-H (M,N).
     MOVE 0 TO SUB2-C-H (M,N).
     MOVE 0 TO SUB2-PC-H (M,N).
     MOVE 0 TO SUB2-CPU-H (M,N).
 B7.
     MOVE ZERO TO SUB-L-H (M).
 B5.
*    COMPUTE DPERKBH = DPERKBH / 100.
*    EXHIBIT NAMED MAX-CORE, MAX-IO, MAX-CP, DPERKBH.
*    READY TRACE.



 RD1.
         READ CARD-IN INTO CD1 AT END GO TO SORT-LIST.
*    EXHIBIT NAMED SUB-CARD-IN.
     ADD 1 TO CNTR.
     IF CNTR > 10 RESET TRACE.
*    NOTE THAT SWT SHOULD = 1 IF UPDATE TRANSACTIONS ARE INCLUDED
*        BUT SWT SHOULD = 0 FOR BY-PASS.
     IF SWT = 0 GO TO BY-PASS.
     IF C-T = 3 MOVE SUB-CARD-IN TO ID-HEAD GO TO RD1.
     IF C-T NOT = 1 GO TO HR-CARD.
     MOVE SUB-CARD-IN TO ABC (I).  ADD 1 TO I.  GO TO RD1.
 HR-CARD.
     EXAMINE SUB-CARD-IN REPLACING ALL SPACES BY ZERO.
     MOVE SUB-CARD-IN  TO CD1.
*    IF HR = 32 MOVE 0 TO SWP  GO TO SORT-LIST.
     IF C-T = 2 MOVE 0 TO SWP    GO TO SORT-LIST.
*    IF ALIAS-HR = SPACES MOVE 0 TO SWP GO TO SORT-LIST.
     IF HR > 24
     GO TO RD1.
     IF TOTF = 0 GO TO RD1.
     IF I = 1 GO TO BY-PASS.  IF NOPWM = 0 GO TO  INITI.



 M-UNM.
     MOVE ABC (I) TO TRANSACTION.
```

```
*     EXHIBIT NAMED YR-DAY-HR, TIME, I, ABC (I).
      IF YR-DAY-HR > TIME     GO TO BY-PASS.
      IF YR-DAY-HR = TIME     GO TO UP-DATE.
      ADD 1 TO I.  IF I > MAXI GO TO BY-PASS. GO TO M-UNM.
 INITI.
      SUBTRACT 1 FROM I. MOVE I TO MAXI. MOVE 1 TO I.  GO TO M-UNM.


 UP-DATE.
      PERFORM BY-PASS THRU BY-P.  MOVE ABC (I) TO TRANSACTION.
*     PERFORM UP-D.  GO TO BY-P2.


 UP-D.
          EXAMINE TRANSACTION REPLACING ALL SPACE BY ZERO.
          COMPUTE ADD-JOB = ADD-IO-R +ADD-CPU-R + ADD-CORE-R.
          IF ADD-JOB = ZERO GO TO STEP2.
      IF ADD-IO-R NOT = ZERO ADD ADD-IO-R TO IO-REV.
      IF ADD-CPU-R NOT = ZERO ADD ADD-CPU-R TO CPU-REV.
      IF ADD-CORE-R NOT = ZERO ADD ADD-CORE-R TO CORE-REV.
      ADD ADD-JOB TO TOTAL-REV.
      IF ADD-JOBS NOT = 0 ADD ADD-JOBS TO JOBS.


 STEP2.
      IF DOWN-TIME NOT = ZERO PERFORM STEP3.
      IF ADD-CORE-R NOT = ZERO MOVE ADD-CORE-R TO AV-CORE, ELSE
           MOVE USER-C TO AV-CORE.
      GO TO BY-P2.
 STEP3.
      MOVE SUB-CARD-IN TO SUB-PRINT-1 MOVE SPACES TO SUB-PRINT-2.
      WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
      COMPUTE RATIO = 60  / DOWN-TIME.
*     MULTIPLY RATIO BY IO-REV.
*     MULTIPLY RATIO BY CPU-REV.
*     MULTIPLY RATIO BY CORE-REV.
*     MULTIPLY RATIO BY TOTAL-REV.
*     MULTIPLY RATIO BY JOBS.
      MULTIPLY RATIO BY PC-CHAN.
      MULTIPLY RATIO BY PC-CPU.
      MULTIPLY RATIO BY KB-HR.


BY-PASS.
     EXAMINE SUB-CARD-IN REPLACING ALL SPACES BY ZERO.
```

```
           MOVE SUB-CARD-IN  TO CD1.
           COMPUTE IO-REV = IOF + IOS.
           COMPUTE CPU-REV = CPUF + CPUS.
           COMPUTE CORE-REV = KBF + KBS.
           COMPUTE TOTAL-REV = TOTF + TOTS.
           COMPUTE JOBS = JOF + JOS.
           COMPUTE PC-CHAN = PCCF + PCCS.
           COMPUTE PC-CPU = PCCPUF + PCCPUS.
           COMPUTE KB-HR = KBHF + KBHS.
       *   COMPUTE WK-DAY = YR-DAY + 3.
           IF YR-DAY > 166 MOVE 170 TO AV-CORE, ELSE
           MOVE 230 TO AV-CORE.
           COMPUTE PC-USE = ((IO-REV + CPU-REV + CORE-REV * DPERKBH)
            / (MAX-IO + MAX-CP + AV-CORE * DPERKBH))  * 100.
           COMPUTE SUMEFF = (PC-CHAN * PC-CPU * KB-HR / AV-CORE) * 100.
       *   EXHIBIT NAMED PC-CHAN, PC-CPU, KB-HR, AV-CORE, SUMEFF.
           COMPUTE M = PC-CHAN * 10 + 1.
       *   EXHIBIT NAMED M, PC-CHAN.
       *   EXHIBIT NAMED SUB-CARD-IN.
       *   EXHIBIT NAMED IOF, IOS, CPUF, CPUS, KBF, KBS, TOTF, TOTS,
       *   JOF, JOS, PCCF, PCCS, PCCPUF, PCCPUS, KBHF, KBHS, YR-DAY
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                                ADD 1 TO SUB2-C-H (HR,M).
           COMPUTE M = PC-CPU * 10 + 1.
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                               ADD 1 TO SUB2-CPU-H (HR,M).
           COMPUTE M = (KB-HR / AV-CORE) * 10  + 1.
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                                     ADD 1 TO SUB2-CORE-H (HR,M).
           COMPUTE M = PC-USE / 10 + 1.
           IF M > 10 EXHIBIT NAMED M MOVE  10 TO M.
                                    ADD 1 TO SUB2-PC-H (HR,M).
        BY-P.
           MOVE SPACES TO SUB-PRINT-2.
       *   COMPUTE WK-DAY = WK-DAY - 7.
       *   IF WK-DAY > 7 GO TO BY-P.
       *   MOVE WK-DAY TO WDAY.
           MOVE WDAY  TO COMP-DAY
       *   MOVE WK-DAY TO COMP-DAY.
           MOVE CD1 TO SUB-PRINT-1.



        BY-P2.
**********  A TEST LIST OF INPUT DATA  **********************
       *   WRITE SUB-PRINT AFTER POSITIONING 1 LINES.



        BUILD-CARD.
           MOVE HR TO OHR                  MOVE YR-DAY TO ODAY
           MOVE YR TO OYR                  MOVE  WDAY TO OWK-DAY
           MOVE IO-REV TO OIO-REV          MOVE CPU-REV TO OCPU-REV
           MOVE CORE-REV TO OCORE-REV  MOVE TOTAL-REV TO  OTOTAL-REV
           MOVE JOBS TO OJOBS
```

```
      COMPUTE OPC-CHAN = PC-CHAN * 100.
      MULTIPLY 100 BY PC-CPU GIVING OPC-CPU.
      MOVE KB-HR TO OKB-HR.
      MOVE PC-USE TO OSYSEFF.
      MOVE SPACES TO SBCR.
      MOVE SUMEFF TO PEFF.



*WRITE SUB-BUILD CARD ONTO CARD, PRINTER, + TAPE
*AS + IF REQUIRED.
      MOVE SUB-BUILD-CARD TO SUB-PRINT-1.
      WRITE SUB-PRINT AFTER POSITIONING 1 LINES.
      WRITE SUB-CAB FROM SUB-BUILD-CARD
      GO TO RD1.
 HEAD1.
      MOVE LOW-VALUES TO SUB-PRINT.
      WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
      MOVE SPACES TO SUB-PRINT-1.
      WRITE SUB-PRINT AFTER POSITIONING 0 LINES.
      MOVE   '* PERIOD    *         REVENUE          *     NUMBER * PE
-     'RCENTAGE USE OF EQUIPMENT  *'
      TO SUB-PRINT-1.
      WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
      MOVE SPACES TO SUB-PRINT-1.
      MOVE   'HR DAY YR      CHAN  CPU  CORE   TOTAL  OF JOBS  ( CH
-     'ANS (CPU  AV.CORE NET EFF *'
      TO SUB-PRINT-1  WRITE SUB-PRINT AFTER POSITIONING 1 LINES.
      MOVE LOW-VALUES TO SUB-PRINT.
      WRITE SUB-PRINT AFTER POSITIONING 2 LINES.
 SORT-LIST.
      IF SWNS = 0 GO TO REP-HISTO.
      DISPLAY 'ENTERING SORT-LIST'.
      PERFORM HEAD1.
      CLOSE CAB.
      SORT MASK ON  DESCENDING KEY MSYSEFF USING CAB GIVING ABCD.
      OPEN INPUT ABCD.
      MOVE LOW-VALUES TO EDIT-1.
      MOVE ZERO TO 1CNT.
      MOVE ZERO TO 2CNT.
 RD2.
      READ ABCD INTO SUB-BUILD-CARD, AT END GO TO EOF.
          MOVE  OHR          TO  H
          MOVE  ODAY         TO  Y-D
          MOVE  OYR          TO  Y
          MOVE  OWK-DAY      TO  W-D
          MOVE  OIO-REV      TO  I-R
          MOVE  OCPU-REV     TO  C-R
          MOVE  OCORE-REV    TO  KB-R
          MOVE  OTOTAL-REV   TO  T-R
          MOVE  OJOBS        TO  J
          MOVE  OPC-CHAN     TO  PC-CH
          MOVE  OPC-CPU          TO  PC-CP
          MOVE  OKB-HR           TO  PC-KB
          MOVE  OSYSEFF          TO  EFF.
      MOVE PEFF TO PIEFF.
```

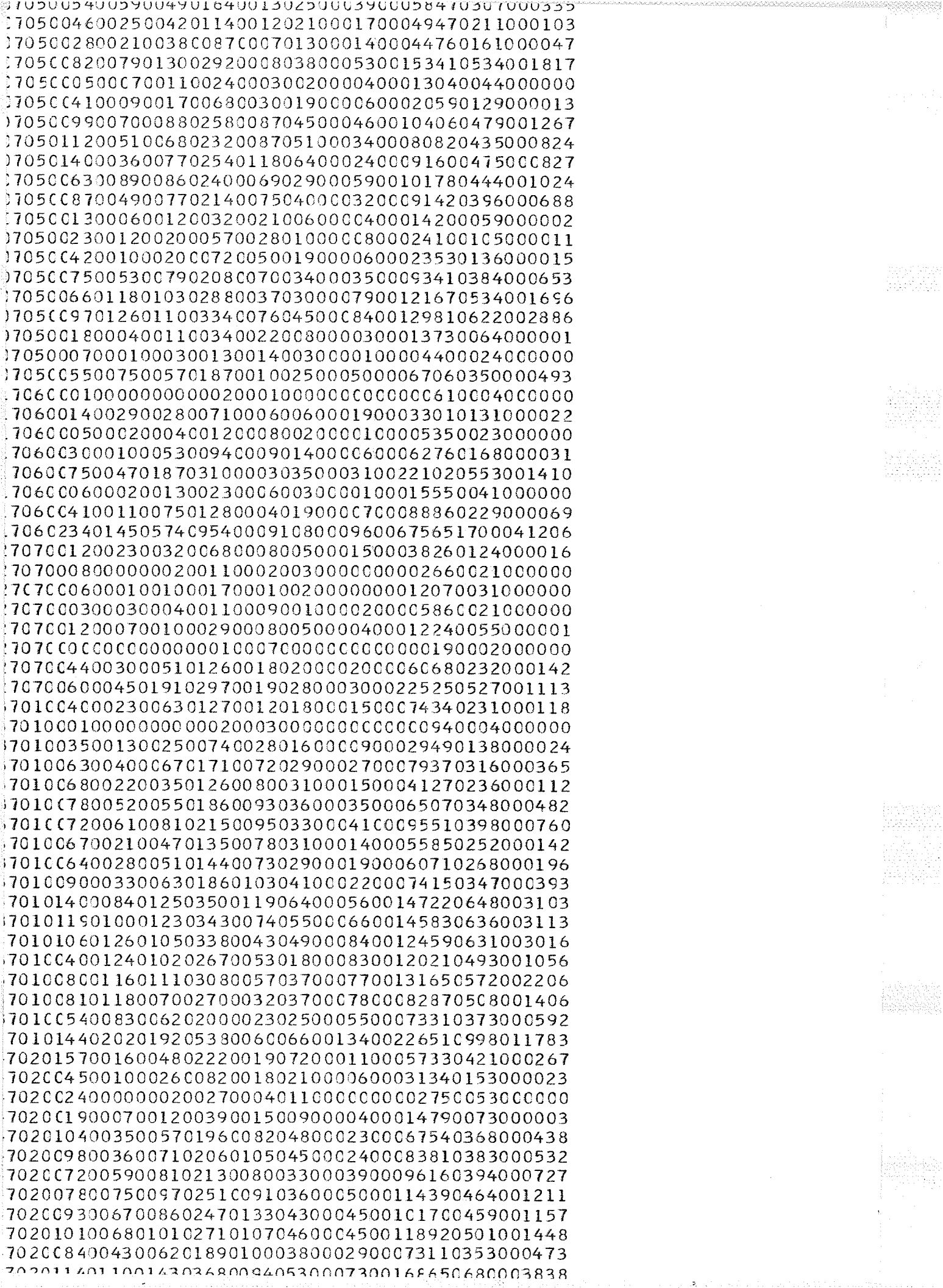
```
           IF ODAY > 166 ADD 1 TO 2CNT, ELSE ADD 1 TO 1CNT.
           MOVE LOW-VALUES TO SUB-PRINT.
           WRITE SUB-PRINT FROM EDIT-1 AFTER POSITIONING 1 LINES.
           GO TO RD2.
       B9.
           MOVE SUB2-C-H (I, 1) TO M1.
           MOVE SUB2-C-H (I, 10) TO M1.
           MOVE SUB2-CPU-H (I, 1) TO M2.
           MOVE SUB2-CPU-H (I, 10) TO M2.
           MOVE SUB2-CORE-H (I, 1) TO M3.
           MOVE SUB2-CORE-H (I, 10) TO M3.
           MOVE SUB2-PC-H (I, 1) TO M4.
           MOVE SUB2-PC-H (I, 10) TO M4.
       *   PERFORM B8 VARYING K FROM 1 BY 1 UNTIL K > 9.
           PERFORM B8 VARYING K FROM 10 BY -1 UNTIL K < 2.
       B8.
           COMPUTE N = K + 1
           COMPUTE N = K - 1.
           ADD SUB2-C-H (I, N) TO M1.
           IF SUB2-C-H (I, N) > 0 MOVE M1 TO SUB2-C-H (I, N).
           ADD SUB2-CPU-H (I, N) TO M2.
           IF SUB2-CPU-H (I, N) > 0 MOVE M2 TO SUB2-CPU-H (I, N).
           ADD SUB2-CORE-H (I, N) TO M3.
           IF SUB2-CORE-H (I, N)   > 0 MOVE M3 TO SUB2-CORE-H (I, N).
           ADD SUB2-PC-H (I, N) TO M4.
           IF SUB2-PC-H (I, N) > 0 MOVE M4 TO SUB2-PC-H (I, N).
       B10.
           IF SUB2-C-H (I, N) > 0 MOVE SUB2-C-H (I, N) TO M.
       B13.
           PERFORM B10 VARYING N FROM 10 BY -1 UNTIL N < 1.
           PERFORM B12 VARYING K FROM 1 BY 1 UNTIL K > 10.
       B12.
           COMPUTE SUB2-C-H (I, K) = SUB2-C-H (I, K) * 100 / M.
           COMPUTE SUB2-CPU-H (I, K) = SUB2-CPU-H (I, K) * 100 / M.
           COMPUTE SUB2-CORE-H (I, K) = SUB2-CORE-H (I, K) * 100 / M.
           COMPUTE SUB2-PC-H (I, K) = SUB2-PC-H (I, K) * 100 / M.
       EOF.
           CLOSE ABCD.
       REP-HISTO.
           MOVE LOW-VALUES TO SUB-PRINT.
           WRITE SUB-PRINT AFTER POSITIONING 0 LINES.
       B18.
           PERFORM B20.
           PERFORM B15.
       B19.
           WRITE   SUB-PRINT AFTER POSITIONING 2 LINES.
       *   WRITE   SUB-PRINT AFTER POSITIONING + LINES.
           MOVE                          '    PERCENT USAGE OF CHANNELS
       -                                 '      PERCENT USAGE OF CPU
       -                                 '      PERCENT USAGE OF CORE
       -                               ' SYSTEM EFFICIENCY'
           TO  SUB-PRINT.
           WRITE   SUB-PRINT AFTER POSITIONING 2 LINES.
           MOVE ' HR  0 10 20 30 40 50 60 70 80 90 100   10 20 30 40 50
       -   '60 70 80 90 100   10 20 30 40 50 60 70 80 90 100   10 20 30
       -   '40 50'  TO SUB-PRINT.
           MOVE '100' TO COMP-DAY.
```

```
      WRITE   SUB-PRINT AFTER POSITIONING 2 LINES.
      PERFORM B1 VARYING M FROM 1 BY 1 UNTIL M > 24.
 B14.
      GO TO B6.
 B20.
      MOVE ID-H TO SUB-PRINT WRITE SUB-PRINT AFTER POSITIONING 2
      LINES.
 B15.
      MOVE                 '                                          F
-                         'R E Q U E N C Y   ( D A Y S  /  D E C I L E
-                         '  /  H R ) .        '  TO SUB-PRINT.
 B16.
      MOVE                 '                        C U M L A T I V E    F
-                         'R E Q U E N C Y   ( D A Y S  /  D E C I L E
-                         '  /  H R ) .        '  TO SUB-PRINT.
 B17.
      MOVE                 '                                          P E R
-                          'C E N T A G E   O F  D A Y S  /  D E C I L E
-                         '  /  H R   .        '  TO SUB-PRINT.
 B1.
      PERFORM B2 VARYING N FROM 1 BY 1 UNTIL N > 10.
      MOVE LIST-HISTO TO L-SUB-L-H.
      MOVE M TO L-HR.
      WRITE SUB-PRINT FROM EDIT-LIST AFTER POSITIONING 2 LINES.
*     WRITE SUB-PRINT FROM LIST-HISTO AFTER POSITIONING 2 LINES.
 B2.
      COMPUTE K = N + 1.
      MOVE SUB2-C-H (M,N) TO SUB-L-H (K).
      COMPUTE K = N + 12
      MOVE SUB2-CPU-H (M,N) TO SUB-L-H (K).
      COMPUTE K = N + 23
      MOVE SUB2-CORE-H (M,N) TO SUB-L-H (K).
      COMPUTE K = N + 34
      MOVE SUB2-PC-H (M,N) TO SUB-L-H (K).
 B6.
      PERFORM B9 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM REP-HISTO.
      PERFORM B20.
      PERFORM B16.  PERFORM B19.
 B11.
*     PERFORM B10 VARYING N FROM 1 BY 1 UNTIL N > 10.
      PERFORM B10  VARYING N FROM 10 BY -1 UNTIL N < 1.
      PERFORM B13 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM  REP-HISTO.
      PERFORM B20.
      PERFORM B17.  PERFORM B19.
      PERFORM TRANS VARYING KK FROM 1 BY 1 UNTIL  KK > 24.
      PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
      PERFORM TRANS2 VARYING KK FROM 25 BY 1 UNTIL KK > 48.
      MOVE 24 TO JJ.
      PERFORM P13 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
      PERFORM TRANS3 VARYING KK FROM 49 BY 1 UNTIL KK > 72.
      MOVE 48 TO JJ.
      PERFORM P13 VARYING I FROM 1 BY 1 UNTIL I > 24.
      PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
      PERFORM TRANS4 VARYING KK FROM 73 BY 1 UNTIL KK > 96.
```

```
        MOVE 72 TO JJ.
        PERFORM P13 VARYING I FROM 1 BY 1 UNTIL I > 24.
        PERFORM YSWING VARYING JJ FROM 1 BY 1 UNTIL JJ > 6.
        GO TO BCH.
    P13.
        ADD 1 TO JJ.  MOVE SUB1PLOT (JJ) TO SUB1PLOT (I).
    TRANS.
        COMPUTE  L = KK
        MOVE SUB1-C-H (L) TO TENARG.
        MOVE 0 TO JJ.
        PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
        EXHIBIT NAMED MASB .
        PERFORM INT THRU P8      VARYING S FROM 1 BY 1 UNTIL S > 6.
        EXHIBIT NAMED SUB1PLOT (KK).
    TRANS2.
        COMPUTE  L = KK - 24.
        MOVE SUB1-CPU-H (L) TO TENARG.
        MOVE  0 TO JJ.
        PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
        EXHIBIT NAMED MASB .
        PERFORM INT THRU P8      VARYING S FROM 1 BY 1 UNTIL S > 6.
        EXHIBIT NAMED SUB1PLOT (KK).
    TRANS3.
        COMPUTE L = KK - 48.
        MOVE SUB1-CORE-H (L) TO TENARG.
        MOVE 0 TO JJ.
        PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
        PERFORM INT THRU P8      VARYING S FROM 1 BY 1 UNTIL S > 6.
                                  EXHIBIT NAMED SUB1PLOT (KK).
    TRANS4.
        COMPUTE L = KK - 72.
        MOVE SUB1-PC-H (L) TO TENARG.
        MOVE 0 TO JJ.
        PERFORM TENTOSIX THRU P7 VARYING I FROM 1 BY 1 UNTIL I > 10.
        PERFORM INT THRU P8      VARYING S FROM 1 BY 1 UNTIL S > 6.
                                  EXHIBIT NAMED SUB1PLOT (KK).
    TENTOSIX.
        IF SUBTARG (I) = 0  GO TO P7.
        ADD 2 TO JJ.
        COMPUTE M = JJ - 1.
        MOVE SUBTARG (I) TO B(M)
        COMPUTE B (JJ) = 10 * I.
        MOVE JJ TO K.
    P7.
        EXIT.
    INT.
        COMPUTE ARG = (6 - S) * 20
        PERFORM INT2 THRU P9   VARYING JJ FROM 2 BY 2 UNTIL JJ > K.
    INT2.
        COMPUTE M = JJ - 1.
*       EXHIBIT NAMED K, S, ARG, JJ, M, B (M).
        IF B (M) < ARG GO TO P9.
        IF B (M) = ARG GO TO P10.
        IF JJ = K GO TO P12.
        COMPUTE M1 = JJ + 1.
        COMPUTE M2 = JJ + 2.
        IF B (M1) > ARG GO TO P9.
```

```
           COMPUTE SUB2PLOT (KK, S) = (B (M) - ARG)
            * (B (M2) - B (JJ))/
           (B (M)      -  B (M1))   + B (JJ).
      *    EXHIBIT NAMED M, M1, M2, JJ, B (M), B (M1), B (M2), B (JJ),
      *     SUB2PLOT (KK, S).
           GO TO P11.
       P12.
           MOVE B (K) TO SUB2PLOT (KK,S)
           GO TO P11.
       P10.
           MOVE B (JJ) TC SUB2PLOT (KK,S).
       P11.
           GO TO P8.
       P9.
           EXIT.
       P8.
           EXIT.
      XSWING.
           MOVE SUB2PLOT (I, JJ) TO CDIM2 (I).
      YSWING.
           PERFORM  XSWING VARYING  I  FROM 1 BY 1 UNTIL I > 24.
                                                  EXHIBIT NAMED CDIM.
           MOVE SPACES  TC SUB-CARD-CUT.
           WRITE SUB-CARD-OUT FRCM CDIM.
      BCH.
           IF SWP = 0 CLOSE CAB OPEN OUTPUT CAB MOVE 1 TO SWP
           MOVE LOW-VALUES TO SUB-PRINT  WRITE SUB-PRINT AFTER
           POSITIONING 0 LINES
           MOVE 0 TO I  GO TO SUB-RD.
           CLOSE  CARD-IN, PRINTER.
           CLOSE CARD-OUT, ABCD.
           STOP RUN.
```

702CC2600340034C0960015012000220004101017700006 3
702C19901120139045200240920007400164350844006581
702018400680081033400230850004500095360630002145
702CC8900290050016900100410001900059030317000270
702010100550068022500950460003600080560420000784
702C09C0024004901640960410001600058210308000224
702CC7900460066019200850360003000078340356000497
702C0660071007202110073030000480008564039200 0725
702CC8400360062018300970390002400073330340000403
702CC8400390073019701130390C026C008633C365000514
702C056006800760201009702600045000895603720006 16
702CC61C09200990254008402800061001171604680011 77
702CC5500960079023100680250006400093970429C00884
702CC6201230092027800650290008200108740518001521
702CC7C012101220314005903200080001436205790021 62
702CC1800060011003600190080CC040001333006700000 2
702CC8E016201450397005604100108001716 5C736004470
702011601650158044C0089054001100018641081600651 3
703005700970092024800400260006500 1C8850458001082
703CC120139C076022900130050009300089840426000245
703C004015200550212001100200101000647703990000 76
703C00601470051020600200030009800060820388000105
703CC07014900550212002000300099000654203990001 14
703C024014900640238001901100099000764 6C448000489
703C00801580108027500130030010500127570508000236
703CC0800100012003100130040CC060001435005700000 2
703CC8000940083025700390370006200097960480001321
703009200510061020500710420003400071780384000602
703010600400071021800660490002600084330406000631
703011800670105029100950540004500124230540001775
703010500640100027000830480004300117920500001431
703CC8100380069018900910370002500C81390351000442
703CC5900210032011300860270001400038180212000084
703CC8E00430069020201260410002900082210376000574
703007500480064C1880109034000320CC760C0349000486
703CC7000530064018801000320003500076350350000503
703CC8001050106C292007103700C700012515054100190 6
703CC9200930094028000600420006200111090521001701
703CC8901180120032800590410007800142110608002673
703012601360156041900380580C0900018372077500564 1
703CC7600790109026500570350005300129060488001408
703C051004100610154008102300027000724802840002 64
704003300410038011300210150002700044730210000106
704CC7501370077028900180340009100090650544001649
704CC8501430102033100170390009500120730618002631
704C029011700770224001301300C78C0C9101041600054 2
704001400010004002000120060000000005580039000000
7C4CC020C0C0C0C0000300020010000C0000009100070000 00
704CC0100000C0000100020000C0CCCC0C0100002000000
704011300300055019900990520002000064720374000395
704CC970069C0940260011004500C460011067048300134 7
704011300990113032600970520006600133550 6C5002696
704C06001250102028800420270008300120680534001590
704010900380065021301020500002500076690398000563
704CC8100290045015600960370001900053380292000220
704CC8100590079022001020370003900093680408000795
704011400790082C27601000520005300C967505160015 68
704C10100590083024400750460003900098620455001040
704CC6501480164037800700300009800193990693003354
704C03500760073018500510160005100C8634034200041 4
704010300640114028300670480004300135070521001639
704CC9C0058009502440041041000390011219045000105 5
704CC6200170041012000340280001100048540224000087
705003700380048012400160170002500056610229000141
705C07800420064018600140360002800075550347000447



705CC2200100022005400110100CCC600C2620010000009
705C043006500790187003001900C430CC92960343000446
706C027000300170048000401200002000208800900000002
706000100CCC0000000100020000CCCCCCCCC0015000300000
706CC3700020008004800180170000100010110C92000001
706CC11C0000002C013001200500000000024400260C00000
706CC0100000001C002000100000CCCCCCCC012100050000000
707000300010001000600020010000C0000228001100C000
707CC0800060007002200080030000400009160041000000
707C02C000500090035C028009000C3000111200660CCCC01
707C02C000500090035002200900003000110700650000001
707C0180003C007002800230080000200C0829C053000000
707CC2100030007003200240100000200008930061000001
707C0C1100020005C019001100500001000059900370000000
707CC090002000300150CC8004CC001CC004410C28C00CC0
707000100000000000010002000000000000060C02000000
707C006008901030199CC8002000C5900121580359000084
701CC0500010003001100C700200001000C424002C0C0000
701CC1800020009003000020080000100011000057000000
701CC040C0CCCC02C00C700010010CCCCCCCC3030C13000000
701004400150022008200270200001000025900154000030
701CC760035005201640052035000230006186030600029 2
701C0900031004501680085041000210005374031500027 2
701C0840037005501780086039000250006535033200037 4
701CC7800800098025700700360005300115970476001301
701CC8200410052017700820380C2700C619703320003 74
701012800940089031200860590006200105100585002261
701C0960092010102890110044CC061CC11884053800187 6
701009000380057018601100410002500C671803480004C5
701CC980083011302950062045000550013391054600194 9
701CC450077007802010059021000510009265037100058 3
701003901220066022700500180008 1C0C77670426000666
701C028010500770211003801200C7000C91220391000450
701C026011800960240C035012 00C790011314044300063 0
701CC1800670051013700220080004400061070254000126
701CC450012002800860041021000080003348016 1000033
702CC0800050005C019001100400003CC006240035000000
702CC040000000200 08C007002000000000305 0C15000000
702CC030002000500100008001000010000621001900000 0
702C07500150022011300360340001C0002637021500005 2
702C0930063006102180061043 00C042000724904090C0770
702011300690069025200850520004600081850473001151
7020C710044005401700067033C002900064280318000361
702CC780039006301800077036000260007414033500040 8
702C10100420063020 6CC970460 0C28C0C7421038600056 2
702010700610097026700860490004000115290494001329
702CC9900590084024400900460003900098920453001043
702CC880107011203090069041000710013289057300227 5
702012300790096030000600570005200113960559001986
702CC5000450065C161004002300030000771702960003 13
702003100700078018000320140004700092210330000356
702C00701420162031100230030C095001906705630003 19
702CC030142016 1C3070009001000940018971055400010 4
702CC0701010111022000120030006700130920398000154
702002001040092021600200090006900108620397000396

F R E Q U E N C Y (D A Y S / D E C I L E / H R) .

RCENT USAGE OF CHANNELS

10	20	30	40	50	60	70	80	90	100
3	6	1				1	1		
6		1	2				1		1
3	2	1	1					1	
4	1	1		1					
4	1								
3	1		1						
3									
4									
	3	2	2						
2		1	2	4					
	1		1	6	3				
1			4	3	2	2			
1		2	5	1	3				
3		1	6	2	1	1			
3	1	2	3	3	2				
3	1		4	5	1				
1	2	1	4	3	1	1			
	1	2	3	2	5				
2	1	3	3	4					
2	6	3	1	2					
4	3	1	4	1					
4	3	1	2	1	1				
3	1	3	2	2					
3	3	5		1	1	1			1

PERCENT USAGE OF CPU

10	20	30	40	50	60	70	80	90	100
3	3	3	1			1		1	
5		2					1		3
3			1	1					3
2	2			1			1		1
3	1								1
3					1				1
2									1
4									
3	2					2			
2		2	1	3			1		
	1	5	4	1					
1	1	5	3	2					
1	1	1	3	3	2	1			
3	2	4	1	1	1	1		1	
3	5	2	1	2		1			
4	2	3	3	1		1			
3		3	4	1	1		1		
1			4		4	2	1		1
2			2		1	2	4	2	
3				2	1	1		4	3
4			1		1		3	2	2
3	1		1	1			2	1	3
2		1	2	1	2	1			2
2	2	2	1	1	2	2			3

PERCENT USAGE OF CORE

10	20	30	40	50	60	70	80	90	100
2	1	3	2	1		2			1
4	1		1	1	2		1		1
3			2		1	1	1		
2		1	3		1				
3		1	1						
3				1					1
2							1		
4									
	5			1	1				
2			1	3		3			
		1	4	5	1				
1			4	3	2	1	1		
1			1	3	4	2	1		
3			3	3	2	1	1	1	
3	1	2	2	3	1	2			
3	1	1	1	2	4	2			
1	1		3	3	3		1	1	
1				1	2	4	1	1	3
2				1	4	1	4		1
2				2	4	3	2		1
3		1		1	2		1	2	3
3	1			2		2	2		2
1	1		2	1	2	1	2		1
1	1	2	2	1	2	1	1		4

SYSTEM EFFICIENCY

10	20	30	40	50			100
1							
3	2	3		2	2		
4	2		1	1	1		2
3			2		1	2	
2		2	2	1			
3	1		1				
3				1	1		
2					1		
4							
	4	1	1	1			
2			3	3	1		
		1	6	4			
1			6	3	2		
1			6	2	2	1	
3		1	6	2	1	1	
3	1	4	2	3	1		
3	1	2	4	2	2		
1	2		5	3	1	1	
1			3	2	4	2	1
2		1	2	2	3	3	
2		1	3	3	3	2	
3	1		3		3	2	1
3	1	1		1	5		1
1	1	2	3	2	1		1
1	2	3	4		2		1

C U M L A T I V E F R E Q U E N C Y (D A Y S / D E C I L E / H R) .

RCENT USAGE OF CHANNELS										PERCENT USAGE OF CPU										PERCENT USAGE OF CORE										SYSTEM EFFICIENCY						
10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50		100
12	9	3				2	1			12	9	6	3			2		1		12	10	9	6	4		3			1	12	9	7		4	2	
11		5	4				2		1	11		6					4		3	11	7		6	5	4		2		1	11	7		5	4	3		2
8	5	3	2					1		8			5	4					3	8			5		3	2	1			8			5		3	2	
7	3	2		1						7	5			3			2		1	7		5	4		1					7		5	3	1		
5	1									5	2								1	5		2	1							5	2		1			
5	2		1							5					2				1	5				2				1		5				2	1	
3										3									1	3						1				3					1	
4										4										4										4						
	7	4	2							7	4					2					7			2	1						7	3	2	1		
9		7	6	4						9		7	5	4			1			9			7	6		3				9			7	4	1	
	11		10	9	3						11	10	5	1							11	10	6	1							11	10	4			
12			11	7	4	2				12	11	10	5	2						12			11	7	4	2	1			12			11	5	2		
12		11	9	4	3					12	11	10	9	6	3	1				12			11	10	7	3	1			12			11	5	3	1	
14		11	10	4	2	1				14	11	9	5	4	3	2		1		14			11	8	5	3	2	1		14		11	10	4	2	1	
14	11	10	8	5	2					14	11	6	4	3		1				14	11	10	8	6	3	2				14	11	10	6	4	1		
14	11		10	6	1					14	10	8	5	2		1				14	11	10	9	8	6	2				14	11	10	8	4	2		
13	12	10	9	5	2	1				13		10	7	3	2		1			13	12		11	8	5		2	1		13	12		10	5	2	1	
	13	12	10	7	5					13			12		8	4	2		1	13			12	11	9	5	4	3		13			12	9	7	3	1
13	11	10	7	4						13			11		9	8	6	2		13			11	10	6	5		1		13		11	10	8	6	3	
14	12	6	3	2						14				11	9	8		7	3	14			12	10	6	3		1		14		12	11	8	5	2	
13	9	6	5	1						13			9		8		7	4	2	13		10		9	8		6	5	3	13	10		9		6	3	1
12	8	5	4	2	1					12	9		8	7			6	4	3	12	9			8		6	4		2	12	9	8		7	6		1
11	8	7	4	2						11		9	8	6	5	3			2	11	10		9	7	6	4	3		1	11	10	9	7	4	2		1
15	12	9		4	3	2			1	15	13	11	9	8	7	5			3	15	14	13	11	9	8	6	5		4	15	14	12	9		5		3

16 TO 30, 1970 CORE 768K

PERCENTAGE OF DAYS / DECILE / HR .

RCENT USAGE OF CHANNELS

10	20	30	40	50	60	70	80	90	100
100	75	25				16	8		
100		45	36				18		9
100	62	37	25					12	
100	42	28		14					
100	20								
100	40		20						
100									
100									
	100	57	28						
100		77	66	44					
	100		90	81	27				
100			91	58	33	16			
100		91	75	33	25				
100		78	71	28	14	7			
100	78	71	57	35	14				
100	78		71	42	7				
100	92	76	69	38	15	7			
	100	92	76	53	38				
100	84	76	53	30					
100	85	42	21	14					
100	69	46	38	7					
100	66	41	33	16	8				
100	72	63	36	18					
100	80	60		26	20	13			6

PERCENT USAGE OF CPU

10	20	30	40	50	60	70	80	90	100
100	75	50	25			16		8	
100		54					36		27
100			62	50					37
100	71			42			28		14
100	40								20
100					40				20
100									33
100									
100	57					28			
100		77	55	44			11		
	100	90	45	9					
100	91	83	41	16					
100	91	83	75	50	25	8			
100	78	64	35	28	21	14		7	
100	78	42	28	21		7			
100	71	57	35	14		7			
100		76	53	23	15		7		
100			92		61	30	15		7
100			84		69	61	46	15	
100				78	64	57		50	21
100			69		61		53	30	15
100	75		66	58			50	33	25
100		81	72	54	45	27			18
100	86	73	60	53	46	33			20

PERCENT USAGE OF CORE

10	20	30	40	50	60	70	80	90	100
100	83	75	50	33		25		8	
100	63		54	45	36		18		9
100			62		37	25	12		
100		71	57		14				
100		40	20						
100				40				20	
100						33			
100									
	100			28	14				
100			77	66		33			
		100	90	54	9				
100			91	58	33	16	8		
100			91	83	58	25	8		
100			78	57	35	21	14	7	
100	78	71	57	42	21	14			
100	78	71	64	57	42	14			
100	92		84	61	38		15	7	
100				92	84	69	38	30	23
100				84	76	46	38		7
100				85	71	42	21		7
100		76		69	61		46	38	23
100	75			66		50	33		16
100	90		81	63	54	36	27		9
100	93	86	73	60	53	40	33		26

SYSTEM EFFICIENCY

10	20	30	40	50					100
100	75	58		33	16				
100	63		45	36	27				18
100			62		37	25			
100		71	42	14					
100	40		20						
100				40	20				
100					33				
100									
	100	42	28	14					
100			77	44	11				
		100	90	36					
100			91	41	16				
100			91	41	25	8			
100		78	71	28	14	7			
100	78	71	42	28	7				
100	78	71	57	28	14				
100	92		76	38	15	7			
100			92	69	53	23	7		
100		84	76	61	46	23			
100		85	78	57	35	14			
100	76		69		46	23	7		
100	75	66		58	50		8		
100	90	81	63	36	18		9		
100	93	80	60		33			20	

APPENDIX 4 -- Program GRAPH

FUNCTION.

1) To plot a percentage cumulative frequency per decile grid for each of the four efficiencies.

2) To superimpose equi-efficiency contours on the grid.

3) To develop and plot for each hour, a cross-section of the previous three-dimensional plan to show resource use and resource waste.

4) To develop and plot mean daily efficiencies for each solar hour.

5) To develop and plot mean month efficiencies.

DATA.

The efficiency figures derived by program RESEVAL.

REFERENCE.

A description of the method is given in section 3D of the text, p. 68-70.

```
      $JOB   WAITLV   DORAN,KP=29
  1         REAL A(24,7), B(24)
  2         REAL X(24)
  3         REAL Y(24)/  1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.,
           1  16.,17.,18.,19.,20.,21.,22.,23.,24./
  4      120 FORMAT (24F3.0)
  5      1   CALL ID ('DORAN',5)
  6          DO 3 J = 1,6
  7          READ (5,120,END=99) (X(I), I = 1,24)
  8          WRITE (6,121) (X(I), I = 1,24)
  9      121 FORMAT (1HO, 24F5.0)
 10          CALL BUILDA (A,X,J)
 11          CALL GRAPH2 (3.,8.)
 12       3  CALL GRAPHO (1,1,X,Y,24,0,100.,0.,24.,1.0)
 13          CALL GRAPH1 (4, 'CHANNELS EFF',20.,1.0,12)
 14          CALL MEAN (A,X,B)
 15          CALL GRAPHO (1,1,B,Y, 24, 3, 100., 0., 24., 1.)
 16          CALL GRAPH4
 17          CALL GRAPH4
 18          CALL GRAPH4
 19          CALL GRAPH4
 20          CALL XSEC (A,I,B,Y)
 21          CALL GRAPH4
 22          CALL GRAPH4
 23          CALL GRAPH4
 24          CALL GRAPH4
 25          DO 4 J=1,6
 26          READ (5,120) (X(I), I = 1,24)
 27          WRITE (6,121) (X(I), I = 1,24)
 28          CALL BUILDA (A,X,J)
 29          CALL GRAPH2 (3.,8.)
 30       4  CALL GRAPHO (1,1,X,Y,24,0,100.,0.,24.,1.)
 31          CALL GRAPH1 (4,'CPU EFF  ',20.,1.0,9)
 32          CALL MEAN (A,X,B)
 33          CALL GRAPHO (1,1,B,Y, 24, 3, 100., 0., 24., 1.)
 34          CALL GRAPH4
 35          CALL GRAPH4
 36          CALL GRAPH4
 37          CALL GRAPH4
 38          CALL XSEC (A,I,B,Y)
 39          CALL GRAPH4
 40          CALL GRAPH4
 41          CALL GRAPH4
 42          CALL GRAPH4
 43          DO 5 J=1,6
 44          READ (5,120) (X(I), I = 1,24)
 45          WRITE (6,121) (X(I), I = 1,24)
 46          CALL BUILDA (A,X,J)
 47          CALL GRAPH2 (3.,8.)
 48       5  CALL GRAPHO (1,1,X,Y,24,0,100.,0.,24.,1.)
 49          CALL GRAPH1 (4,'CORE EFF',20.,1.0,8)
 50          CALL MEAN (A,X,B)
 51          CALL GRAPHO (1,1,B,Y, 24, 3, 100., 0., 24., 1.)
 52          CALL GRAPH4
 53          CALL GRAPH4
 54          CALL GRAPH4
 55          CALL GRAPH4
 56          CALL XSEC (A,I,B,Y)
 57          CALL GRAPH4
 58          CALL GRAPH4
```

GRAPH

```
112           D(8) = I + .4 * Z
113           D(9) = I + .2 * Z
114           D(10) = I
115           C(5) = A(I,1)
116           C(6) = A(I,2)
117           C(7) = A(I,3)
118           C(8) = A(I,4)
119           C(9) = A(I,5)
120           C(10) = A(I,6)
121           WRITE (6,133) (C(K), K=1,10), (D(K), K = 1,10)
122       133 FORMAT (1HO,20F6.2)
123           SUM = 0.
124           DO 23 L = 1,24
125        23 SUM = SUM + B(L)
126           SUM = SUM / 24
127           DO 24 L = 1,24
128        24 BB(L) = SUM
129           CALL GRAPH2 (3.,8.)
130       22  CALL GRAPHO (1,1,C,D,10,0,100.,0.,24.,1.)
131           CALL GRAPHO (1,1,B,Y,24,3,100.,0.,24.,1.)
132           CALL GRAPHO (1,1,BB,Y,24,0,100.,0.,24.,1.)
133           RETURN
134           END

      $ENTRY
WARNING**  SPECIFICATION STATEMENT DOES NOT PRECEDE STATEMENT FUNCTION DEFINITIONS OR EXECUTABLE STATEMENTS
WARNING**  SPECIFICATION STATEMENT DOES NOT PRECEDE STATEMENT FUNCTION DEFINITIONS OR EXECUTABLE STATEMENTS

0.  10.  10.  10.  10.  10.  10.  10.  20.  10.  20.  10.  10.  10.  10.  10.  10.  20.  10.  10.  10.  10.  10.  10.
8.  17.  15.  13.  12.  13.  10.  10.  24.  27.  50.  43.  36.  28.  19.  19.  27.  37.  25.  21.  16.  15.  17.  20.
3.  24.  20.  16.  15.  16.  10.  10.  29.  42.  53.  49.  43.  42.  37.  43.  42.  46.  36.  25.  23.  22.  31.  30.
7.  35.  28.  21.  17.  20.  10.  10.  35.  50.  57.  57.  48.  47.  47.  50.  49.  58.  45.  30.  37.  31.  38.  41.
2.  75.  59.  41.  20.  40.  10.  10.  40.  50.  60.  67.  60.  55.  57.  56.  57.  60.  50.  41.  45.  47.  48.  60.
0. 100.  90.  50.  20.  40.  10.  10.  40.  50.  60.  70.  60.  70.  60.  60.  70.  60.  50.  50.  50.  60.  50. 100.
3.  41.  34.  24.  16.  23.  10.  10.  32.  40.  52.  51.  44.  42.  39.  41.  43.  48.  37.  29.  30.  30.  33.  41.

        PLOT NUMBER    1  COMPLETED

        PLOT NUMBER    2  COMPLETED

        PLOT NUMBER    3  COMPLETED

        PLOT NUMBER    4  COMPLETED

.00100.00  0.00  0.00 10.00 18.00 23.00 27.00 52.00 80.00  1.75  1.00  1.00  1.75  1.75  1.60  1.45  1.30  1.15  1.00
.00100.00  0.00  0.00 10.00 17.00 24.00 35.00 75.00100.00  2.75  2.00  2.00  2.75  2.75  2.60  2.45  2.30  2.15  2.00
.00100.00  0.00  0.00 10.00 15.00 20.00 28.00 59.00 90.00  3.75  3.00  3.00  3.75  3.75  3.60  3.45  3.30  3.15  3.00
.00100.00  0.00  0.00 10.00 13.00 16.00 21.00 41.00 50.00  4.75  4.00  4.00  4.75  4.75  4.60  4.45  4.30  4.15  4.00
.00100.00  0.00  0.00 10.00 12.00 15.00 17.00 20.00 20.00  5.75  5.00  5.00  5.75  5.75  5.60  5.45  5.30  5.15  5.00
```

.00100.00 0.00 0.00 10.00 13.00 16.00 20.00 40.00 40.00 6.75 6.00 6.00 6.75 6.75 6.60 6.45 6.30 6.15 6.00

.00100.00 0.00 0.00 10.00 10.00 10.00 10.00 10.00 10.00 7.75 7.00 7.00 7.75 7.75 7.60 7.45 7.30 7.15 7.00

.00100.00 0.00 0.00 10.00 10.00 10.00 10.00 10.00 10.00 8.75 8.00 8.00 8.75 8.75 8.60 8.45 8.30 8.15 8.00

.00100.00 0.00 0.00 20.00 24.00 29.00 35.00 40.00 40.00 9.75 9.00 9.00 9.75 9.75 9.60 9.45 9.30 9.15 9.00

.00100.00 0.00 0.00 10.00 27.00 42.00 50.00 50.00 50.00 10.75 10.00 10.00 10.75 10.75 10.60 10.45 10.30 10.15 10.00

.00100.00 0.00 0.00 20.00 50.00 53.00 57.00 60.00 60.00 11.75 11.00 11.00 11.75 11.75 11.60 11.45 11.30 11.15 11.00

.00100.00 0.00 0.00 10.00 43.00 49.00 57.00 67.00 70.00 12.75 12.00 12.00 12.75 12.75 12.60 12.45 12.30 12.15 12.00

.00100.00 0.00 0.00 10.00 36.00 43.00 48.00 60.00 60.00 13.75 13.00 13.00 13.75 13.75 13.60 13.45 13.30 13.15 13.00

.00100.00 0.00 0.00 10.00 28.00 42.00 47.00 55.00 70.00 14.75 14.00 14.00 14.75 14.75 14.60 14.45 14.30 14.15 14.00

.00100.00 0.00 0.00 10.00 19.00 37.00 47.00 57.00 60.00 15.75 15.00 15.00 15.75 15.75 15.60 15.45 15.30 15.15 15.00

.00100.00 0.00 0.00 10.00 19.00 43.00 50.00 56.00 60.00 16.75 16.00 16.00 16.75 16.75 16.60 16.45 16.30 16.15 16.00

.00100.00 0.00 0.00 10.00 27.00 42.00 49.00 57.00 70.00 17.75 17.00 17.00 17.75 17.75 17.60 17.45 17.30 17.15 17.00

.00100.00 0.00 0.00 20.00 37.00 46.00 58.00 60.00 60.00 18.75 18.00 18.00 18.75 18.75 18.60 18.45 18.30 18.15 18.00

.00100.00 0.00 0.00 10.00 25.00 36.00 45.00 50.00 50.00 19.75 19.00 19.00 19.75 19.75 19.60 19.45 19.30 19.15 19.00

.00100.00 0.00 0.00 10.00 21.00 25.00 30.00 41.00 50.00 20.75 20.00 20.00 20.75 20.75 20.60 20.45 20.30 20.15 20.00

.00100.00 0.00 0.00 10.00 16.00 23.00 37.00 45.00 50.00 21.75 21.00 21.00 21.75 21.75 21.60 21.45 21.30 21.15 21.00

.00100.00 0.00 0.00 10.00 15.00 22.00 31.00 47.00 60.00 22.75 22.00 22.00 22.75 22.75 22.60 22.45 22.30 22.15 22.00

.00100.00 0.00 0.00 10.00 15.00 22.00 31.00 47.00 60.00 22.75 22.00 22.00 22.75 22.75 22.60 22.45 22.30 22.15 22.00

.00100.00 0.00 0.00 10.00 17.00 31.00 38.00 48.00 50.00 23.75 23.00 23.00 23.75 23.75 23.60 23.45 23.30 23.15 23.00

PLOT NUMBER 5 COMPLETED

PLOT NUMBER 6 COMPLETED

PLOT NUMBER 7 COMPLETED

PLOT NUMBER 8 COMPLETED

0. 10. 10. 10. 10. 10. 10. 10. 10. 10. 20. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10.

3. 18. 25. 16. 13. 26. 36. 10. 14. 27. 32. 30. 33. 19. 19. 16. 26. 47. 45. 46. 29. 18. 31. 24.

6. 27. 41. 31. 16. 43. 63. 10. 19. 37. 36. 35. 46. 31. 25. 27. 36. 60. 70. 65. 62. 47. 46. 40.

4. 68. 88. 54. 20. 60. 90. 10. 49. 53. 41. 40. 54. 38. 31. 37. 44. 66. 81. 93. 85. 85. 62. 64.

5. 100. 100. 91. 100. 100. 100. 10. 70. 71. 46. 48. 62. 61. 51. 47. 53. 76. 88. 100. 96. 100. 93. 100.

0. 100. 100. 100. 100. 100. 100. 10. 70. 80. 50. 50. 70. 90. 70. 70. 80. 100. 90. 100. 100. 100. 100. 100.

7. 54. 62. 49. 41. 57. 69. 10. 38. 47. 38. 37. 47. 40. 33. 33. 41. 61. 67. 72. 65. 61. 57. 57.

PLOT NUMBER 9 COMPLETED

PLOT NUMBER 10 COMPLETED

PLOT NUMBER 11 COMPLETED

PLOT NUMBER 12 COMPLETED

.00100.00 0.00 0.00 10.00 18.00 26.00 34.00 56.00 90.00 1.75 1.00 1.00 1.75 1.75 1.60 1.45 1.30 1.15 1.00

.00100.00 0.00 0.00 10.00 18.00 27.00 68.00100.00100.00 2.75 2.00 2.00 2.75 2.75 2.60 2.45 2.30 2.15 2.00

.00100.00 0.00 0.00 10.00 25.00 41.00 88.00100.00100.00 3.75 3.00 3.00 3.75 3.75 3.60 3.45 3.30 3.15 3.00

.00100.00 0.00 0.00 10.00 16.00 31.00 54.00 91.00100.00 4.75 4.00 4.00 4.75 4.75 4.60 4.45 4.30 4.15 4.00

.00100.00 0.00 0.00 10.00 13.00 16.00 20.00100.00100.00 5.75 5.00 5.00 5.75 5.75 5.60 5.45 5.30 5.15 5.00

.00100.00 0.00 0.00 10.00 26.00 43.00 60.00100.00100.00 6.75 6.00 6.00 6.75 6.75 6.60 6.45 6.30 6.15 6.00

.00100.00 0.00 0.00 10.00 36.00 63.00 90.00100.00100.00 7.75 7.00 7.00 7.75 7.75 7.60 7.45 7.30 7.15 7.00

.00100.00 0.00 0.00 10.00 10.00 10.00 10.00 10.00 10.00 8.75 8.00 8.00 8.75 8.75 8.60 8.45 8.30 8.15 8.00

.00100.00 0.00 0.00 10.00 14.00 19.00 49.00 70.00 70.00 9.75 9.00 9.00 9.75 9.75 9.60 9.45 9.30 9.15 9.00

.00100.00 0.00 0.00 10.00 27.00 37.00 53.00 71.00 80.00 10.75 10.00 10.00 10.75 10.75 10.60 10.45 10.30 10.15 10.00

.00100.00 0.00 0.00 20.00 32.00 36.00 41.00 46.00 50.00 11.75 11.00 11.00 11.75 11.75 11.60 11.45 11.30 11.15 11.00

.00100.00 0.00 0.00 10.00 30.00 35.00 40.00 48.00 50.00 12.75 12.00 12.00 12.75 12.75 12.60 12.45 12.30 12.15 12.00

.00100.00 0.00 0.00 10.00 33.00 46.00 54.00 62.00 70.00 13.75 13.00 13.00 13.75 13.75 13.60 13.45 13.30 13.15 13.00

.00100.00 0.00 0.00 10.00 19.00 31.00 38.00 61.00 90.00 14.75 14.00 14.00 14.75 14.75 14.60 14.45 14.30 14.15 14.00

.00100.00 0.00 0.00 10.00 19.00 25.00 31.00 51.00 70.00 15.75 15.00 15.00 15.75 15.75 15.60 15.45 15.30 15.15 15.00

.00100.00 0.00 0.00 10.00 16.00 27.00 37.00 47.00 70.00 16.75 16.00 16.00 16.75 16.75 16.60 16.45 16.30 16.15 16.00

.00100.00 0.00 0.00 10.00 26.00 36.00 44.00 53.00 80.00 17.75 17.00 17.00 17.75 17.75 17.60 17.45 17.30 17.15 17.00

.00100.00 0.00 0.00 10.00 47.00 60.00 66.00 76.00100.00 18.75 18.00 18.00 18.75 18.75 18.60 18.45 18.30 18.15 18.00

.00100.00 0.00 0.00 10.00 45.00 70.00 81.00 88.00 90.00 19.75 19.00 19.00 19.75 19.75 19.60 19.45 19.30 19.15 19.00

.00100.00 0.00 0.00 10.00 46.00 65.00 93.00100.00100.00 20.75 20.00 20.00 20.75 20.75 20.60 20.45 20.30 20.15 20.00

.00100.00 0.00 0.00 10.00 29.00 62.00 85.00 96.00100.00 21.75 21.00 21.00 21.75 21.75 21.60 21.45 21.30 21.15 21.00

.00100.00 0.00 0.00 10.00 18.00 47.00 85.00100.00100.00 22.75 22.00 22.00 22.75 22.75 22.60 22.45 22.30 22.15 22.00

.00100.00 0.00 0.00 10.00 31.00 46.00 62.00 93.00100.00 23.75 23.00 23.00 23.75 23.75 23.60 23.45 23.30 23.15 23.00

PLOT NUMBER 13 COMPLETED

PLOT NUMBER 14 COMPLETED

PLOT NUMBER 15 COMPLETED

PLOT NUMBER 16 COMPLETED

PLOT NUMBER 6

PLOT NUMBER 5

JUN 16'70 PLOT NUMBER 4

PLOT NUMBER 3

PLOT NUMBER 2

OORAN JUN 16'70 PLOT NUMBER 1

CPU EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

JUN 15'70

PLOT NUMBER 9

PLOT NUMBER 10

PLOT NUMBER 11

JUN 15'70 PLOT NUMBER 12

PLOT NUMBER 13

PLOT NUMBER 14

PLOT NUMBER 23

PLOT NUMBER 22

PLOT NUMBER 21

JUN 15'70

PLOT NUMBER 20

PLOT NUMBER 19

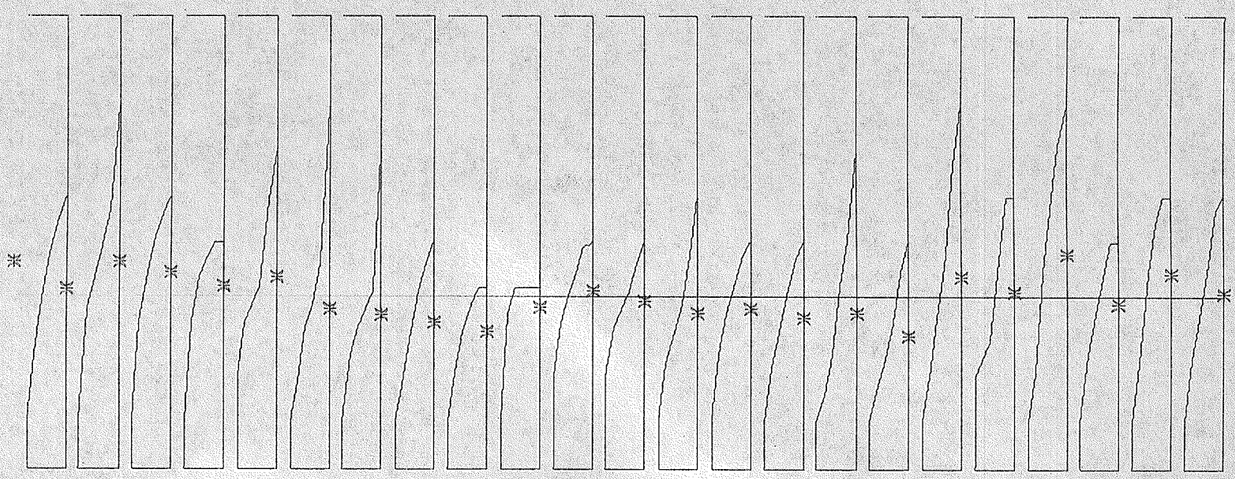
PLOT NUMBER 18

JUN 15'70 PLOT NUMBER 17

CORE EFF

PLOT NUMBER 31

PLOT NUMBER 30

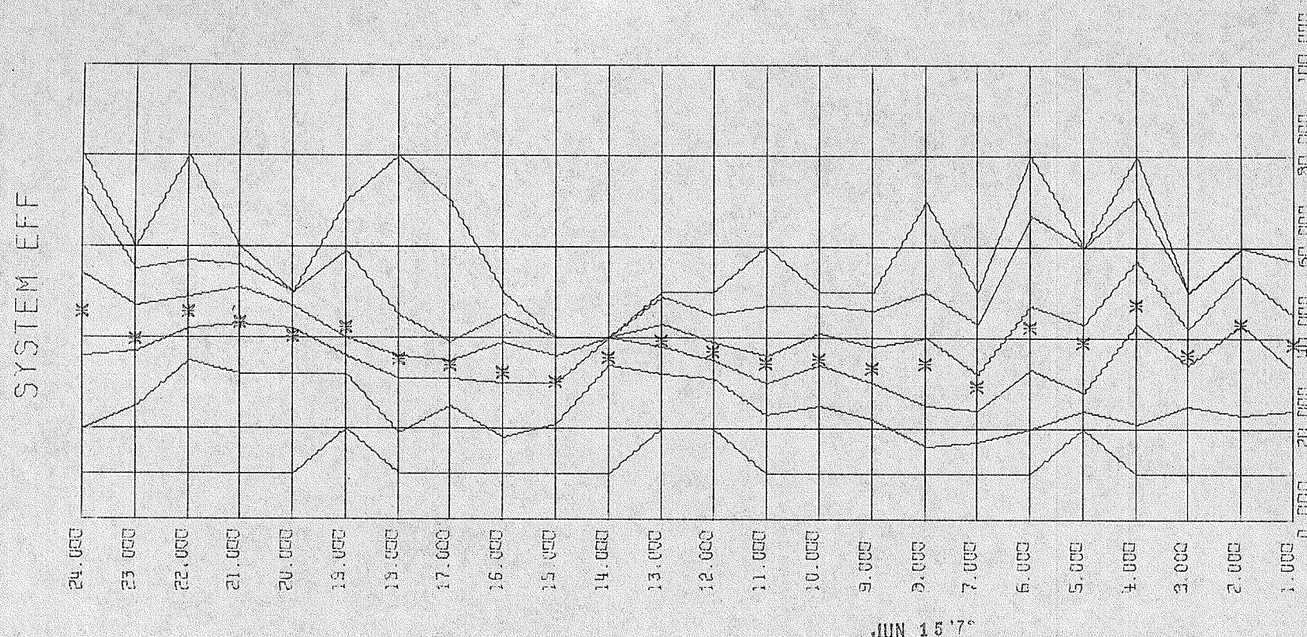


JUN 15'70 PLOT NUMBER 29

PLOT NUMBER 28

PLOT NUMBER 27

PLOT NUMBER 26



JUN 15'70 PLOT NUMBER 25

CHANNELS EFF

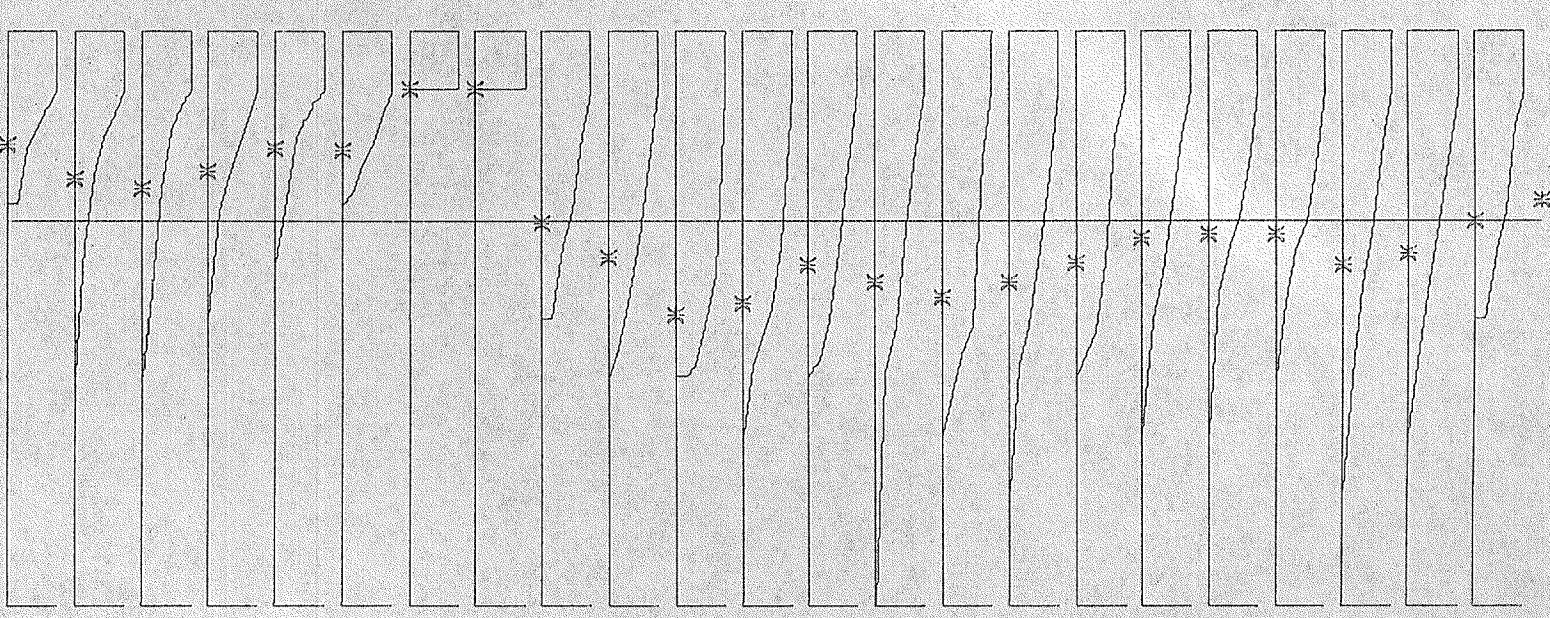
PLOT NUMBER 66

PLOT NUMBER 67

PLOT NUMBER 68

PLOT NUMBER 69

JUL '70



PLOT NUMBER 70

PLOT NUMBER 71

CPU EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

JUL '70

PLOT NUMBER 73

PLOT NUMBER 74

PLOT NUMBER 75

PLOT NUMBER 76

JUL '70

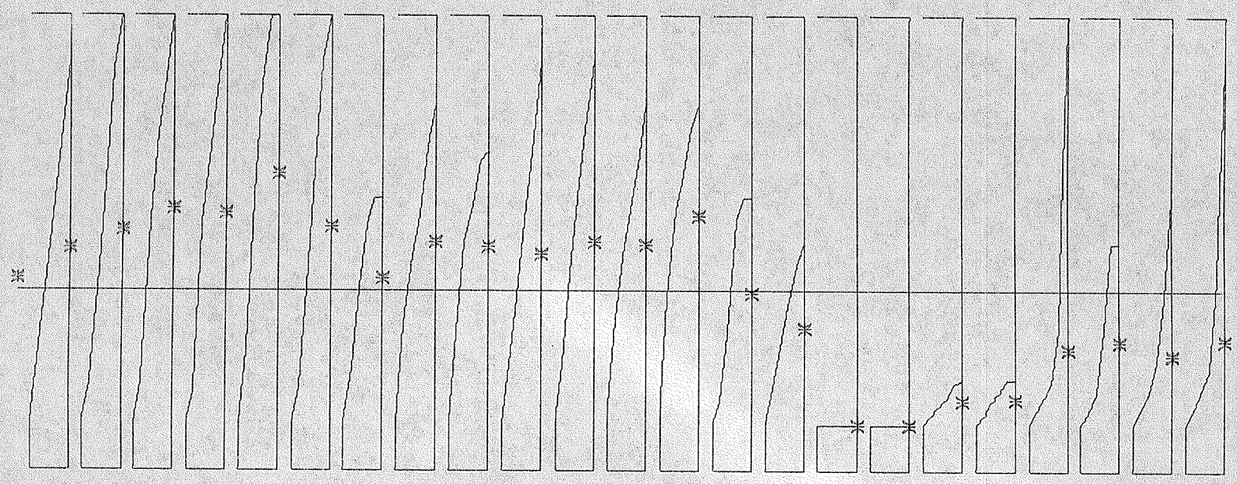
PLOT NUMBER 77

PLOT NUMBER 78

PLOT NUMBER 79

PLOT NUMBER 67

PLOT NUMBER 65



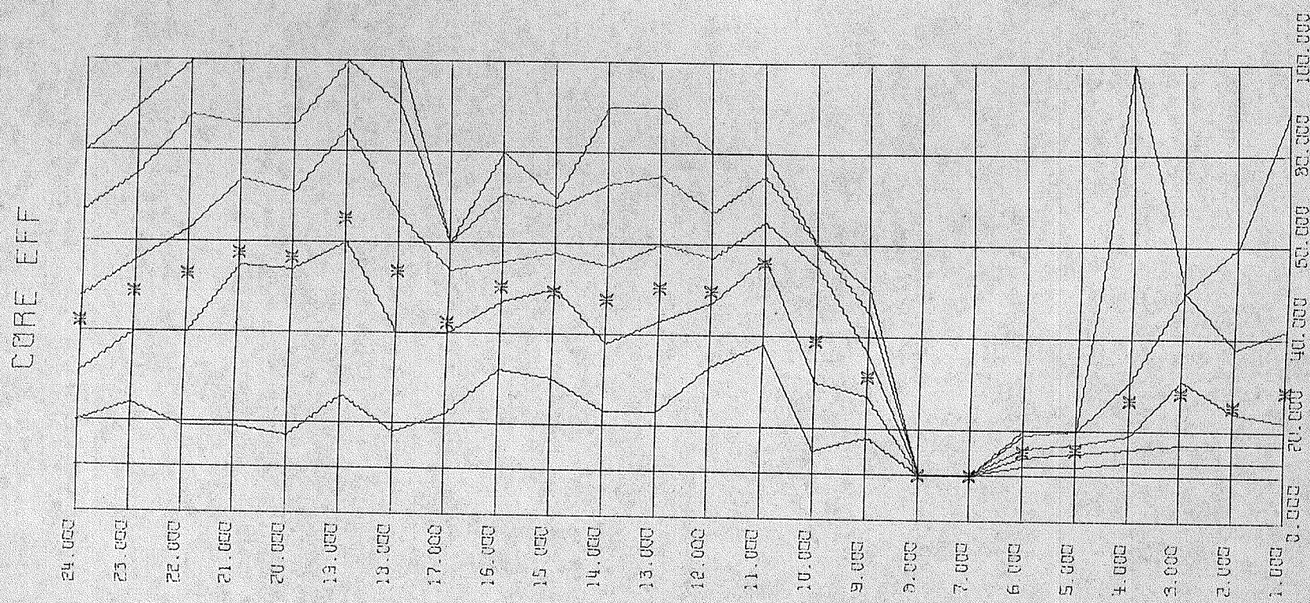
JUL '70

PLOT NUMBER 65

PLOT NUMBER 64

PLOT NUMBER 63

PLOT NUMBER 62



JUL '70

PLOT NUMBER 61

PLOT NUMBER 60

PLOT NUMBER 89

PLOT NUMBER 90

PLOT NUMBER 91

PLOT NUMBER 92

PLOT NUMBER 93

PLOT NUMBER 94

PLOT NUMBER 95

CHANNELS EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

DORAN AUG '70 PLOT NUMBER 1

PLOT NUMBER 2

PLOT NUMBER 3

PLOT NUMBER 4

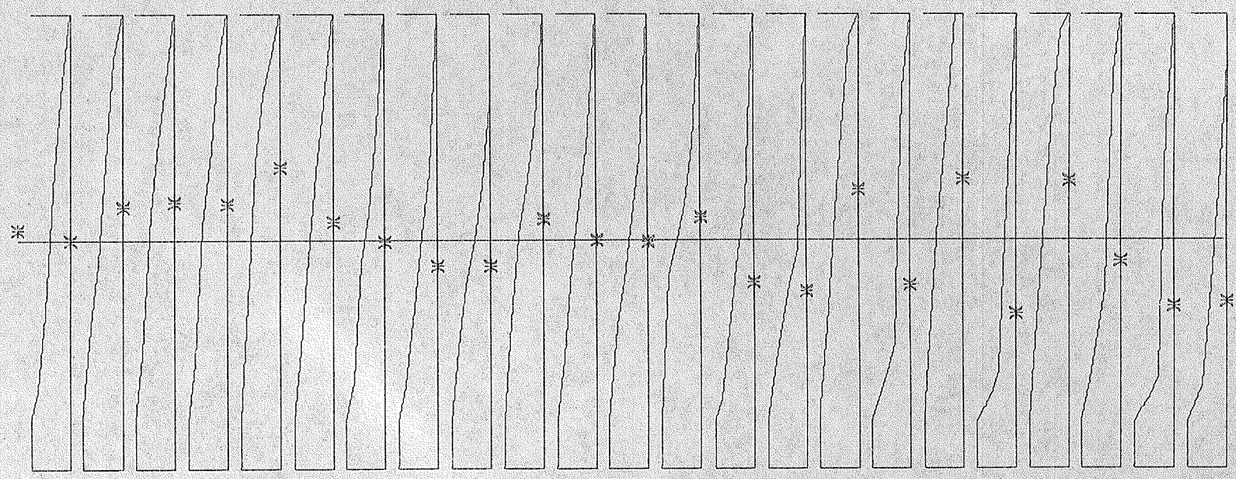
AUG '70 PLOT NUMBER 5

PLOT NUMBER 5

PLOT NUMBER 7

PLOT NUMBER 15

PLOT NUMBER 14



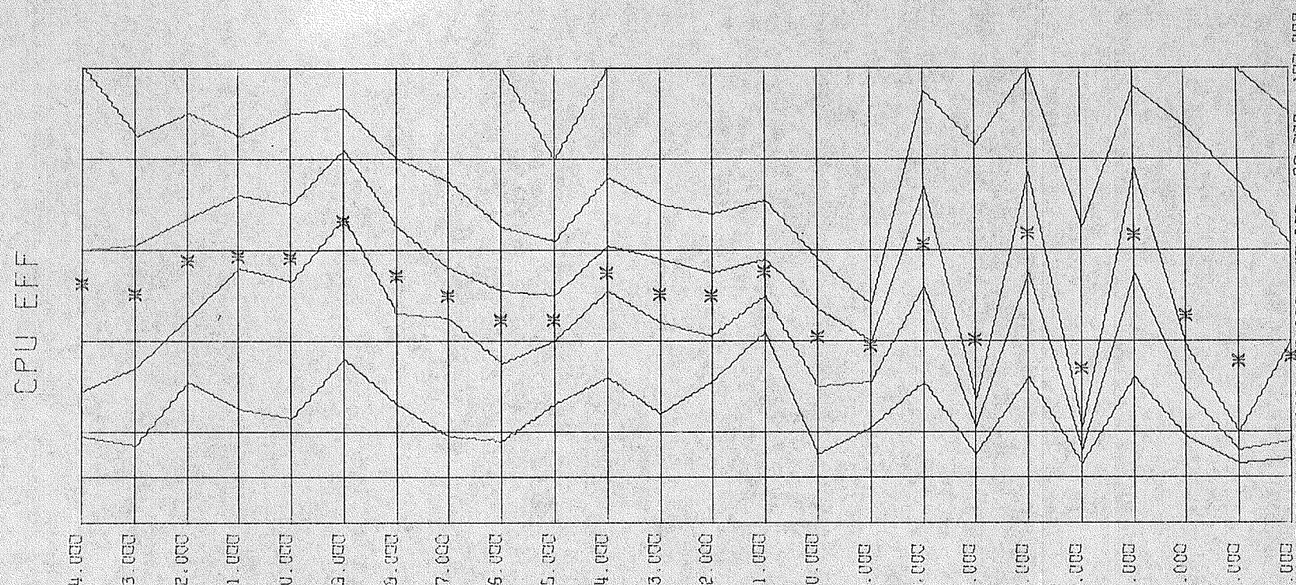
AUG '70

PLOT NUMBER 13

PLOT NUMBER 12

PLOT NUMBER 11

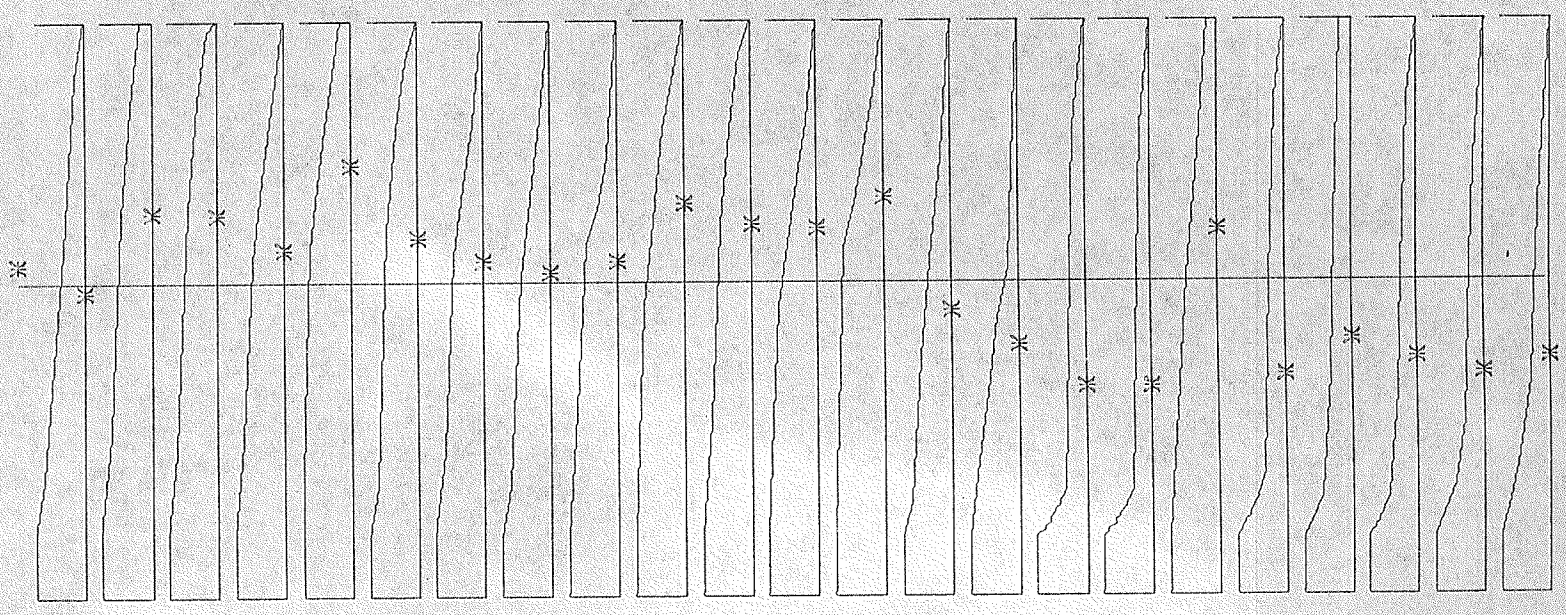
PLOT NUMBER 10



AUG '70

PLOT NUMBER 9

PLOT NUMBER 22

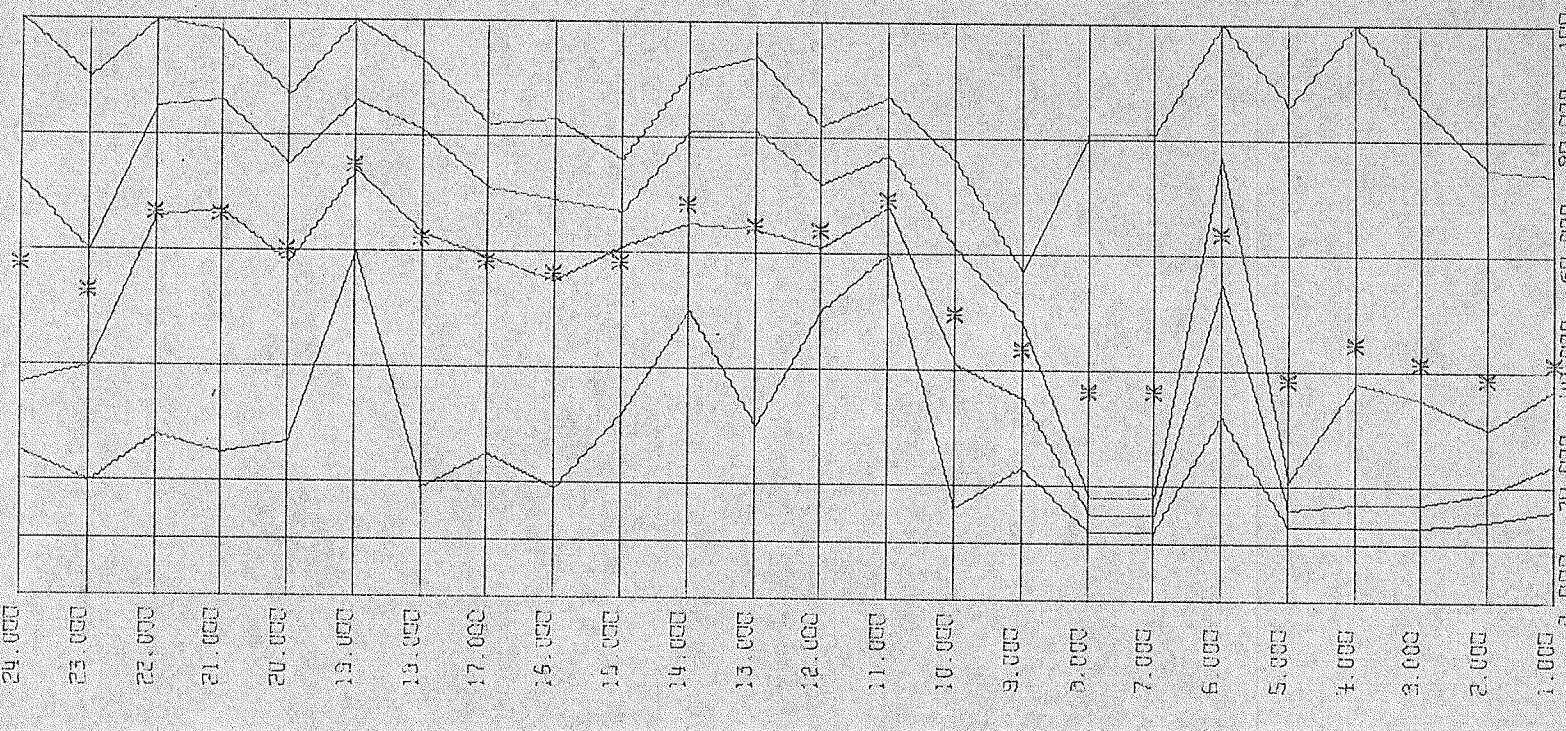


PLOT NUMBER 21

PLOT NUMBER 20

PLOT NUMBER 19

PLOT NUMBER 18



AUG '70

PLOT NUMBER 17

PLOT NUMBER 16

PLOT NUMBER 31

PLOT NUMBER 30

PLOT NUMBER 29

AUG '70

PLOT NUMBER 28

PLOT NUMBER 27

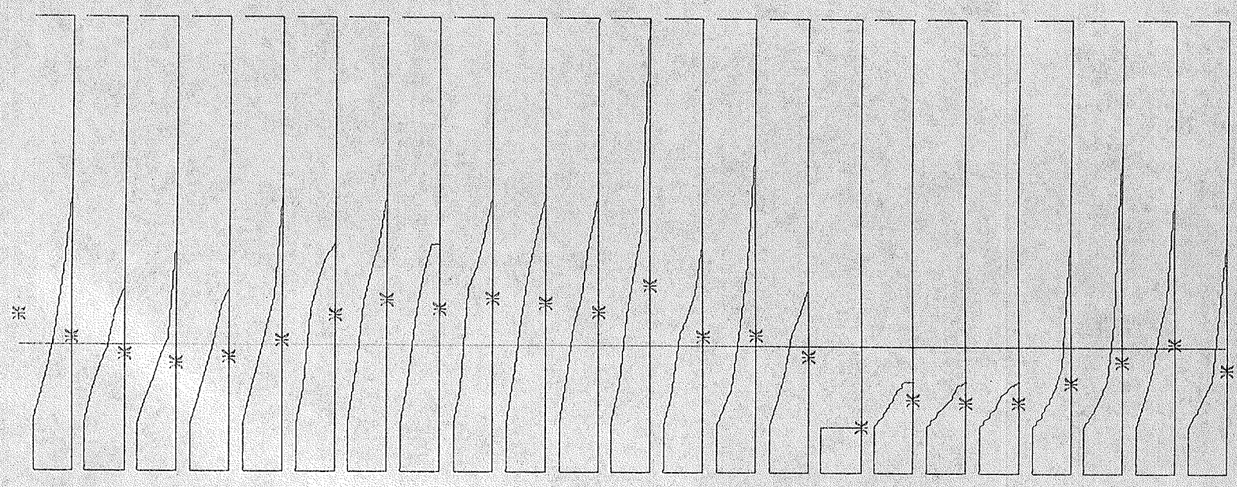
PLOT NUMBER 25

AUG '70 PLOT NUMBER 25

SYSTEM EFF

PLOT NUMBER 39

PLOT NUMBER 38

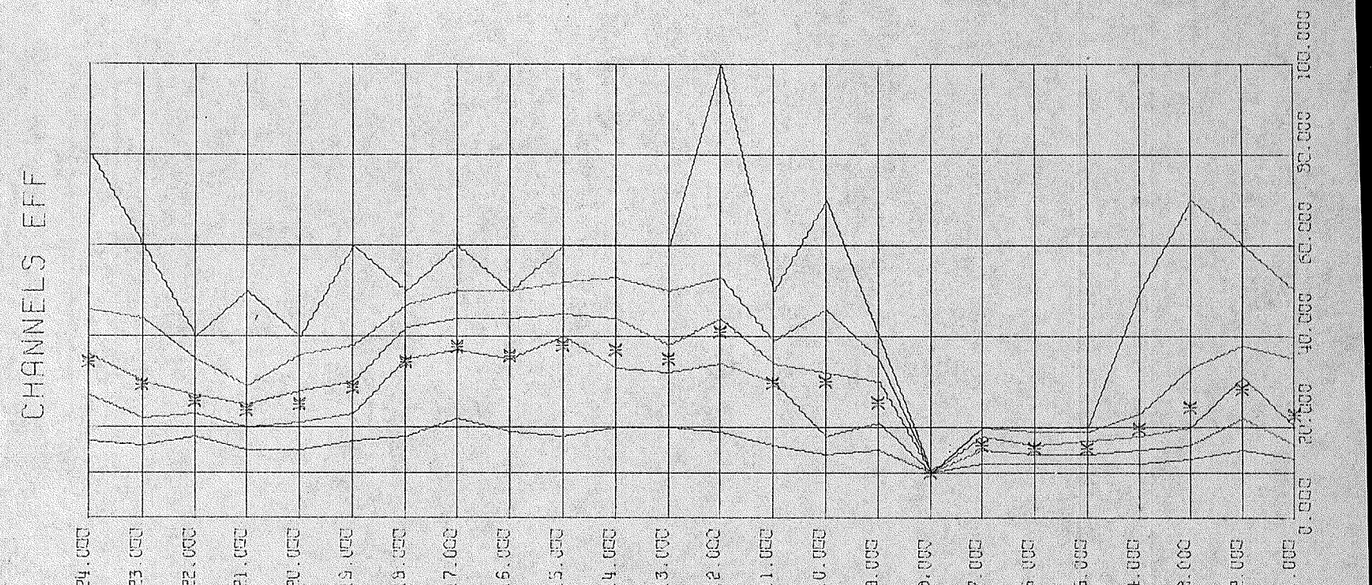


SEP '70 PLOT NUMBER 37

PLOT NUMBER 36

PLOT NUMBER 35

PLOT NUMBER 34



ORAN SEP '70 PLOT NUMBER 33

CPU EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

SEP '70

PLOT NUMBER 41

PLOT NUMBER 42

PLOT NUMBER 43

PLOT NUMBER 44

SEP '70

PLOT NUMBER 45

PLOT NUMBER 46

PLOT NUMBER 47

CORE EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

SEP '70

PLOT NUMBER 49

PLOT NUMBER 50

PLOT NUMBER 51

PLOT NUMBER 52

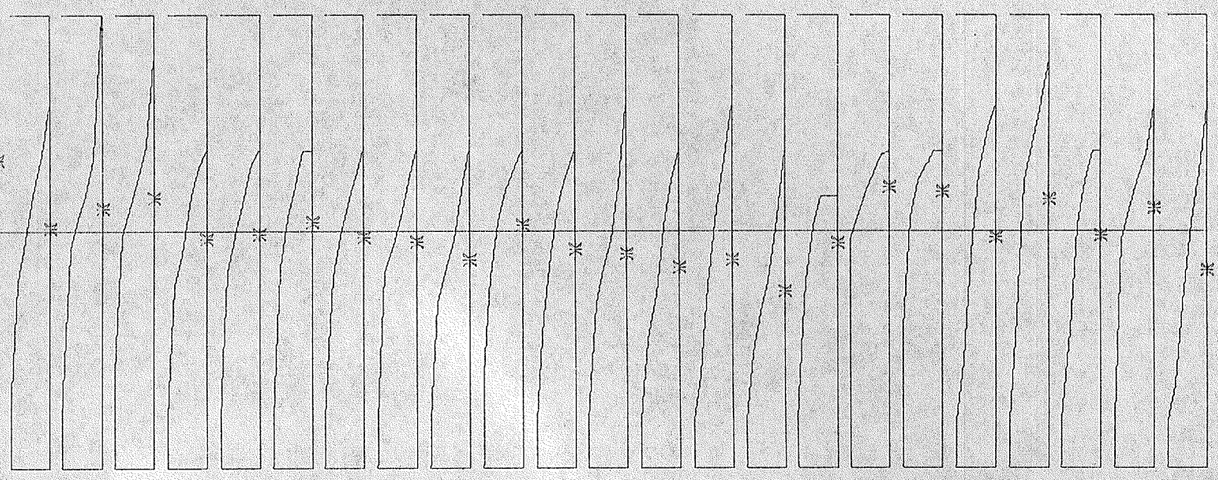
SEP '70

PLOT NUMBER 53

PLOT NUMBER 54

PLOT NUMBER 63

PLOT NUMBER 62



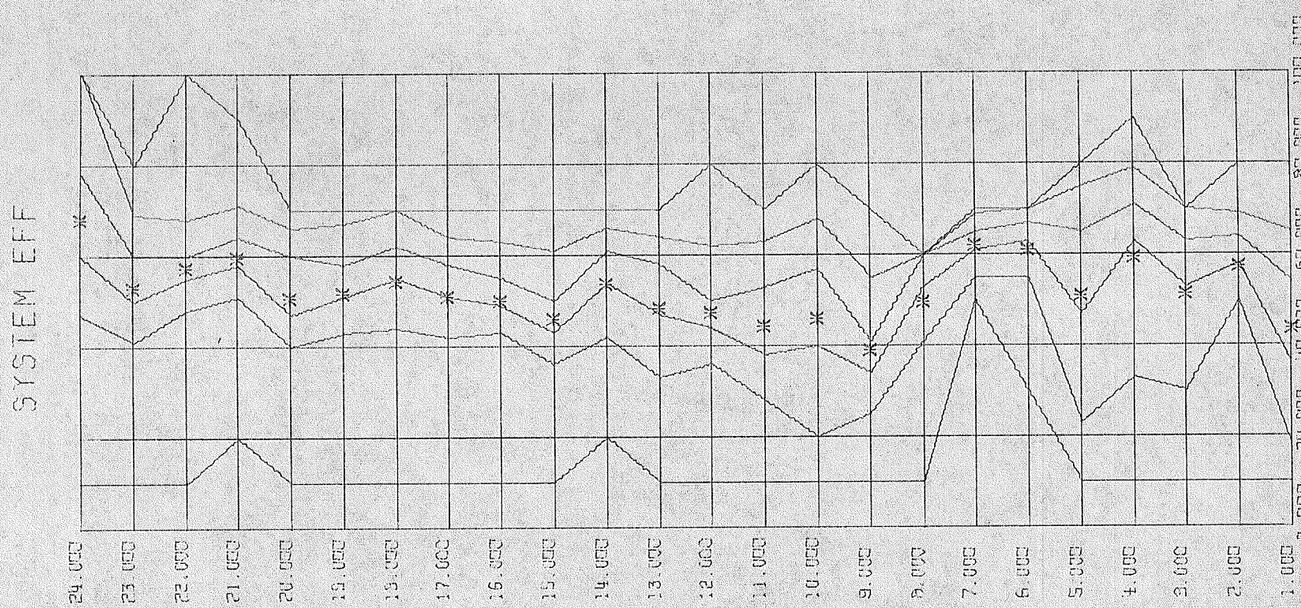
SEP '70

PLOT NUMBER 61

PLOT NUMBER 60

PLOT NUMBER 59

PLOT NUMBER 58



SEP '70

PLOT NUMBER 57

CHANNELS EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

COBRA

02, 130

PLOT NUMBER 54

PLOT NUMBER 55

PLOT NUMBER 56

PLOT NUMBER 57

PLOT NUMBER 58

02, 130

PLOT NUMBER 59

PLOT NUMBER 70

CPU EFF

07, '20

PLOT NUMBER 73

07, '20

PLOT NUMBER 74

07, '20

PLOT NUMBER 75

07, '20

PLOT NUMBER 76

07, '20

PLOT NUMBER 77

07, '20

PLOT NUMBER 78

CORE EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

OCT '10

PLOT NUMBER S1

PLOT NUMBER S2

PLOT NUMBER S3

PLOT NUMBER S4

OCT '10

PLOT NUMBER S5

PLOT NUMBER S5

SYSTEM EFF

24.000
23.000
22.000
21.000
20.000
19.000
18.000
17.000
16.000
15.000
14.000
13.000
12.000
11.000
10.000
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000

0.000 20.000 40.000 60.000 80.000 100.000

OCT '20

PLOT NUMBER 89

PLOT NUMBER 90

PLOT NUMBER 91

PLOT NUMBER 92

OCT '20

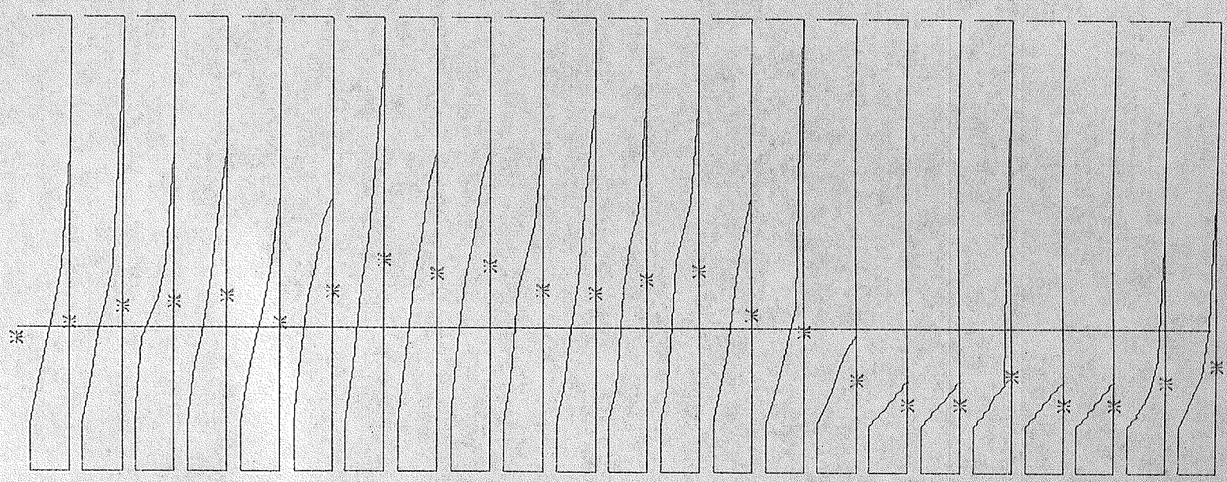
PLOT NUMBER 93

PLOT NUMBER 94

PLOT NUMBER 95

PLOT NUMBER 7

PLOT NUMBER 6

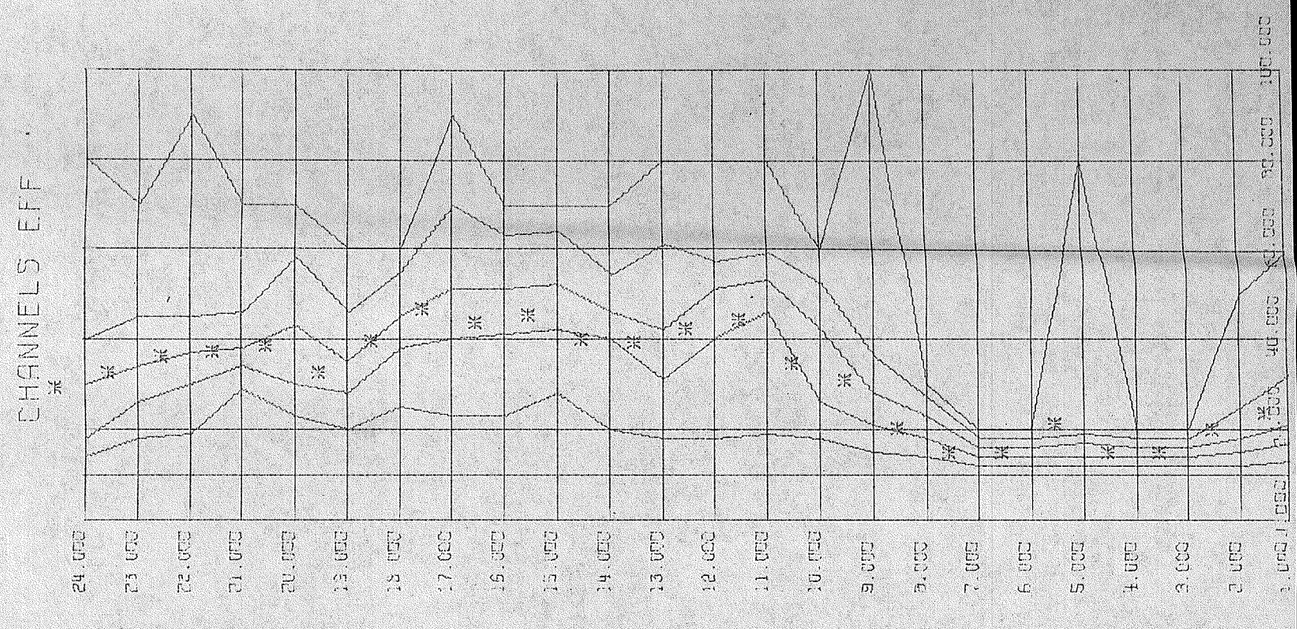


PLOT NUMBER 5

PLOT NUMBER 4

PLOT NUMBER 3

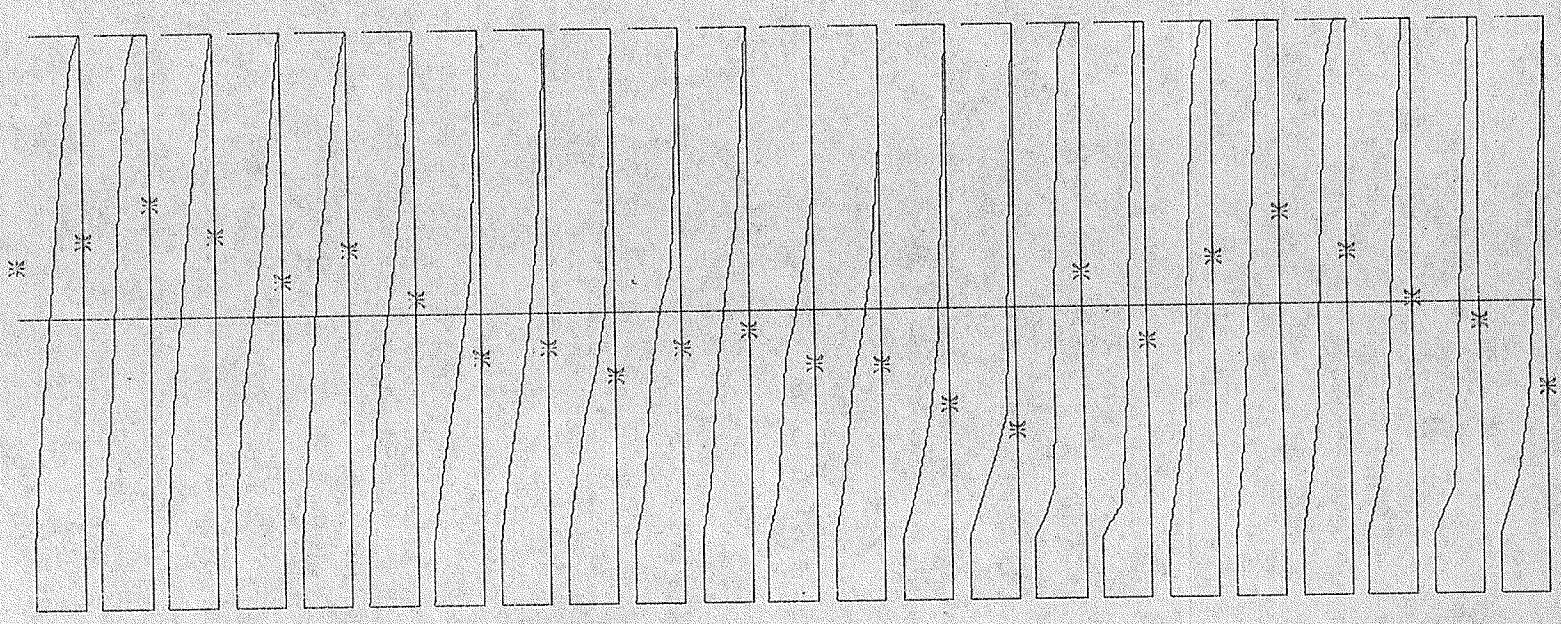
PLOT NUMBER 2



OCEAN NOV ..'70 PLOT NUMBER 1

PLOT NUMBER 15

PLOT NUMBER 14

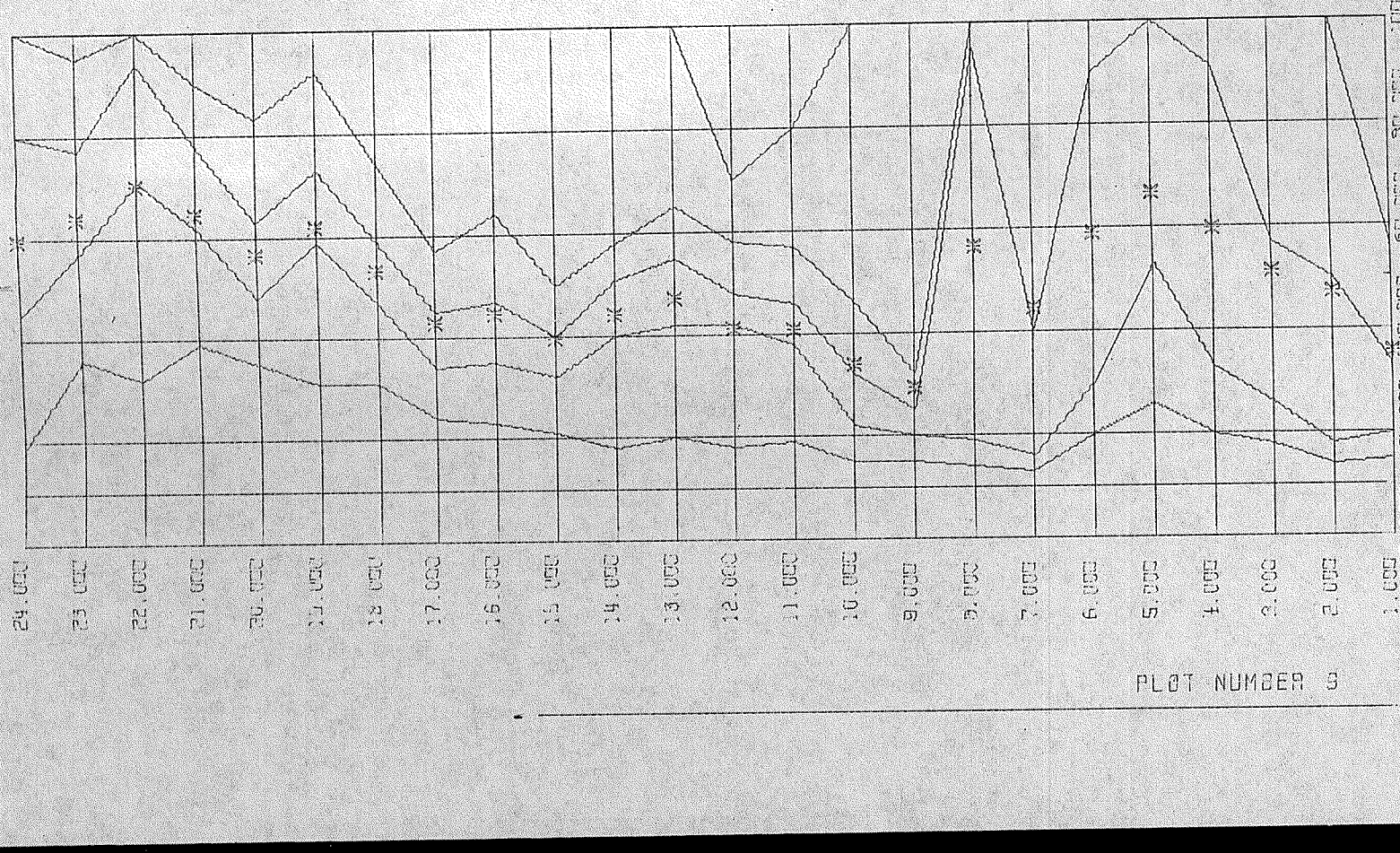


PLOT NUMBER 13

PLOT NUMBER 12

PLOT NUMBER 11

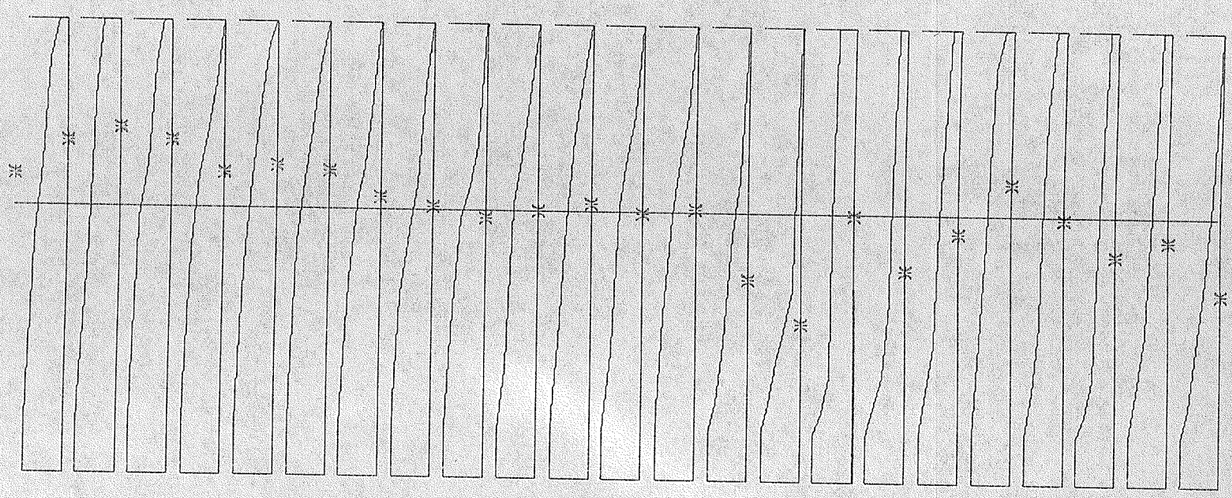
PLOT NUMBER 10



PLOT NUMBER 9

PLOT NUMBER 23

PLOT NUMBER 22

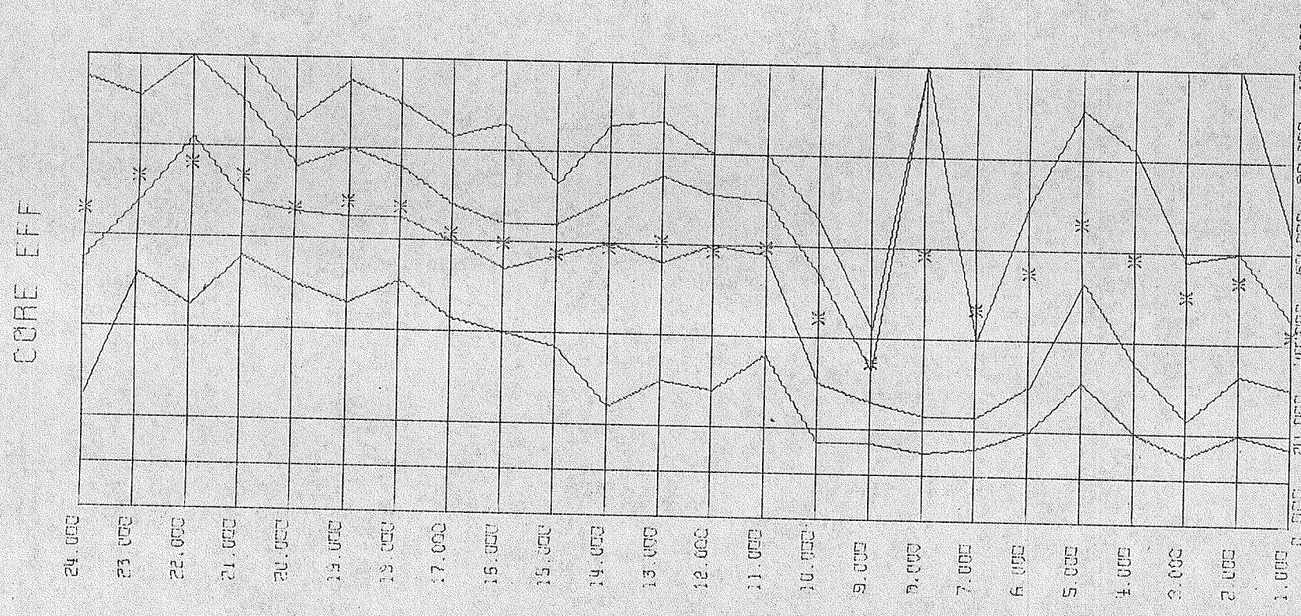


PLOT NUMBER 21

PLOT NUMBER 20

PLOT NUMBER 19

PLOT NUMBER 18



PLOT NUMBER 17

SYSTEM EFF

PLOT NUMBER 24

NON - '70

PLOT NUMBER 25

PLOT NUMBER 26

PLOT NUMBER 27

PLOT NUMBER 28

PLOT NUMBER 29

PLOT NUMBER 30

APPENDIX 5 -- Flow Chart and Program WASTE.

FUNCTION.

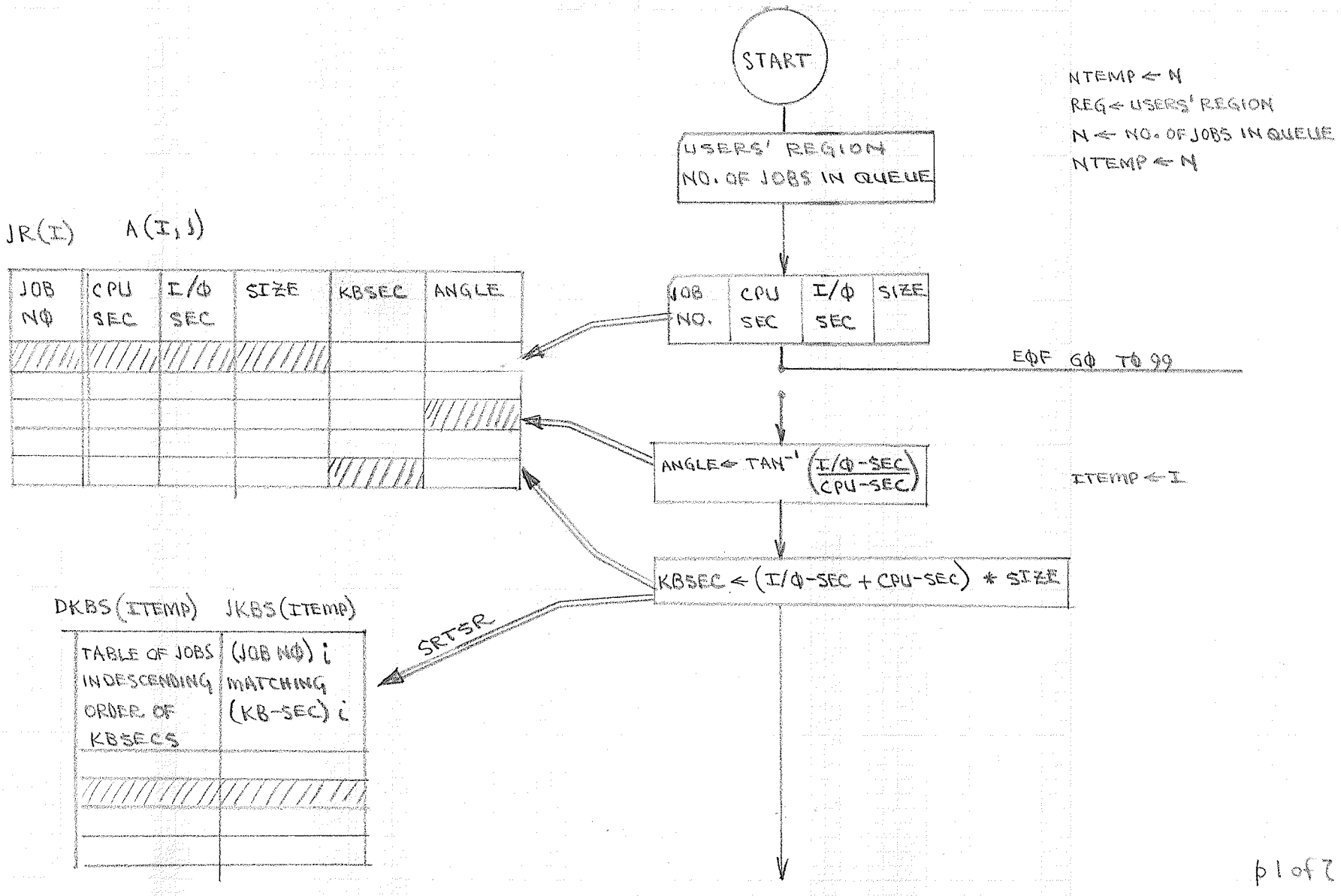
To examine the forecast characteristics of all jobs in the input queue and to determine how the jobs should be best matched, at execution time, to minimize waste of computer resources.

DATA.

Job cards preceding each job submitted to the input queue.

REFERENCE.

The method is shown on the flow chart immediately following. A tutorial is submitted in Chapter 4 of the text, pp. 71-122.

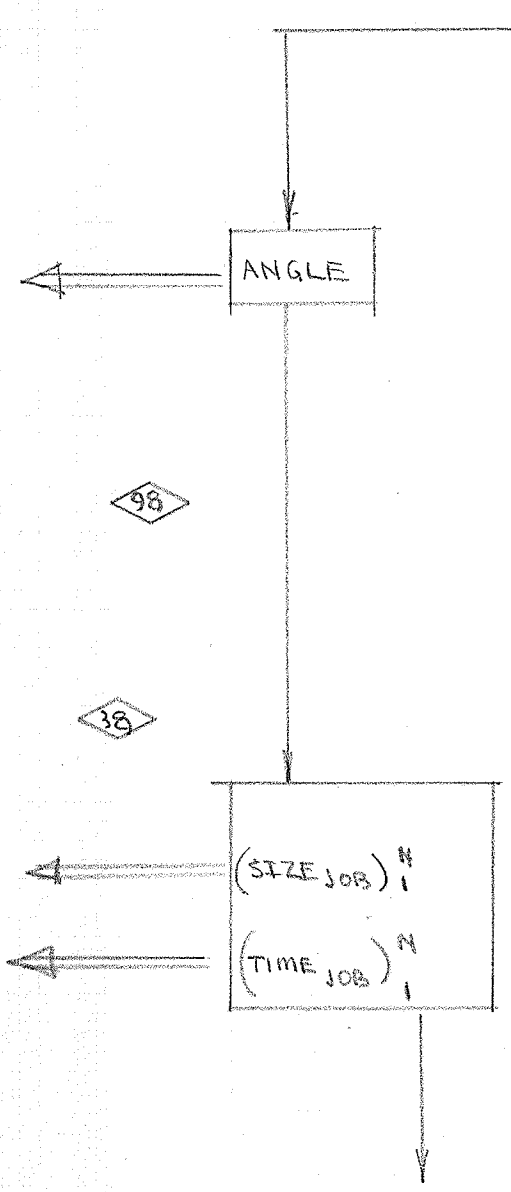


p 1 of 2

FLOWCHART OF PROGRAM 'WASTE'

DTHETA(ITEMP) JTHETA(ITEMP)

TABLE OF ANGLE IN DESCENDING ORDER	JOB NO. MATCHING ANGLE



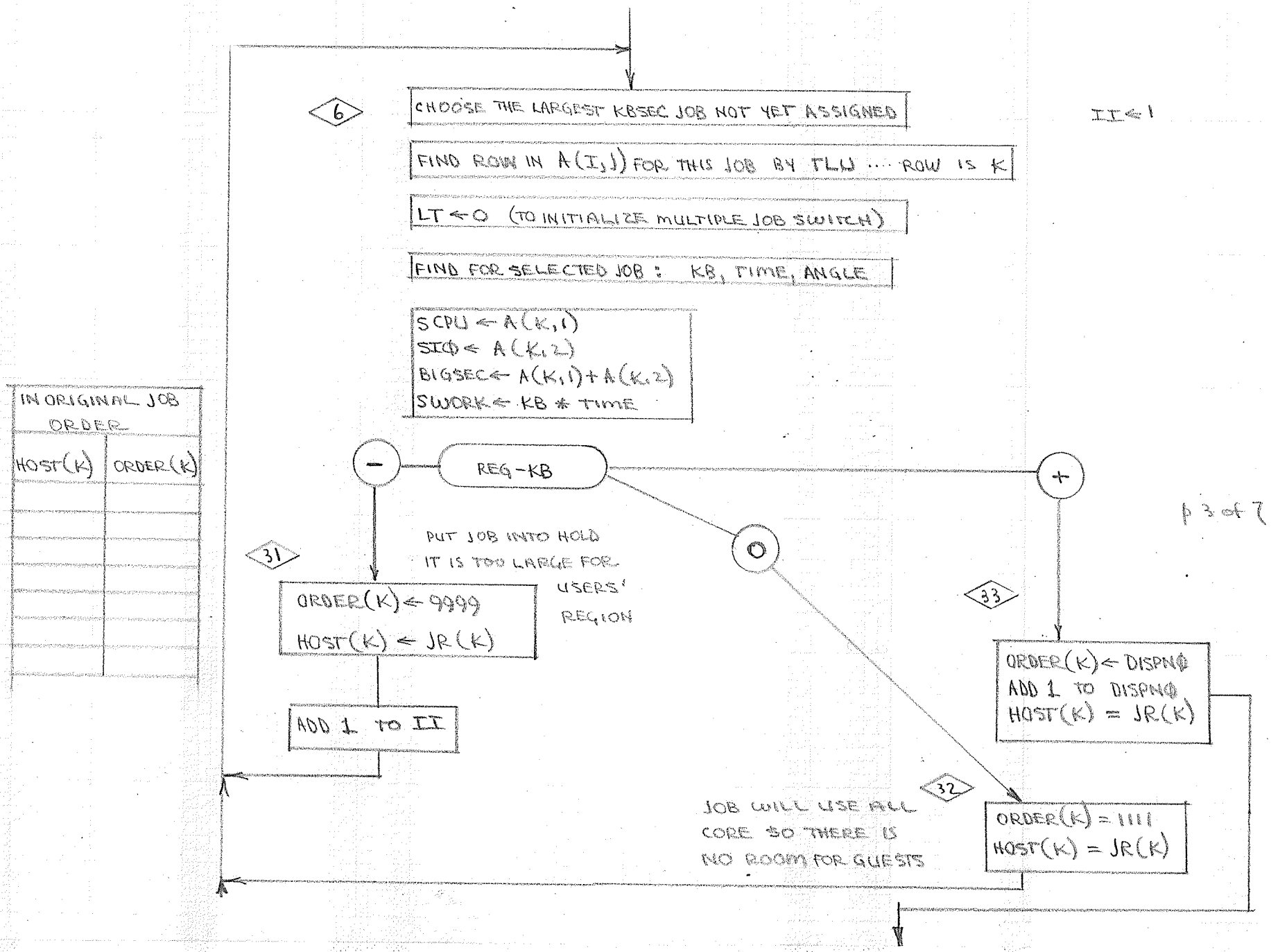
ITEMP ← I

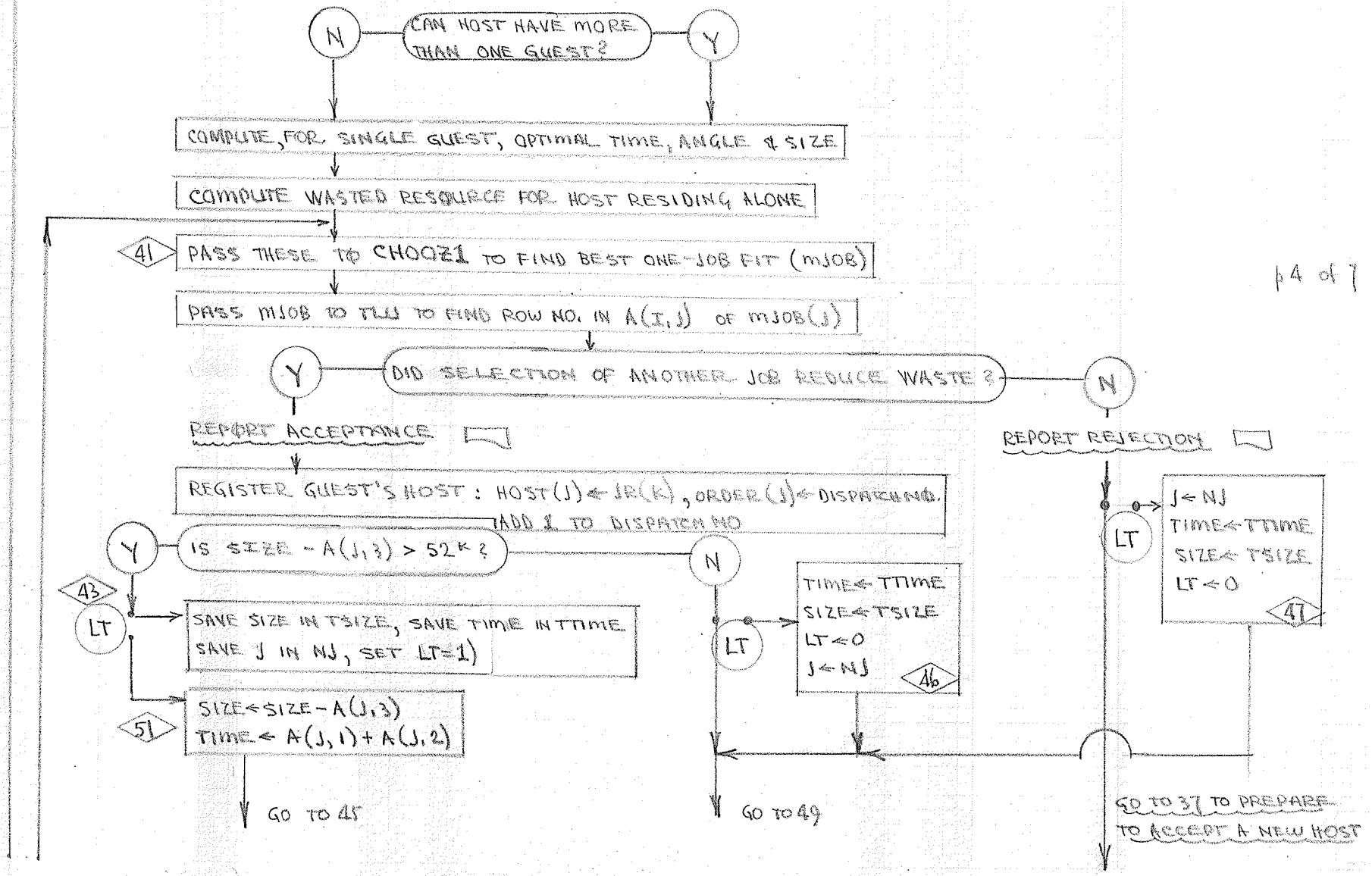
I ← N

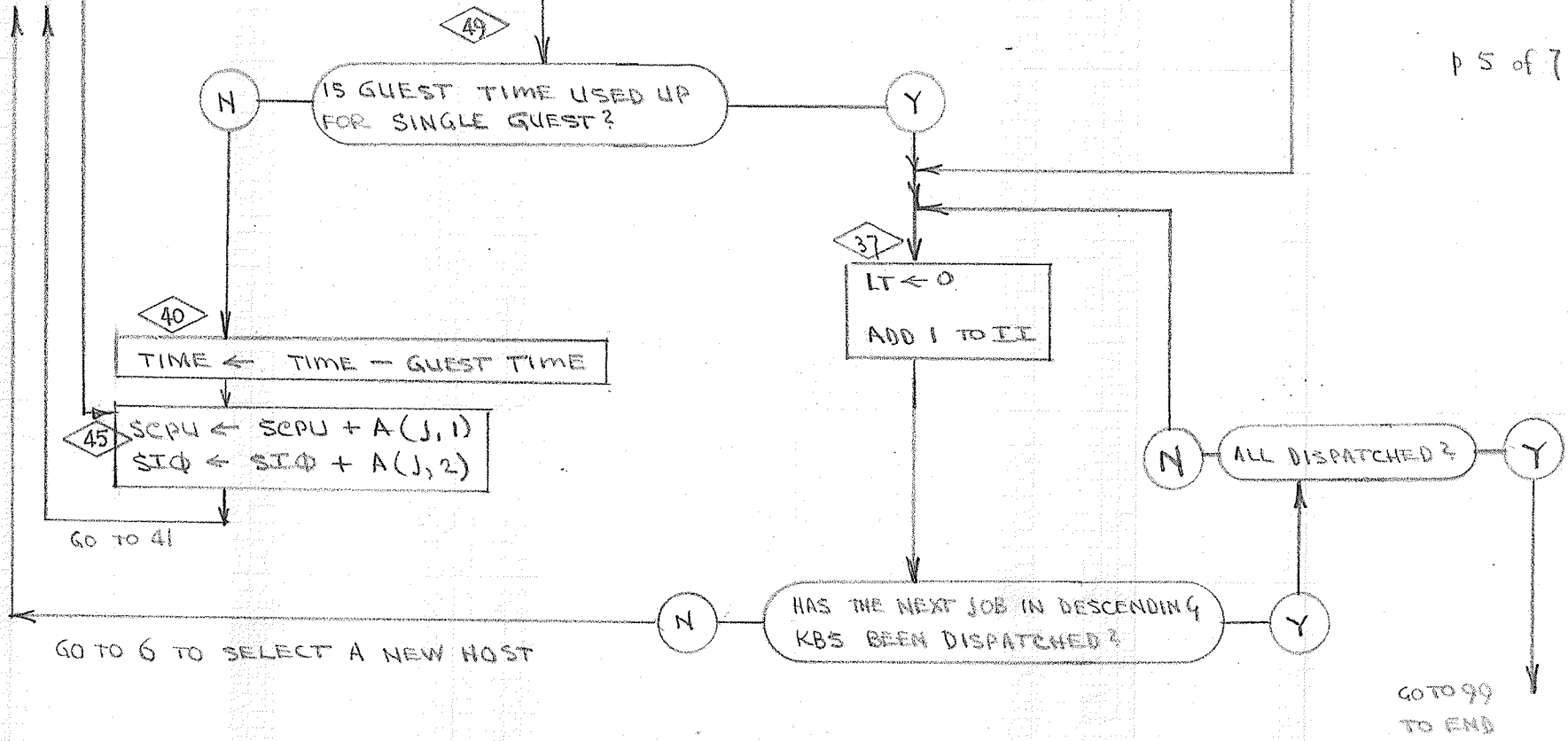
I ← ITEMP

RTHETA(L) TTHETA(L)

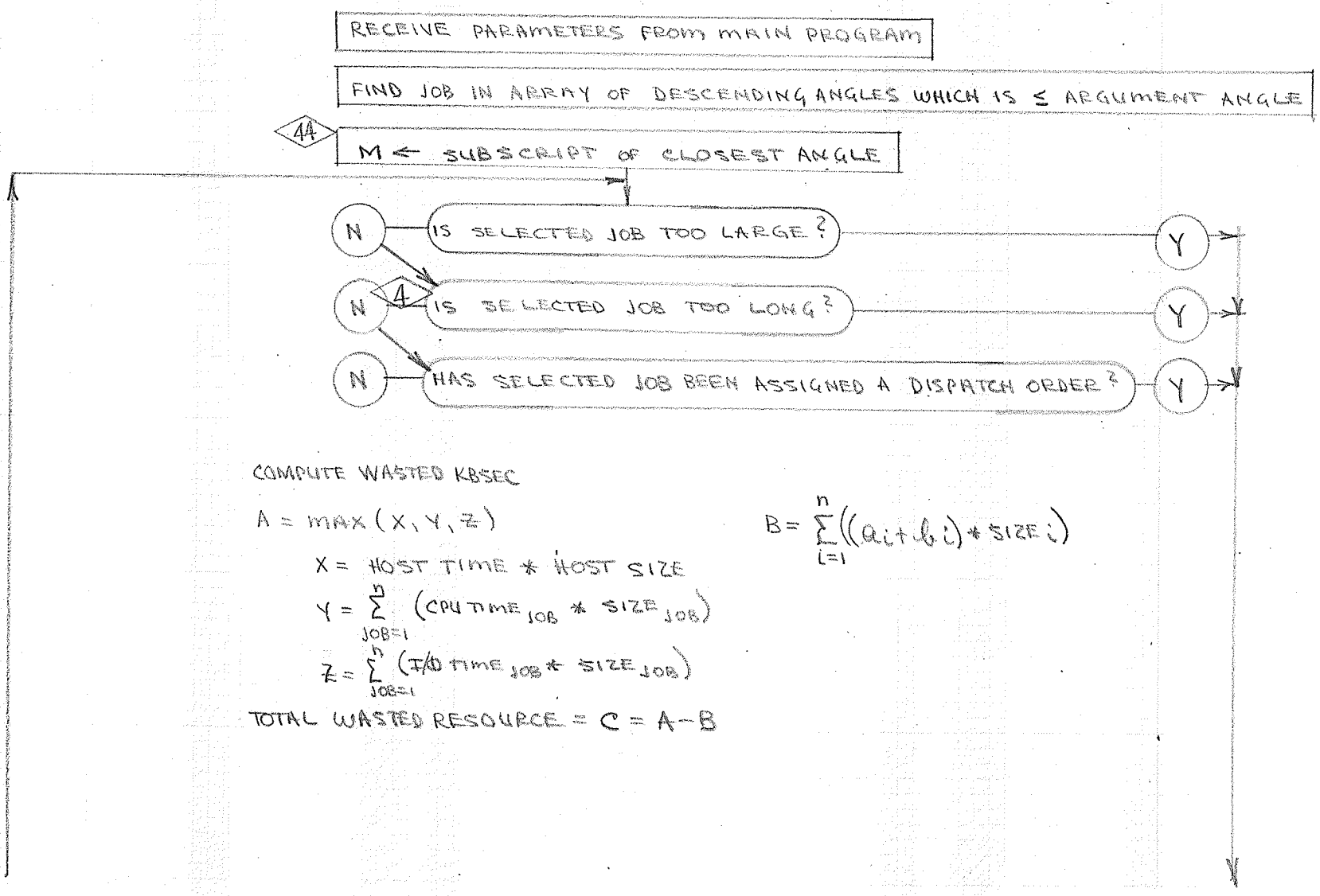
TABLE OF JOB REGION IN DESCENDING ORDER OF Θ	TABLE OF JOB TIME IN DESCENDING ORDER OF Θ


6
CHOOSE THE LARGEST KBSEC JOB NOT YET ASSIGNED
II ← 1
FIND ROW IN A(I,J) FOR THIS JOB BY TLU ··· ROW IS K
LT ← 0 (TO INITIALIZE MULTIPLE JOB SWITCH)
FIND FOR SELECTED JOB : KB, TIME, ANGLE
SCPU ← A(K,1)
SIO ← A(K,2)
BIGSEC ← A(K,1) + A(K,2)
SWORK ← KB * TIME
IN ORIGINAL JOB ORDER
HOST(K)
ORDER(K)
REQ-KB
PUT JOB INTO HOLD
IT IS TOO LARGE FOR USERS' REGION
31
ORDER(K) ← 9999
HOST(K) ← JR(K)
ADD 1 TO II
33
ORDER(K) ← DISPNO
ADD 1 TO DISPNO
HOST(K) = JR(K)
32
JOB WILL USE ALL CORE SO THERE IS NO ROOM FOR GUESTS
ORDER(K) = 1111
HOST(K) = JR(K)


N
CAN HOST HAVE MORE THAN ONE GUEST?
Y
COMPUTE, FOR SINGLE GUEST, OPTIMAL TIME, ANGLE & SIZE
COMPUTE WASTED RESOURCE FOR HOST RESIDING ALONE
41
PASS THESE TO CHOOZ1 TO FIND BEST ONE-JOB FIT (mJOB)
PASS MJOB TO TLU TO FIND ROW NO. IN A(I,J) OF mJOB(J)
Y
DID SELECTION OF ANOTHER JOB REDUCE WASTE?
N
REPORT ACCEPTANCE
REPORT REJECTION
REGISTER GUEST'S HOST: HOST(J)←IR(K), ORDER(J)←DISPATCH NO.
ADD 1 TO DISPATCH NO
Y
IS SIZE - A(J,3) > 52K?
N
43
LT
SAVE SIZE IN TSIZE, SAVE TIME IN TTIME
SAVE J IN NJ, SET LT=1)
51
SIZE←SIZE-A(J,3)
TIME←A(J,1)+A(J,2)
GO TO 45
LT
TIME←TTIME
SIZE←TSIZE
LT←0
J←NJ
46
GO TO 49
LT
J←NJ
TIME←TTIME
SIZE←TSIZE
LT←0
47
GO TO 37 TO PREPARE TO ACCEPT A NEW HOST


49
IS QUEST TIME USED UP FOR SINGLE QUEST?
N
Y
37
LT ← 0
ADD 1 TO II
40
TIME ← TIME − QUEST TIME
45
SCPU ← SCPU + A(J,1)
SIO ← SIO + A(J,2)
GO TO 41
N
ALL DISPATCHED?
Y
N
HAS THE NEXT JOB IN DESCENDING KBS BEEN DISPATCHED?
Y
GO TO 6 TO SELECT A NEW HOST
GO TO 99 TO END

SUBROUTINE CHØØZ1



COMPUTE WASTED KBSEC

$A = \max(X, Y, Z)$

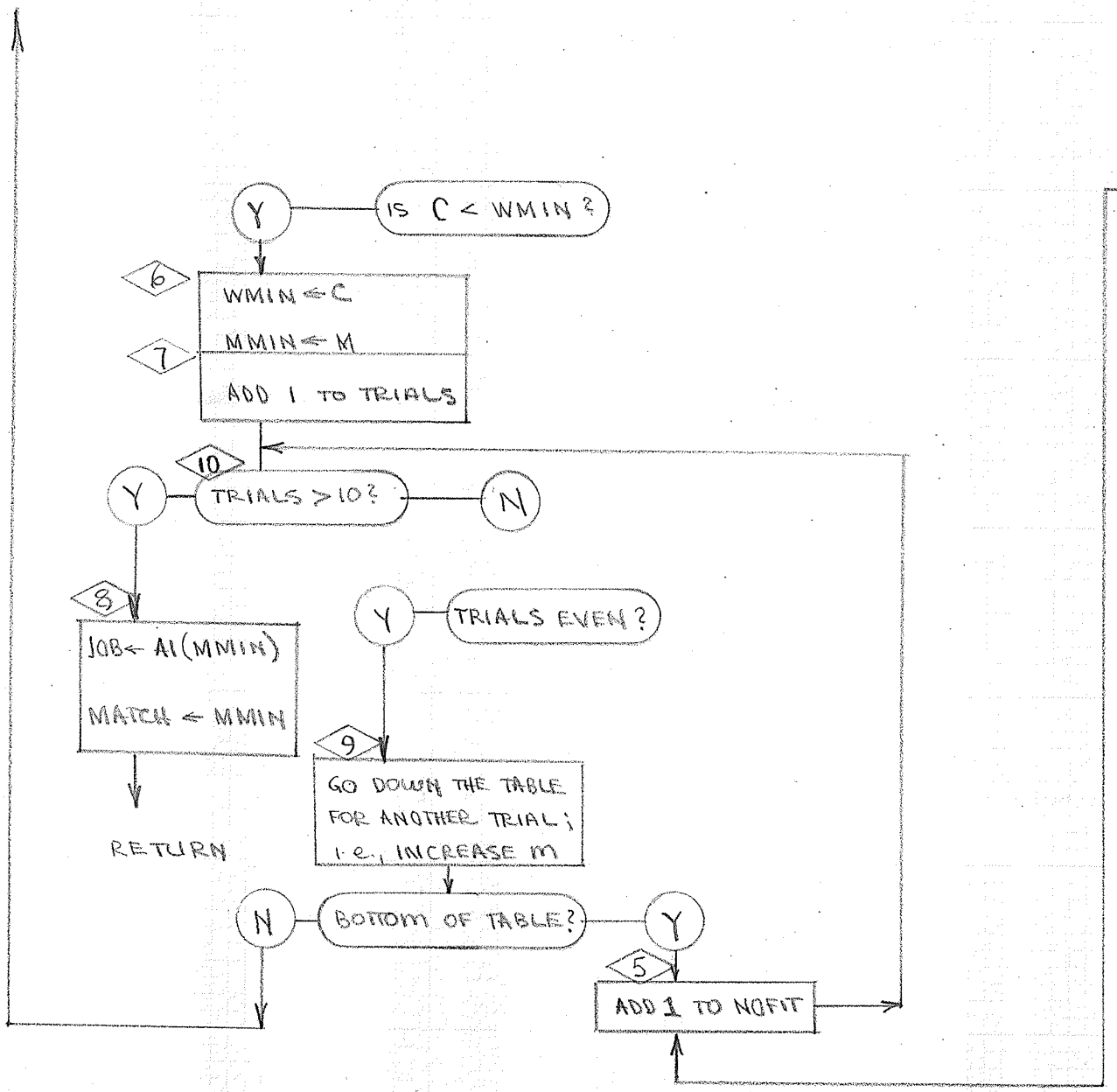
$X =$ HOST TIME * HOST SIZE

$Y = \sum_{JOB=1}^{n} (\text{CPU TIME}_{JOB} * \text{SIZE}_{JOB})$

$Z = \sum_{JOB=1}^{n} (\text{I/O TIME}_{JOB} * \text{SIZE}_{JOB})$

TOTAL WASTED RESOURCE $= C = A - B$

$$B = \sum_{i=1}^{n} ((a_i + b_i) * \text{SIZE}_i)$$


Y
IS C < WMIN ?
6
WMIN ← C
7
MMIN ← M
ADD 1 TO TRIALS
10
Y
TRIALS > 10 ?
N
8
JOB ← A1(MMIN)
MATCH ← MMIN
RETURN
Y
TRIALS EVEN ?
9
GO DOWN THE TABLE FOR ANOTHER TRIAL; i.e., INCREASE m
N
BOTTOM OF TABLE?
Y
5
ADD 1 TO NOFIT

```
$JOB              DUMAN,TIME=10,PAGES=30
      INTEGER SUBSCR,                              DISPNO/1/,JKBS(100),
     1JP(100),JTHETA(100),HOST(100),OPDEF(100)/100*0/,ERROR
      REAL A(100,5),          DTHETA(100)/100*0./,DKBS(100)/100*0./,
     1  TTHETA(100), RTHETA(100),CUMWST(100)/100*0./
      REG=270.
      N=100
      READ(5,101)REG,N
  101   FORMAT (F10.0,I5,65X)
      NTEMP=N
      DO 27 I=1,NTEMP
C*************************************************************
C       READ JOBNO,CPU TIME, I/O TIME, KB CORE REQ'D
C*************************************************************
        READ(5,102,END=98) JP(I),(A(I,J),J=1,3)
  102  FORMAT(I5,2X,2F10.2,F10.0)
C*************************************************************
C         COMPUTE KBSECS
C*************************************************************
        A(I,4)=(A(I,1)+A(I,2))*A(I,3)
C*************************************************************
C         STORE KBSEC IN DESCENDING ORDERED ARRAY DKBS(WASTE) &
C          REORDER ATTENDING JOB NOS IN JKBS
C*************************************************************
      ITEMP=I
      CALL SPTSTR(A(I,4),DKBS,JP(I),JKBS,ITEMP)
C*************************************************************
C         COMPUTE JOB VECTOR ANGLE & STORE IN A(I,5)
C*************************************************************
        A(I,5)=ATAN(A(I,2)/A(I,1))
C*************************************************************
C         STORE ANGLE IN DESCENDING ORDERED ARRAY STHETA &
C         REORDER ATTENDING JOB NO IN JTHETA.
C*************************************************************
      ITEMP=I
C.....................................................................
C      WRITE(6,118)A(I,5),(DTHETA(K),K=1,N),JP(I),
C     1    (JTHETA(K),K=1,N),I
C 118 FORMAT(1H0,13F10.7/11X,12F10.7/11X,3F10.7/I5,6X,
C     1  12I10/11X,12I10/11X,3I10/' I=  '/I5)
C.....................................................................
       CALL SRTSTR.(A(I,5),DTHETA,JP(I),JTHETA,ITEMP)
   27  CONTINUE
98     CONTINUE
      I = N
      WRITE(6,118)A(I,5),(DTHETA(K),K=1,N),JP(I),
     1    (JTHETA(K),K=1,N),I
  118 FORMAT(1H0,13F10.7/11X,12F10.7/11X,3F10.7/I5,6X,
     1  12I10/11X,12I10/11X,3I10/' I=  '/I5)
C*************************************************************
C         BUILD ARRAYS RTHETA,TTHETA
C*************************************************************
      ITEMP = I
      DO 38 L=1,ITEMP
C*************************************************************
C         FIND JR SUBSCRIPT FOR JTHETA(L)
C*************************************************************
      CALL TLU(JTHETA(L),  JP, ITEMP, J, ERROR)
       RTHETA(L)=A(J,3)
   38  TTHETA(L)=A(J,1)+A(J,2)
```

SUPPLEMENT 6 APPENDIX 5

program WASTE forms family groups for minimum waste

```
      WRITE(6,114)
  114 FORMAT(1H1,'JOB NO',T12,'REGION',T22,'SECONDS',
     1T32,'RADIANS')
      DO 39 L=1,I
   39 WRITE(6,119)JTHETA(L),RTHETA(L),TTHETA(L),DTHETA(L)
  119 FORMAT (1H , I6,F10.1,F10.2, F10.5)
C*****************************************************************
C       PRINT DATA ARRAYS
C*****************************************************************
      WRITE(6,103)
  103 FORMAT(1H1,'JOB NO.',T17,'CPU-SEC',T26,'CHAN-SEC',
     1T37,'CORE-KB',T47,'KBSEC',T57,'THETA')
      ITEMP=I
      DO 2 J=1,ITEMP
    2 WRITE(6,104)JR(J),(A(J,K),K=1,5)
  104 FORMAT(1H ,I6,4X,5F10.2)
      WRITE(6,105)
  105 FORMAT(1H1,'JOB NO',T16,'KB-SEC')
      ITEMP=I
      DO 4 J=1,ITEMP
    4 WRITE(6,104) JKBS(J),DKBS(J)
      WRITE(6,107)
  107 FORMAT(1H1,'JOB NO',T17,'THETA')
      ITEMP=I
      DO 5 J=1,ITEMP
    5 WRITE(6,104) JTHETA(J),DTHETA(J)
C*****************************************************************
C           ASSIGN A DISPATCH CODE TO ALL JOBS. I.E.,
C           GROUP JOBS FOR CO-HABITATION
C*****************************************************************
      II=1
    6 CONTINUE
C*****************************************************************
C           FROM LIST OF JOBS NOT YET ASSIGNED A DISPATCH
C           CODE CHOOSE THE ONE WITH THE LARGEST KBSEC
C           REQUIREMENT: I.E., THE JOB HIGHEST IN THE
C           DKBS LIST. ITS CORE REQUIREMENT MAY BE
C           > USERS' REGION. IF SO,PUT IT INTO A HOLD
C           LIST. FIND ARRAY'A' ROW 'K' FOR SELECTED 'KBSEC'
C       IF JKBS(II
C*****************************************************************
      CALL TLU(JKBS(II),JR,I,K,ERROR)
      LT=0
      BIGSEC = A(K,1) + A(K,2)
      SCPU = A(K,1)
      SIO = A(K,2)
      SWORK = BIGSEC * A(K,3)
      WMIN = 1.E20
C*****************************************************************
C       IF(ERROR.EQ.0) GO TO 30
C           K IS SUBSCRIPT OF 'A' ARRAY FOR MASTER JOB
C           FIND KB REQ'D OF JOB (A(K,3))
C           FIND TIME REQ'D OF JOB; A+B;(A(K,1)+A(K,2))
C           FIND ANGLE OF JOB (A(K,5))
C*****************************************************************
      IF(REG-A(K,3))31,32,33
C*****************************************************************
C   31      PUT JOB K INTO HOLD; TOO BIG FOR USERS'REGION
C*****************************************************************
   31 ORDER(K)=99999
```

```
      WRITE (6,120)  JR(K), A(K,3), REG
  120 FORMAT(1HO,'JOB NO.  ',I5,'   OF SIZE',F10.0,'  TOO BIG FOR REG',
     1    F10.5)
   34  HOST(K)=JR(K)
       CUMWST(K) = (REG - A(K,3)) * (A(K,1) + A(K,2))
        II=II+1
        GO TO 6
C*****************************************************************
C   32       JOB REQUIRES ALL OF USERS' REGION, SO IT
C            MUST STAND ALONE. IT'S EFFICIENCY IS NOT
C        .   A FUNCTION OF DISPATCH ORDER
C*****************************************************************
   32  ORDER(K)=11111
       HOST(K) = JR(K)
       WRITE (6,121)  JR(K), A(K,3), REG
  121 FORMAT(1HO,'JOB NO.',I5,'   OF SIZE',F10.0,'  USES ALL REG',
     1  F10.0)
       GO TO 34
C*****************************************************************
C   30       JOB JKBS(II) WAS NOT FOUND IN JR
C*****************************************************************
   30  WRITE(6,108) JKBS(II)
  108  FORMAT(1H-,'JOB NO.',I8       ,'NOT IN JOB LIST')
       II=II+1
       GO TO 6
   33  CONTINUE
       HOST(K)=JR(K)
       CUMWST(K) = (REG - A(K,3)) * (A(K,1) + A(K,2))
       ORDER(K)=DISPNO
       DISPNO=DISPNO+1
C*****************************************************************
C            IF KB REQ'D IS WITHIN 52K OF USERS' REGION
C            BY EXTANT OSMVT RULES IT MUST STAND ALONE
C*****************************************************************
      IF(REG-52.-A(K,3)) 32,36,36
C*****************************************************************
C            IF KB REQ'D IS WITHIN 104K OF USERS' REGION
C            BY EXTANT OSMVT RULES ONLY ONE JOB WILL CO-HABIT
C*****************************************************************
   36 IF((REG-104.).GT.A(K,3)) GO TO 44
C*****************************************************************
C            ONLY ONE JOB CAN COHABIT WITH JOB JR(K) AT
C            ANY GIVEN MOMENT BECAUSE OF SPACE RESTRICTIONS
C            FIND THE BEST MATCH. IT WILL HAVE REG-A(K,3)
C            KB, PI/2-A(K,5) ANGLE & REQUIRE NOT
C            MORE THAN (A(K,1)+A(K,2))TIME.
C            CHOOSE A LIKELY JOB,CHECK WASTE.CHOOSE
C            ANOTHER; COMPARE
C*****************************************************************
   44 TIME1=A(K,1)+A(K,2)
       TIME=TIME1
       WASTE=TIME*(REG-A(K,3))
       CUMWST(K) = WASTE
       ANGLE=1.5708-A(K,5)
       SIZE=REG-A(K,3)
C*****************************************************************
C            SET UP HEADERS FOR REPORT OF TRIALS
C*****************************************************************
       WRITE(6,109)REG
  109  FORMAT(1H-,6X,'REG=',F5.0)
```

```
      WRITE(6,110)
  110 FORMAT(1H1,T2,'TRIAL',T17,'JOB NO',T27,'ANGLE',
     1T38,'SIZE',T48,'TIME',T57,'WASTE',T77,'ANGLE',
     2T88,'SIZE',T98,'TIME',T112,'SUBSCRIPT')
C*****************************************************************
C          REPORT HOST JOB CHARACTERISTICS.
C*****************************************************************
      WRITE(6,111)JR(K),A(K,5),A(K,3),TIME1,WASTE,ANGLE,SIZE,
     1TIME,M
  111 FORMAT(1H-,3X,'0',10X,I6,F10.3,2F10.2,F10.0,7X,'OPT',
     1F10.3,2F10.2,10X,I6)
      WRITE(6,112)
  112 FORMAT(1HO,T26,'A',T36,'B')
      WRITE(6,113)
  113 FORMAT(1H+,T26,'A',T36,'B',/)
C*****************************************************************
C          GO TO CHOOZ1 TO FIND BEST FIT
C*****************************************************************
   41 CONTINUE
      CALL CHOOZ1(REG,I,ANGLE,SIZE,TIME,JTHETA,RTHETA,
     . 1 TTHETA,DTHETA,JR,ORDER,BIGSEC,SCPU,SIC,SWORK,TRIALC,SUBSCR,MJOB,
     2 WMIN)
      IF (TRIALC.GE.WASTE) GO TO 42
      WASTE = TRIALC
C*****************************************************************
C          STORE HOST NO IN HOST(+MJOB)& ASSIGN DISPATCH
C          ORDER NO TO ORDER(+HOST)& ORDER(+MJOB)
C          FIND ROW NO OF MJOB
C*****************************************************************
      CALL TLU(MJOB,JR,I,J,ERROR)
C*****************************************************************
C     IF(ERROR.EQ.0) GO TO 30
C*****************************************************************
      WRITE (6,124) JR(J), JR(K), WASTE
  124 FORMAT (1HO,'JOB NO',I4,' IS ACCEPTED FOR DISPATCH WITH JOB NO',
     1  I4,' WASTE NOW REDUCED TO',F10.0,' KBS')
      HOST(J)=JR(K)
      ORDER(J)=DISPNO
      CUMWST(J) = WASTE
      DISPNO=DISPNO+1
      WRITE (6,122)
     1 (JTHETA(L),L = 1,12), (RTHETA(L), L=1,12), (TTHETA(L),L=1,12),
     2 (DTHETA(L),L=1,12), (JR(L), L=1,12),(ORDER(L),L=1,12),SUBSCR,MJOB
  122 FORMAT
     1                    ( '-JTHETA',12I10/ ' RTHETA', 12F10.2/
     2 ' TTHETA', 12F10.4/' DTHETA', 12F10.4/' JR    ', 12I10/
     3 ' ORDER ', 12I10/ ' SUBSC=', I10, ' MJOB=', I10)
      WRITE (6,122)
     1 (JTHETA(L),L =13,24), (RTHETA(L),L=13,24), (TTHETA(L),L=13,24),
     2 (DTHETA(L),L=13,24),(JR(L),L=13,24),
     3  (ORDER(L),L=13,24),SUBSCR,MJOB
C*****************************************************************
C          DELETE DISPATCHED RECORDS FROM THETA TABLES
C          REDUCE THETA TABLE I BY NO. DISPATCHED
C          IF JOBS ARE NOT ALL DISPATCHED GO BACK TO 6
C          GUEST TIME MAY NOT BE ALL USED UP
C*****************************************************************
      IF((SIZE-A(J,3)).GT.52.) GO TO 43
      IF(LT.EQ.1) GO TO 46
   49 IF(TIME-A(J,1)-A(J,2))37,37,40
```

```
   40   TIME=TIME-A(J,1)-A(J,2)
   45 SCPU=SCPU+A(J,1)
      SIO = SIO + A(J,2)
      SWORK = SWORK + (A(J,1) + A(J,2)) * A(J,3)
      WRITE(6,123) ANGLE, SIZE, TIME, M
  123 FORMAT (1H-,65X,'OPT', 3F10.2,I6)
       GO TO 41
   42 WRITE(6,125) MJOB, JR(K)
  125 FORMAT (1HO,'NO OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB',
     1  I4)
      IF(LT.EQ.1) GO TO 47
      GO TO 37
   43 IF(LT.EQ.0) GO TO 50
   51 SIZE=SIZE-A(J,3)
      TIME=A(J,1)+A(J,2)
      GO TO 45
   50 TSIZE=SIZE
      TTIME=TIME
      NJ=J
      LT=1
      GO TO 51
   47 J=NJ
      TIME=TTIME
      SIZE=TSIZE
      LT = 0
      GO TO 49
   46 TIME=TTIME
      SIZE=TSIZE
      LT=0
      J=NJ
      GO TO 48
   37  II=II+1
      LT=0
      IF (II.GT.I)  GO TO 99
C*************************************************************************
C           IS JKBS(II)DISPATCHED?
C*************************************************************************
       CALL TLU(JKBS(II),JR,I,J,ERROR)
C*************************************************************************
C       IF ERROR =0 GO TO 30
C*************************************************************************
       IF(ORDER(J).NE.0) GO TO 37
       IF(I-II)99,99,6
    99  WRITE(6,115)
  115  FORMAT(1H1,'JOB NO',T10,'HOST',T27,'ORDER',T33,'CUM WASTE')
      ITEMP=I
      DO 52 L=1,ITEMP
      LL = L
      DO 58 N = 1,LL
      IF (N.EQ.L) GO TO 57
      IF (HOST(N). EQ .HOST(L)) GO TO 52
   57 M = HOST(L)
      KK = LL
   58 CONTINUE
      DO 59 K = KK,ITEMP
      IF(HOST(K).EQ.M) GO TO 117
      GO TO 59
  126 FORMAT (1H ,9I4)
  117 WRITE(6,116) JR(K),HOST(K),ORDER(K),CUMWST(K)
   59 CONTINUE
```

```
   52   CONTINUE
  116   FORMAT(1H0,I6,4X,2I10,F10.0)
        STOP
        END

        SUBROUTINE CHOOZ1(REG,I,ANGLE,SIZE,TIME,A1,A2,A3,A4,
       1JR,ORDER,BIGSEC,SCPU,SIO,SWORK,TRIALC,MATCH,JOB,WMIN)
C*****************************************************************
C            GIVEN USERS' REGION, ENTRY COUNT,
C                  REST ANGLE SIZE&TIME, SELECT 'MATCH' FROM
C            ARRAYS OF JOBNO,JOBKB,JOBMIN,JOBANGLE,ALL ORDERED
C            ON DESCENDING JOBANGLE. 'I' IS ENTRIES COUNT
C*****************************************************************
        INTEGER JR(I),ORDER(I),A1(I),ERROR,END,TRIALS,NOFIT,TOP,BTM
        REAL A2(I),A3(I),A4(I),WMIN
        END = 0
        TRIALS = 1
        NOFIT = 0
        TOP = 0
        BTM = 0
        DO 2 MATCH=1,I
           IF(ANGLE.GE.A4(MATCH)) GO TO 44
    2   CONTINUE
        GO TO 3
   44   M=MATCH
        GO TO 4
C*****************************************************************
C    3       ANGLE IS < ANY THETA ENTRY. WORK UP FROM
C            BOTTOM OF TABLE
C*****************************************************************
    3   END=1
        WRITE(6,200) ANGLE,SIZE,TIME
  200   FORMAT(1H-,'ANGLE ENTRY TO CHOOZ1 < ANY ENTRY',
       13F10.5)
        M = I
        GO TO 4
C*****************************************************************
C    4       IF END =0 A REASONABLE ANGLE MATCH HAS
C            BEEN FOUND. IF END=1, END OF TABLE WAS FOUND.
C*****************************************************************
    4   IF(A3(M).GT.TIME) GO TO 5
        IF(A2(M).GT.SIZE) GO TO 5
        CALL TLU(A1(M),JR,I,J,ERROR)
        IF(ORDER(J).NE.0) GO TO 5
C*****************************************************************
C           COMPUTE WASTE = 'C' KBSEC
C*****************************************************************
       TRCPU = A3(M) / (1 + TAN(A4(M)))
       TRIO = A3(M) - TRCPU
       A = AMAX1(BIGSEC, SCPU + TRCPU, SIO + TRIO) * REG
       B = SWORK + A3(M) * A2(M)
        C=A-B
       IF(C.LT.WMIN) GO TO 6
C*****************************************************************
C           REPORT RESULT OF THIS TRIAL EXCURSION.
C*****************************************************************
    7   WRITE(6,201)TRIALS,A1(M),A,B,C,A4(M),A2(M),A3(M),M
  201   FORMAT(1H0,I4,10X,I6,F10.0,F10.2,10X,F10.0,4X,'ACTUAL',3X,
       1F10.2,2F10.2,10X,I6)
C*****************************************************************
```

```
C           KEEP COUNT OF NO. OF EXCURSIONS
C*****************************************************************
      TRIALS=TRIALS+1
   10 IF(TRIALS.GT.10) GO TO 9
      IF(((TRIALS+NOFIT)/2)*2.EQ.(TRIALS+NOFIT))GO TO 9
C*****************************************************************
C           TRIALS IS ODD, GO UP THE TABLE NEXT TO
C           M=MATCH-(TRY/2)=MATCH-(TRIALS+NOFIT)/2
C*****************************************************************
      M=MATCH-(TRIALS+NOFIT)/2
      IF(M.LE.0) GO TO 12
      GO TO 4
   12 TOP = 1
      IF  (BTM.EQ.1) GO TO 8
      GO TO 5
    9 CONTINUE
C*****************************************************************
C           TRIALS IS EVEN,GO DOWN THE TABLE NEXT TO
C           M=MATCH+(TRY/2)=MATCH+((TRIALS+NOFIT)/2)
C*****************************************************************
      M=MATCH+((TRIALS+NOFIT)/2)
      IF(M.GT.I) GO TO 11
      GO TO 4
   11 BTM = 1
      IF (TOP.EQ.1) GO TO 8
      GO TO 5
    5 CONTINUE
C*****************************************************************
C           THIS TRIAL JOB IS TOO LARGE IN SOME RESPECT
C*****************************************************************
      NOFIT=NOFIT+1
      GO TO 10
C*****************************************************************
C           SAVE CHARACTERISTICS OF BEST TRIAL
C*****************************************************************
    6 WMIN=C
      MMIN=M
C*****************************************************************
C           RETURN FOR ANOTHER TRIAL
C*****************************************************************
      GO TO 7
    8 JOB=A1(MMIN)
      MATCH=MMIN
      TRIALC = WMIN
      RETURN
      END

      SUBROUTINE TLU(JOBNO,ARRAY,I,J,ERROR)
C*****************************************************************
C           GIVEN A 'JORNO',FIND ITS ROW NO,'J',IN'ARRAY'
C           OF LENGTH 'I'.IF NO MATCH ERROR WILL=0
C*****************************************************************
      INTEGER ARRAY(I),ERROR
      ERROR=1
      DO 1 J=1,I
      IF(JOBNO.GT.ARRAY(J))GO TO 1
      IF(JOBNO.EQ.ARRAY(J))GO TO 2
      ERROR=0
      GO TO 2
    1 CONTINUE
```

```
      2  RETURN
         END

         SUBROUTINE SRTSTR (ARG,ARRAY1,JOBNO,ARRAY2,I)
C*************************************************************************
C              STORE ARG IN DESCENDING ORDERIN ARRAY1 &
C              JOB NO IN DESCENDING ORDER IN ARRAY2
C              I IS TOTAL NO OF ARRAY ENTRIES INCLUDING THIS ONE
C*************************************************************************
         INTEGER ARRAY2(I)
         REAL ARRAY1(I)
C*************************************************************************
C              FIND SLOT & SLIDE ALL REMAINING ENTRIES DOWN ONE
C*************************************************************************
         DO 7 M=1,I
         IF(ARG.GT.ARRAY1(M))GO TO 3
      7  CONTINUE
      3  IF (ARRAY1(M) .EQ .0.) GO TO 9
C*************************************************************************
C              MOVE BOTTOM ELEMENT     DO 8 J=I-1,M-1,-1
C              DOWN ONE & FOLLOW WITH      ARRAY1(J+1)=ARRAY(J)
C              OTHERS TILL A SLOT          8    ARRAY2(J+1)=ARRAY(J)
C              IS LEFT FOR INSERTION
C*************************************************************************
         K = I-1
         DO 8 J = M,K
          IHI=I-(J-M)
          ILO=IHI-1
          ARRAY1(IHI)=ARRAY1(ILO)
      8   ARRAY2(IHI)=ARRAY2(ILO)
      9   ARRAY1(M)=ARG
          ARRAY2(M)=JOBNO
          RETURN
          END

$ENTRY

556  1.5208370  1.4711270  1.4056470  1.2120240  1.2120240  1.1071480  1.1071480
     0.5880024  0.5880025  0.5191461  0.4993465  0.4182243  0.3805063  0.3805063
     0.1973956  0.1973955  0.0166651
           29         30         40         31         39         38         50
           36         49         33         45         44         42         48
           54         28         51
```

REGION	SECONDS	RADIANS
60.0	2.10	1.52084
52.0	1.10	1.47113
66.0	0.70	1.40565
65.0	1.10	1.21202
95.0	1.10	1.21202
64.0	1.50	1.10715
76.0	0.30	1.10715
78.0	0.30	1.10715
105.0	0.70	0.92730
54.0	1.10	0.87606
85.0	1.50	0.85197
72.0	1.70	0.72664
58.0	1.50	0.59800
125.0	0.50	0.59800
62.0	1.10	0.51915
115.0	1.70	0.49935
70.0	1.30	0.41822
68.0	0.70	0.38051
74.0	0.70	0.38051
75.0	1.50	0.34877
80.0	0.40	0.32175
110.0	0.90	0.27830
120.0	0.90	0.27830
56.0	1.10	0.21067
135.0	0.60	0.19740
100.0	6.00	0.19740
130.0	6.10	0.01667

CPU-SEC	CHAR-SEC	CORE-KB	KBSEC	THETA
5.00	1.00	100.00	600.00	0.20
0.10	2.00	60.00	126.00	1.52
0.10	1.00	52.00	57.20	1.47
0.30	0.80	65.00	71.50	1.21
0.50	0.60	54.00	59.40	0.88
0.70	0.40	62.00	68.20	0.52
0.90	0.20	56.00	61.60	0.22
1.10	0.40	75.00	112.50	0.35
0.90	0.60	58.00	87.00	0.59
0.70	0.90	85.00	127.50	0.85
0.50	1.00	64.00	96.00	1.11
0.30	0.50	95.00	104.50	1.21
0.10	0.60	66.00	46.20	1.41
0.30	0.40	105.00	73.50	0.93
0.50	0.20	68.00	47.60	0.38
0.70	0.20	110.00	99.00	0.28
0.90	0.40	70.00	91.00	0.42
1.10	0.60	115.00	195.50	0.50
0.90	0.80	72.00	122.40	0.73
0.70	0.20	120.00	108.00	0.28
0.50	0.20	74.00	51.80	0.38
0.30	0.20	125.00	62.50	0.59
0.10	0.20	76.00	22.80	1.11
6.00	0.10	130.00	793.00	0.02
0.10	0.20	78.00	23.40	1.11
0.30	0.10	80.00	32.00	0.32
0.50	0.10	135.00	81.00	0.20

KB/SEC
703.00
400.00
195.50
127.50
126.00
122.40
112.50
108.00
104.50
99.00
96.00
91.00
87.00
81.00
73.50
71.50
68.20
62.50
61.60
59.40
57.20
51.80
47.60
46.20
32.00
23.40
22.80

1.52
1.47
1.41
1.21
1.21
1.11
1.11
1.11
0.93
0.88
0.85
0.73
0.59
0.59
0.52
0.50
0.42
0.38
0.30
0.35
0.32
0.28
0.28
0.22
0.20
0.20
0.02

EG= 300.

RIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SEQUENCE
0	51	0.017	130.00	6.10	1037.	OPT	1.554	170.00	6.10	UUUUUU
		A	B							
1	29	1830.	919.00		911.	ACTUAL	1.52	60.00	2.10	1
2	30	1830.	850.20		980.	ACTUAL	1.47	52.00	1.10	2
3	40	1830.	839.20		991.	ACTUAL	1.41	66.00	0.70	3
4	31	1890.	864.50		1026.	ACTUAL	1.21	65.00	1.10	4
5	39	1890.	897.50		993.	ACTUAL	1.21	95.00	1.10	5
6	38	1950.	889.00		1061.	ACTUAL	1.11	64.00	1.50	6
7	50	1830.	815.80		1014.	ACTUAL	1.11	76.00	0.30	7
8	52	1830.	816.40		1014.	ACTUAL	1.11	78.00	0.30	8
9	41	1890.	866.50		1024.	ACTUAL	0.93	105.00	0.70	9
10	32	1950.	852.40		1098.	ACTUAL	0.88	54.00	1.10	10

JOB NO 29 IS ACCEPTED FOR DISPATCH WITH JOB NO 51 WASTE NOW REDUCED TO 911. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	0	2	0	0	0	0	0	0	0	0	0	0

UBSC= 1 MJOB= 29

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	0	0	0	0	0	0	0	0	0	0	0	1

UBSC= 1 MJOB= 29

					OPT	1.55	110.00	2.10	UUUUUUU	
1	30	1860.	976.20		884.	ACTUAL	1.47	52.00	1.10	2
2	40	1860.	965.20		895.	ACTUAL	1.41	66.00	0.70	3
3	31	1920.	990.50		930.	ACTUAL	1.21	65.00	1.10	4
4	39	1920.	1023.50		897.	ACTUAL	1.21	95.00	1.10	5
5	38	1980.	1015.00		965.	ACTUAL	1.11	64.00	1.50	6
6	50	1860.	941.80		918.	ACTUAL	1.11	76.00	0.30	7
7	52	1860.	942.40		918.	ACTUAL	1.11	78.00	0.30	8
8	41	1920.	992.50		928.	ACTUAL	0.93	105.00	0.70	9
9	32	1980.	978.40		1002.	ACTUAL	0.88	54.00	1.10	10

10.	37	2040.	1048.50		968.	ACTUAL	0.89	85.00	1.50		11	

OB NO 30 IS ACCEPTED FOR DISPATCH WITH JOB NO 51 WASTE NOW REDUCED TO 884. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	0	2	3	0	0	0	0	0	0	0	0	0
UBSC=	2 MJOB=	30										
THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	0	0	0	0	0	0	0	0	0	0	0	1
UBSC=	2 MJOB=	30										

OPT 1.55 58.00 1.10UUUUUU

1	32	2010.	1035.60	974.	ACTUAL	0.88	54.00	1.10	10
2	34	2130.	1037.80	1092.	ACTUAL	0.22	56.00	1.10	24

0 OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 30

0 OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 51

OPT 1.55 170.00 4.00UUUUUU

1	40	1920.	1148.40	772.	ACTUAL	1.41	66.00	0.70	3
2	31	1980.	1173.70	806.	ACTUAL	1.21	65.00	1.10	4
3	39	1980.	1206.70	773.	ACTUAL	1.21	95.00	1.10	5
4	38	2040.	1198.20	842.	ACTUAL	1.11	64.00	1.50	6
5	50	1920.	1125.00	795.	ACTUAL	1.11	76.00	0.30	7
6	52	1920.	1125.60	794.	ACTUAL	1.11	78.00	0.30	8
7	41	1980.	1175.70	804.	ACTUAL	0.93	105.00	0.70	9
8	32	2040.	1161.60	878.	ACTUAL	0.88	54.00	1.10	10
9	37	2100.	1229.70	870.	ACTUAL	0.85	85.00	1.50	11
10	46	2160.	1224.60	935.	ACTUAL	0.73	72.00	1.70	12

OB NO 40 IS ACCEPTED FOR DISPATCH WITH JOB NO 51 WASTE NOW REDUCED TO 772. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	0	2	3	0	0	0	0	0	0	0	0	0
UBSC=	3 MJOB=	40										
THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187

| | | |

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 1

					OPT	1.55	170.00	3.00UUUUUU	
1	31	2250.	1312.90	937.	ACTUAL	1.21	65.00	1.10	4
2	39	2250.	1345.90	904.	ACTUAL	1.21	95.00	1.10	5
3	38	2310.	1337.40	973.	ACTUAL	1.11	64.00	1.50	6
4	50	2070.	1264.20	806.	ACTUAL	1.11	76.00	0.30	7
5	41	2130.	1314.90	815.	ACTUAL	0.93	105.00	0.70	9
6	32	2190.	1300.80	889.	ACTUAL	0.88	54.00	1.10	10
7	37	2250.	1368.90	881.	ACTUAL	0.85	85.00	1.50	11
8	46	2280.	1363.80	916.	ACTUAL	0.73	72.00	1.70	12
9	36	2280.	1328.40	952.	ACTUAL	0.59	58.00	1.50	13
10	49	2100.	1303.90	796.	ACTUAL	0.59	125.00	0.50	14

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 52

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 51
REG= 300.

RIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SUBSCRIPT
0	28	0.197	100.00	6.00	1200.	OPT	1.373	200.00	6.00	UUUUUU
		A	B							
1	31	1800.	671.50		1129.	ACTUAL	1.21	65.00	1.10	4
2	39	1800.	704.50		1096.	ACTUAL	1.21	95.00	1.10	5
3	38	1800.	696.00		1104.	ACTUAL	1.11	64.00	1.50	6
4	50	1800.	622.80		1177.	ACTUAL	1.11	76.00	0.30	7
5	41	1800.	673.50		1127.	ACTUAL	0.93	105.00	0.70	9
6	32	1800.	659.40		1141.	ACTUAL	0.88	54.00	1.10	10
7	37	1800.	727.50		1073.	ACTUAL	0.85	85.00	1.50	11
8	46	1800.	722.40		1078.	ACTUAL	0.73	72.00	1.70	12
9	36	1800.	687.00		1113.	ACTUAL	0.59	58.00	1.50	13
10	49	1800.	662.50		1138.	ACTUAL	0.59	125.00	0.50	14

OB NO 37 IS ACCEPTED FOR DISPATCH WITH JOB NO 28 WASTE NOW REDUCED TO 1073. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	0	0	0	0	0	0	7	0	0

UBSC= 11 MJOB= 37

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	0	0	0	0	0	0	0	0	0	0	1

UBSC= 11 MJOB= 37

OPT 1.37 115.00 1.50UUUUUU

1	31	1800.	799.00	1001.	ACTUAL	1.21	65.00	1.10	4
2	39	1800.	832.00	968.	ACTUAL	1.21	95.00	1.10	5
3	38	1860.	823.50	1036.	ACTUAL	1.11	64.00	1.50	6
4	50	1800.	750.30	1050.	ACTUAL	1.11	76.00	0.30	7
5	41	1800.	801.00	999.	ACTUAL	0.93	105.00	0.70	9
6	32	1860.	786.90	1073.	ACTUAL	0.88	54.00	1.10	10
7	36	1980.	814.50	1165.	ACTUAL	0.59	58.00	1.50	13
8	33	1920.	795.70	1124.	ACTUAL	0.52	62.00	1.10	15
9	44	1980.	818.50	1162.	ACTUAL	0.42	70.00	1.30	17

1084. ACTUAL 0.50 55.00 0.70 18

OB NO 39 IS ACCEPTED FOR DISPATCH WITH JOB NO 28 WASTE NOW REDUCED TO 968. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	0	0	0	0	0	0	7	0	8
UBSC=	5 MJOB=	39										
THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	0	0	0	0	0	0	0	0	0	0	1
UBSC=	5 MJOB=	39										

OPT 1.37 200.00 4.50UUUUUU

1	31	2010.	926.50	1084.	ACTUAL	1.21	65.00	1.10	4
2	38	2070.	951.00	1119.	ACTUAL	1.11	64.00	1.50	6
3	50	1950.	877.80	1072.	ACTUAL	1.11	76.00	0.30	7
4	41	2010.	928.50	1081.	ACTUAL	0.93	105.00	0.70	9
5	32	2070.	914.40	1156.	ACTUAL	0.88	54.00	1.10	10
6	46	2190.	977.40	1213.	ACTUAL	0.73	72.00	1.70	12
7	36	2190.	942.00	1248.	ACTUAL	0.59	58.00	1.50	13
8	49	2010.	917.50	1092.	ACTUAL	0.59	125.00	0.50	14
9	33	2130.	923.20	1207.	ACTUAL	0.52	62.00	1.10	15
10	45	2250.	1050.50	1199.	ACTUAL	0.50	115.00	1.70	16

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 39

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 28
REG= 300.

0	45	0.499	115.00	1.70	314.	OPT	1.071	185.00	1.70	UUUUUU	
		A	B								
1	41	510.	269.00		241.	ACTUAL	0.93	105.00	0.70	9	
2	32	510.	254.90		255.	ACTUAL	0.88	54.00	1.10	10	
3	50	510.	218.30		292.	ACTUAL	1.11	76.00	0.30	7	
4	46	600.	317.90		282.	ACTUAL	0.73	72.00	1.70	12	
5	38	510.	291.50		219.	ACTUAL	1.11	64.00	1.50	6	
6	36	600.	282.50		318.	ACTUAL	0.59	58.00	1.50	13	
7	49	510.	258.00		252.	ACTUAL	0.59	125.00	0.50	14	
8	31	510.	267.00		243.	ACTUAL	1.21	65.00	1.10	4	
9	33	540.	263.70		276.	ACTUAL	0.52	62.00	1.10	15	
10	44	600.	286.50		314.	ACTUAL	0.42	70.00	1.30	17	

JOB NO 38 IS ACCEPTED FOR DISPATCH WITH JOB NO 45 WASTE NOW REDUCED TO 219. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	0	0	0	0	0	0	7	10	8

UBSC= 6 MJOB= 38

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	0	0	0	0	9	0	0	0	0	0	1

UBSC= 6 MJOB= 38

				OPT	1.07	121.00	1.50UUUUUU			
1	41	600.	365.00	235.	ACTUAL	0.93	105.00	0.70	9	
2	32	660.	350.90	309.	ACTUAL	0.88	54.00	1.10	10	
3	50	540.	314.30	226.	ACTUAL	1.11	76.00	0.30	7	
4	36	750.	378.50	372.	ACTUAL	0.59	58.00	1.50	13	
5	31	720.	363.00	357.	ACTUAL	1.21	65.00	1.10	4	
6	33	690.	359.70	330.	ACTUAL	0.52	62.00	1.10	15	
7	44	750.	382.50	368.	ACTUAL	0.42	70.00	1.30	17	
8	42	630.	339.10	291.	ACTUAL	0.38	68.00	0.70	18	
9	48	630.	343.30	287.	ACTUAL	0.38	74.00	0.70	19	

10 [illegible] 510. 404.00 408. ACTUAL 0.55 15.00 1.90 20

0 OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 38

0 OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 45

OPT 1.07 185.00 0.20UUUUUU

0 OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 38

0 OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 45

REG= 300.

RIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SUBSCRIPT
0	46	0.727	72.00	1.70	388.	OPT	0.844	228.00	1.70	UUUUUU

A B

1	36	540.	209.40	331.	ACTUAL	0.59	58.00	1.50	13
2	49	510.	184.90	325.	ACTUAL	0.59	125.00	0.50	14
3	32	510.	181.80	328.	ACTUAL	0.88	54.00	1.10	10
4	33	510.	190.60	319.	ACTUAL	0.52	62.00	1.10	15
5	41	510.	195.90	314.	ACTUAL	0.93	105.00	0.70	9
6	44	540.	213.40	327.	ACTUAL	0.42	70.00	1.30	17
7	50	510.	145.20	365.	ACTUAL	1.11	76.00	0.30	7
8	42	510.	170.00	340.	ACTUAL	0.38	68.00	0.70	18
9	48	510.	174.20	336.	ACTUAL	0.38	74.00	0.70	19
10	35	600.	234.90	365.	ACTUAL	0.35	75.00	1.50	20

OB NO 41 IS ACCEPTED FOR DISPATCH WITH JOB NO 46 WASTE NOW REDUCED TO 314. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	0	0	0	0	0	0	7	10	8

UBSC= 9 MJOB= 41

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	0	0	9	11	0	0	0	0	1

UBSC= 9 MJOB= 41

OPT 0.84 123.00 0.70UUUUUU

1	50	510.	218.70	291.	ACTUAL	1.11	76.00	0.30	7
2	42	510.	243.50	266.	ACTUAL	0.38	68.00	0.70	18
3	48	510.	247.70	262.	ACTUAL	0.38	74.00	0.70	19
4	53	510.	227.90	282.	ACTUAL	0.32	80.00	0.40	21

OB NO 48 IS ACCEPTED FOR DISPATCH WITH JOB NO 46 WASTE NOW REDUCED TO 262. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	0	0	0	0	0	0	7	10	8

UBSC= 19 MJOB= 48

THETA	36	49	33	45	44	42	48	35	53	43	47	34

THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	0	0	9	11	0	13	0	0	1

UBSC= 19 MJOB= 48

OPT 0.84 228.00 1.00UUUUUU

1	49	540.	331.90	208.	ACTUAL	0.59	125.00	0.50	14
2	50	540.	292.20	248.	ACTUAL	1.11	76.00	0.30	7
3	42	600.	317.00	283.	ACTUAL	0.38	68.00	0.70	18
4	53	540.	301.40	239.	ACTUAL	0.32	80.00	0.40	21
5	43	660.	368.40	292.	ACTUAL	0.28	110.00	0.90	22
6	47	660.	377.40	283.	ACTUAL	0.28	120.00	0.90	23
7	54	600.	350.40	250.	ACTUAL	0.20	135.00	0.60	25

OB NO 49 IS ACCEPTED FOR DISPATCH WITH JOB NO 46 WASTE NOW REDUCED TO 208. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	0	0	0	0	0	0	7	10	8

UBSC= 14 MJOB= 49

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	0	0	9	11	0	13	14	0	1

UBSC= 14 MJOB= 49

OPT 0.84 103.00 0.50UUUUUU

1	50	600.	354.70	245.	ACTUAL	1.11	76.00	0.30	7
2	53	630.	363.90	266.	ACTUAL	0.32	80.00	0.40	21

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 49

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 46

OPT 0.84 228.00 0.50UUUUUU

1	50	660.	417.20	243.	ACTUAL	1.11	76.00	0.30	7
2	53	720.	426.40	294.	ACTUAL	0.32	80.00	0.40	21

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 49

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 46
REG= 300.

RIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SUBSCRIPT	
0	35	0.349	75.00	1.50	338.	OPT	1.222	225.00	1.50	UUUUUU	
		A	B								
1	31	450.	184.00		266.	ACTUAL	1.21	65.00	1.10		4
2	50	450.	135.30		315.	ACTUAL	1.11	76.00	0.30		7
3	32	480.	171.90		308.	ACTUAL	0.88	54.00	1.10		10
4	36	600.	199.50		401.	ACTUAL	0.59	58.00	1.50		13
5	33	540.	180.70		359.	ACTUAL	0.52	62.00	1.10		15
6	44	600.	203.50		397.	ACTUAL	0.42	70.00	1.30		17
7	42	480.	160.10		320.	ACTUAL	0.38	68.00	0.70		18
8	53	450.	144.50		306.	ACTUAL	0.32	80.00	0.40		21
9	43	540.	211.50		329.	ACTUAL	0.28	110.00	0.90		22
10	47	540.	220.50		320.	ACTUAL	0.28	120.00	0.90		23

JOB NO 31 IS ACCEPTED FOR DISPATCH WITH JOB NO 35 WASTE NOW REDUCED TO 266. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	16	0	0	0	15	0	7	10	8
UBSC=	4	MJOB=	31									
THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	0	0	9	11	0	13	14	0	1
UBSC=	4	MJOB=	31									

				OPT	1.22	160.00	1.10UUUUUU			
1	50	450.	206.80	243.	ACTUAL	1.11	76.00	0.30		7
2	32	570.	243.40	327.	ACTUAL	0.88	54.00	1.10		10
3	33	630.	252.20	378.	ACTUAL	0.52	62.00	1.10		15
4	42	570.	231.60	338.	ACTUAL	0.38	68.00	0.70		18
5	53	510.	216.00	294.	ACTUAL	0.32	80.00	0.40		21
6	43	630.	283.00	347.	ACTUAL	0.28	110.00	0.90		22
7	47	630.	292.00	338.	ACTUAL	0.28	120.00	0.90		23
8	34	690.	245.60	444.	ACTUAL	0.22	56.00	1.10		24
9	54	570.	265.00	305.	ACTUAL	0.20	135.00	0.60		25

JOB NO. 50 IS ACCEPTED FOR DISPATCH WITH JOB NO. 35 WASTE NOW REDUCED TO 245. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	16	0	0	0	15	0	7	10	8

UBSC= 7 MJOB= 50

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	0	0	9	11	0	13	14	17	1

UBSC= 7 MJOB= 50

OPT 1.22 84.00 0.30UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 50

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 35

OPT 1.22 225.00 0.40UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 50

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 35

REG= 300.

TRIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		SINGLE	SIZE	TIME	SUBSCRIPT
0	47	0.278	120.00	0.90	162.	OPT	1.292	180.00	0.90	UUUUUU
		A	B							
1	42	360.	155.60		204.	ACTUAL	0.38	68.00	0.70	18
2	53	300.	140.00		160.	ACTUAL	0.32	80.00	0.40	21
3	43	420.	207.00		213.	ACTUAL	0.28	110.00	0.90	22
4	54	360.	189.00		171.	ACTUAL	0.20	135.00	0.60	25

JOB NO 53 IS ACCEPTED FOR DISPATCH WITH JOB NO 47 WASTE NOW REDUCED TO 160. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	16	0	0	0	15	0	7	10	8

UBSC= 21 MJOB= 53

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3213	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	0	0	9	11	18	13	14	17	1

UBSC= 21 MJOB= 53

	OPT	1.29	100.00	0.40UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 53

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 47

	OPT	1.29	180.00	0.50UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 53

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 47
REG= 300.

TAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SUBSCRIPT
0	43	0.278	110.00	0.90	171.	OPT	1.292	190.00	0.90	000000
		A	B							
1	42	360.	146.60		213.	ACTUAL	0.38	68.00	0.70	18
2	54	360.	180.00		180.	ACTUAL	0.20	135.00	0.60	25

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 54

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 43
REG= 300.

RIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SUBSCRIPT
0	44	0.418	70.00	1.30	299.	OPT	1.153	230.00	1.30	UUUUUU

A B

1	32	420.	150.40	270.	ACTUAL	0.88	54.00	1.10	10
2	33	480.	159.20	321.	ACTUAL	0.52	62.00	1.10	15
3	42	420.	138.60	281.	ACTUAL	0.38	68.00	0.70	18
4	34	540.	152.60	387.	ACTUAL	0.22	56.00	1.10	24
5	54	420.	172.00	248.	ACTUAL	0.20	135.00	0.60	25

OB NO 54 IS ACCEPTED FOR DISPATCH WITH JOB NO 44 WASTE NOW REDUCED TO 248. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	16	0	0	0	15	0	7	10	8
UBSC=	25 MJOB=	54										
THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	20	21	9	11	18	13	14	17	1
UBSC=	25 MJOB=	54										

	OPT	1.15	95.00	0.60UUUUUU
	OPT	1.15	230.00	0.70UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 54

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 44

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 54

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 44

REQ= 300.

RTAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SUBSCRIPT
0	36	0.588	58.00	1.50	363.	OPT	0.983	242.00	1.50	UUUUUU

A B

1	32	450.	146.40	304.	ACTUAL	0.88	54.00	1.10	10	
2	33	480.	155.20	325.	ACTUAL	0.52	62.00	1.10	15	
3	42	450.	134.60	315.	ACTUAL	0.38	68.00	0.70	18	
4	34	540.	148.60	391.	ACTUAL	0.22	56.00	1.10	24	

OB NO 32 IS ACCEPTED FOR DISPATCH WITH JOB NO 36 WASTE NOW REDUCED TO 304. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	16	24	0	0	15	23	7	10	8
UBSC=	10 MJOB=		32									
THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	0	20	21	9	11	18	13	14	17	1
UBSC=	10 MJOB=		32									

					OPT	0.98	188.00	1.10UUUUUU	
1	33	630.	214.60	415.	ACTUAL	0.52	62.00	1.10	15
2	42	570.	194.00	376.	ACTUAL	0.38	68.00	0.70	18
3	34	690.	208.00	482.	ACTUAL	0.22	56.00	1.10	24

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 32

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 36

OPT	0.98	242.00	0.40UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 32

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 36
REG= 300.

RIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		SIZE	TIME	SUBSCRIPT
0	33	0.519	62.00	1.10	262.	OPT	1.052	238.00	1.10	UUUUUU

	A	B

1	42	360.	115.80	244.	ACTUAL	0.38	68.00	0.70	18
2	34	480.	129.80	350.	ACTUAL	0.22	56.00	1.10	24

OB NO 42 IS ACCEPTED FOR DISPATCH WITH JOB NO 33 WASTE NOW REDUCED TO 244. KBS

THETA	29	30	40	31	39	38	50	52	41	32	37	46
THETA	60.00	52.00	66.00	65.00	95.00	64.00	76.00	78.00	105.00	54.00	85.00	72.00
THETA	2.1000	1.1000	0.7000	1.1000	1.1000	1.5000	0.3000	0.3000	0.7000	1.1000	1.5000	1.7000
THETA	1.5208	1.4711	1.4056	1.2120	1.2120	1.1071	1.1071	1.1071	0.9273	0.8761	0.8520	0.7266
R	28	29	30	31	32	33	34	35	36	37	38	39
RDER	6	2	3	16	24	25	0	15	23	7	10	8

UBSC= 18 MJOB= 42

THETA	36	49	33	45	44	42	48	35	53	43	47	34
THETA	58.00	125.00	62.00	115.00	70.00	68.00	74.00	75.00	80.00	110.00	120.00	56.00
THETA	1.5000	0.5000	1.1000	1.7000	1.3000	0.7000	0.7000	1.5000	0.4000	0.9000	0.9000	1.1000
THETA	0.5880	0.5880	0.5191	0.4993	0.4182	0.3805	0.3805	0.3488	0.3218	0.2783	0.2783	0.2187
R	40	41	42	43	44	45	46	47	48	49	50	51
RDER	4	12	26	20	21	9	11	18	13	14	17	1

UBSC= 18 MJOB= 42

OPT 1.05 170.00 0.70UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 42

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 33

OPT 1.05 238.00 0.40UUUUUU

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 42

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 33

REG= 300.

RIAL	JOB NO	ANGLE	SIZE	TIME	WASTE		ANGLE	SIZE	TIME	SUBSCRIPT
0	34	6.219	56.00	1.10	268.	OPT	1.352	244.00	1.10	UUUUUU
		A	B							

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 42

O OTHER JOB REDUCES WASTE WHEN DISPATCHED WITH JOB 34

SUB NO	POST	LEVEL	COST
28	28	6	1200.
37	28	7	1073.
39	28	8	968.
29	51	2	911.
30	51	3	884.
40	51	4	772.
51	51	1	1037.
52	51	5	762.
31	35	16	266.
35	35	15	338.
50	35	17	243.
32	36	24	304.
36	36	23	363.
33	33	25	262.
42	33	26	244.
34	34	27	268.
38	45	10	219.
45	45	9	314.
41	46	12	314.
46	46	11	388.
48	46	13	262.
49	46	14	208.
43	43	20	171.
44	44	21	299.
54	44	22	248.
47	47	18	162.
53	47	19	160.

ORE USAGE OBJECT CODE= 13224 BYTES,ARRAY AREA= 6000 BYTES,TOTAL AREA AVAILABLE= 34912 BYTES

OMPILE TIME= 1.46 SEC,EXECUTION TIME= 3.16 SEC,DATE= 70/363