A STUDY OF THE SYSTEMS AND TECHNIQUES OF SCHEDULING A PUBLIC TRANSIT COMPANY WITH THE USE OF A DIGITAL COMPUTER

by

PRABODH G. BAKSHI

B. E., Victoria Jubilee Technical Institute
University of Bombay, India, 1960

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1965

Approved by

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

DEFINITIONS

Piece of work	Any work assigned to an operator which pays less than 8 hours.
Straight run	A regular run having no unpaid breaks.
Split run	A regular run having unpaid breaks.
Pull out time	The starting time of a piece of work, straight run, or a split run.
Pull in time	The finishing time of a piece of work, straight run, or a split run.
Travel time	The actual running time from the garage to point of relief.
Platform time	The time during which an operator is in charge of a bus.
Turn-in	A specific amount of time which the operator receives for every straight run and each piece of work in his assignment.
Guarantee time	The time paid for but not worked, to make up a minimum guarantee for certain classes of work.
Allowance	A time equal to either the travel time, the guarantee time, or both.
Pay time	The total pay time for a specific run.
Spread time	The total working time plus the unpaid time of a split run.

INTRODUCTION

Since the close of World War II, the public has made it abundantly clear that they prefer the use of automobiles to any local system of mass transportation. This is an established trend, and until this trend is reversed, the use of mass transportation will fall off.[1] The rapid increase in the number of automobiles in use is not only causing alarm in the transit industry but is also threatening its very existence. Automobiles have captured today's transportation market for three important reasons which may be stated as:

1. They give complete door-to-door service in one vehicle.

2. They allow flexible routing for direct services from origin to destination.

3. They offer prompt, comfortable and convenient service.

This imbalance between public and private transportation must be checked before it is too late.

City planners are greatly concerned about the growth in the number of automobiles in use today. They also realize the importance of having public transportation to service downtown areas. New cities of the future have to be planned with adequate facilities for efficient transit operations; however, the transit industry must also offer complete, or nearly complete, origin-to-destination services for a large number of passenger trips. The aim of the transit authorities must be to carry both those who have to use their services and those who elect to use them, particularly during rush hours.

---

1. "What Transit Companies are Doing to Improve Ridership." Metropolitan Transportation and Planning, July, 1963, Vol. 59:4, pp. 34-35.

It is obvious that the task of the transit companies is becoming immeasurably greater and more complex. One of the urgent needs of today is that of increasing the efficiency of the transit services and to make them more attractive to the public. In view of the constant increase in the cost of labor, which constitutes almost 85 per cent of the total revenue[2], there is an urgent need for reducing operating costs.

Proper scheduling techniques yielding efficient and effective schedules would play a major role in reducing operating costs. Hence, the aim of transit companies has been rescheduling services as a means of maintaining operations on a profitable basis in face of strong competition from the automobile users. The best manual methods of scheduling, however, fall short of solving the problem because computations cannot be made quickly enough to keep pace with the rapidly changing conditions. In order to overcome the computational difficulties, researchers are turning to the use of high-speed electronic computers. This report is based on the theoretical analysis completed in previous research on the use of computers in scheduling[3].

The transit scheduling problem is divided into three main areas:

1. Determination of the passenger load and vehicle requirements by dividing the city into areas. Routes to service these areas are then fixed.

2. Preparation of a headway sheet giving the timings and the frequency of services in the various areas. This is dependent upon the requirements of the transit services during peak hours and slack periods.

---

2. Lassow, Bill, "Why Do We Need Transportation Research?" Metropolitan Transportation and Planning, November, 1963, Vol. 59:6, pp. 20-22.

3. Elias, Samy E. G., "A Digital Computer Solution to the Transit Operation Assignment Problem." Unpublished Doctoral Thesis, Oklahoma State University, 1960.

3. Preparation of operator assignments and establishment of a daily work schedule.

In this report a computer solution to the assignment of the daily work schedule is demonstrated.

HYPOTHESIS

The objective of this report was to develop computer programs to construct the daily work schedule of the Cincinnati Transit Company, Ohio. The computer approach is primarily aimed at reducing both laborious manual work and the cost involved in constructing daily schedules. The solution is based on previous research on the use of computers for transit company scheduling[4].

As a further step toward reducing manual work, computer programs were developed to select the most economical combinations of split runs from among several alternatives.

The IBM 1620 computer was used for solving the problem. All programs were written in the FORTRAN (Formula Translation System) language. The data for this problem was obtained from the Cincinnati Transit Company, Ohio.

SCHEDULING OF THE CINCINNATI TRANSIT COMPANY

The computer approach to scheduling public transit was put to test on the data from four transit companies. It was the author's responsibility to develop the computer programs for the Cincinnati Transit Company, Ohio, as an objective of this report. The computer approach of preparing the transit company schedule is a simulation of the manual method now in use. Therefore, the computer programs follow the manual method procedure very closely.

---

4. Ibid.

The number of variables and restrictions is the main difficulty encountered in effectively programming a daily work schedule for each operator. Operating variables, trip times, and restrictions of the union-management agreement are a few of the factors which have to be considered in making an effective daily schedule.

One of the biggest problems faced by the transit industry is the variable nature of the daily demand on their services. Traffic peaks in the mornings and late afternoons tend to fall off in the early morning hours and on weekends. As can be seen from the graph in Fig. 1, the volume of traffic is almost 300 per cent more during peak hours than during slack periods. This vast fluctuation in the maximum and minimum demands forces the transit companies to maintain large fleets of buses and an equally large number of operators. It may be of interest to point out that 85 per cent[5] of the revenue of transit companies goes out as wages to the employees. In order to remove traffic bottlenecks during peak hours, some buses are put on shorter trips while others make extremely long runs. This variation in trip distance and time involved leads to many complications in making operator assignments.

Some of the restrictions and constraints of the union-management agreement of the Cincinnati Transit Company are:

1. No regular run shall pay less than eight hours.

2. Time and one-half is paid as overtime for all work beyond eight hours.

3. Forty-six per cent (45%) of all runs shall be straight runs.

4. Five minutes pay shall be allowed for operators for turn-in and will be considered as part of the regular time.

---

5. Lassow, op. cit.

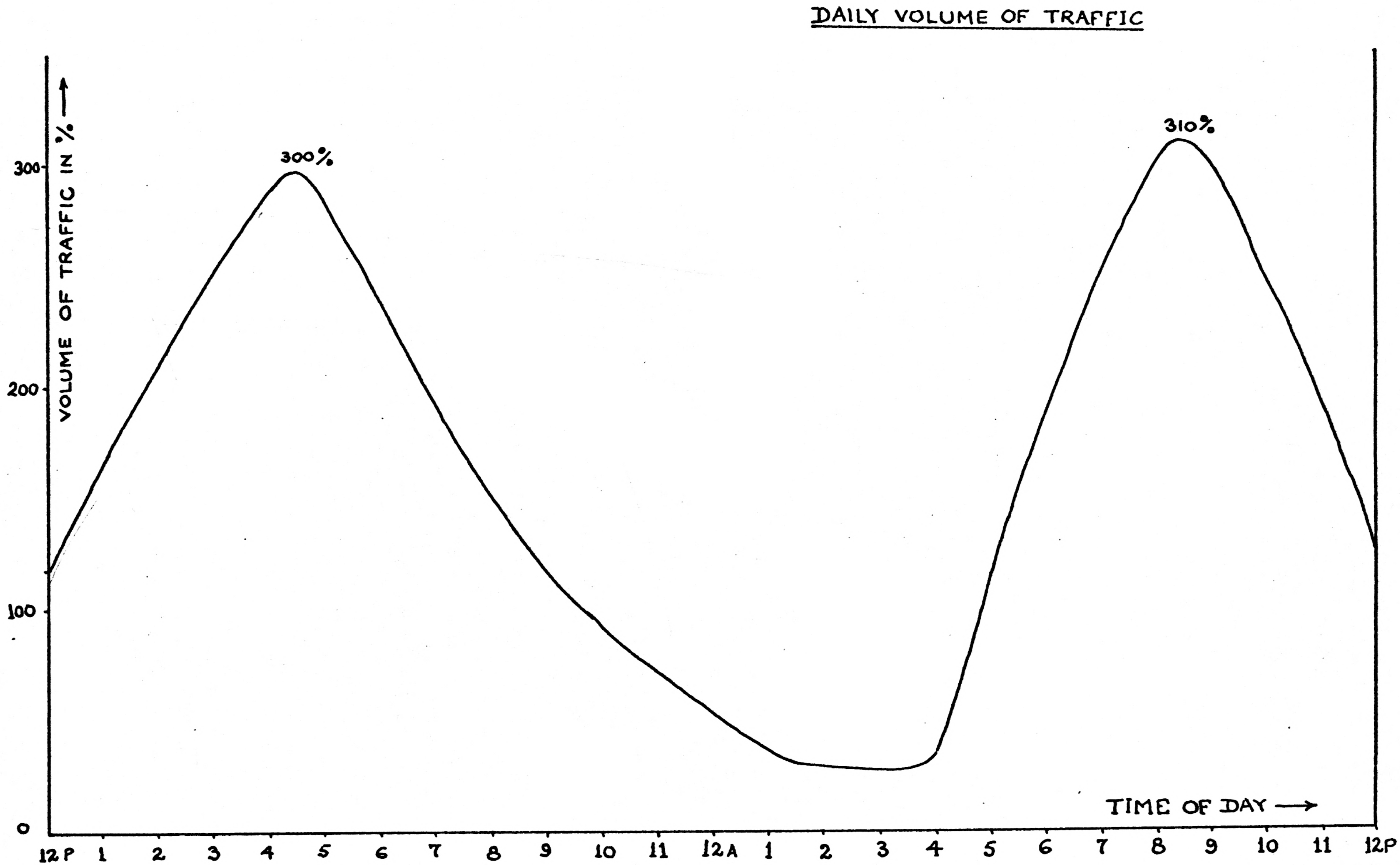
DAILY VOLUME OF TRAFFIC
VOLUME OF TRAFFIC IN %
300%
310%
300
200
100
0
TIME OF DAY
12P 1 2 3 4 5 6 7 8 9 10 11 12A 1 2 3 4 5 6 7 8 9 10 11 12P

Fig. 1

5. No less than fifteen minutes shall be allowed for operators for meal relief.

6. If the operator is relieved on the road, he will be allowed actual running time from the garage to the point of relief plus five minutes.

7. If the intervening time between taking out parts of the run amounts to one or less, such intervening time shall be paid for as a part of the regular run.

To develop an effective schedule within the above restrictions is a difficult and tedious task. An ideal schedule would be one having all straight runs; however, as this is practically impossible, a three-step approach is used. These steps, the same in both the manual and the computer approaches to daily scheduling, are:

1. Developing all possible straight runs.

2. Constructing split runs by combining pieces of work.

3. Combining pieces of work not used in step 2 with straight runs not utilized from step 1. The straight runs are broken into two or three pieces, depending upon the requirements.

The reason for following a three-step approach is that the computer storage capacity limits the use of one program to do all the steps.

STRAIGHT RUN MAKING

In this step both the manual and the computer methods have the same objectives - to make all possible straight runs. As an example, the data for Route 26 and Block 3 is taken. Details regarding the route number, block number, pull-out time, pull-in time, and relief times are obtained from the headway sheet. The details of Route 26 and Block 3 are:

Pull out of garage at	5:25 AM		
Pull into garage at	5:57 PM		
Relief times	5:25,	7:44,	8:59 ------
	12:25,	12:58,	2:21 ------
	4:45,	5:52,	5:57 ------

All the data received from the Cincinnati Transit Company was in hours and minutes. In order to make the data compatible for use on the computer it had to be converted onto a 24-hour clock and into hours and decimals. A conversion program was used for this purpose. The converted computer output for Route 26 and Block 3 was punched out in the following form:

No/reliefs	Travel	Route	Block	P/out	P/in	Reliefs:
13	0.0	26	3	5.41	17.95	5.41, 7.73, 8.98, ----
						12.46, 12.96, 14.35, ----
						16.75, 17.86, 17.95, ----

The steps followed in constructing straight runs are:

1. The machine finds the total trip time for the block and compares this with eight hours, the minimum permissible working time for a straight run. 17.95 - 5.41 = 12.54 hours.

2. If the total trip time is more than 16 hours, two straight runs can be developed, one from the front of the block and another from the tail end of the block, leaving a piece of work in between. The logic behind this procedure is to avoid having a late piece of work which would be difficult to use in the next step.

If the total trip time is larger than seven hours but less than 16 hours, a straight run and a piece of work will be developed. If the total trip time is less than seven hours, a piece of work will be punched out.

It was found that for any block having a total trip time of exactly seven hours, it would be cheaper for the company to pay guarantee time and make a straight run rather than paying spread-time penalty and making a split run. Hence, it was the company's policy to use a seven hour limit in deciding whether to make a straight run or a piece of work. The total time for the example falls in the second category.

3. The platform time for the straight run is now computed. The union-management agreement states that the company has to consider the five minutes turn-in, travel time, and five minutes travel allowance, if any, as a part of the regular time. Therefore, to avoid paying overtime, the platform time is computed as eight hours minus turn-in time (five minutes), travel time and travel allowance, if any. For Route 26 and Block 3, used in the example, there is no travel time. Therefore, the platform time equals 7.91 hours (8.00 less 5 minutes turn-in time).

The machine now checks the relief times on the block, starting from the pull-out side, and looks for one that breaks the block into two pieces, one of them being either equal to or slightly larger or slightly smaller than the platform time. In the example, if the block is broken at the relief 1296 we get a piece having a platform time of 7.55 hours, whereas the next relief time 1435 gives a platform time of 8.94 hours.

4. The next step is to decide which of these two relief times to select as the pull-in time of the straight run. This selection is done on the basis of cost. In the case of relief time 1296, the company has to pay 0.36 hours as a guarantee time but relief time 1435 pays an overtime of 0.56 hours. Of these two alternatives, the one most economical is chosen. Therefore, relief time 1296 becomes the pull-in time for the straight run.

5. The machine next checks the trip time of the remainder of the block. In our example this is 4.99 hours (1795 minus 1296). It is not enough for another straight run so a piece of wo1k is punched out.

6. Having developed a straight run and a piece of work from the forward direction, the machine now constructs another straight run from the tail end of the block. This is done to give the schedule maker the choice between selecting an early or a late straight run on the same block.

The straight run output for Route 26 and Block 3 is shown in Fig. 2. A flow diagram of the first step is given in Fig. 3.

CONSTRUCTING SPLIT RUNS

The next step in constructing the schedule is that of combining the pieces of work. To explain this step, the following two pieces of work will be used:

	Route	Block	TF	P/out	P/in	TB	Platform
No 1	1	12	0	606	951	0	345
No 2	4	4	0	1406	1826	0	420

Here again, both the manual and the computer methods follow almost identical steps. The computer goes through the following sequence of operations:

1. The cards containing the information about the pieces of work are first read by the machine. This information is then stored in the memory storage area of the machine.

2. The computer selects the first piece of work and makes all possible split run combinations with the other pieces. In our example, the first piece to be chosen would be (1-12). Before constructing a split run, the computer checks to see if all restrictions are fulfilled. These are:

a. The pull-out and pull-in times of the two pieces being combined are checked. Both the pull-out and the pull-in times of piece (4-4)

COMPUTER OUTPUT - STRAIGHT RUN MAKING

ROUTE 26 BLOCK 3

NO.	ROUTE	BLOCK	TF	P/OUT	P/IN	TB	PLAT	PENLT	TURN IN	PAY
1	26	3	0	541	1296	0	755	36	8.3	800
	26	3	0	1296	1795	0	499			
2	26	3	0	898	1795	0	897	0	8.3	905
	26	3	0	541	898	0	357			

Fig. 2

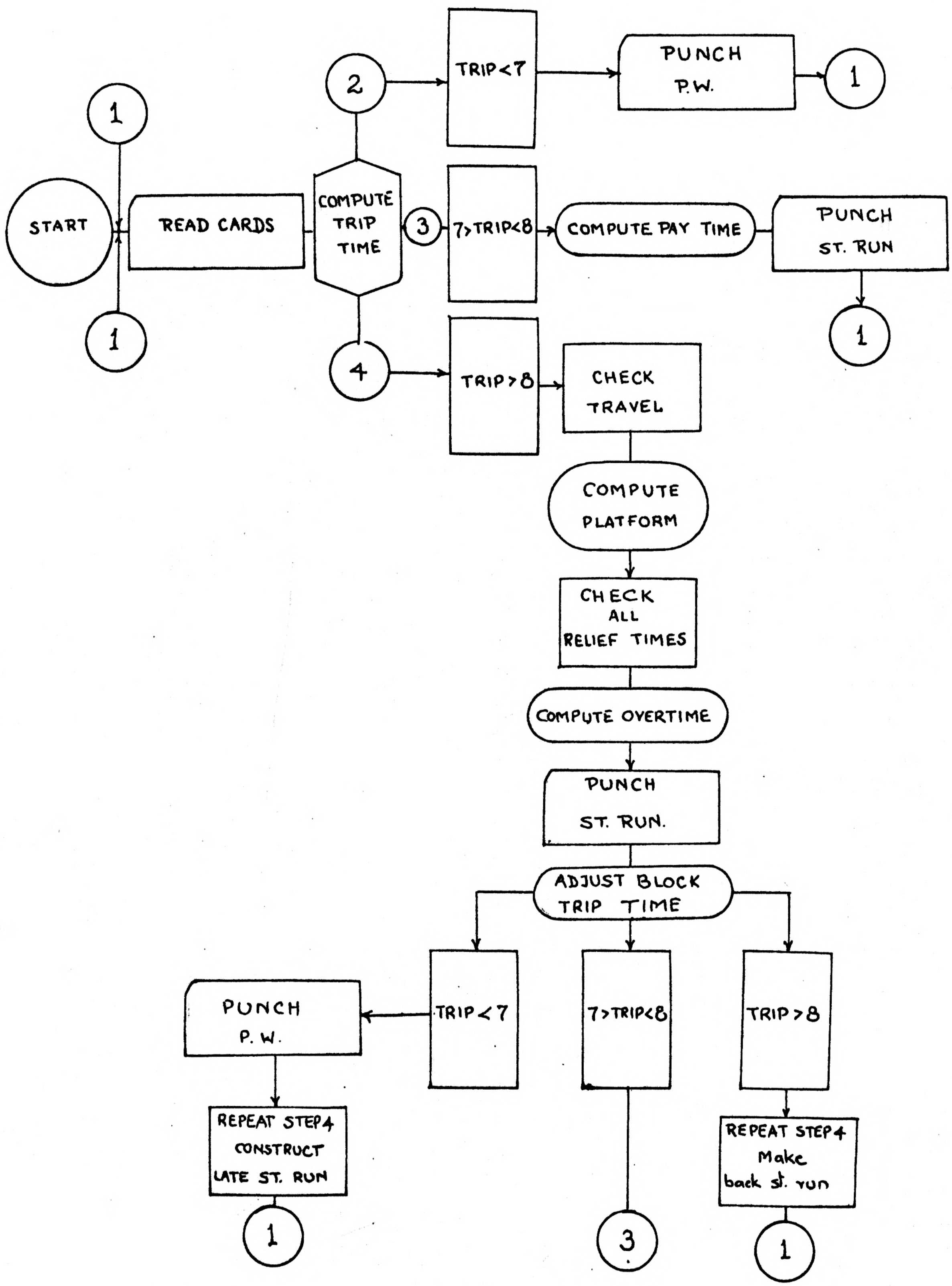

Fig. 3. Flow Diagram - Straight Run Making

must either be earlier or later than the pull-out and pull-in times of piece (1-12). In the example this condition is fulfilled; the times of piece (4-4) are later than those of piece (1-12).

b. A gap of a minimum of five minutes plus travel, if any, must exist between the pull-in time of piece (1-12) and the pull-out time of piece (4-4), if any. The gap between the two pieces being used is 4.55, well beyond the minimum limit.

c. Spread time is computed. There is a maximum limit of 14.0 hours on spread time. However, any time beyond 11.0 hours is paid for at half the regular rate.

Spread time = (pull-in time of (4-4) - travel + turn-in)

- (pull-out time of (1-12) - travel + turn-in)

or (1826 - 0 + 8.3) - (606 - 0 + 8.3) = 12.20

This figure (12.20) is within the maximum limit of 14.0 hours.

d. All the pieces used for making split runs must be at least one hour long. The amount by which a piece falls short of this limit is paid for by the company as guarantee time. The pieces of work used in the example satisfy this restriction.

e. The last check before a split run is constructed is for the platform time. The total platform time must lie between 7.50 and 9.00 hours. Platform time for the example (1-12) + (4-4) is 7.81 hours.

As the two pieces satisfy all the restrictions, a split run is developed. The computer output is in the following format:

No.	Route	Block	TF	P/out	P/in	TB	Spread	Platform	Penalty	Turn-in	Pay
	1	12	0	606	951	0		345			
1	4	4	0	1406	1826	0	128.3	420	18.34	16.67	800

In a similar manner, the machine picks one piece at a time and constructs all possible split run combinations with the other remaining pieces. Thus, the computer approach gives every possible split run combination that can be constructed. The schedule maker has now to make a selection from the split runs developed.

A part of the objective of this report was to develop a computer technique to do the selection. A detailed description of the various approaches developed is given in the latter part of this report. Figure 4 gives the flow diagram of the computer program used in this step.

COMBINING PIECES OF STRAIGHT RUNS AND PIECES OF WORK

After completing selections from the several hundred alternatives of split runs constructed in the preceding step, there were invariably some pieces of work that were unused. To avoid paying a high overtime, these are now combined with pieces of straight runs.

According to the union-management agreement, a daily schedule must contain at least 46 per cent straight run. The straight runs which are broken and used in this step are the ones that are in excess after this minimum restriction has been satisfied. These straight runs are broken into two or three pieces and combined with the unused pieces from the preceding step. The machine is programmed to do the following sequence of operations:

1. All the pieces of work are read and retained in the memory storage.

2. One straight run is read at a time and is combined as follows:

a. The straight run is broken at every relief point until one section can be combined with one or more pieces of work. This section is

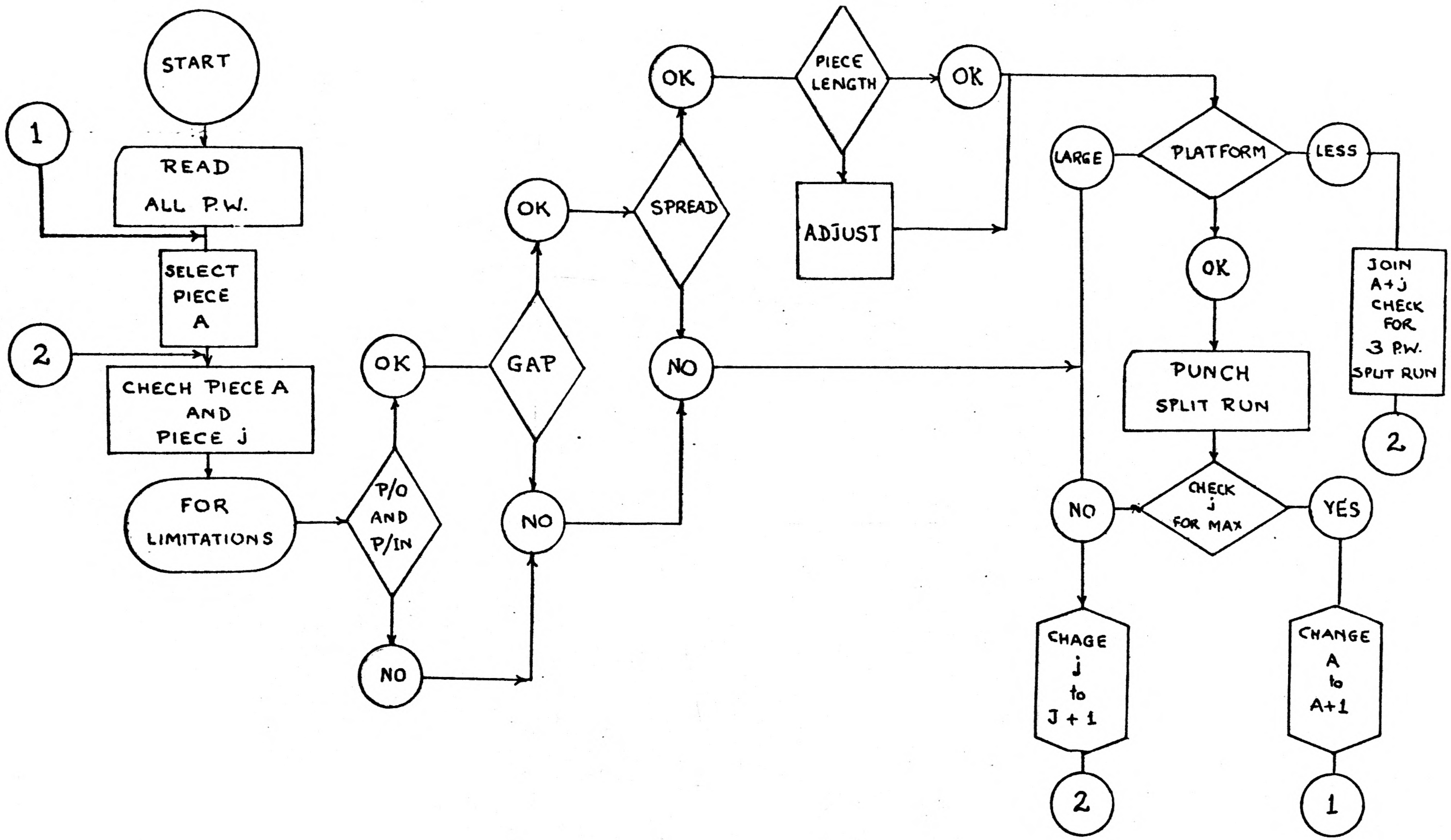

Fig. 4. Flow Diagram - Split Run Making

called piece No. 1 to facilitate the description of the procedure. All the split run combinations are punched out.

b. The remaining portion of the straight run is now broken until a second section is developed with which one or more pieces of work can be combined. Again, all the combinations are punched out. The piece of the straight run used is designated as piece No. 2.

c. The computer now makes all the possible split run combinations with the third and last piece of the straight run.

d. On completion of step c, the machine goes back to piece No. 2 and increases its length by one relief time. The new piece No. 2 is again combined with the same pieces of work. Step c is repeated but with a smaller piece No. 3. This cycle is repeated until the length of piece No. 2 becomes equal to the straight run minus piece No. 1.

e. The machine now branches back to step a and increases the length of piece No. 1 by one relief time. The sequence of operations is repeated. This cycle ends as soon as the length of piece No. 1 reaches the maximum limit of six hours.

3. The machine now reads another straight run and repeats the whole sequence of operations described above.

A flow diagram for this part of the computer program is shown in Fig. 5. A sample output of this step is shown in Fig. 6.

SPLIT RUN SELECTION

As mentioned in the hypothesis, one of the objectives of this report was to present a computer approach to the selection problem in order to eliminate manual work. In the case of small companies, selection from the split

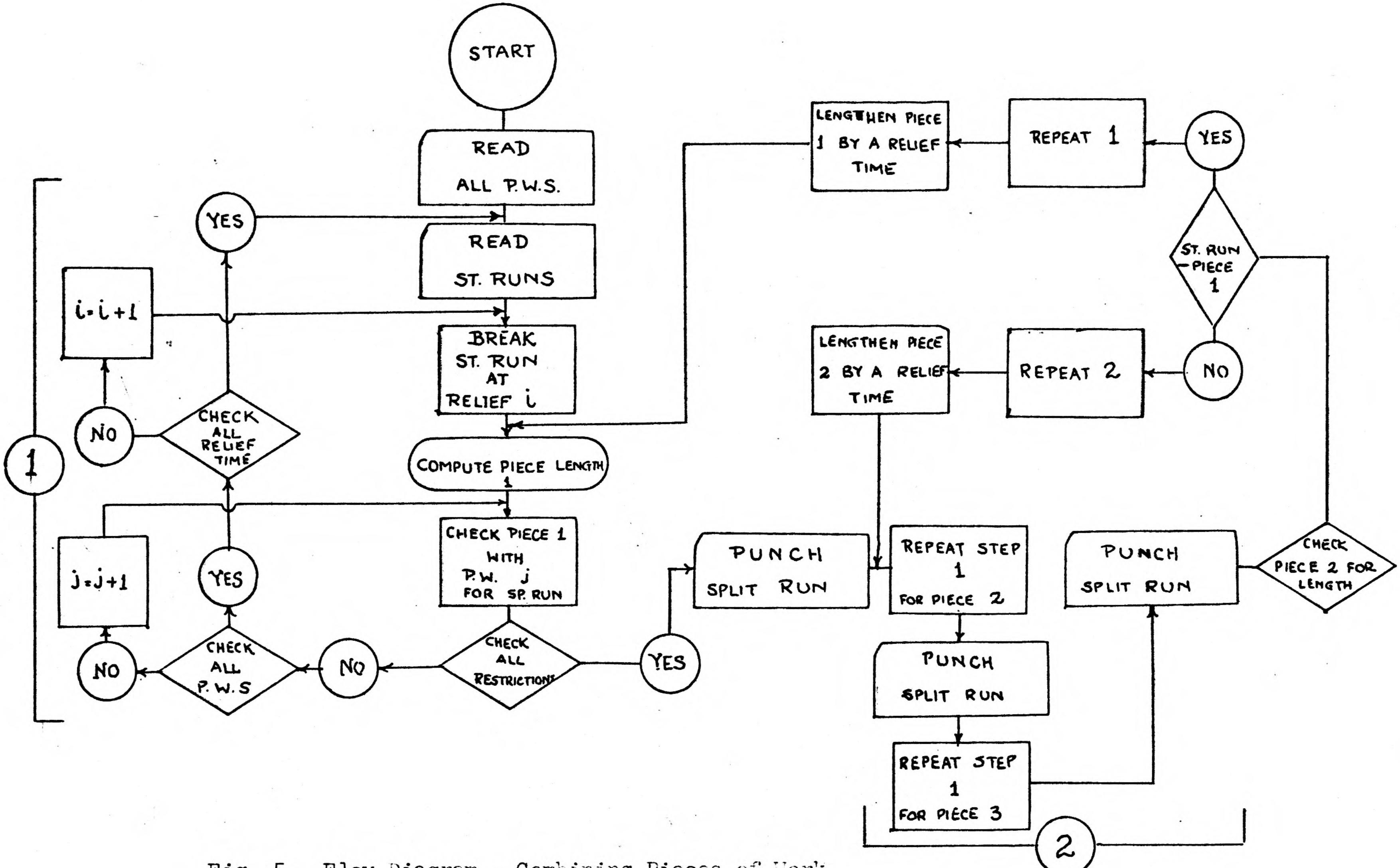

Fig. 5. Flow Diagram - Combining Pieces of Work and Pieces of Straight Runs

MACHINE OUTPUT - COMBINING PIECES OF STRAIGHT RUNS AND PIECES

ROUTE NO. 44 BLOCKO NO. 2

NO.	RN	BN	TF	P/OUT	P/IN	TB	SPREAD	TRIP	PENLTY	TURN	PAY
	1	9	0.0	1440	1946	0.0		506			
1001	44	2	16.6	651	888	0.0	110.0	237	40.3	16.6	800.0
	1	64	0.0	1523	1883	0.0		360			
1002	644	402	0.0	625	1083	0.0	83.1	458	0.0	16.6	834.6
	3	33	0.0	1475	1843	0.0		368			
1003	644	402	0.0	625	1083	0.0	126.3	458	0.0	16.6	842.6
	4	3	0.0	1386	1740	0.0		354			
1004	644	402	0.0	625	1083	0.0	0.0	458	0.0	16.6	828.6
	5	56	0.0	1498	1818	0.0		320			
1005	644	402	0.0	625	1083	0.0	101.3	458	5.3	16.6	800.0
	10	51	0.0	1446	1796	0.0		350			
1006	644	402	0.0	625	1083	0.0	0.0	458	0.0	16.6	824.6
	10	52	0.0	1480	1816	0.0		336			
1007	644	402	0.0	625	1083	0.0	0.0	458	0.0	16.6	810.6
	24	9	0.0	1515	1868	0.0		353			
1008	644	402	0.0	625	1083	0.0	75.6	458	0.0	16.6	827.6
	24	57	0.0	1518	1838	0.0		320			
1009	644	402	0.0	625	1083	0.0	121.3	458	5.3	16.6	800.0
	68	11	0.0	1485	1841	0.0		356			
1010	644	402	0.0	625	1083	0.0	124.3	458	0.0	16.6	830.6
	68	58	0.0	1601	1935	0.0		334			
1011	644	402	0.0	625	1083	0.0	109.1	458	0.0	16.6	808.6
	68	59	0.0	1480	1798	0.0		318			
1012	644	402	0.0	625	1083	0.0	0.0	458	7.3	16.6	800.0
	44	2		1083	1281		198				
	1	9	0.0	1440	1946	0.0		506			
1013	44	2	0.0	888	1156	0.0	0.0	268	9.3	16.6	800.0
	44	2		1156	1281		125				
	1	9	0.0	1440	1946	0.0		506			
1014	44	2	16.6	651	935	0.0	110.0	284	16.6	16.6	823.3
	3	1	36.6	1468	1870	0.0		402			
1015	44	2	16.6	651	935	0.0	90.3	284	97.3	16.6	800.0
	1	9	0.0	1440	1946	0.0		506			
1016	44	2	0.0	935	1208	0.0	0.0	273	4.3	16.6	800.0
	44	2		1208	1281		73				

Fig. 6

run alternatives developed is not a problem, but as the size of the company grows, this becomes a difficult and tedious task. The difficulties arise from the fact that the number of split run alternatives to select from becomes very large, sometimes running into several hundred. The number of split runs developed for the Cincinnati Transit Company was 803.

Selection criteria vary from company to company. The manual selection method developed in previous research[6] and three other computer approaches developed in this report are described in detail. These methods are:

1. Manual Method - Summary Table
2. Computer Method - Total Pay-time Basis
3. Computer Method - Modified Summary Table based on Pay-time
4. Computer Method - Linear Programming Approach

MANUAL METHOD - SUMMARY TABLE

The split run output from the computer method consists of all the possible combinations that can be made. To make the output more manageable, a summary table, Fig. 7, was constructed. This table was made by the machine. Figure 7 gives the table for the 803 combinations made for the Cincinnati Transit Company.

On this table, a letter "1" is indicated for every split run combination made. The number of combinations made with each piece is indicated on the extreme right and bottom of the table. The use of the summary table is now illustrated.

The criterion for selection is to use as many pieces as possible. The

---

6. Elias, Samy E. G., "The Use of Digital Computers in the Economic Scheduling for both Man and Machine in Public Transportation." Special Report 49, Kansas State University Engineering Experiment Station.

```
     I                  22266664    4  116  12     2422 2   22    22266  126    I
     I1111144444455788468888841138445  834 91333544686833389133544888 11 68     I
     I  56   566 55  6 5  555        5555 655 555555555 455 56365 5515 6556     I
     I1678 24891313112 311567354261361291 31378974413428464243159751 89242      I
-----------------------------------------------------------------------------------
  111I1  1  1      111 1 1111 111       11   111 1 1            1  11       I 24
  112I1 11  1      111 1 11111111       11   111 1 1            1  11  1    I 27
  304I1 111 1  1     1 11111111   1 1  11  1 1 11 1        1 1  1  1  1     I 29
  305*   1  111 1  1          11111    1   1 1     1    1  1 1              I 18
  306I  1 1 1 111    1  1  1    11111    1 1 1       1  1 1 1  11 1  1      I 25
  309I 1  1 1      111 11 111 11            1 1 1                 1   1     I 18
  310I1 1 1 1  1     1  1 111111  1 1  1   1   1       1 1  1  1            I 22
  311I 1 1 1 1     111 11 111 11        1 1  1 1 1 1                1   1   I 22
  502I1 111 1 111    1  11 11 11  1111111  1 1 1       1 1  11 11 1         I 32
  503I       11  1             1 1 11          1     111     1              I 12
  504I 1   1 1   1     1              11                                    I  7
  508I1  1  1      111 1 1111 111       11    1  1 1               11       I 21
  701I 1   1 1   1     1              11                                    I  7
 1001I1 1 1 1 111    1  1   1  1  11111    1   1           1 1  11 1        I 23
 1002I11 1  1      111 11 111 111       11    1  1 1               11       I 22
 1003I1  1  1      111 1 1111 111       11   111 1 1           1  11  1     I 25
 2404I 1   1 1   1     1              11                                    I  7
 2405I 1   1 1     111 11 111 11        1    1 1 1 1               1  1     I 20
 2406I11 1  1      111 11 111 111       111  111 1 1               11  1    I 26
 2601I 1   1 1   1 111 11                      1                   1        I 11
 2602I1 111 1  1  11111 11111111        11  111111 1       1 1 1  11  1  1  I 35
 2805I1 1   1      11  1 11111111       1                                   I 15
 2806I1 11  1      11111 11111111       11    1 11 1               11       I 25
 2902I1 111 1  1  11111 11111111        11  111111 1       1 1 1  11  1  1  I 35
 6802I1 11  1      111 1 11111111       11   111 1 1           1  11  1     I 27
 6806I 1   1 1     111 11 111 11            1 1 1 1            1  1     I 19
 6807I1 1 1 1  1     1  11111111  1 1  11  1 1 1           1 1  1  1        I 25
 6810I1 111 1 111    1  11 11 11  1111111  1 1 1  1        1 1  11 11 1     I 33
 6809I1 111 1  1     1  11111111  1 1  11  1 1 11 1        1 1  1  1  1     I 29
  901I 1   1 1   1     1              11                                    I  7
 4403I  1 1  111     1  1   1     11111    1 1 1  1  1  1 1 1 111111 1     I 28
 2807I 1   1 1     111 11  1  1           1 1 1 1              1   1    I 16
  110I11 1  1      111 11 111 111       11   1 1 1 1               1        I 22
 4404I 1 1   1     111 11 111 11        1                                   I 14
  604I 1   1 1   1     11               1   1                   1         I  9
 2604I 1   1 1   1     1              11                                    I  7
 2901I            1          11                 1    111111                 I 10
 6813I 1   1 1   1     1              11                                    I  7
  113I1     1      11  1 1 11 111       1                                   I 12
  505I 1    1      111 11 11  11                                            I 11
  307I 1   1 1   1     1               1                                    I  6
 2408I 1   1 1   1     1               1                                    I  6
 2803* 1   1 1   1     1               1                                    I  6
  806I                                                              1       I  1
-----------------------------------------------------------------------------------
     I22111121  11 22112212221222  1 1  21 1111111 1        1 1    1 11  1  I
     I 1572618772131193219344545 12 7 772681237538371 1522311 2 39772366 11  I
```

Fig. 7

logical approach, in such a case, would be to select first the pieces having the least number of combinations.

In the summary table, Fig. 7, it is observed that piece (8-6) was scheduled only once. In checking for a complementary piece, it was noted that piece (26-52) was also combined only once. Thus, this split run combination would be the first choice.

After making the first selection, the next step would be to eliminate these two pieces from the summary table. New totals are then computed and the next least combined piece is chosen. In this manner the complete selection is made. This method is time-consuming and does not take into consideration the cost factor at all. For example, if the schedule maker is faced with two split run alternatives to choose from, he has no ready means of knowing which is the most economical.

COMPUTER METHOD - TOTAL TIME BASIS

In order to computerize the selection and to incorporate the cost factor, the following approach was adopted. The criterion used in this approach was that of selecting the least costly combination first.

The initial step in this approach was to arrange all the 803 split runs in their ascending order with respect to total pay time. This was done by means of a computer program. To explain the steps, the following example is described in detail.

Assume that three pieces, A, B, and C, were fed into the machine in the following order:

Piece	Pay time
A	800
B	840
C	820

1. The machine stores this information in the memory storage area.

2. The pay time of pieces A and B is compared. If the computer finds that the pay time of A is greater than that of B, it then interchanges their positions. In this case, A is less than B so the machine moved to the next two pieces, B and C.

3. On comparing these pieces, the machine finds that piece B has a larger pay time. It interchanges the position of pieces B and C. The order of the pieces in the storage area is now shown below:

Piece	Pay time
A	800
C	820
B	840

4. On completing the comparison of the last pair, the machine starts with the first two pieces again. It now compares A and C, then C and B. As no interchange takes place, the arranging is complete.

5. The arranged pieces are punched out with serial numbers assigned by the computer.

After the split run input was arranged and punched out, the machine shifted to manual control. At this stage the schedule maker could instruct the computer to select the split run of his choice. This permits the inclusion of a particular split run which has to be used in the schedule. Upon receiving this information, the machine goes about selecting the least costly combinations first. The program has checks to ensure that no piece of work is used more than once. As soon as one selection is complete, the information is punched out and the computer shifts back to manual control, ready to make another selection. The selection cycles take only ten minutes.

The results from this approach are given in Fig. 8. On comparing these with those selected by the company, it was observed that all but eight pieces are the same. In place of the eight pieces not used by the company, other pieces of similar pull-out time, pull-in time, and trip times were utilized. There was a saving of almost eleven hours or 4.1 per cent in the schedule. The company-selected runs paid 268.93 hours whereas the computer selections paid only 258.22 hours.

COMPUTER METHOD - MODIFIED SUMMARY TABLE

In spite of saving almost eleven hours by using the preceiding method, it was felt that a little more flexibility in making a selection was needed. Almost every transit company has a few routes on which it invariably makes a loss. In order to minimize the losses, the company attempts to use the least costly split run combinations on such routes. This modified summary table approach was so developed that split runs from such blocks can be selected first. The selection depends on the order in which the blocks are fed into the machine.

In the summary table, Fig. 7, no distinction was made on the basis of pay time. The letter "1" simply indicated that a split run had been developed. To improve on this, a new modified summary table was constructed.

The new modified table is shown in Fig. 9. Prior to feeding the split runs into the computer to construct the new table, the data had to be ranked. This ranking was done on the machine in the following manner:

1. All the split run combinations were read by the machine and

SELECTION BASED ON TOTAL PAYTIME

AM/P	PM/P	PAY
6802	107	800
2406	302	800
806	2652	800
2602	6801	802
503	4401	816
311	104	818
2807	2603	836
6810	403	840
2404	2851	845
2902	2460	852
2604	408	855
4403	6859	857
2405	751	858
305	553	859
502	459	859
112	404	864
701	106	865
2408	2805	868
6809	1051	871
6806	801	881
1003	101	881
1001	461	887
304	160	890
306	556	896
309	6857	897
111	301	900
2806	557	901
901	402	904
6807	463	908
504	501	912

TOTAL= 258.22 HOURS

Fig. 8

```
I                                  2 2 2 2 2 4 6 6 6 6 4          2 4   2       2 2           1 1 2 2 2 6 6 6       1 1 2 2 6 2     I
I 1 1 1 1 1 1 1 3 3 3 4 4 4 4 4 4 5 5 5 7 8 8 4 4 6 6 8 4 8 8 8 8 4 1 1 3   8 8 4 6 8 3 3 3 8 9 1 3 3 4 5 5     4 4 8 8 8 8 1 1 3 4       6 9 8 8   I
I           5 6 5 5 5           5 6 6     5 5 5       5 6     5 5 5       5 5 5     5     5     4 5 5     5 6 3 6     5 5 5 5     5 5 1 5 5     6     6 5 5 5 5 6     I
I 1 3 6 7 8     7 8 9 2 4 8 9 1 3 1 3 7 1 1 2 4     3 1 1 4 1 5 6 7 3 5 4 2   6 3 1 4 2 8 4 6 4 2 4 3 1 3 5 6 1 2 9 7 5 1 8 9 9 2 1     3 4 2 1     5   I
----------------------------------------------------------------------------------------------------------------------------------------------
111I 4        11  20 923    3                 17192122  1513       11218 7  16 6 2                                              14      2410 5 8                 I 24
112I 5       113  221126    4                 19212425  1716       21420 9 718 8 3                                              15      2712 610          23     I 27
113I 4                      3                 1012        8        1  11 7   9 5 2                                                          6                   I 12
304I17 8    124 6  18      16         7     11        29 5  26     325282315272113                      1410   2     4  12     9      192022                   I 29
305I  14        13               5 716    118          17         2                             3          15    9    61110   8  12   4                         I 18
306I  15    9  14  24      23    1 418     21          19          16        22                 7     11  2017   5    310 813 6  12   2  25                     I 25
309I     16       11  1218  17                3 4 910   1  15       8 7 5   6 2                                                    14                  13        I 18
310I17 8    1   6          16         7     11          5          3  222015211913                      1410   2     4  12     9          18                     I 22
311I     19  10  14  1621  20                 6 81113   21218        9 7 4   5 1                                                   17   3  22          15        I 22
502I2413    13111  25      23    3 416     19          17        1232   3022   2821                     1814   2    8 5 715 9   1020 6  262729                   I 32
503I                             810      2                     4                       1       6 3 5                   9  12  11       7                         I 12
504I     3                4   2         5                   1                                                                                    7        6      I  7
505I    10                   11                  3 4 8     1   9      7 6      5 2                                                                              I 11
508I 4      12      9      3                  16181920  1413        11117 7  15 5 2                                                     2110 6 8                 I 21
701I     3                4   2         5                   1                                                                              6        7            I  7
1001I2313    5  12           22    1 316     19          17         14        21      20                 1815   2    7 8 611 9   10     4                         I 23
1002I 4  22  12     10      3                 15171819    91421       716 6  13 2 1                                                  2011 5 8                    I 22
1003I 4      12  201024     3                 17192223  1413        1 918 7  16 5 2                                     15       2511 6 8           21           I 25
2404I     3               4   2         5                   1                                                                                7        6          I  7
2405I    18        13  1520  19                3 71011    21217        9 8 5    6 1                                                  16    4              14         I 20
2406I 4  25  12  181022     3                 15172021    91424       716 6   13 2 1                                                 2311 5 826          19          I 26
2601I     6             810      9        11    2 3 4      1   5                                                                       7                             I 11
2602I14 7   123 5311535  11         6     1028303334 32625       2242922122720 4                 13 9     18          17 8        161921            32           I 35
2805I 5     1                4               1315        11        21014 9 712 8 3                                                      6                       I 15
2806I 6     115     12       5              720222324 41817        2162114 81910 3                                                 2513 911                       I 25
2902I14 7   123 5311535  11         6     1028303334 32625       2242922122720 4                 13 9     18          17 8        161921            32           I 35
6802I 5     113  221126      4              19212425   1716        21420 9 618 8 3                                        15       2712 710          23          I 27
6806I    17        12  1419  18                3 4 910    11116       8 7 5   6 2                                                  15                  13          I 19
6807I17 8   1   6  18      16         7     11          5          323252214242012                 1510   2     4  13     9          1921                       I 25
6810I2412    23111  25      23    3 416     19          17  33     1332   3022   2821              1814   1    8 5 715 9   1020 6   262729                      I 33
6809I17 8   124 6  18      16         7     11        29 5  26     325282314272112                 1510   2     4  13     9         192022                       I 29
901I     3                4   2         5                   1                                                                                7        6          I  7
4403I  16  11  15  26          2 419     23          20            17        24        22       7     9  2118    625 310 813 5271214 1  28                        I 28
307I     3                4   2         5                   1                                                                                         6          I  6
2807I    13        9  1015  14                 2 3 6 7    1  12        5       4                                                       8             1611        I 16
110I 4   22  11  16  19     3                 12141718    71521        813 6   9 2 1                                                   20    510                 I 22
4404I    13  10              14                 6 911      2  12        7 8 4    5 1                                                       3                     I 14
604I     3            6 7    5         8                1   2                                                                          4            9            I  9
2408I     3               4   2         5                                                                                                          6    1        I  6
2604I     3               4   2         5                   1                                                                              7        6            I  7
2803I     3               4   2         5                   1                                                                                       6            I  6
2901I                                    3                      8           1   2   6 9 4 510 7                                                                     I 10
6813I     3               4   2         5                   1                                                                              6        7            I  7
806I                                                                                                                                                 1           I  1
----------------------------------------------------------------------------------------------------------------------------------------------
I 2 1 2 1 1 1 1 1 1 1 2 1       1 1   1 2 2 1 1 1 2 1 2   1 2 2 2 1 2 2 2                   1 1   1     1           1     1 1 2 1         1 1     I
I   2 1 5 7 2 3 7 5 6 1 8 7 7 2 1 3 3 1 1 9 8 3 2 7   3 9 3 4 4 5 4 5   1 1 2   1 5 2 2 3 1 1 2     3 7   7 9 7 7 2 3 7 6 6 2 6 8   1 1 1 1     I
```

Fig. 9

stored in the memory section.

2. The computer then picked the first block used and found all the split run combinations made in that block. These were arranged with respect to total pay time. A program similar to the one described under Computer Method - Total Pay time Basis was used.

3. The arranged and ranked output is punched out. In a similar fashion all the split runs were arranged and ranked within each block.

4. This arranged and ranked output was used in making the modified summary table, Fig. 9..

The program for constructing the modified summary table is given in the appendix. The table is constructed as follows:

1. The machine reads all the pieces of work used for the construction of the split runs.

2. The A. M. pieces are arranged along the vertical axis of the table and the P. M. pieces along the horizontal axis.

3. Each split run is now read. The machine locates the cell contained by the two pieces used in this run and puts the rank into the storage area. Similarly, all the split runs are arranged in the memory of the machine in a matrix form.

4. On reading the last split run, the machine calculates the number of combinations constructed with each piece. The table is then punched out. This is shown in Fig. 9..

The modified summary table made selection easier and gave a ready means of telling which alternative was least costly.

A program to make the selection was written. On receiving the instruction of the first split run to select, the machine moved from block to block, picking the least costly combination possible. The permutation and combinations of the various blocks would yield innumerable selections. The piece (68-9) was combined 29 times. Twenty-nine sets of selections were made, each having one combination of piece (68-9) as the first choice. On an average, 38 split runs were selected in each set. The total time varied from a minimum of 333.10 hours to a maximum of 335.30 hours.

A comparison between the least costly selection and the company's set is shown in Fig. 10. Thirteen pieces of work used in the company's schedule were not included in the computer solution. The total time paid by the company was 268.93 hours as against 251.03 hours for the computer selection. This shows a saving of almost 18 hours or 6.7 per cent.

COMPUTER METHOD - LINEAR PROGRAMMING APPROACH

This method of selection was based on the "Assignment Problem Technique"[7] used in linear programming. An assignment problem is one in which there is a matrix of effectiveness which associates each of a number of origins with each of the same number of destinations. The desired solution is an optimal assignment wherein an origin is associated with one and only one destination in such a way as to minimize (maximize) the summed effectiveness. This technique is demonstrated by the following example.

A situation exists where there are four job vacancies and an equal

---

7. Sasieni, M., <u>Operations Research Methods and Problems</u>. New York: John Wiley and Sons, Inc., 1959, pp. 183-192.

COMPARISON OF RESULTS

COMPANY SELECTION VS MODIFIED TABLE MACHINE SELECTED

MACHINE SELECTED			COMPANY SELECTED		
AM/P	PM/P	PAY	AM/P	PM/P	PAY
6809	107	803	6802	404	818
2806	6801	836	310	107	838
304	403	846	503	553	849
503	4401	816	2902	160	861
6807	1051	888	2405	801	864
311	104	818	901	408	868
6802	302	800	113	6801	871
1003	404	873	1001	459	874
111	101	886	505	2603	877
2805	301	935	311	2454	883
508	6857	916	2404	501	884
2807	2503	836	6806	6856	885
310	2460	910	2602	358	893
2405	751	858	6807	2460	895
604	2851	892	2806	101	897
6810	459	850	2406	357	900
2406	6805	854	304	557	901
2901	806	800	701	402	901
2902	160	861	502	103	904
2602	463	842	6810	463	910
112	4453	889	309	802	912
110	105	919	6809	4453	914
2404	408	855	305	4454	929
2604	106	864	111	6805	932
502	461	873	112	2651	934
701	402	901	2601	106	935
1002	460	932	508	158	937
504	501	912	1002	751	950
4403	801	881	2805	6857	950
6813	1053	946	1003	359	1018

TOTAL= 251.03 HOURS   TOTAL= 268.93 HOURS

Fig. 10

number of eligible applicants. Each applicant states his preference for each job by ranking the job 1 for his first choice and so on to 4, his last choice. Similarly, the four supervisors state their preferences for each applicant in each job. The result is the following table[8].

Jobs

Applicants		A	B	C	D
	a	1, 3	2, 2	3, 1	4, 3
	b	1, 4	2, 3	3, 2	4, 4
	c	3, 1	1, 4	2, 3	4, 2
	d	2, 2	3, 1	1, 4	4, 1

(The first digit indicates applicant's preference, the second, the job preference for each applicant as given by the supervisors.)

To reach the optimum solution, the following steps are required:

1. Add the two ranks as shown in the matrix.

	A	B	C	D
a	4	4	4	7
b	5	5	5	8
c	4	5	5	6
d	4	4	5	5

2. Subtract the minimum cell in each row from all the elements in its row. This results in the following matrix.

	A	B	C	D
a	0	0	0	3
b	0	0	0	3
c	0	1	1	2
d	0	0	1	1

3. Subtract the minimum cell in each column resulting from step 2 from all the elements in its column.

	A	B	C	D
a	0	0	0	2
b	0	0	0	2
c	0	1	1	1
d	0	0	1	0

4a. The row or column having only one zero is selected as this is the only possible assignment that can be made. After selecting the first assignment, the row and column enclosing this cell (D, d) are deleted from the matrix.

	A	B	C	D
a	0	0	0	2
b	0	0	0	2
c	0	1	1	1
d	0	0	1	0

4b. This step is repeated and the cell (A, c) is assigned.

	A	B	C	D
a	0	0	0	2
b	0	0	0	2
c	0	1	1	1
d	0	0	1	0

5. The remaining matrix contains all zeros. This indicates the possibility of two optimal solutions, either (B, a) and (C, b) or (B, b) and (C, a).

In summary, there are two final optimal solutions:

First Solution: Assign Job A to applicant c
Job B to applicant b
Job C to applicant a
Job D to applicant d

Second Solution: Assign Job A to applicant c
Job B to applicant a
Job C to applicant b
Job D to applicant d

Comparing this example with our selection problem, it will be noticed that the applicants are analogous to the A. M. pieces and the jobs analogous to the P. M. pieces. The objective in the example was to find the most satisfactory combination of applicant and job assignment; the goal of the selection problem was to combine the A. M. piece with the P. M. piece that gave the least costly combination of split runs.

The similarity between the problem of developing the split run selection and this example led the author to investigate the application of the assignment problem technique. The split run data was ranked horizontally (Fig. 9) and vertically (Fig. 11). Steps similar to those used in solving the example[9] in the text (assignment problem technique) were performed. A solution containing 30 split runs was obtained. When compared with the company's selection of 30 split runs, it was noted that this solution paid a total time of 261.99 hours as against 268.93 hours paid by the company. This was a saving of approximately seven hours or 2.6 per cent per day.

9. Ibid.

```
 I                                          2 2 2 2 2 4 6 6 6 6 4             2 4   2         2 2              1 1 2 2 2 6 6 6           1 1 2 2 6 2    I
 I 1 1 1 1 1 1 1 3 3 3 4 4 4 4 4 4 5 5 5 7 8 8 4 4 6 6 8 4 8 8 8 8 4 1 1 3  8 8 4 6 8 3 3 3 8 9 1 3 3 4 5 5   4 4 8 8 8 8 1 1 3 4     6 9 8 8    I
 I           5 6 5 5 5           5 6 6   5 5 5     5 6   5 5 5     5 5 5      5   5   4 5 5   5 6 3 6   5 5 5 5   5 5 1 5 5   6   6 5 5 5 5 6        I
 I 1 3 6 7 8   7 8 9 2 4 8 9 1 3 1 3 7 1 1 2 4   3 1 1 4 1 5 6 7 3 5 4 2  6 3 1 4 2 8 4 6 4 2 4 3 1 3 5 6 1 2 9 7 5 1 8 9 9 2 1   3 4 2 1   5    I
------------------------------------------------------------------------------------------------------------------------------------------------------------
 111I 6       6  11 713    6              15151512  16 9     6131411  1413 6                                   6      14 7 7 6             I 24
 112I 3     3 4  10 412    4              14141411  15 7     4 913 9 41310 3                                   4      13 4 5 3         9   I 27
 113I13                   13              2121       22      12  2018  201913                                                16                 I 12
 304I15 4   715 4  12     15       4      5       18 4  16   11202220 7222115                    5 4  3    3   3    5    121813                 I 29
 305I  12       12             6 612   313          13          1                          3       12  10   610 6   6  12   6                   I 18
 306I  11  15  11  17     21   5 511    12          12       19        15                  2     2 1111  9   4 9 5 9 4  11   4  16              I 25
 309I   16       7   616  16              6 6 5 6    6  15     6 6 6   6 6                                                 7               7   I 18
 310I17 8  10    8        17        8     7          7       14  242210242317                    8 8  6    6   6     8          20              I 22
 311I   13   1   1   113  13              1 1 1 1    1 112      1 1 1   1 1                                                3   1   7       1   I 22
 502I19 6  1217 6  15     19   2 2 7     9          9        1623  2412  2519                    6 6  4    3 4 2 4 3   6 2 3  142216             I 32
 503I                          7 7     2                     3                       2     5 2 2               7   7   7      7                  I 12
 504I   4              5   4         5                     4                                                                       4     5     I  7
 505I   18                  17                8 8 8     9  17   11 7       811                                                                   I 11
 508I 8       9       8    9               16161613   1711     7141513   1514 8                                              15 8 9 8           I 21
 701I   3              3   3        3                      3                                                                       3     3     I  7
1001I2010  13   10        20    4 410    11          11       18        14     20                 1010   8    5 8 4 8 5   10     5              I 23
1002I 9  21   10      9   10              121212 9   131020     101110   11 810                                               11 911 9          I 22
1003I 5       5   9 611   5               13131310   14 6     5 812 7   12 9 5                                 5              12 6 6 5      8   I 25
2404I   1              1   1        1                       1                                                                   1     1        I  7
2405I   14       3   414  14              2 2 2 2    4 413     3 2 4    2 3                                                   4   4        3   I 20
2406I 2  17   2    2 2 7   2              5 5 6 4    7 216     2 5 2    3 2 1                                                 8 2 2 2 8    2   I 26
2601I    11          811  11        11    7 7 7      8  10                                                                    5                I 11
2602I 4 1  4 8 112 314   3         1     117171714 11812     2151614  21615 4                    1 1    1          2 1         310 4       10   I 35
2805I12     9             12               2020      21      9181917 9191812                                                    15                I 15
2806I10     612     10    11              418181815 51913    8161715 6171611                                               16101310                I 25
2902I11 2   213  213 515   8       2     219191916 22014     3171816 31817 9                    2 2    2          3 2          51411       11   I 35
6802I 1      1 3    6 1 9   1              9 9 9 7   10 3    1 4 8 3 1 7 4 2                                   1          9 1 3 1       6   I 27
6806I    15       5  515  15              4 4 4 5    5 514     5 4 5    5 5                                                 6                 5   I 19
6807I16 7   8   7  13     16       5     6         6         13212321 8232216                    7 7  5    5   5     7         1914             I 25
6810I18 5  1116 5  14     18    1 1 6    8         8  17     1522  2311  2418                    3 5  1   2 1 1 1 2   3 1 2   132115             I 33
6809I14 3   514 3  11     14       3     3       17 3  15    10192119 5212014                    4 3  2     2   2     4       111712             I 29
 901I     6            4   6        4                     6                                                                         6    4     I  7
4403I    9  14    9  16         3 3 9    10        10          17       13       1       1     1    9 9    7 3 1 7 3 7 1 7 9 3 1   15                I 28
 307I    8            8   8        8                       7                                                                              8     I  6
2807I    12        4   212  12            3 3 3 3    2  11       3     4                                                    1          11 4    I 16
 110I 7  20    7   8  10   7              111111 8  12 819     710 8   10 7 7                                               10   8 7           I 22
4404I    19  11             18             101010      11  18   12 912     912                                                    12           I 14
 604I     9          310  10         10                3   9                                                               2           10       I  9
2408I     7             6   7        6                                                                                                   6   1   I  6
2604I     2             2   2        2                     2                                                                   2       2       I  7
2803I    10             9   9        9                     8                                                                           9       I  6
2901I                               1                        2       1   1   1 4 1 1 3 1                                                        I 10
6813I      5             7   5        7                     5                                                                   5      7       I  7
 806I                                                                                                                                   1      I  1
------------------------------------------------------------------------------------------------------------------------------------------------------------
 I 2 1 2 1 1 1 1 1 1 1 2 1     1 1   1 2 2 1 1 1 2 1 2   1 2 2 2 1 2 2 2                1 1   1     1         1   1 1 2 1       1 1        I
 I   2 1 5 7 2 3 7 5 6 1 8 7 7 2 1 3 3 1 1 9 8 3 2 7   3 9 3 4 4 5 4 5    1 1 2   1 5 2 2 3 1 1 2     3 7   7 9 7 7 2 3 7 6 6 2 6 8   1 1 1 1    I
```

Fig. 11

To utilize the computer on this assignment problem technique approach was not possible due to the nonsymmetry of the effectiveness matrix and the limited capacity of the machine. However, upon studying the basis of the technique, it was felt that the following approach would yield equally good results.

The horizontal and vertical ranks were added as before. However, the decision was made by first selecting the split run combination which ranked the least. This, in effect, was the same as selecting the cells having all zeros in the effectiveness matrix. A set of 30 split runs was selected using this procedure. The total pay time for this combination was 259.51 hours comparing withing half a per cent ($\frac{1}{2}\%$) of those obtained by using the assignment problem technique.

The closeness of these two results prompted the author to test another alternative approach. This time the horizontal and vertical ranks were multiplied and a decision again made by selecting the combination ranked the least. Figure 12 shows the results of using the assignment problem technique, adding approach and multiplying approach. As seen from the totals of the pay times, there is a variation of less than half a per cent ($\frac{1}{2}\%$) in these solutions. It is therefore felt that either of the methods could be used. These methods, the adding approach and the multiplying approach, involve little or no manual work as computer programs have been written to do all the steps from ranking to the final selection.

SUMMARY AND CONCLUSIONS

The computer method, on comparison, was found to be more advantageous than the manual method in the following respects:

ASSIGNMENT PROBLEM TECHNIQUE RESULTS

METHODS

ASSIGNMENT APPROACH			ADDING APPROACH			MULTIPLYING APPROACH		
AM/P	PM/P	PAY	AM/P	PM/P	PAY	AM/P	PM/P	PAY
806	2652	800	311	104	818	311	104	818
112	107	800	6802	107	800	6802	107	800
304	403	846	6810	403	840	6810	403	840
311	104	818	2406	302	800	2406	302	800
502	459	859	2901	806	800	2901	806	800
508	6801	833	2404	2851	845	2404	2851	845
2406	302	800	4403	6859	857	4403	6859	857
307	2851	874	503	4401	816	503	4401	816
309	801	893	2807	2603	836	2807	2603	836
310	1051	903	305	4454	929	305	4454	929
1002	460	932	2602	2460	829	2602	2460	829
2806	557	901	2604	408	855	2408	2805	868
604	2603	839	2408	2805	868	2604	408	855
505	105	914	2405	751	858	1001	459	874
2409	2805	868	2902	6801	810	1003	6801	820
305	4454	929	502	459	859	2405	751	858
504	408	866	6809	1051	871	6809	1051	871
2404	106	864	604	109	914	604	109	914
6806	2651	920	701	106	865	502	461	873
4403	6859	949	2601	802	919	701	106	865
111	404	878	1001	461	887	2902	160	861
2602	2460	829	306	556	896	2601	802	919
503	4401	816	1003	404	873	306	556	896
2901	2802	907	112	101	872	112	101	872
306	556	896	6806	801	881	6806	801	881
2807	751	864	901	402	904	901	402	904
901	402	904	304	160	890	111	404	878
701	1053	935	504	501	912	504	501	912
2601	802	919	2806	557	901	2806	557	901
6802	4453	843	6813	1053	946	304	463	894
TOTAL=	261.99		TOTAL=	259.51		TOTAL=	258.86	

Fig. 12

1. Manual work, both computational and clerical, was reduced to the bare minimum.

2. The computer solution not only developed the company's present schedule but also several alternatives. The speed with which the computer programs work enables the schedule maker to incorporate changes in schedules almost instantaneously.

3. The techniques developed for choosing split runs were based on the cost criterion, which is the main objective.

4. The time taken in preparing the schedule for the Cincinnati Transit Company with the computer was approximately three hours as against several days when manual methods were used.

In conclusion, the author feels confident that this computer method will give efficient and effective schedules and may possibly be an important step in solving a segment of the transit industry's crisis.

The author feels that, although this method looks efficient, there is still ample room for improvement. An interesting problem would be to develope a program with which it would be possible to prepare the complete schedule in one step as against the three steps used in this report.

ACKNOWLEDGMENTS

The writer gratefully acknowledges the guidance and encouragement of his major professor, Dr. Samy E. G. Elias during the execution of the entire project. Appreciation is also expressed to the Cincinnati Transit Company, Ohio, for making available the data used in this investigation.

REFERENCES

Elias, Samy E. G., "A Digital Computer Solution to the Transit Operation Assignment Problem." Unpublished Doctoral Thesis, Oklahoma State University, 1960.

Elias, Samy E. G., "The Use of Digital Computers in the Economic Scheduling for Both Man and Machine in Public Transportation." Special Report 49, Kansas State University Engineering Experiment Station.

Lassow, Bill, "Why Do We Need Transportation Research?" Metropolitan Transportation and Planning, November, 1963, Vol. 59:6, pp. 20-22.

Sasieni, M., Operations Research Methods and Problems. New York: John Wiley and Sons, Inc., 1959, pp. 183-192.

Teichroew, Daniel, An Introduction to Management Science - Deterministic Models, New York: John Wiley and Sons, Inc., 1964, pp. 520-526.

"What Transit Companies are Doing to Improve Ridership." A geographical analysis made by Metropolitan Transportation and Planning researchers. Metropolitan Transportation and Planning, July, 1963, Vol. 59:4, pp. 34-35.

APPENDIX

```
      PROGRAM NO. 1

      CONVERSION PROGRAM FOR HOURS MINUTES TO HOURS DECIMALS
      DIMENSION REL(100)
 1111 READ1112,TRAVEL
 1112 FORMAT(F2.0)
      TRAVEL=TIME(TRAVEL,1.667)
  111 N1=1
      N2=13
      READ1,LINE,NOTRIP,POUT,PIN,(REL(I),I=N1,N2)
    1 FORMAT(2I2,15F5.0)
      IF(LINE)100,100,2
    2 IF(REL(N2))3,4,3
    3 IF(REL(N2)-PIN)5,6,5
    5 N1=1+N2
      N2=N2+15
      READ1,LINE,NOTRIP,(REL(I),I=N1,N2)
      GO TO 2
    4 DO 7 I=1,N2
      IF(REL(I))7,8,7
    7 CONTINUE
    8 I=I-1
      GO TO 9
    6 I=N2
    9 DO 10 J=1,I
   10 REL(J)=TIME(REL(J),1.667)
      POUTC=TIME(POUT,1.667)
      POINC=TIME(PIN, 1.667)
   17 PUNCH18,I,TRAVEL,LINE,NOTRIP,POUTC,POINC,(REL(J),J=1,I)
       GO TO 111
  100 PAUSE
      GO TO 1111
   18 FORMAT(I3,F5.1,2I3,11F6.0/26X,9F6.0/26X,9F6.0/26X,9F6.0/26X,9F6.0/
     126X,9F6.0/26X,9F6.0)
      END
** TIME FUNCTION
      FUNCTION TIME(X,Y)
      IF(X)1,2,2
    1 X=-X+1200.
    2 IX=X/100.
      HR=IX*100
      TIME=(X-HR)*Y+HR
      RETURN
      END
```

PROGRAM NO. 2

A STRAIGHT RUN MAKING

```
      DIMENSION  REL(100)
    1 K=0
    2 TURN=8.33
      TF=0
      TB=0
   10 DO 201 I=1,100
  201 REL(I)=0
      PUNCH 202
  202 FORMAT(//)
      READ11,M,TRAVEL,ROUTE,BLOCK,TIMON,TIMOFF,(REL(L),L=1,M)
   14 BACK=0
      FRONT=0
      POUT=TIMON
      POIN=TIMOFF
   18 EIGHT=800.-TURN
   20 ALLOW=0
   21 IF(TIMOFF-TIMON-800.)24,24,22
   22 BACK=1
   24 IF(TIMOFF-TIMON)10,10,25
   25 TRIPT=TIMOFF-TIMON
   26 IF(TIMON-POUT)27,28,27
   27 TF=TRAVEL
   28 IF(TIMOFF-POIN)29,30,29
   29 TB=TRAVEL
   30 IF(TRIPT-700.)31,136,136
  136 IF(TRIPT-800.)36,36,150
  150 IF(FRONT)9000,50,200
   31 IF(TRIPT-190.)331,331,431
  331 PUNCH 332,ROUTE,BLOCK,TF,TIMON,TIMOFF,TB,TRIPT
  332 FORMAT(9X,2I4,F5.1,2I6,F5.1,5X,I6,29X,1H-)
      GO TO 33
  431 PUNCH32,ROUTE,BLOCK,TF,TIMON,TIMOFF,TB,TRIPT
   33 TF=0
      TB=0
      IF(BACK)9000,10,100
   36 IF(TIMON-POUT)37,38,37
   37 TF=TRAVEL
   38 IF(TIMOFF-POIN)39,40,39
   39 TB=TRAVEL
   40 ALLOW=TF+TB
      K=K+1
      TOTAL=TRIPT+TURN+TF+TB
   42 PUNCH43,K,ROUTE,BLOCK,TF,TIMON,TIMOFF,TB,TRIPT,ALLOW,TURN,TOTAL
      TF=0
      TB=0
      FRONT=1
      IF(BACK)9000,10,100
```

```
50 PLAT=TIMON+800.-TRAVEL-TURN
51 DO 53 L=1,M
52 IF(PLAT-REL(L))65,54,53
53 CONTINUE
   IF(L-M)53,31,31
54 TB=TRAVEL
55 TOTAL=REL(L)-TIMON+TURN+TRAVEL
56 ALLOW=TF+TB
   TRIPT=REL(L)-TIMON

   K=K+1
59 PUNCH43,K,ROUTE,BLOCK,TF,TIMON,REL(L),TB,TRIPT,ALLOW,TURN,TOTAL
   FRONT=1
   TF=0
   TB=0
   TIMON=REL(L)
63 EIGHT=800.-TURN-TRAVEL
   GO TO 24
65 PWO=(REL(L)-PLAT)*.5
66 PWP=PLAT-REL(L-1)
67 IF(PWO-PWP)68,68,74
68 K=K+1
   TB=TRAVEL
   ALLOW=TF+TB
   TOTAL=REL(L)-TIMON+TURN+TF+TB
   TRIPT=REL(L)-TIMON
   GO TO 59
74 K=K+1
   TB=TRAVEL
   ALLOW=TF+TB+PWP
   TOTAL=REL(L-1)-TIMON+TURN+ALLOW
   TRIPT=REL(L-1)-TIMON
   PUNCH43,K,ROUTE,BLOCK,TF,TIMON,REL(L-1),TB,TRIPT,ALLOW,TURN,TOTAL
   TIMON=REL(L-1)
   TB=0
   FRONT=1
   GO TO 63
99 TB=TRAVEL
100 TIMON=POUT
   TIMOFF=POIN
200 PLAT=TIMOFF-800.+TRAVEL+TURN
101 DO 103 L=1,M
   IF(PLAT-REL(L))121,104,103
103 CONTINUE
   IF(L-M)103,31,31
104 TF=TRAVEL
   ALLOW=TF+TB
   TOTAL=800.
```

```
      K=K+1
      TRIPT=TIMOFF-REL(L)
  109 PUNCH 43,K,ROUTE,BLOCK,TF,REL(L),TIMOFF,TB,TRIPT,ALLOW,TURN,TOTAL
      FRONT=0
  111 TF=0
  113 TIMOFF=REL(L)
  114 TIME=TIMOFF-TIMON
  115 IF(TIME)9000,10,116
  116  IF(TIME-(800.-TRAVEL-TRAVEL))117,119,199
  199  TB=TRAVEL
      GO TO 50
  117 BACK=0
      GO TO 25
  119 BACK=0
  120 GO TO 36
  121 PWO=(PLAT-REL(L-1))*.5
  122 PWP=REL(L)-PLAT
  123 IF(PWO-PWP)124,124,134
  124 TF=TRAVEL
      ALLOW=TF+TB
      TOTAL=TIMOFF-REL(L-1)+TURN+ALLOW
      K=K+1
      TRIPT=TIMOFF-REL(L-1)
      PUNCH 43,K,ROUTE,BLOCK,TF,REL(L-1),TIMOFF,TB,TRIPT,ALLOW,TURN,TOTA
     1L
      FRONT=0
      TF=0
      TIMOFF=REL(L-1)
      GO TO 114
  134 TF=TRAVEL
      ALLOW=TF+TB+PWP
      TOTAL=TIMOFF-REL(L)+TURN+ALLOW
      K=K+1
      TRIPT=TIMOFF-REL(L)
      GO TO 109
 9000 STOP
   32  FORMAT(9X,2I4,F5.1,2I6,F5.1,5X,I6)
   43 FORMAT(I5,6X,2I4,F5.1,2I6,F5.1,5X,I6,2F6.1,F7.0)
   11 FORMAT(I5,F5.1,2F3.0,11F6.0/26X,9F6.0/26X,9F6.0/26X,9F6.0/26X,9F6.
     10/26X,9F6.0)
      END
```

PROGRAM NO. 3

B SPLIT RUN MAKING

```
      DIMENSION ROUTE(200),BLOCK(200),TF(200),TIMON(200),TIMOFF(200),TB(
     1200),TRIPT(200)
    1 READ2,IPW
    2 FORMAT(3XI3)
      TURN2=16.67
    3 TURN=8.33
    4 FIVE=8.33
      M=0
    6 DO 9 I=1,IPW
    7  READ8,ROUTE(I),BLOCK(I),TF(I),TIMON(I),TIMOFF(I),TB(I),TRIPT(I)
      IF(TIMON(I)-1100.)5,5,13
    5 IK=IK+1
      IAM(IK)=ROUTE(I)*100.+BLOCK(I)
      GO TO 9
   13 JK=JK+1
       IPM(JK)=ROUTE(I)*100.+BLOCK(I)
    9 CONTINUE
   10  L=1
   11 DO 1000 J=1,IPW
   12  PENLT1=0
      PENLT2=0
      PENLT3=0
   16 IF(TIMON(J)-TIMON(L))1000,1000,17
   17 IF(TIMOFF(J)-TIMOFF(L))1000,1000,18
   18 IF(TIMON(J)-TIMOFF(L))1000,1000,19
      N=0
   19 GAP=TIMON(J)-TIMOFF(L)
   20 BREAK=TB(L)+TF(J)+FIVE
   21 IF(GAP-BREAK)1000,22,22
   22 IF(GAP-100.)500,23,23
   23 SPREAD=TIMOFF(J)-TIMON(L)+TF(L)+TB(J)+TURN
   24 IF(1400.-SPREAD)1000,25,124
  124 IF(SPREAD-1100.)26,26,25
   25 PENLT3=(SPREAD-1100.)
   26 PW1=TRIPT(L)+TF(L)+TB(L)
   27 PW2=TRIPT(J)+TF(J)+TB(J)
   28 IF(PW1-100.)29,31,31
   29 PENLT1=100.-PW1
   30 PW1=100.
   31 IF(PW2-100.)32,34,34
   32 PENLT2=100.-PW2
   33 PW2=100.
   34 TOTAL=PW1+PW2+2.*TURN
   35 IF(TOTAL-750.)1000,36,45
   36 PENLTY=50.
   37 PAYTIM=TOTAL+PENLTY
      ALLOW=PENLTY+PENLT1+PENLT2+ TF(L)+TB(J) +TB(L)+TF(J)
      M=M+1
   40 PUNCH41,ROUTE(L),BLOCK(L), TF(L),TIMON(L),TIMOFF(L),TB(L),TRIPT(L)
```

```
 42 PUNCH43,M,ROUTE(J),BLOCK(J),TF(J),TIMON(J),TIMOFF(J),TB(J),PENLT3,
   1TRIPT(J),ALLOW,TURN2,PAYTIM
    GO TO 1000
 45 IF(800.-TOTAL)50,46,46
 46 PENLTY=800.-TOTAL
 47 PAYTIM=PENLTY+TOTAL
 48 ALLOW=PENLTY+PENLT1+PENLT2+TF(L)+TB(L)+TF(J)+TB(J)
 49 M=M+1
    GO TO 40
 50 IF(900.-TOTAL)1000,51,51
 51 PENLTY=0
    GO TO 47
500 SUM=TRIPT(L)+TRIPT(J)+TF(L)+TB(J)+TURN+GAP
501 IF(SUM-750.)1000,502,511
502 ALLOW=GAP-TB(L)-TF(J)+800.-SUM
503 N=N+1
    TOTAL=SUM-TURN
    PAYTIM=800.
505 BLOCKX=BLOCK(L)*100.+BLOCK(J)
    ROUTEX=ROUTE(L)*100.+ROUTE(J)
    PUNCH509,N,ROUTEX,BLOCKX,TF(L),TIMON(L),TIMOFF(J),TB(J),TOTAL,ALLO
   1W,TURN,PAYTIM
    GO TO 701
511 IF(SUM-800.)502,502,512
512 IF(SUM-900.)513,513,1000
513 ALLOW=GAP-TB(L)-TF(J)
    N=N+1
    TOTAL=SUM-TURN
    PAYTIM=SUM
    GO TO 505
1000 CONTINUE
    L=L+1
    IF(L-IPW)11,11,1005
1005 STOP
  8 FORMAT(9X,2I4,F5.1,2F6.0,F5.1,5X,F6.0)
 41 FORMAT(3X,2I5,F5.1,2I5,F5.1,7X,F6.0)
509 FORMAT(I5,2I5,F5.1, 2I5,F5.1,7X,  F6.0,2F7.2,F7.1)
 43 FORMAT(I5,2I5,F5.1, 2I5,F5.1,F7.1,F6.0,2F7.2,F7.1)
    END
```

PROGRAM NO. 4

```
C   COMBINING STRAIGHT RUN PIECES AND PIECES OF WORK
      DIMENSION LINEP(200),BLOCKP(200),PPF1(200),POINP(200),POUTP(200),
     1PPF2(200),TRIPTP(200),REL(100),PW(2),OUT(2),PIN(2),FF(2),FB(2)
      DIMENSION PIECE(30)
      STR=500
      KX=1000
      SONOF=16.67
      TEN=16.67
    1 READ2,IPW,NSP,NST
    2 FORMAT(3I4)
      DO 3 I=1,IPW
    4 FORMAT(9X,2I4,F5.1,2F6.0,F5.1,5X,F6.0)
    3 READ4,LINEP(I),BLOCKP(I),PPF1(I),POUTP(I),POINP(I),PPF2(I),TRIPTP(
     1 I)
    5 KNST=0
    6 KNST=KNST+1
    7 IF(KNST-NST)8,8,9000
    8 READ9,M,TRAVEL,LINEO,BLOCKO,POUTO,POINO,(REL(K),K=1,M)
    9 FORMAT(I3,F5.1,3X,2I3,5X,2I6,6X,8I5/40X,8I5)
      PUNCH888,LINEO,BLOCKO
  888 FORMAT(14H  ROUTE  NO.  ,I5,14H  BLOCKO  NO.  ,I5)
      PUNCH777,
      CHECKC=0
      CHECK=0
      CHECKB=0
      KTWOA=0
   10 MONE=1
   11 POUT=POUTO
      POIN=POINO
      FR1S=16.67
      FR2S=8.33
   16 DO1000K=MONE,M
 1016 SPW=REL(K)-POUT
      IF(SPW-150.)1000,1000,18
   18 SPOUT=POUT
      IF(SPW-600.)19,19,116
  116 IF(SPOUT-POUTO)19,6,19
   19 SPIN=REL(K)
      SBLOCK=BLOCKO
      SLINE=LINEO
      SPF1=FR1S
      SPF2=TRAVEL
   24 DO 999 N=1,IPW
   26 PENLT1=0
      PENLT2=0
      PENLT3=0
      PENLT4=0
```

```
30 L=1
   J=2
34 PW(L)=SPW
   OUT(L)=SPOUT
   PIN(L)=SPIN
   FF(L)=SPF1
   FF(J)=PPF1(N)
   FB(L)=SPF2
   FB(J)=PPF2(N)
   PW(J)=TRIPTP(N)
   OUT(J)=POUTP(N)
   PIN(J)=POINP(N)
44 IF(OUT(2)-OUT(1))45,999,49
45 L=2
   J=1
   GO TO 34
49 IF(PIN(2)-PIN(1))999,999,50
50 IF(OUT(2)-PIN(1))999,999,51
51 GAP=(OUT(2)-PIN(1))
52 IF(GAP-TEN)999,53,53
53 SPREAD=PIN(2)-OUT(1)+FF(1)+FB(1)+FF(2)+FB(2)+SONOF/2.
   IF(1325.-SPREAD)999,55,58
55 PENLT1=(SPREAD-1100.)*.5
   SPREAD=1300.
   GO TO 62
58 IF(1250.-SPREAD)55,59,59
59 IF(1200.-SPREAD)60,61,61
60 PENLT1=(SPREAD-1100.)
61 SPREAD=1250.
62 IF(GAP-100.)200,200,63
63 PW1T=PW(1)+FF(1)+FB(1)
   PW2T=PW(2)+FF(2)+FB(2)
   TIME=PW1T+PW2T+SONOF
173 IF(PW1T-100.)73,175,175
73 PENLT2=100.-PW1T
175 IF(PW2T-100.)75,76,76
75 PENLT3=100.-PW2T
76 TIMET=TIME+PENLT2+PENLT3
   IF(TIMET-750.)999,78,82
78 PENLTY=PENLT2+PENLT3+PPF1(N)+PPF2(N)+SPF1+SPF2+50.
   PAYTIM=TIME+PENLT2+PENLT3+50.
   GO TO 90
82 IF(800.-TIMET)86,83,83
83 PENLTY=PENLT2+PENLT3+800.-TIMET+PPF1(N)+PPF2(N)+SPF1+SPF2
   PAYTIM=TIME+PENLT2+PENLT3+ 800.-TIMET
   GO TO 90
86 IF(850.-TIMET)999,87,87
87 PENLTY=PENLT2+PENLT3+PPF1(N)+PPF2(N)+SPF1+SPF2
   PAYTIM=TIME+PENLT2+PENLT3
```

```
 90 KX=KX+1
    MKSX=KX
221 PUNCH79,LINEP(N),BLOCKP(N),PPF1(N),POUTP(N),POINP(N),PPF2(N),
   1TRIPTP(N)
    PUNCH80,MKSX,SLINE,SBLOCK,SPF1,SPOUT,SPIN,SPF2,PENLT1,SPW,PENLTY,
   1SONOF,PAYTIM
 79 FORMAT(4X,2I5,F5.1,2I5,F5.1,7XI5)
 80 FORMAT(I4,2I5,F5.1,2I5,F5.1,F7.1,I5,3F7.1)
    IF(IPW-N)1999,195,1999
195 IF(CHECK)9000,92,401
 92 CHECK=CHECK+1.
    KONE=K
 95 LEFT=POINO-REL(K)
 96 IF(LEFT)9000,99,97
 97 IF(LEFT-200)98,98,103
 98 PUNCH199,LINEO,BLOCKO,REL(K),POINO,LEFT
 99 IF(KONE+3-M)100,6,6
100 CHECK=0
101 MONE=KONE +1
    PUNCH777
777 FORMAT(///)
102 GO TO 11
103 MONE=K+1
    POUT=REL(K)
    POUTB=POUT
    POIN=POINO
    FR1S=TRAVEL
    FR2S=8.33
    GO TO 16
999 CONTINUE
    IF(CHECKC)801,1000,801
801 CHECKC=0
    GO TO 195
1000 CONTINUE
1001 IF(CHECKB)9000,99,408
1999 IF(CHECKC)999,800,999
800 CHECKC=CHECKC+1.
    GO TO 999
9000 STOP
200 SUM=TRIPTP(N)+SPW+FF(1)+FB(2)+GAP+SONOF
    IF(SUM-750.)300,202,209
300 IF(SBLOCK-100.)333,999,999
333 SBLOCK=BLOCKP(N)*100.+SBLOCK
        XLINE=LINEP(N)
    SLINE=XLINE*100.+SLINE
    SPW=PIN(2)-OUT(1)
    SPOUT=OUT(1)
    SPIN=PIN(2)
    SPF1=FF(1)
    SPF2=FB(2)
    GO TO 24
202 PENLTY=800.-SUM +PPF1(N)+PPF2(N)+SPF1+SPF2 +GAP
```

```
      TIME=SUM
      TRIP=TRIPTP(N)+SPW
      STR=STR+1.
      OVERT=0
      PAYTIM=800.
      MKSX=STR
      GO TO 220
  209 IF(800.-SUM)210,202,202
  210 IF(SUM-850.)211,211,999
  211 SFB1=0
      TRIP=TRIPTP(N)+SPW
      SFF2=0
      TIME=SUM
      PENLTY=PPF1(N)+PPF2(N)+SPF1+SPF2 +GAP
      PAYTIM=TIME+FF(2)+FB(1)
      STR=STR+1.
      MKSX=STR
  220 GO TO 221
  401 IF(CHECKB)9000,402,406
  402 CHECKB=CHECKB+1.
      KTWO=K
  404 IF(KTWOA-KTWO)405,499,405
  405 KTWOA=KTWO
  406 LEFT=POINO-REL(K)
      IF(LEFT)9000,408,412
  412 IF(LEFT-200) 498,498,413
  498 PUNCH199, LINEO,BLOCKO,REL(K),POINO,LEFT
  408 IF(KTWO+2-M)409,499,499
  409 CHECKB=0
      MONE=KTWO+1
      POIN=POINO
      POUT=POUTB
      FR1S=TRAVEL
      KTWOA=0
      PUNCH778,
  778 FORMAT(/)
  411 GO TO 16
  413 POUT=REL(K)
      K=M
      REL(K)=POINO
      FR1S=TRAVEL
      GO TO 1016
  499 CHECKB=0
      GO TO 99
  199 FORMAT(4X2I5,5X2I5,5XI5)
      END
```

PROGRAM NO. 5

ARRANGING AND SELECTING PROGRAM - TOTAL PAYTIME

```
      DIMENSION BLOKX(1000),ROUTX(1000),PAYX(1000),BLOK1(100),ROUT1(100)
     1,PAY1(100)
      READ1,NL
    1 FORMAT(I5)
       DO 22 I=1,NL
   22 READ11,BLOKX(I),ROUTX(I),PAYX(I)
   11 FORMAT(4X,2I6,1X,I4)
      KOUNT=NL
      ISUM1=0
      IND=0
      INDEX=KOUNT-1
   13 DO45 I=1,INDEX
      IF(PAYX(I+1)-PAYX(I))16,45,45
   16 SAVE=PAYX(I+1)
      CAVE=BLOKX(I+1)
      RAVE=ROUTX(I+1)
      PAYX(I+1)=PAYX(I)
      BLOKX(I+1)=BLOKX(I)
      ROUTX(I+1)=ROUTX(I)
      PAYX(I)=SAVE
      ROUTX(I)=RAVE
      BLOKX(I)=CAVE
      IHOLD=I
      IND=1
   45 CONTINUE
      IF(IND)9,8,9
    9 INDEX=IHOLD
      IND=0
      GO TO13
    8 DO41I=1,KOUNT
      ISUM1=ISUM1+1
   41 PUNCH7,BLOKX(I),ROUTX(I),PAYX(I),ISUM1
    7 FORMAT(4I6)
  111 READ101,L
  101 FORMAT(I5)
      CHECK=0.0
       MX=0
      DO4 J=1,NL
      IF(BLOKX(L)-BLOKX(J))2,4,2
    2 IF(ROUTX(L)-ROUTX(J))44,4,44
   44 IF(CHECK)15,55,6
    6 DO5 M=1,JK
      IF(ROUTX(J)-ROUT1(M))25,4,25
   25 IF(ROUTX(J)-BLOK1(M))5,4,5
    5 CONTINUE
      DO17 M=1,JK
      IF(BLOKX(J)-BLOK1(M))117,4,117
  117 IF(BLOKX(J)-ROUT1(M))17,4,17
   17 CONTINUE
```

```
55 MX=MX+1
   BLOK1(MX)=BLOKX(J)
   ROUT1(MX)=ROUTX(J)
   PAY1(MX)=PAYX(J)
   JK=MX
   CHECK=1.
 4 CONTINUE
   PUNCH12,BLOKX(L),ROUTX(L),PAYX(L)
   IF(MX)15,15,34
34 PUNCH12,(BLOK1(MX),ROUT1(MX),PAY1(MX),MX=1,JK)
12 FORMAT(20X,3I5,45X)
   GO TO 111
15 STOP
   END
```

PROGRAM NO. 6

ARRANGING AND RANKING BLOCKWISE

```
      DIMENSION BLOCK(1000),ROUT(1000),PAY(1000),BLOCKX(50),ROUTX(50),PAYX
     1(50),USEDB(100)
      READ1,NL
    1 FORMAT(I5)
      DO22 I=1,NL
   22 READ11,BLOCK(I),ROUT(I),PAY(I)
   11 FORMAT(5X,I4,I4,4X,I4)
      L=1
      ISUM=0
      M=1
      USEDB(M)=BLOCK(L)
      KJ=M
  101 DO102 J=1,KJ
      IF(BLOCK(L)-USEDB(J))102,105,102
  102 CONTINUE
      GO TO 108
  105 IF(L-NL)106,110,110
  106 L=L+1
      J=0
      GO TO 101
  108 M=M+1
      USEDB(M)=BLOCK(L)
      KJ=M
      IF(L-NL)106,110 ,110
  110 DO500 K=1,KJ
      MX=0
  100 DO55 J=1,NL
      IF(USEDB(K)-BLOCK(J))55,33,55
   33 MX=MX+1
      BLOCKX(MX)=BLOCK(J)
      ROUTX(MX)=ROUT(J)
      PAYX(MX)=PAY(J)
      KOUNT=MX
   55 CONTINUE
      ISUM1=0
      IND=0
      INDEX=KOUNT-1
    3 DO 5 I=1,INDEX
      IF(PAYX(I+1)-PAYX(I))6,5,5
    6 SAVE=PAYX(I+1)
      CAVE=BLOCKX(I+1)
      RAVE=ROUTX(I+1)
      PAYX(I+1)=PAYX(I)
      BLOCKX(I+1)=BLOCKX(I)
      ROUTX(I+1)=ROUTX(I)
```

```
      PAYX(I)=SAVE
      ROUTX(I)=RAVE
      BLOCKX(I)=CAVE
      IHOLD=I
      IND=1
    5 CONTINUE
      IF(IND)9,8,9
    9 INDEX=IHOLD
      IND=0
      GO TO 3
    8 DO4 L=1,MX
      ISUM1=ISUM1+1
      ISUM=ISUM+1
    4 PUNCH7,BLOCKX(L),ROUTX(L),PAYX(L),ISUM1
    7 FORMAT(4I6)
      I=0
      KOUNT=0
      L=0
      IF(ISUM-NL)500,999,999
  500 CONTINUE
  999 STOP
       END
```

PROGRAM NO. 7

MODIFIED SUMMARY TABLE PROGRAM - RANKED TABLE

```
      DIMENSION          IAM(75),IPM(75),LX(4,75),LX1(3,75),ISUM1(75),
     1K(75,75)
      DO 400 I=1,75
      DO 400 J=1,75
  400 K(I,J)=0
      N=0
      DO 401 I=1,75
      DO 401 J=1,75
      LX1(I,J)=0
  401 LX(I,J)=0
      IK=0
      JK=0
      READ500,N
  500 FORMAT(I3)
      DO2 I=1,N
      READ1,IROUTE,IBLOCK,POUT
    1 FORMAT(11XI2,2XI2,7XF4.0)
      IBLOC=IROUTE*100+IBLOCK
      IF (POUT-1000.)3,3,4
    3 IK=IK+1
      IAM(IK)=IBLOC
      GO TO 2
    4 JK=JK+1
      IPM(JK)=IBLOC
    2 CONTINUE
    5 READ6,ISPRN1,ISPRN2,ML
    6 FORMAT(2I6,6X,I6)
      IF (ISPRN2)7,7,8
    8 DO 9 I=1,IK
      IF (ISPRN1-IAM(I))9,10,9
    9 CONTINUE
      DO 11 I=1,JK
      IF(ISPRN1-IPM(I)) 11,12,11
   11 CONTINUE
      GO TO 5
   12 IK=IK+1
      IAM(IK) =IPM(I)
      I=IK
   10 DO 13 J=1,JK
      IF(ISPRN2-IPM(J))13,14,13
   13 CONTINUE
      DO 15 J=1,IK
      IF(ISPRN2-IAM(J))15,16,15
```

```
 15 CONTINUE
    GO TO 5
 16 JK=JK+ 1
    IPM(JK)=IAM(J)
    J=JK
 14 K(I,J)=ML
    GO TO 5
  7 DO 103 J=1,JK
     LX(1,J)=IPM(J)/1000
    LX(2,J)=(IPM(J)-LX(1,J)*1000)/100
    LX(3,J)=(IPM(J)-LX(1,J)*1000-LX(2,J)*100)/10
103  LX(4,J)=IPM(J)-LX(1,J)*1000-LX(2,J)*100-LX(3,J)*10
    PUNCH102,((LX(I,J),J=1,71),I=1,4)
102 FORMAT(4X,1HI,24I3,/,5X,24I3,/,5X,23I3,1HI)
    PUNCH203
101 FORMAT(I4,1HI,24I3,/,5X,24I3,/,5X,23I3,1HI,I3)
203 FORMAT(77H----I---------------------------------------------------
   1---------------------,/,5X,72H-----------------------------------
   2-------------------------------------,/,5X,73H---------------------
   3------------------------------------------------I---)
    DO 107 I=1,IK
    ISUM=0
    DO 106 J=1,JK
     IF(K(I,J))106,106,105
105 ISUM=ISUM+1
106 CONTINUE
107 PUNCH101,IAM(I),(K(I,J),J=1,71),ISUM
    PUNCH203
    DO 204 J=1,JK
     ISUM1(J)=0
    DO205 I=1,IK
    IF(K(I,J))205,205,209
209 ISUM1(J)=ISUM1(J)+1
205 CONTINUE
204 CONTINUE
     DO 206 J=1,JK
    LX1(1,J)=ISUM1(J)/10
206 LX1(2,J)=ISUM1(J)-LX1(1,J)*10
     PUNCH102,((LX1(I,J),J=1,71),I=1,2)
    STOP
    END
```

PROGRAM NO. 6

PROGRAM FOR ADDING AND MULTIPLYING RANKS

```
      DIMENSION BLOK(900),ROUT(900),PAY(900),NO(900)
  999 ISUM=0
      READ1,N
    1 FORMAT(I3)
      DO22 I=1,N
      READ4,BLOK(I),ROUT(I),PAY(I),NO(I)
    4 FORMAT(4I6)
   22 CONTINUE
  888 READ5,BLOCK,ROUTE,PAYX,INO
    5 FORMAT(4I6)
      DO55 J=1,N
      IF(BLOCK-BLOK(J))55,56,55
   56 IF(ROUTE-ROUT(J))55,57,55
   55 CONTINUE
   57 NOA=INO+NO(J)
      NOM=INO*NO(J)
      PUNCH6,BLOCK,ROUTE,PAYX,NOA
      PUNCH7,BLOCK,ROUTE,PAYX,NOM
    6 FORMAT(4I6)
    7 FORMAT(4I6,55X,1H-)
      ISUM=ISUM+1
      IF(ISUM-N)888,999,1000
 1000 STOP
      END
```

PROGRAM NO. 9

SELECTION PROGRAM FOR- 1. MODIFIED SUMMARY TABLE

2. ADDING APPROACH
3. MULTIPLYING APPROACH

```
      DIMENSION BLOCK1(900),ROUT1(900),PAY(900),ML(900),BLOCKX(75),ROUTX(
     175),PAYX(75),NO(75)
       READ13,NL
   13 FORMAT(I5)
      DO22 I=1,NL
      READ1,BLOCK1(I),ROUT1(I),PAY(I),ML(I)
    1 FORMAT(2X,I4,2X,I4,2X,I4,4X,I2)
   22 CONTINUE
  444 READ14,N
   14 FORMAT(I3)
      DO333 L=1,N
      MX=0
      CHECK=0.0
      DO4 J=1,NL
      IF(BLOCK1(L)-BLOCK1(J))2,4,2
    2 IF(ROUT1(L)-ROUT1(J))44,4,44
   44 IF(CHECK)15,55,6
    6 DO5 M=1,JK
      IF(ROUT1(J)-ROUTX(M))77,4,77
   77 IF(ROUT1(J)-BLOCKX(M))5,4,5
    5 CONTINUE
      DO7 M=1,JK
      IF(BLOCK1(J)-BLOCKX(M))88,4,88
   88 IF(BLOCK1(J)-ROUTX(M))7,4,7
    7 CONTINUE
   55 MX=MX+1
   66 BLOCKX(MX)=BLOCK1(J)
      ROUTX(MX)=ROUT1(J)
      PAYX(MX)=PAY(J)
      NO(MX)=ML(J)
      JK=MX
      CHECK=1.0
    4 CONTINUE
      PUNCH12,BLOCK1(L),ROUT1(L),PAY(L),ML(L)
      IF(MX)15,15,34
   34 PUNCH12,(BLOCKX(MX),ROUTX(MX),PAYX(MX),NO(MX),MX=1,JK)
   12 FORMAT(20X,4I5,40X)
      PUNCH 16
   16 FORMAT(///)
  333 CONTINUE
      GO TO 444
   15 STOP
      END
```

A STUDY OF THE SYSTEMS AND TECHNIQUES OF SCHEDULING A PUBLIC TRANSIT COMPANY WITH THE USE OF A DIGITAL COMPUTER

by

PRABODH G. BAKSHI

B. E., Victoria Jubilee Technical Institute
University of Bombay, India, 1960

AN ABSTRACT OF A MASTER' REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1965

The objective of this report was to develop computer programs to construct the daily work schedule of the Cincinnati Transit Company, Ohio. The computer approach was primarily aimed at reducing laborious manual work and cost involved in preparing the daily schedule. The machine was also programmed to make a split runs selection based on the cost criterion.

The preparing of the daily work schedule was done in three steps on the IBM 1620 computer. Computer programs were developed and tested successfully. On comparing the results of the computer approach with those of the manual method, the following conclusions were made:

1. Manual work, both computational and clerical, was reduced to the bare minimum.

2. The computer solution not only developed the company's present schedule but also several alternatives. The speed with which the computer programs work enables the schedule maker to incorporate changes in schedules almost instantaneously.

3. The techniques developed for choosing split runs were based on the cost criterion, which is the main objective.

4. The time taken in preparing the schedule for the Cincinnati Transit Company with the computer was approximately three hours as against several days when the manual methods were used.