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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>HYPOTHESIS</td>
<td>3</td>
</tr>
<tr>
<td>SCHEDULING OF THE CINCINNATI TRANSIT COMPANY</td>
<td>3</td>
</tr>
<tr>
<td>STRAIGHT RUN MAKING</td>
<td>6</td>
</tr>
<tr>
<td>CONSTRUCTING SPLIT RUNS</td>
<td>9</td>
</tr>
<tr>
<td>COMBINING PIECES OF STRAIGHT RUNS AND PIECES OF WORK</td>
<td>13</td>
</tr>
<tr>
<td>SPLIT RUN SELECTION</td>
<td>15</td>
</tr>
<tr>
<td>MANUAL METHOD - SUMMARY TABLE</td>
<td>18</td>
</tr>
<tr>
<td>COMPUTER METHOD - TOTAL TIME BASIS</td>
<td>20</td>
</tr>
<tr>
<td>COMPUTER METHOD - MODIFIED SUMMARY TABLE</td>
<td>22</td>
</tr>
<tr>
<td>COMPUTER METHOD - LINEAR PROGRAMMING APPROACH</td>
<td>26</td>
</tr>
<tr>
<td>SUMMARY AND CONCLUSIONS</td>
<td>32</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>35</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>36</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>37</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1. Daily Volume of Traffic Curve .............................................. 5
2. Computer Output - Straight Run Making .................................. 10
3. Flow Diagram - Straight Run Making ....................................... 11
4. Flow Diagram - Split Run Making ........................................... 14
5. Flow Diagram - Combining Pieces of Work and Pieces of Straight Run 16
6. Machine Output - Combining Pieces of Straight Run and Pieces of Work ..................................................... 17
7. Summary Table ........................................................................ 19
8. Selection Based on Total Paytime ............................................. 23
9. Modified Table ......................................................................... 24
10. Comparison of Results, Company Selection vs Modified Table Machine Selected .................................................... 27
11. Vertical Ranking of Split Run Data ........................................... 31
12. Assignment Problem Technique Results .................................... 33
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piece of work</td>
<td>Any work assigned to an operator which pays less than 8 hours.</td>
</tr>
<tr>
<td>Straight run</td>
<td>A regular run having no unpaid breaks.</td>
</tr>
<tr>
<td>Split run</td>
<td>A regular run having unpaid breaks.</td>
</tr>
<tr>
<td>Pull out time</td>
<td>The starting time of a piece of work, straight run, or a split run.</td>
</tr>
<tr>
<td>Pull in time</td>
<td>The finishing time of a piece of work, straight run, or a split run.</td>
</tr>
<tr>
<td>Travel time</td>
<td>The actual running time from the garage to point of relief.</td>
</tr>
<tr>
<td>Platform time</td>
<td>The time during which an operator is in charge of a bus.</td>
</tr>
<tr>
<td>Turn-in</td>
<td>A specific amount of time which the operator receives for every straight run and each piece of work in his assignment.</td>
</tr>
<tr>
<td>Guarantee time</td>
<td>The time paid for but not worked, to make up a minimum guarantee for certain classes of work.</td>
</tr>
<tr>
<td>Allowance</td>
<td>A time equal to either the travel time, the guarantee time, or both.</td>
</tr>
<tr>
<td>Pay time</td>
<td>The total pay time for a specific run.</td>
</tr>
<tr>
<td>Spread time</td>
<td>The total working time plus the unpaid time of a split run.</td>
</tr>
</tbody>
</table>
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.

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.

---


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 scheduling4.

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 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 (46%) 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.
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:

<table>
<thead>
<tr>
<th>No/reliefs</th>
<th>Travel</th>
<th>Route</th>
<th>Block</th>
<th>P/out</th>
<th>P/in</th>
<th>Reliefs:</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0.0</td>
<td>26</td>
<td>3</td>
<td>5:41</td>
<td>17:05</td>
<td>5:41, 7:23, 8:09, ----</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12:46, 12:06, 14:35, ----</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16:06, 17:86, 17:95, ----</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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. 17:05 - 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 work 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:

<table>
<thead>
<tr>
<th>Route</th>
<th>Block</th>
<th>TF</th>
<th>P/out</th>
<th>P/in</th>
<th>TB</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>No 1</td>
<td>1</td>
<td>12</td>
<td>0</td>
<td>606</td>
<td>251</td>
<td>0</td>
</tr>
<tr>
<td>No 2</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>1406</td>
<td>1826</td>
<td>0</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>No.</th>
<th>ROUTE</th>
<th>BLOCK</th>
<th>TF</th>
<th>P/OUT</th>
<th>P/IN</th>
<th>TB</th>
<th>PLAT</th>
<th>PENLT</th>
<th>TURN IN</th>
<th>PAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26</td>
<td>3</td>
<td>0</td>
<td>541</td>
<td>1296</td>
<td>0</td>
<td>755</td>
<td>36</td>
<td>8.3</td>
<td>800</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>3</td>
<td>0</td>
<td>1296</td>
<td>1795</td>
<td>0</td>
<td>499</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>3</td>
<td>0</td>
<td>898</td>
<td>1795</td>
<td>0</td>
<td>897</td>
<td>0</td>
<td>8.3</td>
<td>905</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>3</td>
<td>0</td>
<td>541</td>
<td>898</td>
<td>0</td>
<td>357</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>No.</th>
<th>Route</th>
<th>Block</th>
<th>TF</th>
<th>P/out</th>
<th>P/in</th>
<th>TD</th>
<th>Spread</th>
<th>Platform</th>
<th>Penalty</th>
<th>Turn-in</th>
<th>Pay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>12</td>
<td>0</td>
<td>606</td>
<td>951</td>
<td>0</td>
<td>31+5</td>
<td></td>
<td>18.34</td>
<td>16.67</td>
<td>800</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>1406</td>
<td>1826</td>
<td>0</td>
<td>128.3</td>
<td>420</td>
<td>18.34</td>
<td>16.67</td>
<td>800</td>
</tr>
</tbody>
</table>
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 BLOCKS NO. 2**

<table>
<thead>
<tr>
<th>NO.</th>
<th>RN</th>
<th>BN</th>
<th>TF</th>
<th>P/CUT</th>
<th>P/IN</th>
<th>TB</th>
<th>SPREAD</th>
<th>TRIP</th>
<th>PENLTY</th>
<th>TURN</th>
<th>PAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>944</td>
<td>0.0</td>
<td>1440</td>
<td>1946</td>
<td>0.0</td>
<td>506</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1N01</td>
<td>44</td>
<td>2 16.6</td>
<td>651</td>
<td>888</td>
<td>0.0</td>
<td>110.0</td>
<td>237</td>
<td>40.3</td>
<td>16.6</td>
<td>800.0</td>
<td></td>
</tr>
<tr>
<td>1N02</td>
<td>644</td>
<td>0.0</td>
<td>1523</td>
<td>1883</td>
<td>0.0</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1N03</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>83.1</td>
<td>458</td>
<td>0.0</td>
<td>16.6</td>
<td>834.6</td>
<td></td>
</tr>
<tr>
<td>1N04</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>126.3</td>
<td>458</td>
<td>0.0</td>
<td>16.6</td>
<td>842.6</td>
<td></td>
</tr>
<tr>
<td>1N05</td>
<td>5 0.0</td>
<td>1498</td>
<td>1818</td>
<td>0.0</td>
<td>354</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1N06</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>101.3</td>
<td>458</td>
<td>5.3</td>
<td>16.6</td>
<td>800.0</td>
<td></td>
</tr>
<tr>
<td>1N07</td>
<td>24</td>
<td>9 0.0</td>
<td>1515</td>
<td>1868</td>
<td>0.0</td>
<td>353</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1N08</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>75.6</td>
<td>458</td>
<td>0.0</td>
<td>16.6</td>
<td>827.6</td>
<td></td>
</tr>
<tr>
<td>1N09</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>121.3</td>
<td>458</td>
<td>5.3</td>
<td>16.6</td>
<td>800.0</td>
<td></td>
</tr>
<tr>
<td>1N10</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>124.3</td>
<td>458</td>
<td>0.0</td>
<td>16.6</td>
<td>830.6</td>
<td></td>
</tr>
<tr>
<td>1N11</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>109.1</td>
<td>458</td>
<td>0.0</td>
<td>16.6</td>
<td>808.6</td>
<td></td>
</tr>
<tr>
<td>1N12</td>
<td>402</td>
<td>0.0</td>
<td>625</td>
<td>1083</td>
<td>0.0</td>
<td>7.3</td>
<td>458</td>
<td>7.3</td>
<td>16.6</td>
<td>800.0</td>
<td></td>
</tr>
</tbody>
</table>

1N13 44 2 0.0 888 1156 0.0 0.0 268 9.3 16.6 800.0
1N14 44 2 16.6 651 935 0.0 110.0 284 16.6 16.6 823.3
1N15 44 2 16.6 651 935 0.0 90.3 284 97.3 16.6 800.0
1N16 44 2 0.0 935 1208 0.0 0.0 273 4.3 16.6 800.0

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 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

---

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:

<table>
<thead>
<tr>
<th>Piece</th>
<th>Pay time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>600</td>
</tr>
<tr>
<td>B</td>
<td>840</td>
</tr>
<tr>
<td>C</td>
<td>820</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Piece</th>
<th>Pay time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>800</td>
</tr>
<tr>
<td>C</td>
<td>820</td>
</tr>
<tr>
<td>B</td>
<td>840</td>
</tr>
</tbody>
</table>

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 preceding 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**

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

**TOTAL** = 258.22 HOURS

*Fig. 8*
Fig. 9
25

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, 39 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" 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

---

COMPARISON OF RESULTS

COMPANY SELECTION VS MODIFIED TABLE MACHINE SELECTED

<table>
<thead>
<tr>
<th></th>
<th>AM/P</th>
<th>PM/P</th>
<th>PAY</th>
<th></th>
<th>AM/P</th>
<th>PM/P</th>
<th>PAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>6809</td>
<td>107</td>
<td>803</td>
<td>6802</td>
<td>404</td>
<td>818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2806</td>
<td>6801</td>
<td>836</td>
<td>310</td>
<td>107</td>
<td>838</td>
<td></td>
<td></td>
</tr>
<tr>
<td>304</td>
<td>403</td>
<td>846</td>
<td>503</td>
<td>553</td>
<td>849</td>
<td></td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>4401</td>
<td>816</td>
<td>2902</td>
<td>160</td>
<td>861</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6807</td>
<td>1051</td>
<td>888</td>
<td>2405</td>
<td>801</td>
<td>864</td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>104</td>
<td>818</td>
<td>901</td>
<td>408</td>
<td>866</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6802</td>
<td>302</td>
<td>800</td>
<td>113</td>
<td>6801</td>
<td>871</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1003</td>
<td>404</td>
<td>873</td>
<td>100</td>
<td>459</td>
<td>874</td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>101</td>
<td>886</td>
<td>505</td>
<td>2603</td>
<td>877</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2805</td>
<td>301</td>
<td>935</td>
<td>311</td>
<td>2454</td>
<td>883</td>
<td></td>
<td></td>
</tr>
<tr>
<td>508</td>
<td>6857</td>
<td>916</td>
<td>2404</td>
<td>501</td>
<td>884</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2807</td>
<td>1503</td>
<td>836</td>
<td>6806</td>
<td>6856</td>
<td>885</td>
<td></td>
<td></td>
</tr>
<tr>
<td>310</td>
<td>2460</td>
<td>910</td>
<td>2602</td>
<td>356</td>
<td>893</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2405</td>
<td>751</td>
<td>858</td>
<td>6807</td>
<td>2460</td>
<td>895</td>
<td></td>
<td></td>
</tr>
<tr>
<td>604</td>
<td>2851</td>
<td>892</td>
<td>2806</td>
<td>101</td>
<td>897</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6810</td>
<td>459</td>
<td>850</td>
<td>2406</td>
<td>357</td>
<td>900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2406</td>
<td>6805</td>
<td>854</td>
<td>304</td>
<td>557</td>
<td>901</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2901</td>
<td>806</td>
<td>800</td>
<td>701</td>
<td>402</td>
<td>901</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2902</td>
<td>160</td>
<td>861</td>
<td>502</td>
<td>103</td>
<td>904</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2602</td>
<td>463</td>
<td>842</td>
<td>6810</td>
<td>463</td>
<td>910</td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>4453</td>
<td>889</td>
<td>309</td>
<td>802</td>
<td>912</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>105</td>
<td>919</td>
<td>6809</td>
<td>4453</td>
<td>914</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2404</td>
<td>408</td>
<td>855</td>
<td>305</td>
<td>4454</td>
<td>929</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2604</td>
<td>106</td>
<td>864</td>
<td>111</td>
<td>6805</td>
<td>932</td>
<td></td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>461</td>
<td>873</td>
<td>112</td>
<td>2651</td>
<td>934</td>
<td></td>
<td></td>
</tr>
<tr>
<td>701</td>
<td>402</td>
<td>901</td>
<td>2601</td>
<td>106</td>
<td>935</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1002</td>
<td>460</td>
<td>932</td>
<td>508</td>
<td>158</td>
<td>937</td>
<td></td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>501</td>
<td>912</td>
<td>1002</td>
<td>751</td>
<td>950</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4403</td>
<td>801</td>
<td>881</td>
<td>2805</td>
<td>6857</td>
<td>950</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6813</td>
<td>1053</td>
<td>946</td>
<td>1003</td>
<td>359</td>
<td>1018</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Jobs</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1, 3</td>
<td>2, 2</td>
<td>3, 1</td>
<td>4, 3</td>
</tr>
<tr>
<td>b</td>
<td>1, 4</td>
<td>2, 3</td>
<td>3, 2</td>
<td>4, 4</td>
</tr>
<tr>
<td>c</td>
<td>3, 1</td>
<td>1, 4</td>
<td>2, 3</td>
<td>4, 2</td>
</tr>
<tr>
<td>d</td>
<td>2, 2</td>
<td>3, 1</td>
<td>1, 4</td>
<td>4, 1</td>
</tr>
</tbody>
</table>

(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.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>b</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>c</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>d</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
3. Subtract the minimum cell in each column resulting from step 2 from all the elements in its column.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

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

\footnote{9. Ibid.}
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

<table>
<thead>
<tr>
<th>ASSIGNMENT APPROACH</th>
<th>ADDING APPROACH</th>
<th>MULTIPLYING APPROACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM/P</td>
<td>PM/P</td>
<td>PAY</td>
</tr>
<tr>
<td>806</td>
<td>2652</td>
<td>800</td>
</tr>
<tr>
<td>112</td>
<td>107</td>
<td>800</td>
</tr>
<tr>
<td>904</td>
<td>403</td>
<td>846</td>
</tr>
<tr>
<td>311</td>
<td>104</td>
<td>818</td>
</tr>
<tr>
<td>502</td>
<td>459</td>
<td>859</td>
</tr>
<tr>
<td>508</td>
<td>6801</td>
<td>833</td>
</tr>
<tr>
<td>2406</td>
<td>302</td>
<td>800</td>
</tr>
<tr>
<td>307</td>
<td>2851</td>
<td>874</td>
</tr>
<tr>
<td>309</td>
<td>601</td>
<td>893</td>
</tr>
<tr>
<td>310</td>
<td>1051</td>
<td>903</td>
</tr>
<tr>
<td>1002</td>
<td>460</td>
<td>932</td>
</tr>
<tr>
<td>2806</td>
<td>557</td>
<td>901</td>
</tr>
<tr>
<td>604</td>
<td>2603</td>
<td>839</td>
</tr>
<tr>
<td>505</td>
<td>103</td>
<td>914</td>
</tr>
<tr>
<td>2409</td>
<td>2805</td>
<td>868</td>
</tr>
<tr>
<td>305</td>
<td>4454</td>
<td>929</td>
</tr>
<tr>
<td>504</td>
<td>408</td>
<td>866</td>
</tr>
<tr>
<td>2404</td>
<td>106</td>
<td>864</td>
</tr>
<tr>
<td>6806</td>
<td>2651</td>
<td>920</td>
</tr>
<tr>
<td>4403</td>
<td>6859</td>
<td>949</td>
</tr>
<tr>
<td>111</td>
<td>404</td>
<td>878</td>
</tr>
<tr>
<td>2602</td>
<td>2460</td>
<td>829</td>
</tr>
<tr>
<td>503</td>
<td>4401</td>
<td>816</td>
</tr>
<tr>
<td>2901</td>
<td>2802</td>
<td>907</td>
</tr>
<tr>
<td>206</td>
<td>536</td>
<td>896</td>
</tr>
<tr>
<td>2807</td>
<td>751</td>
<td>864</td>
</tr>
<tr>
<td>901</td>
<td>402</td>
<td>904</td>
</tr>
<tr>
<td>701</td>
<td>1053</td>
<td>935</td>
</tr>
<tr>
<td>2601</td>
<td>802</td>
<td>919</td>
</tr>
<tr>
<td>6802</td>
<td>4453</td>
<td>843</td>
</tr>
<tr>
<td>TOTAL= 261.99</td>
<td>TOTAL= 259.51</td>
<td>TOTAL= 258.86</td>
</tr>
</tbody>
</table>

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 develop 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


PROGRAM NO. 1

CONVERSION PROGRAM FOR HOURS MINUTES TO HOURS DECIMALS
DIMENSION REL(100)
111 READ1112,TRAVEL
112 FORMAT(F2,0)
   TRAVEL=TIME(TRAVEL,1.667)
111 N1=1
N2=1
1 READ1,LINEX,NCT,PCUT,PIN,(REL(I),I=N1,N2)
1 FORMAT(2I2,15F5.0)
   IF(LIN1)100,100,2
2 IF(REL(N2))3,4,5
3 IF(REL(N2)-PIN)5,6,5
5 N1=1+N2
N2=N2+15
1 READ1,LINEX,NCT,PCUT,(REL(I),I=N1,N2)
   GO TO 2
4 DC 7 I=1,N2
   IF(PFL(I))7,8,7
7 CONTINUE
8 I=1-1
   GO TO 9
6 I=N2
9 DC 10 J=1,1
10 REL(J)=TIME(REL(J),1.667)
   PCUTC=TIME(PCUT,1.667)
   POINC=TIME(PIN,1.667)
17 PUNCH18,I,TRAVEL,LINEX,NCT,PCUT,POINC,(REL(J),J=1,1)
   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,9F6.0/26X,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*10.0
   TIME=(X-HR)*Y+HR
   RETURN
END
PROGRAM NO. 2

A STRAIGHT RUN MAKING

DIMENSION REL(100)
1 K=0
2 TURN=8.33
3 TF=C
4 TB=0
10 DO 201 I=1,100
201 REL(I)=0
PUNCH 202
202 FORMAT(/1)
READ11,M,TRAVEL,ROUTE,BLOCK,TIMON,TIMOFF,(REL(L),L=1,M)
14 BACK=0
FRONT=0
PUNCH=TIMON
PUNCH=TIMOFF
18 EIGHT=BACK=TURN
20 ALLOW=C
21 IF(TIMOFF-TIMON-800.,24,24,22
22 BACK=I
24 IF(TIMOFF-TIMON)10,10,25
25 TRIP1=TIMOFF-TIMON
26 IF(TIMON-POUT)27,28,27
27 TF=TRAVEL
28 IF(TIMOFF-POIN)29,30,29
29 TB=TRAVEL
30 IF(TRY-750.,31,136,136
136 IF(TRY-800..)36,36,136
156 IF(FRONT)'9000.'30,30,156
31 IF(TRY-150.,32,33,43
33 PUNCH 32,ROUTE,BLOCK,TF,TIMON,TIMOFF,TF,TRIP1
34 FORMAT(9X,214,F5.1,216,F5.1,5X,25X,1H-)
GO TO 33
43 PUNCH 32,ROUTE,BLOCK,TF,TIMON,TIMOFF,TF,TRIP1
33 TF=0
TB=C
1 IF(BACK)'9000.'10,10,10
36 IF(TIMON-POUT)37,37,37
37 TF=TRAVEL
38 IF(TIMOFF-POIN)39,40,39
39 TB=TRAVEL
40 ALLOW=TF+TB
K=K+1
TOTAL=TRIP1+TURN+TF+TB
42 PUNCH 43,K,ROUTE,BLOCK,TF,TIMON,TIMOFF,TF,TRIP1,ALLOW,TURN,TOTAL
TF=C
TB=0
FRONT=1
IF(BACK)'9000.'10,10,10
H_AT=Ti...N+60o.-TkAVr.L-TURN
L=19M
`..NTINUE
^4
TB=TRAVL
^4
TOTAL=REL(L)-TImON+TURN+TRAVEL
FS
ALLO'=TF+TB
TRIPT=REL(L)-TImON
K=K+1
K=K+1
TB=TRAVL
ALLOW=TF+TB
TRIPT=REL(L)-TImON+TURN+TF+TB
EIGHT=600.-TURN-TRAVEL
GC TO 24
PWC=(REL(L)-PLAT)*.5
PWP=PLAT-REL(L-1)
IF(PWC-PWP)68,68,74
K=K+1
TB=TRAVL
ALLOW=TF+TB
TRIPT=REL(L)-TImON+TURN+TF+TB
EIGHT=600.-TURN-TRAVEL
GC TO 59
K=K+1
TB=TRAVL
ALLOW=TF+TB+PWP
TRIPT=REL(L-1)-TImON+TURN+ALLOW
PUNCH43,K,ROUTE,BLOCK,TF,TIMON,REL(L-1),TB,TRIPT,ALLOW,TURN,TOTAL
TIMON=REL(L-1)
TF=U
FRONT=1
GC TO 63
TB=TRAVL
TIMON=POUT
TIMOFF=PCIN
200 PLAT=TIMOFF-800.+TRAVEL+TURN
103 DC 103 L=1,M
103 CONTINUE
IF(L-M)103,31,31
104 TF=TRAVL
ALLOW=TF+TB
TOTAL=800.
\begin{verbatim}
K=K+1
TRIPT=TIMOFF-REL(L)
109 PUNCH 43,K,ROUTE,BLOCK,TF,REL(L-1),TIMOFF,TC,TRIPT,ALLOW,TURN,TOTAL
F R O N T = 0
111 TF=U
113 TIMOFF=REL(L)
114 TIME=TIMOFF-TIMON
115 IF (TIME) 90,10,116
116 IF (TIME=(86.0-TRAVEL-TRAVEL)) 117,119,199
199 TF=TRAVEL
GO TO 20
117 BACK=0
GO TO 26
119 BACK=0
120 GO TO 56
121 P&V=(PLAT-REL(L-1))*.5
122 P&P=REL(L)-PLAT
124 TF=TRAVEL
ALLOW=TF+TP
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,TC,TRIPT,ALLOW,TURN,TOTAL
1 L
F R O N T = 0
TF=U
TIMOFF=REL(L-1)
GO TO 114
134 TF=TRAVEL
ALLOW=TF+TR+P&P
TOTAL=TIMOFF-REL(L-1)+TURN+ALLOW
K=K+1
TRIPT=TIMOFF-REL(L)
GO TO 109
END
\end{verbatim}
PROGRAM NO. 3

SPLIT RUN MAKING

DIMENSION ROUTE(200), BLOCK(200), TF(200), TIMON(200), TIMOFF(200), TB(200), TRIPT(200)

1 READ2, IPW
2 FORMAT(3X13)
   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 I=I+1
   IAM(I)=ROUTE(I)*100.+BLOCK(I)
   GO TO 9
12 J=J+1
   IPW(J)=ROUTE(I)*100.+BLOCK(I)
9 CONTINUE
10 L=1
11 DC 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(TIMON(J)-TIMOFF(L))1000, 25, 124
124 IF(SPREAD=1100.)26, 26, 25
25 PENLT3=(SPREAD-1100.)
26 PW1=TRIPT(J)+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 PAYT1=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 PRINT 49, ROUTE(J), BLOCK(J), TF(J), TIMCN(J), TIOMFF(J), TB(J), PENLT3, TRIFT(J), ALLOW, TURN2, PAYTIM
GO TO 1000
45 IF (SUM = TOTAL) 50, 500, 46
46 PENLT1 = SUM - TOTAL
47 PAYTIM = PENLT1 + TOTAL
48 ALLOW = PENLT1 + PENLT2 + TF(L) + TD(L) + TR(J) + TB(J)
49 N = N + 1
GO TO 41
50 IF (SUM = TOTAL) 1000, 51, 51
51 PENLT1 = 0
GO TO 47
52 SUM = TRIFT(L) + TRIFT(J) + TF(L) + TB(J) + TURN + GAP
53 IF (SUM < 750) 1000, 502, 511
54 ALLOW = GAP - TD(L) - TF(J) + 600 - SUM
55 N = N + 1
TOTAL = SUM - TURN
PAYTIM = 600
56 BLOCKX = BLOCK(L) * 100 + BLOCK(J)
ROUTEX = ROUTE(L) * 100 + ROUTE(J)
PUNCH5US, N, ROUTEX, BLOCKX, TF(L), TIMCN(L), TIMOFF(J), TB(J), TOTAL, ALLOW
1000 TURN, PAYTIM
GO TO 701
511 IF (SUM = TOTAL) 502, 502, 512
512 IF (SUM = 600) 513, 513, 1000
513 ALLOW = GAP - TD(L) - TF(J)
N = N + 1
TOTAL = SUM - TURN
PAYTIM = 600
GO TO 503
500 CONTINUE
L = L + 1
1000 IF (L = IPW) 11, 11, 1005
1005 STOP
8 FORMAT (9X, 2I4, F6.0, F6.0, F6.0, 3X, F6.0)
41 FORMAT (3X, 2I5, 2I5, F6.0, F6.0, 3X, F6.0)
509 FORMAT (15, 2I5, 2I5, F6.0, F7.1, 1X, F6.0, 2F7.1)
43 FORMAT (15, 2I5, 2I5, F6.0, F7.1, 1X, F6.0, 2F7.1)
END
PROGRAM NO. 4

C COMBINING STRAIGHT RUN PIECES AND PIECES OF WORK

DIMENSION LINEP(200), BLOCKP(200), PPF1(200), PCINP(200), POUTP(200),
1 PPF2(200), TRIP TP(200), REL(100), PW(2), OUT(2), PIN(2), FF(2), FB(2)
DIMENSION PIECE(30)
STR=500
KX=100
SCNCF=16.67
TEN=16.67
1 READ2, IPW, NSP, NST
2 FORMAT(3I4)
   DC 3 I=1, IPW
4 FORMAT(9X,2I4,F5.1,2F6.0,F5.1,5X,F6.0)
3 READ4,INEP(I),BLOCKP(I),PPF1(I),POUTP(I),PCINP(I),PPF2(I),TRIP TP(I)
1 I
5 KNST=0
6 KNST=KNST+1
7 IF(KNST-NST)8,8,9000
8 READ9,*,TRAVEL,LINES,BLOCKC,POUTC,PINC,(REL(K),K=1,M)
9 FORMAT(13,F5.1,3X9213,5X1213,6X,815/40X,815)
PUNCH888,LINES,BLOCKC
888 FORMAT(14H ROUTE NO., ,I5,14H BLOCKC NO. , ,I5)
PUNCH777,
   CHECKC=0
   CHECK=0
   CHECKB=0
   KTWCA=0
10 MCNE=1
11 POUT=POUTC
   PCIN=PCINC
   FR1S=16.67
   FR2S=8.33
16 DC1000K=MCNE,M
1016 SPW=REL(K)-POUT
   IF(S PW-150.)1000,1000,18
18 SP CUT=POUT
   IF(S PW-600.)19,19,116
116 IF(SPCUT-POUTC)19,6,19
19 SPIN=REL(K)
   SBLOCK=BLOCKC
   SLINE=LINES
   SPF1=FR1S
   SPF2=TRAVEL
24 DC 999 N=1, IPW
26 PENLT1=0
   PENLT2=0
   PENLT3=0
   PENLT4=0
L = 1
J = 2

PW(L) = SPW
CUT(L) = SCUT
PIN(L) = SPIN
FF(L) = SPF1
FB(L) = SPF2
PW(J) = TRIP(N)
CUT(J) = PCUT(N)
PIN(J) = PCINP(N)

IF(CUT(2) - CUT(1)) 45, 999, 49

L = 2
J = 1

GC TO 34

IF(PIN(2) - PIN(1)) 999, 999, 50
IF(CUT(2) - PIN(1)) 999, 999, 51
GAP = (CUT(2) - PIN(1))
IF(GAP - TEN) 999, 55, 53
SPEA = PIN(2) - CUT(1) + FF(1) + FB(1) + FF(2) + FB(2) + SCONC/2.
IF(1325. - SPEA) 999, 55, 58
PENLT1 = (SPEA - 1100.)* .5
SPEA = 1300.

GC TO 62

IF(125. - SPEA) 55, 59, 59
IF(1200. - SPEA) 60, 61, 61
PENLT1 = (SPEA - 1100.)
SPEA = 1250.

PW1T = PW(1) + FF(1) + FB(1)
PW2T = PW(2) + FF(2) + FB(2)
TIME = PW1T + PW2T + SPF1 + SPF2 + 50.

IF(TIME - TIMET) 750. 78, 82
PENLTY = PENLT2 + PENLT3 + PPF1(N) + PPF2(N) + SPF1 + SPF2 + 50.
PAYTIM = TIME + PENLT2 + PENLT3 + 50.

GC TO 90

IF(800. - TIMET) 86, 83, 83
PENLTY = PENLT2 + PENLT3 + 800. - TIMET + PPF1(N) + PPF2(N) + SPF1 + SPF2
PAYTIM = TIME + PENLT2 + PENLT3 + 800. - TIMET

GC TO 90

IF(850. - TIMET) 999, 87, 87
PENLTY = PENLT2 + PENLT3 + PPF1(N) + PPF2(N) + SPF1 + SPF2
PAYTIM = TIME + PENLT2 + PENLT3
90  \text{KX} = \text{KX} + 1
\text{MKSX} = \text{KX}

221 \text{PUNCH79, LINEP(N), BLCKP(N), PP1(N), PCUTP(N), PCINP(N), PPF2(N),}
\text{TRIPTP(N)}
\text{PUNCH80, MKSX, SLINE, SBLCK, SPF1, SPCHT, SPIN, SPF2, PENDL1, SPW, PENLTY,}
\text{1SCNOF, PAYTIM}

79 \text{FORMAT}(4X, 2I5, 1E15, 1F5, 1, 7X15)
80 \text{FORMAT}(14, 2I5, 1, 2I5, 1E15, 1F5, 1, 7I5, 1, 3F7, 1)
195 \text{IF(IPW-N)1999, 195, 1999}
92 \text{CHECK} = \text{CHECK} + 1.
\text{KCNE} = K
95 \text{LEFT} = \text{PCINO} - \text{REL(K)}
96 \text{IF(LEFT)9000, 99, 97}
97 \text{IF(LEFT-200)98, 98, 103}
98 \text{PUNCH199, LINEC, SBLCKC, REL(K), PCINC, LEFT}
99 \text{IF(KCNE +3-M)100, 6, 6}
100 \text{CHECK} = 0
101 \text{MCNE} = \text{KCNE} + 1
\text{PUNCH777}
777 \text{FORMAT(////)}
102 \text{GC TO 11}
103 \text{MCNE} = K + 1
\text{PCUT} = \text{REL(K)}
\text{PCUTB} = \text{PCUT}
\text{PCIN} = \text{PCINC}
\text{FR1S} = \text{TRAVEL}
\text{FR2S} = 8.33
\text{GC TO 16}
999 \text{CONTINUE}
\text{IF(CHECKC)801, 1000, 801}
801 \text{CHECKC} = 0
\text{GC TO 195}
1000 \text{CONTINUE}
1001 \text{IF(CHECKB)9000, 99, 408}
1999 \text{IF(CHECKC)999, 800, 999}
800 \text{CHECKC} = \text{CHECKC} + 1.
\text{GC TO 999}
9000 \text{STOP}
200 \text{SUM=TRIPTP(N)+SPW+FF(1)+FB(2)+GAP+SONOF}
\text{IF(SUM-750., 1300, 202, 209}
300 \text{IF(SBLCK-100.)333, 999, 999}
333 \text{SBLCK} = \text{BLCKP(N)*100} + \text{SBLCK}
\text{XLINE=LINEP(N)}
\text{SLINE=XLINE*100} + \text{SLINE}
\text{SPW=PIN(2)-OUT(1)}
\text{SPCUT=OUT(1)}
\text{SPIN=PIN(2)}
\text{SPF1=FF(1)}
\text{SPF2=FB(2)}
\text{GC TO 24}
202 \text{PENLTY} = 800. - \text{SUM} + \text{PPF1(N)+PPF2(N)+SPF1+SPF2} + \text{GAP}
TIME=SUM
TRIP=TRIPTP(N)+SPW
STR=STR+1.
CVERT=0
PAYTIM=800.
MKSX=STR
GC 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 GC TO 221

401 IF(CHECKB)9000,402,406
402 CHECKB=CHECKB+1.

403 K=K

404 IF(KTWCA-KTWC)405,499,405
405 KTWCA=KTWC

406 LEFT=PCINC-REL(K)
412 IF(LEFT<200)409,499,412

408 IF(KTWCA+2-M)409,499,499

409 CHECKB=0
MONE=KTWC+1
PCIN=PCINC
POUT=PCUTB
FR1S=TRAVEL
KTWCA=V
PUNCH778,

778 FORMAT(/)

411 GC TO 16

413 POUT=REL(K)
K=M
REL(K)=PCINC
FR1S=TRAVEL
GC TO 1016

499 CHECKB=0
GC 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)
   KCOUNT=NL
   ISUM1=0
   IND=0
   INDEX=KCOUNT-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
   GC TO13
8 DO41 I=1, KCOUNT
   ISUM1=ISUM1+1
41 FORMAT(4I6)
111 READ101, L
101 FORMAT(I5)
   CHECK=0, 0
   MX=0
   DC4 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 DC5 M=1, JK
   IF(ROUTX(J)-ROUT1(M))25, 4, 25
25 IF(ROUTX(J)-BLOK1(M))5, 4, 5
5 CONTINUE
17 CONTINUE
55 MX=MX+1
   BLCK1(MX)=BLCKX(J)
   ROUT1(MX)=ROUTX(J)
   PAY1(MX)=PAYX(J)
   JK=MX
   CHECK=1.
4 CONTINUE
   PUNCH12, BLOCKX(L), ROUTX(L), PAYX(L)
   IF(MX)15,15,34
34 PUNCH12, (BLOCK1(MX), ROUT1(MX), PAY1(MX), MX=1, JK)
12 FORMAT(20X,3I5,45X)
   GC TO 111
15 STOP
END
PROGRAM NO. 6

ARRANGING AND RANKING BLOCKWISE

DIMENSION BLCK(1000), RCUT(1000), PAY(1000), BLCKX(50), RCUTX(50), PAYX(50), USEDB(100)
READ 1, NL
1 FORMAT (I5)
DC 22 I = 1, NL
22 READ 11, BLCK(I), RCUT(I), PAY(I)
11 FORMAT (5X, I4, 14, 4X, I4)

L = 1
ISUM = 0
M = 1
USEDB(M) = BLCK(L)
KJ = M

101 DC 102 J = 1, KJ
102 IF (BLCK(L) - USEDB(J)) 102, 105, 102
105 CONTINUE
GC TO 108
106 L = L + 1
J = 0
GO TO 101
108 M = M + 1
USEDB(M) = BLCK(L)
KJ = M
IF (L - NL) 106, 110, 110
110 DC 500 K = 1, KJ
MX = 0

100 DC 55 J = 1, NL
105 IF (USEDB(K) - BLCK(J)) 55, 33, 55
33 MX = MX + 1
BLCKX(MX) = BLCK(J)
RCUTX(MX) = RCUT(J)
PAYX(MX) = PAY(J)
KCOUNT = MX
55 CONTINUE
ISUM1 = 0
IND = 0
INDEX = KCOUNT - 1
3 DC 5 I = 1, INDEX
5 IF (PAYX(I + 1) - PAYX(I)) 6, 5, 5
6 SAVE = PAYX(I + 1)
CAVE = BLCKX(I + 1)
RAVE = RCUTX(I + 1)
PAYX(I + 1) = PAYX(I)
BLCKX(I + 1) = BLCKX(I)
RCUTX(I + 1) = RCUTX(I)
PAYX(I)=SAVE
ROUTX(I)=RAVE
BLCKX(I)=CAVE
IHOLD=I
IND=1
5 CONTINUE
   IF(IND)9,8,9
9 INDEX=IHOLD
   IND=0
   GO TO 3
8 DC4 L=1,MX
   ISUM1=ISUM1+1
   ISUM=ISUM+1
4 PUNCH7,BLCKX(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

IDENTIFIED SUMMARY TABLE PROGRAM = RANKED TABLE

DIMENSION IAM(75), IPM(75), LA(4,75), LA1(2,75), ISUM1(75),
I(75,75), DC(400, I=1,75), DC(400, J=1,75)

DO 100 K(I,J)=.
   M=.
   DO 200 I=1,75
      DO 200 J=1,75
         LX(I,J)=0
   200 CONTINUE
   READ50C,N

500 FORMAT(I3)
   DC2 I=1,N
   READ(I), ROUTE, IBLOCK, POUT
   1 FORMAT(11X12,2X12,7XF4.0)
   IBL=p=1, ROUTE#100+IBLOCK.
   IF (POUT-1000.13,3,4
   3 I=I+1
   IAM(I)=IBLOC
   GO TO 7
   4 JK=JK+1
   IPM(JK)=IBLOC
   2 CONTINUE
   5 READ6, ISPRN1, ISPRN2, ML
   6 FORMAT(216,6X,16)
      IF (ISPRN2) 7, 8, 8
   7 DC 9 I=1, IK
      IF (ISPRN1-IAM(I)) 9, 10, 9
   8 CONTINUE
   DC 11 I=1, JK
      IF (ISPRN1-IPM(I)) 11, 12, 11
   CONTINUE
   KG TO 5
   12 IK=IK+1
      IAM(IK)=IPM(I)
      *=IK
   13 DC 13 J=1, JK
      IF (ISPRN2-IPM(J)) 13, 14, 13
   CONTINUE
   DC 15 J=1, IK
      IF (ISPRN2-IAM(J)) 15, 16, 15
15 CONTINUE
16 GO TO 5
14 JK=JK+1
15 P(L(JK))=LAM(J)
16 J=JK
14 K(I,J)=M
15 GO TO 5
7 DC 107 J=1,JK
10 LX(1,J)=P(L(J))/1000
11 LX(2,J)={(P(L(J))-LX(1,J))*1000}/100
102 LX(4,J)=P(L(J))-LX(1,J)*100-LX(2,J)*100-LX(3,J)*10
102 FORMAT(4X,1HI,2413+/5X,2413+/5X,2413+/5X,2413+/5X,2413+/5X,2413+/5X)
102 PUNCH2:3
102 FORMAT(/in---i------------------------------------------------------
102 2-------------------------------------------------------/JX,7CH------
102 3-------------------------------------------------------/JX,7CH------
102 DC 107 J=1,JK
108 ISUM=S
106 GO TO 106 J=1,JK
105 IF(K(I,J))106,106,105
105 ISUM=ISUM+1
104 CONTINUE
107 PUNCH2:3
106 CONTINUE
108 IF(K(I,J))205,205,209
209 ISUM1(J)=ISUM1(J)+1
208 CONTINUE
204 CONTINUE
206 DC 206 J=1,JK
207 LX(1,J)=ISUM1(J)/10
206 LX(2,J)=ISUM1(J)-LX(1,J)*10
209 PUNCH2:3,((LX(1,J),J=1,71),1=1,2)
3 STOP
4 END
PROGRAM NO. 6

PROGRAM FOR ADDING AND MULTIPLYING RANKS

DIMENSION BLOCK(900), ROUT(900), PAY(900), INC(900)

DO 10 I = 1, N
  REA0, - BLOCK(I), ROUT(I), PAY(I), INC(I)
  FORMAT(416)
  CONTINUE

DO 50 J = 1, N
  IF(BLOCK(J) .EQ. 0) GO TO 53
  IF(ROUTE(J) .EQ. 0) GO TO 53

53 CONTINUE

END
PROGRAM NO. 9

SELECTION PROGRAM FOR
1. MODIFIED SUMMARY TABLE
2. ADDING APPROACH
3. MULTIPLYING APPROACH

DIMENSION BLCK1(900),ROUT1(900),PAY(900),ML(900),BLCKX(75),ROUTX(175),PAYX(75),NC(75)
READ13,NL
13 FORMAT(I5)
DC22 I=1,NL
READ1,BLCK1(I),ROUT1(I),PAY(I),ML(I)
1 FORMAT(2X,I4,2X,I4,2X9I4,9I2)
22 CONTINUE
444 READ14,N
14 FORMAT(I3)
DC333 L=1,N
MX=0
CHECK=0.0
DC4 J=1,NL
IF(BLCK1(L)-BLCK1(J))2,4,2
2 IF(ROUT1(L)-ROUT1(J))44,4,44
44 IF(CHECK)15,55,6
6 DC5 M=1,JK
IF(ROUT1(J)-ROUTX(M))77,4,77
77 IF(BLCK1(J)-BLCKX(M))55,55
5 CONTINUE
DC7 M=1,JK
IF(BLCK1(J)-BLCKX(M))88,4,88
88 CONTINUE
55 MX=MX+1
66 BLCKX(MX)=BLCK1(J)
ROUTX(MX)=ROUT1(J)
PAYX(MX)=PAY(J)
NC(MX)=ML(J)
JK=MX
CHECK=0.0
4 CONTINUE
PUNCH12,BLCK1(L),ROUT1(L),PAY(L),ML(L)
IF(MX)15,15,34
34 PUNCH12,BLCKX(MX),ROUTX(MX),PAYX(MX),NC(MX),MX=1,JK
12 FORMAT(20X,4I5,40X)
PUNCH 16
16 FORMAT(/ /)
333 CONTINUE
GC 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.