CHAPTER 4

TRANSIT FACILITY MANAGEMENT

4.0 Mass Transit

Although traffic systems exhibit many of the special characteristics of service industry, it must be increasingly recognised that the fundamental relationships between the generative process and its regulation are similar to those found with many types of production process. The introduction of the modern data processing software has demonstrated the general interdependence of partial systems and elements along the lines of control circuit with its continual exchange condition and instruction data.

However this present day assessment of the situation was not always so obvious and did not become so without difficulty. The path leading to the use of modern data processing in traffic systems began with marked over optimism concerning its possibilities. However the basic belief that a modern logistics system could bring about decisive improvements in traffic operations is certainly more than ever warranted today.

In general the following two basic principles apply to control of traffic process.

1. The scope of an information system for the control of traffic process must be fixed exclusively by specific process requirements and not by technical possibilities.
2. The organisation of an information system for the control of a traffic process must meet the requirements necessary for optimal human participation.

Man and Machine

The generally applicable requirements for optimal potential human participation in control system can be listed as follows.

The structure of an information system for control purposes should be laid out in a hierarchy in which the data manipulation carried out at each individual level always lies within the scope of normal human understanding. It is not sufficient that only the system developer knows and considers as correct the internal set of rules for data conversion.
The personnel relating to the control system must also be able to understand at least in general what conversion processes are going on, since if this is not the case, their interest in collaboration drops or disappears completely.

Information regarding the state of the system provided to a person as the basis of the control decision must in form, content and density correspond to his or her optimum respectively. With regard to form, for example graphically presented information is more easily absorbed than alpha numeric chains of information. Information content should once again reveal the underlying process in as meaningful a form as possible. That is, with as little as possible abstraction and with pauses between the individual pieces of information sufficient to enable the process changes which must be deduced from that be grasped.

The choice is to be based on process condition information and should be limited to those specifically useful to the process and presented in a sequence determined by their effect. One approach which tends to satisfy this requirements is the use of so called "Menu technique" - Text processing or recording system which leads the operator through the work in a distance sequence of steps each of which offers different alternatives for proceeding.

Adequate time should be allowed to permit the operator to reach each and every decision. This need for distinct reaction time is clearly recognisable. As system interconnections become more complex as the information on conditions becomes more abstract and as the variety of information provided increases, longer reaction time must be allowed. One solution is to offer as far as possible the information on conditions in small modifying steps each of which does not directly require a decision but whose accumulated chronological development makes the direction and importance of the decision to be taken recognisable.

For all traffic operations data manipulations are very important part of the process. Even the smallest transportation service has a mass of incoming and outgoing data to process and disseminate including time-table and service bulletins. This printed output information for customers and company personnel is based on similar voluminous input
data such as traffic studies, tables of trip times, regulations concerning periods of duty, company agreements and so on. In the majority of traffic operations, the transformation of input data into output data is considered to be time consuming, difficult to grasp, inflexible and prone to errors.

Analysing the causes of this problem, it is clear that the pure logic of data processing procedure is in no way complicated. The connections between traffic volume, vehicle capacity, time-table fixed cycle, line times and around trip times can all be understood with the help of very basic calculations.

The use of computer in the comprehensive programming of complex planning process and the automation of data processing procedures does offer particular advantages here. In computer systems for the organisation of operations, all the planning process relationships are stored in automatic programs where the transposition of incoming and outgoing data also take place, thus relieving the planner of all purely routine transfer and representation process and leaving him or her free for truly creative planning activities.

The most important function of the new computer data operations system is to store all relevant data so that after a decision is made by the operator at his or her monitor, all task tables or graphics necessary for operations are immediately available without loss of time or errors. In addition the advantage can be taken of man’s inclination to play because the plan from which subsequent data derived can first be worked out. A first trial version can present worthwhile alternatives immediately side by side for comparison. Small changes can be tried out. The interactive nature of the planning process thus permits improved consultation with third parties, which could be an advantage in service planning. The development of the following data operation capabilities seems worthwhile.

That is programs for the administration of data on services.

- line data
- run trip data and so on.
In the overall design of data operations systems the entire program package must be developed so that each program can be used independently or in combinations with others. Only through a completely modular package can be the specific needs of a given transportation service be satisfied without burdening it with unnecessary procedures.

4.1 Computer Evaluation Systems

The conversion of data operations programs into decision or evaluation can be of use for two planning problems. First they can provide definite assistance when a choice must be made between numerous planning variations of apparently slight significant difference. In this sense decision and evaluation capabilities which may improve the efficiency of the service include programs for the analysis of transportation demand for uniform periods for a given area, network structure and time (yearly, weekly, and daily)

Programs to establish the readiness for service of vehicles and personnel in each planning period.

Evaluation programs for the disposition of staff and facilities in the event of a deviation from a planned operations (incidental demand, breakdown, and so on)

timetable alternations
vehicle schedule alternations.

Every traffic operation includes far more areas of work than just the actual driving from workshops, provisions of materials to the hiring of external services in all of which new short and long term decisions must be made continuously.

Computer optimisation systems

Some planning process at the level of entire traffic network may require the testing of so many planning variants that processing by way of decision programs based on single characteristic values becomes too costly. For task of this sort optimisation programs can be introduced into an appropriate computer evaluation system In the area of local public
transportation the following programs concerned with service efficiency may lead to the application of more effective methods of operational research.

programs for the optimisation of network paths
service network and so on.

Programs for the optimisation of network plans to achieve efficient services
Vehicle schedules personnel duty schedule.

The optimisation models for the solution of traffic problems has shown that it is often insufficient to produce one single solution. Demand requirements and other factors are so varied and subject to so many alternations that uninterrupted adaptation of plans is necessary. The independent use of a computer data operations program obviously cannot guarantee optimisation, but it does ensure much faster and less error-prone planning process. With such a system, the time required for manual planning can be reduced by approximately ninety percent, and procedure become clearer and more flexible.

The field of public transport has of course benefited, as the development of the computer techniques and formalised solution procedures. Mathematical programming approaches include algorithms that are directly based on mathematical model. Dantzig - Ramser formulate vehicle routing problem as a mathematical model in which two interrelated components, one the travelling salesman problem and the other is assignment problem. Christofides discuss Lagrangean relaxation procedure for the routing of vehicles. Christofides also discusses a successful integration of delivery decisions with issues relating to customer service and fleet size determination. Here the method of Optimisation techniques to solve the problems facing by public transport has been discussed.

Vehicle routing and Scheduling

The routing and scheduling of vehicles and their crews is an area important to both operation researchers and transportation planners. Research in this field includes problem formulations and implementation of solution procedures.
The flow chart shows the automatic scheduling process for bus and their crews.

From a practical standpoint the effective routing and scheduling of vehicles and crews can save government and the industry crores of rupees.

First we define what we mean by routing and scheduling of vehicles. A vehicle route is a sequence of pickup and or delivery points which the vehicle must traverse in the order of starting and at a depot. A vehicle schedule is sequence of pickup and delivery points.
together with an associated set of arrival and departure times. The vehicle must traverse the points in the designated order at the specified times. When arrival times at nodes and or at arcs are fixed in advance we refer the problem as a scheduling problem. When arrival times are unspecified the problem is a straightforward routing problem.

4.2 Single Garage Multiple vehicle routing problem

Trip is performed by a bus running from one terminal of a bus route to another block. It is composed of one or several trips normally carried out on a single bus route. Combined block is set a of blocks that can be prompted by the same vehicle

Description of the problem

The transit company has several garages, each characterised by a particular location on the territory and by a specific capacity. Blocks and combined blocks of vehicle must be assigned to each garage in order to minimise the operational cost which are in this case fixed cost of operating the vehicle, the dead head cost (both for crew and vehicle) incurred by each combined block out of this location and garage operating cost related to arrival and departure of vehicles.

Considering the cost structure of this problem. It is clear that the assignment of blocks to garage is interrelated with vehicle scheduling problem. In fact to minimise number of buses to operate a schedule we may consider the problem of allocating the blocks to a minimum number of vehicles. The result however may generate high dead cost to and from the home garage. On the other hand assigning blocks to garages first to minimise deadhead cost may result in using greater number of buses than necessary. The trade off between these two elements must be considered and may be different from one transit organisation to another depending on the amount of subsidies for new buses, the availability of buses and the financing of operating deficits.

The problem of assigning blocks to garages and scheduling vehicle for single garage operation is given below.
The single garage multiple vehicle routing problem asks for a set of delivery routes for vehicles housed at central garage to minimise the total distance to travel. The demand at each node is assumed to be deterministic and each vehicle has a known capacity.
Mathematical model

Minimize \( z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \)

Subject to

\[
\sum_{i=1}^{n} x_{ij} = 1 \quad j = 2 \quad n \quad 1.1
\]
\[
\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1 \quad n \quad 1.2
\]
\[
\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} = 1 \quad v = 1 \ldots \quad nv \quad 1.3
\]
\[
\sum_{i=1}^{n} d_i \sum_{j=1}^{n} x_{ij} \leq k_v \quad v = 1 \ldots \quad nv \quad 1.4
\]
\[
\sum_{i=1}^{n} t_i \sum_{j=1}^{n} x_{ij} + \sum_{j=1}^{n} t_{ij} x_{ij} < T_v \quad 1.5
\]
\[
\sum_{i=1}^{n} x_{ij} < 1 \quad v = 1 \ldots \quad nv \quad 1.6
\]
\[
\sum_{i=2}^{n} x_{ii} < 1 \quad v = 1 \ldots \quad nv \quad 1.7
\]

where \( n \) = number of nodes.
\( nv \) = no of vehicles.
\( K_v \) = capacity of vehicle V.
\( T_v \) = maximum time allowed for a route of a vehicle V.
\( d_i \) = demand at node i.
\( T_i \) = time required for a vehicle to cover the node i
\( t_{ij} \) = travel time for vehicle from node i to node j
\( C_{ij} \) = cost of travel from node i to node j
\( x_{ij} = 1 \) if arc i-j is traversed by vehicle v 0 otherwise

The objective function states that the total cost is to be minimized. The equation 1.1 ensure that each demand node i served by exactly by one vehicle. Equation 1.3 represent the capacity of the vehicle. Equation 1.5 and 1.6 guarantee that vehicle availability is not
exceeded. We assume that the demand at each node does not exceed the capacity of the
system. Now consider there be n nodes to service each demanding v_i (i = 1, 2, n) the
transportation of v_i passengers. Vehicles are stationed at the depot B. Assume all vehicles
have same capacity v_i and when servicing all must start and finish their trips at point B.
Let capacity of any vehicle be greater than demand and each point is serviced by only one
vehicle or one vehicle can service several points.

Determine the set of routes to be used by the vehicles when in service so that the total
distance covered by the entire fleet of vehicles is at a minimum. Keep the point B as
fixed and the n points to be serviced. Let one vehicle service one point at the beginning.
This means at the beginning n vehicles leaves point B service n points and return point
B. The total distance covered by all n vehicles is

\[ 2d(B,1) + 2d(B,2) + 2d(B,3) + \cdots + 2d(B,n) \]

ie \( 2 \sum_{i=1}^{n} d(B,i) \)

where \( d(B,i), i=1,2,3, \ldots n \) is the distance between the points B and point i. If one vehicle
should service two points instead of one let's say i and j then there is a saving mode.

\[ S(i,j) = 2d(B,i) + 2d(B,j) - [d(B,i) + d(i,j) + d(B,j)] \]

\[ = d(B,i) + d(B,j) - d(i,j) \]

Quantitatively \( S(i,j) \) is obtained by joining points i and j into one route. It is clear that the
larger \( S(i,j) \) becomes the better it is to join i and j into one route. Points i and j can not be
joined into one trip if doing so violates one of the constraints in the problem.
Algorithm

Step 1: Calculate $S(i,j) = d(B,i) + d(B,j) - d(i,j)$ for every pair $(i,j)$ of points to be serviced

Step 2: Arrange all $S(i,j)$ in descending order

Step 3: When examining $S(i,j)$ corresponding branch $(i,j)$ is included in the route if so does not violate one of the given constraints

a. neither point $i$ nor point $j$ has been included in a route

b. either point $i$ or point $j$ is already included in a route if that point is not an internal point on the route.

c. both points $i$ and $j$ are included in different routes and neither one is an internal route point (both are external) in which case the route can be joined together

Step 4: if the list of $S(i,j)$ (after formation of the first route) is not completely used up return to step 3 and start from the beginning with the largest unsaving. When the list is used up then the algorithm is finished since all the route have been formed

The distance between the nodes are given below.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\infty$</td>
<td>50</td>
<td>45</td>
<td>70</td>
<td>40</td>
<td>65</td>
<td>40</td>
<td>82</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>$\infty$</td>
<td>55</td>
<td>95</td>
<td>40</td>
<td>90</td>
<td>75</td>
<td>95</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>55</td>
<td>$\infty$</td>
<td>35</td>
<td>70</td>
<td>60</td>
<td>70</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>4</td>
<td>70</td>
<td>95</td>
<td>35</td>
<td>$\infty$</td>
<td>95</td>
<td>45</td>
<td>95</td>
<td>75</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>40</td>
<td>70</td>
<td>95</td>
<td>$\infty$</td>
<td>90</td>
<td>55</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>6</td>
<td>65</td>
<td>90</td>
<td>60</td>
<td>45</td>
<td>90</td>
<td>$\infty$</td>
<td>75</td>
<td>85</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>75</td>
<td>70</td>
<td>95</td>
<td>55</td>
<td>75</td>
<td>$\infty$</td>
<td>80</td>
<td>55</td>
</tr>
<tr>
<td>8</td>
<td>80</td>
<td>95</td>
<td>85</td>
<td>75</td>
<td>95</td>
<td>45</td>
<td>80</td>
<td>$\infty$</td>
<td>90</td>
</tr>
<tr>
<td>9</td>
<td>70</td>
<td>50</td>
<td>85</td>
<td>85</td>
<td>95</td>
<td>60</td>
<td>55</td>
<td>90</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

The vehicle servicing these points have a capacity of $V=70$. The estimated passengers from each node 2,3,4,5,6,7,8,9 are given below

<table>
<thead>
<tr>
<th>node i</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>quant.</td>
<td>Vi</td>
<td>30</td>
<td>10</td>
<td>15</td>
<td>25</td>
<td>40</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>
Calculate the first saving using the formula
\[ S(i,j) = d(1,i) + d(1,j) - d(i,j) \]
Here \[ S(4,6) = d(1,4) + d(1,6) - d(4,6) = 70 + 65 - 45 = 90 \]
Corresponding savings are calculated for all pairs of nodes. The savings are taken in descending order.

<table>
<thead>
<tr>
<th>Branch ( (i,j) )</th>
<th>( S(i,j) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4,6)</td>
<td>90</td>
</tr>
<tr>
<td>(3,4)</td>
<td>80</td>
</tr>
<tr>
<td>(6,9)</td>
<td>75</td>
</tr>
<tr>
<td>(4,8)</td>
<td>75</td>
</tr>
<tr>
<td>(2,9)</td>
<td>70</td>
</tr>
<tr>
<td>(6,8)</td>
<td>60</td>
</tr>
<tr>
<td>(8,9)</td>
<td>60</td>
</tr>
<tr>
<td>(4,9)</td>
<td>55</td>
</tr>
<tr>
<td>(7,9)</td>
<td>55</td>
</tr>
<tr>
<td>(5,6)</td>
<td>50</td>
</tr>
<tr>
<td>(2,5)</td>
<td>50</td>
</tr>
<tr>
<td>(7,8)</td>
<td>40</td>
</tr>
<tr>
<td>(3,8)</td>
<td>40</td>
</tr>
<tr>
<td>(2,3)</td>
<td>40</td>
</tr>
<tr>
<td>(2,8)</td>
<td>35</td>
</tr>
<tr>
<td>(3,9)</td>
<td>30</td>
</tr>
<tr>
<td>(6,7)</td>
<td>30</td>
</tr>
<tr>
<td>(5,7)</td>
<td>25</td>
</tr>
<tr>
<td>(5,8)</td>
<td>25</td>
</tr>
<tr>
<td>(2,4)</td>
<td>25</td>
</tr>
<tr>
<td>(2,6)</td>
<td>20</td>
</tr>
<tr>
<td>(2,7)</td>
<td>15</td>
</tr>
<tr>
<td>(3,5)</td>
<td>15</td>
</tr>
<tr>
<td>(3,7)</td>
<td>15</td>
</tr>
<tr>
<td>(4,5)</td>
<td>15</td>
</tr>
<tr>
<td>(4,7)</td>
<td>15</td>
</tr>
<tr>
<td>(5,6)</td>
<td>15</td>
</tr>
<tr>
<td>(5,9)</td>
<td>15</td>
</tr>
</tbody>
</table>

Here branch \( (4,6) \), has the greatest saving as 90. Therefore the first route is \((1-6-4-1)\).
The no of passenger in the vehicle will be \( V_4 + V_6 = 15 + 40 = 55 < V \), node 4 & 6 can be
joined since this does not violate any constraint concerning the vehicle capacity size. The second order of saving is branch (3,4). Node 4 could be included in the route since the node 4 is not an internal point. Check whether node 3 in the route violate the capacity constraint. so we have $V_3 + V_6 + V_1 = 65 < V$ and conclude that 6 can be included in the route, so that our route is changed to (1-3-4-6-1). Branch (6,9) is next by order of savings. The node (6&9) can not be joined in the route. Since it violate the capacity constraint. But 6 is the internal point of a route. So we cannot start a new bus route using the node (6 & 9). So ignore it. The next saving is the node (4,8). Here also the capacity constraint of the vehicle is violated for the previous route. We can not start a new route, the point 4 is an internal point of the previous route. So ignore it.

The next highest saving is the node (2,9). Here neither 2 nor 9 is an internal point. So we can start a new route, if the vehicle capacity constraint is satisfied.

$$V_2 + V_9 = 30 + 20 = 50 < V$$

So the new route is (1-2-9-1). The next saving is (6,8). This cannot be included in the route since 6 is an internal point. The next saving is node (8,9). Here 9 is not an external point. The node (8&9) can be included in the route if the capacity constraint is satisfied. $V_2 + V_9 + V_8 = 30 + 20 + 10 = 60 < V$. So the route is (1-2-9-8-1). The next saving is the node (4,9). Here the node 4 is an internal point. So it cannot be included in the route. The next saving is the node (7&9). Here 9 is not an external point. But the capacity constraint is violated. i.e $V_2 + V_9 + V_8 + V_7 = 60 + 15 = 75 > V$. So it can not be included in the route. Ignore the next highest savings (3,6),(2,5),(7,8),(3,8),(2,3),(2,8),(3,9),(6,7). Since these are the internal points of the previous route. The next highest saving is (5,7). Neither 5 nor 7 is an internal point. So start a new route.

$$V_5 + V_7 = 35 + 15 = 60 < V.$$ So the route is (1-5-1-7)
Pseudo code in Pascal

Read DistMatrix d(i,j) ∀ i,j where i ≠ j and i & j are nodes
Let maxnodes be the total no. of nodes.
Let savings Array [n] stores values in the order node1, node2, savings.
Let k=0
For i = 1 to maxnodes-1 do
  For j = i+1 to maxnodes do
    begin
      sij = d(i,j)+d(j,i)-d(i,j)
      k = k+1
      Savings[k] = Values of (i,j,sij)
    end
Let n = k
Sort savings Array[n] in descending order of savings
Read passenger(l) for each node
Let Route be the set of nodes
Let iRouteArray[1 to max] be the array of routes
Let l = 1, totalP = 0, x = vehicle capacity
Let Routeindex = 1: RouteArray[Routeindex] = {}
While savingsArray[i].savings < x and route ≠ {} begin

 Route = SavingsArray[i]. Node1 ∪ SavingsArray[i]. Node2;
i = i+1;
end;
While i < index do begin

 Node1 = SavingsArray[i]. Node1
 Node2 = SavingsArray[i]. Node2
If(Node1 ∈ RouteArray(1 to RouteIndex)) and (Node 2 ∈ RouteArray(1 to Route Index)) then i = i+1
else if (node1 ∈ route array(p) where i < p ≤ route index) then begin

 totalpassenger = \sum_{x \in routearray[p]} passenger[x]
totalpassenger = passenger[Node2] + totalpassenger
if total passenger ≤ x then
 routearray[p] = routearray[p] ∪ node2
i = i+1
end
else if (node2 ∈ routearray[p] where 1 < p ≤ route index) then begin

 totalpassenger = \sum_{x \in routearray[p]} passenger[x]
totalpassenger = totalpassenger + passenger[node1]
if totalpassenger ≤ x then
routearray[p]=routearray[p]∪node1
i = i+1
end

elseif(node1 ∈ routearray[p] and node2 ∈ routearray[p])
where 1 < p ≤ routeindex
begin

totalpassenger = ∑passenger[x] | x ∈ routearray[p]
totalpassenger = passenger[node1] + passenger[node2] + totalpassenger
if totalpassenger ≤ x then
routearray[p] = routearray[p] ∪ node1 ∪ node2
else begin
routearray[routeindex] = routearray[routeindex] ∪ [1]
routeindex = routeindex+1
routearray[routeindex] = node1 ∪ node2
end
i = i+1
end

end

4.3 Multi Garage Vehicle Routing.

Slight modification may be made in the single garage vehicle routing problem. Letting nodes 1,2,3,4,......... M denotes the garage. A straightforward extension of the problem
just discussed is to allow vehicles to reside at more than one garage and to seek minimum number of vehicles needed to cover all the task. The problem over the entire network disregarding the depot that would house each vehicle has been discussed.

Mathematical Model

Minimise \( z = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{v=1}^{n_v} c_{ij} x_{ijv} \)

Subject to

\[ \sum_{j=1}^{m} \sum_{v=1}^{n_v} x_{ijv} = 1 \quad j = m+1, m+2 \quad n \quad 2.2.1 \]

\[ \sum_{i=1}^{n} \sum_{v=1}^{n_v} x_{ijv} = 1 \quad i = m+1, m+2 \quad n \quad 2.2.2 \]

\[ \sum_{i=1}^{n} x_{ijv} - \sum_{j=1}^{m} x_{ijv} = 0 \quad v = 1, \ldots, n_v \quad 2.2.3 \]

\[ y_{ij} = 1, 2, n \quad 2.2.4 \]

\[ \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ijv} \leq k_v \quad v = 1, \ldots, n_v \quad 2.2.5 \]

\[ \sum_{i=1}^{n} l_i \sum_{j=m+1}^{m} x_{ijv} + \sum_{i=1}^{n} \sum_{j=1}^{m} l_{ijv} x_{ijv} < T_v \quad 2.2.6 \]

\[ \sum_{i=1}^{n} \sum_{j=m+1}^{m} x_{ijv} \leq 1 \quad v = 1, 2, 3 \quad n_v \quad 2.2.7 \]

Algorithm

The problem of routing vehicles when there are several depots is even more complex than the same problem with one depot. When there are several depots the problem appears of joining points which are serviced by individual depots. The problem of vehicle routing on a network with several depots is most often solved in two steps. In the first step individual depots are joined to groups of points to be serviced. The second step solves the problem of vehicles each depot and its corresponding group of points. The method is as follows. First the following relation is calculated for each point i to be
serviced $a_i = d_1(i)/d_2(i)$ where $d_1(i)$ and $d_2(i)$ are the distance between point $i$ first closest depot and the second closest depot. The number $x$ is introduced in the process for which $0 < x < 1$. The value of $x$ is arbitrarily chosen and then compared to $a_i$. If $a_i < x$ then the point is joined to the nearest depot. (A vehicle from the nearest depot will service it.). If $a_i > x$ then the point is left for further consideration. When all the points for which $a_i < x$ are joined to corresponding depots. The points for which $a_i > x$ are taken into consideration. These points are joined to depots as follows. Let there be two points, $b$ & $c$ joined to depot $Bp$ then we increase the route length starting from depot $Bp$ by

$$d_{bc} = d_{bc} + d_{ac} - d_{bc}$$

It is clear that we will join the point $a$ to the depot where its addition will cause least increase in the route length starting from this depot. When all the points have been joined to depots in this manner, then the algorithm developed for the case of one depot is applied.

The table represents the distance between the individual nodes.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\infty$</td>
<td>45</td>
<td>38</td>
<td>70</td>
<td>56</td>
<td>92</td>
<td>37</td>
<td>25</td>
<td>16</td>
<td>75</td>
<td>68</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>$\infty$</td>
<td>25</td>
<td>67</td>
<td>48</td>
<td>54</td>
<td>16</td>
<td>92</td>
<td>97</td>
<td>36</td>
<td>48</td>
<td>52</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>25</td>
<td>$\infty$</td>
<td>25</td>
<td>68</td>
<td>34</td>
<td>92</td>
<td>16</td>
<td>47</td>
<td>72</td>
<td>35</td>
<td>87</td>
</tr>
<tr>
<td>4</td>
<td>70</td>
<td>67</td>
<td>25</td>
<td>$\infty$</td>
<td>34</td>
<td>47</td>
<td>28</td>
<td>34</td>
<td>43</td>
<td>66</td>
<td>14</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>56</td>
<td>48</td>
<td>68</td>
<td>34</td>
<td>$\infty$</td>
<td>86</td>
<td>45</td>
<td>56</td>
<td>62</td>
<td>55</td>
<td>23</td>
<td>74</td>
</tr>
<tr>
<td>6</td>
<td>92</td>
<td>54</td>
<td>34</td>
<td>47</td>
<td>86</td>
<td>$\infty$</td>
<td>54</td>
<td>84</td>
<td>59</td>
<td>93</td>
<td>75</td>
<td>44</td>
</tr>
<tr>
<td>7</td>
<td>37</td>
<td>16</td>
<td>92</td>
<td>28</td>
<td>45</td>
<td>54</td>
<td>$\infty$</td>
<td>43</td>
<td>74</td>
<td>25</td>
<td>83</td>
<td>63</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>92</td>
<td>16</td>
<td>34</td>
<td>56</td>
<td>84</td>
<td>43</td>
<td>$\infty$</td>
<td>87</td>
<td>37</td>
<td>92</td>
<td>74</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>97</td>
<td>47</td>
<td>43</td>
<td>62</td>
<td>59</td>
<td>74</td>
<td>87</td>
<td>$\infty$</td>
<td>18</td>
<td>57</td>
<td>94</td>
</tr>
<tr>
<td>10</td>
<td>75</td>
<td>36</td>
<td>72</td>
<td>16</td>
<td>55</td>
<td>93</td>
<td>25</td>
<td>37</td>
<td>18</td>
<td>$\infty$</td>
<td>65</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>68</td>
<td>48</td>
<td>35</td>
<td>14</td>
<td>23</td>
<td>75</td>
<td>83</td>
<td>92</td>
<td>57</td>
<td>65</td>
<td>$\infty$</td>
<td>77</td>
</tr>
<tr>
<td>12</td>
<td>25</td>
<td>52</td>
<td>87</td>
<td>96</td>
<td>74</td>
<td>44</td>
<td>63</td>
<td>74</td>
<td>94</td>
<td>18</td>
<td>77</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node i</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1(i)$</td>
<td>25</td>
<td>48</td>
<td>34</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>36</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>$d_2(i)$</td>
<td>67</td>
<td>56</td>
<td>54</td>
<td>37</td>
<td>25</td>
<td>47</td>
<td>72</td>
<td>48</td>
<td>52</td>
</tr>
</tbody>
</table>
The table represents the ratio of first nearest depot and second nearest depot.

<table>
<thead>
<tr>
<th>Node i</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_i )</td>
<td>.37</td>
<td>.85</td>
<td>.62</td>
<td>.43</td>
<td>.64</td>
<td>.34</td>
<td>.5</td>
<td>.72</td>
<td>.48</td>
</tr>
</tbody>
</table>

Take the arbitrary value for \( x = 0.65 \). By stated algorithm we will allocate the nodes to the nearest garage.

<table>
<thead>
<tr>
<th>Node i</th>
<th>depot i</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
</tr>
</tbody>
</table>

i.e the node 4, 6, 8 are joined in depot 3, the node 7, 10 are joined in depot 2 and the nodes 9, 12 are joined in depot 1. The nodes 5 & 11 are not joined in any depot. Since it violates the algorithm variable value \( x \). If the above nodes are joined in any depot causes additional increase in distance. In order to minimise the increase in distance we follow the following procedure.

Find out increase in distance if the node is joined in depot 1

\[
d_0(9,12) = d(9,5) + d(5,12) - d(9,12) = 62 + 74 - 94 = 42
\]
Pseudo algorithm

let distance [row, col] be the distance matrix
for i = 4 to row
a[i] = distance[1,i]/distance[2,i] assuming 3 depots
let b be the set, keeping some row no. route[1], route[2], route[3] etc.
for i = 4 to row begin
if ar[i] ≤ x then
begin
a = distance[1,i]
b = distance[2,i]
c = distance[3,i]
if a < b then
if a < c then route[1] = i ∪ route[1]
else if b < c then route[2] = i ∪ route[2]
end
else b = b \cup i
end
\forall x, x \in b do begin
Min distance = max
\forall route[i]
begin
if single(route [i]) then begin
    p1 \in route[i], p2 = i do step 8
else \forall p1,p2 \mid p1 \in route[i] \& p2 \in route[i] \& p1 \neq p2 begin
    distp1p2 = distance[p1,x] + distance[p2,x] - distance[p1,p2]
if mindist > distp1p2 then begin
    index = 1, mindist = distp1p2, node1 = p1, node2 = p2
end
end
end
stop
4.4 Bus Scheduling Problem

Formulating the bus scheduling problem as a quasi assignment model. Denote i be the index set of short trips and define the linking cost \( C_{ij} \) for each feasible pair \((i,j)\) of trips.

The quasi assignment model explicitly represents the depot as a short trip which is given the index \( n+1 \). Hence cost relative to linkages from or to the depot are fixed as follows.

\[
\begin{align*}
C_{i,n+1} &= d = \frac{D}{2} & i &= 1, 2, 3 \quad n \\
C_{n+1,j} &= d = \frac{D}{2} & j &= 1, 2, 3 \quad \text{.........}n \\
C_{n+1,n+1} &= 0
\end{align*}
\]

The cost relative to unfeasible linkages are made infinite. Therefore if the trips are ordered by increasing value of starting time, the cost matrix becomes

\[
\begin{pmatrix}
1 & 2 & 3.4 & n & n+1 \\
1 & c_{12} & c_{13} & c_{14} & c_{1n} & d \\
2 & & c_{23} & c_{24} & c_{2n} & d \\
3 & & & c_{3n} & d \\
4 & & & & c_{4n} & d \\
n & & & & C_{nn} & d \\
n+1 & d & d & d & d & 0
\end{pmatrix}
\]

The decision variable are defined as follows

\[
x_{ij} = \begin{cases} 
1 & \text{if trip } i \text{ directly connected to } j \\
0 & \text{otherwise}
\end{cases}
\]

\[
x_{n+1,j} = \begin{cases} 
1 & \text{if the depot directly supplies a bus for trip } j \\
0 & \text{otherwise}
\end{cases}
\]

\[
x_{i,n+1} = \begin{cases} 
1 & \text{if after trip } i \text{ the bus immediately returns to the depot} \\
0 & \text{otherwise}
\end{cases}
\]

\[
x_{n+1,n+1} = \text{number of buses remaining idle at the depot}
\]
The problem becomes
\[
\min z = \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} c_{ij} x_{ij}
\]
\[
\sum_{j=1}^{n+1} x_{ij} = 1 \quad i = 1, 2, 3, \ldots, n
\]
\[
\sum_{i=1}^{n+1} x_{ij} = 1 \quad j = 1, 2, 3, \ldots, n
\]
\[
\sum_{i=1}^{n} x_{i, n+1} = n
\]
\[
\sum_{j=1}^{n+1} x_{i, n+1} = n
\]
\[
x_{ij} \geq 0 \text{ and } x_{ij} \in \{0, 1\} \quad (i, j = 1, 2, 3, \ldots, n+1)
\]

**The Graph associated with the problem**

A directed graph \( G = (V, \Lambda) \) associated with the problem can be defined as follows. \( V = \{1, 2, 3, \ldots, n+1\} \) where \( n+1 \) stands for the depot. The arcs in \( \Lambda \) represent feasible linkages between the depot and trips with a cost \( c_{ij} \) on each arc \((i, j)\) as well as linkages between the depot and trips whose cost are set to the fixed value \( D/2 \). The vertices are numbered so that only arcs \((i, j)\) with \( i < j \) exist, except for the depot which is connected in both directions to every vertex in \( i \). The bus scheduling problem consists of finding the minimum cost set of hamiltonian circuits passing through the depot and covering every vertex in \( i \). This graph has a very special structure, since no return arcs exist except for the depot. Hence the problem can be approached as quasi assignment model where the typical assignment constraints are valid for all the vertices except one. This vertex corresponding to the depots is such that all the arcs incident to it have identical cost.
Algorithm

step1: Reduce the cost matrix in order to obtain at least one null cost entry for each row and column

\[
\text{set } v_i = 0 \text{ for } i = 1, 2, 3 \quad n+1
\]
\[
\text{set } w_j = 0 \text{ for } j = 1, 2, 3 \quad n+1
\]

For every row \( i = 1, 2, 3 \) such that \( w_j = 0 \) and \( c_{ij} = 0 \) set \( v_i \) equal to the index of the first column say \( j(i) \) such that

\[
w_{j(i)} = 0 \text{ and } c_{j(i)i} = 0 \text{ then set } w_{j(i)} = i \text{ if } j(i) \leq n
\]

For every column \( j = 1, 2, 3 \) such that \( w_j = 0 \) and \( c_{n+1,j} = 0 \) set \( w_j = n+1 \)

step2: Set \( A = \{ i \leq n : v_i = 0 \} = \emptyset \)

\( A' = \{ 1 \leq i \leq n : v_i = n+1 \} \)

\( B' = \{ 1 \leq j \leq n : w_j = n+1 \} \)

Compute \( nr = |A'|, nc = |B'| \) if \( nr \geq nc \) go to 2.b otherwise goto 2.c

b if \( nr = nc \) and \( |A| = 0 \) goto 3 otherwise do \( A = A \cup A' \cup \{ n+1 \} \)

\( B = B \cup B' \cup \{ n+1 \} \) goto 2.c

c Search for a column \( j \not\in B \) such that \( c_{ij} = 0 \) and \( i \in A \)

if such a column does not exist goto 2.d

if a column \( j \) is found with \( w_j \in \{0,n+1\} \) goto 2.c

otherwise, update \( A = A \cup \{ w_j \} \)
\( B = B \cup j \)
repeat 2.c

d Compute \( C_{\min} = \min C_{ij} \) where \( i \in A, j \not\in B \) and update cost matrix as follows

\[
C_{ij} = C_{ij} - C_{\min} \quad i \in A, \ j \not\in B
\]
\[ C_{ij} + C_{\text{min}} \quad i \notin A, \ j \in B \]

\[ C_{ij} \quad \text{otherwise} \]

goto 2.c

Perform a transfer of assignment \( v_{n+1} = w_{n+1} = 0 \)

Step 3

The optimal assignment which is given either by array v or by array w has been obtained.

Example

To illustrate the algorithm described above, a small example corresponding to six short trip problem with cost matrix is given below. The symbol "-" stands for non admissible linkages and the cost incurred by each bus is 40 d(20)

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 2 & 4 & 7 & 20 & \\
2 & 1 & 5 & 20 & \\
3 & 3 & 2 & 20 & \\
4 & 2 & 20 & \\
5 & 1 & 20 & \\
6 & & & 20 & \\
7 & & & .20 & 20 & 20 & 20 & 20 & 0 \\
\end{array}
\]

The optimal solution is reached with one transfer of assignment and four updates over the cost matrix. After step 1 an initial assignment is obtained with cost equal to 68.

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 3 & 0^* & 2 & 5 & 18 & \\
2 & 0 & 0 & 4 & 19 & \\
3 & 4 & 0^* & 0 & 18 & \\
4 & 5 & 0^* & 18 & \\
5 & 6 & 0^* & 19 & \\
6 & 7 & & 0^* & \\
7 & 0 & 0^* & 20 & 19 & 20 & 20 & 0 \\
\end{array}
\]
The $0^* \text{ position } (i,j)$ means that row $i$ is assigned to column $j$. $v_i = j$ and $w_j = i$. A null value is inputted to $v_{i+1}$ or $w_{j+1}$.

In step 2.a, $\Lambda = \{2\}$, $B = \emptyset$, $n_r = 1$, and $n_c = 2$. Since $n_r < n_c$ and rows 1, 3, 4, 5, 6, 7 are crossed over. Proceeding to 2.c, the column $j = 3 \not\in B$ is found with $C_{23} = 0$. Since $w_3 = 1 \neq 0$ then $A = A \cup \{1\} = \{2, 1\}$, $B = \{3\}$, that is the line over row 1 is removed and column 3 is now covered.

Repeating 2.c no more columns $j \not\in B$ with $C_{ij} = 0$ and $i \in A$ are found. Next following 2.d a minimum uncovered value equal to 2 is found at position (1,5). After updating the cost matrix and returning to 2.c, the column $j = 5 \not\in B$ with $C_{15} = 0$ and $i = 1 \in A$ is selected. Since $w_5 = 4$ update $A = A \cup \{4\} = \{2, 1, 4\}$ and $B = B \cup \{5\} = \{3, 5\}$

Repeating 2.c & 2.d, no more columns $j \not\in B$ with $C_{ij} = 0$ and $i \in A$, are found. Next following 2.d a minimum uncovered value equal to 2 is found at position (1,5). After updating the cost matrix and returning to 2.c, the column $j = 5 \not\in B$ and $C_{15} = 0$ and $i = 1 \in A$ is selected. Since $w_5 = 4$ update $A = A \cup \{4\}$ and $B = B \cup \{5\} = \{3, 5\}$

\[ \begin{array}{ccccccccc} \\
| & 1 & 2 & 3 & 4 & 5 & 6 & 7 |
\end{array} \]

\[ \begin{array}{ccccccccc} \\
1 & 3 & 0^* & 2 & 5 & 18 |
2 & 0 & 0 & 4 & 19 |
3 & 4 & 0^* & 0 & 18 |
4 & 5 & 0^* & 18 |
5 & 6 & 0^* & 19 |
6 & 7 & 0^* & 0 |
7 & 0 & 0^* & 20 & 19 & 20 & 20 & 0 |
\end{array} \]
Repeating 2.c and 2.d no more column $j \notin B$ with $C_{ij} = 0$ and $i \in A$ are found and a minimum uncovered entry value is obtained at (2,6). After updating the matrix the set $A$ and $B$ are modified $A = \{2,1,4,5\}$, $B = \{3,5,6\}$.

Following 2.d the minimum uncovered element 14 is found at (1,7). After updating the cost matrix and returning to 2.e the column $j = 7$ with $C_{77} = 0$ and $1 \in A$ is selected. In this case $w_7 = 0$ and a transfer is assignment is performed in step 2.c.
Returning to 2.a $\Lambda = \Phi$, $B = \Phi$ and $nr = nc = 2$. From 2.b proceeds to step 3 where the algorithm stops with the optimal solution given by $v$ or $w$ having a cost equal to 87.

**Pseudo Code**

Cost matrix $C[n,n]$  
Augment one more column & row to the matrix  
All newly added elements have a value $d$  
$C[n+1,n+1] = 0$ ie No buses are idle  
For row = 1 to n do begin  
Find $k = \min C[row,j]$  
Subract $k$ from $C[i,j]$ for $i = row$ & $j = 1$ to $n+1$
end

For $i = 1$ to $n+1$ do begin $v[i] = 0$ ; $w[i] = 0$ end
For col = 1 to n do begin
Find $k = \min[i,col]$  
Subract $k$ from $C[i,j]$ for $j = col$ & $i = 1,2, n+1$
end

For $i = 1$ to n do begin
Find the first column say $j$ such that $w[j] = 0$  
$C[i,j] = 0$  
$w[j] = i$ & $v[i] = j$
For $j = 1$ to n do
if $w[j] = 0$ and $C[n+1,j] = 0$ then $w[j] = n+1$
Let $A,B,a,b$ be sets  
$A = \{ \}$  
$B = \phi$  
$a = \{ \}$  
$b = \{ \}$
For $i = 1$ to n begin
if $v[i] = 0$ then include $i$ in $A$
if $v[i] = n+1$ then include in $a$
if $w[j] = n+1$ then include $j$ in $b$
compute $nr =$ no of elements in $a$
\( nc = \text{no of elements in } b \)

\[ \text{if } nr < nc \text{ then goto step 10} \]

\[ \text{if } nr = nc \text{ and } A = \{ \} \text{ then goto 12 (optimal)} \]

else begin

\[ A = A \cup a \cup \{n+1\} \]
\[ B = B \cup b \cup \{n+1\} \]

end

end

\[ \text{found} = \text{found2} = \text{false} \]

For \( j = 1 \) to \( n+1 \) do begin

if \((j \in B) \text{ and } (j \in A) \text{ and } C[i,j] = 0\) then begin

\[ \text{found} := \text{true}; \text{ found2 }:= \text{true} \]

end

if (found2) then

if \( w[j] = 0 \text{ or } w[j] = n+1 \) goto 11

else begin

include \( w[j] \in A \)

include \( j \in B \)

found2 := false

end

end

if not found begin

\[ c_{\text{min}} = \text{max} \]

for \( i = 1 \) to \( n+1 \) begin

for \( j = 1 \) to \( n+1 \)

if \( C[i,j] < \text{max} \) AND \( (i \in A) \text{ and } (j \in B) \) then
Vehicle and crew scheduling problems can be thought of as route scheduling problems with additional constraints having to do with the times when various activities may be carried out. In general vehicle and crew scheduling problems interact with one another, the specification of vehicle schedules will set certain constraint on the crew schedules and vice versa. Ideally therefore one would solve the two problems simultaneously. The input to vehicle and crew scheduling problems is a set of tasks. Each task has a specified start time, end time, start location and end location. The cost function consists of components that might include vehicle operating cost and crew operating cost. The fleet of vehicles and the set of crews may be limited and may be housed at one or more depots. The type of scheduling problems that evolves is a function of the constraints imposed upon the formation of schedules the type of tasks being serviced and the locations where these tasks must be carried out.
In the above example, each task has the same start and end location (the depot). The start and end times of each task are given within the node representing the task. A solid branch between two nodes indicates that these two nodes are on the same vehicle schedule and that vehicle schedule will follow the orientation of the branch. The dotted branches indicate feasible connections which are not used in the solution. A branch is drawn from node i to j if the start time of the task j is greater than the end time of the task i and if the start time of the task j is less than or equal to end time of the task i plus one hour. There is no branch from node 6 to node 5. Since the start time of node 5 is less than the end time of node 6 and no branch from node 6 to node 8. Since the start time of node 8 is greater than the end time of node 6 plus one hour. Each vehicle schedule is assumed to be no longer than 8 hours.

We would examine the relationship between crew and vehicle scheduling. Each individual schedule has a set of point where one crew can relieve another. In the mass transit setting, the relief point is a designated stop along a transit line. Each vehicle schedule is split into pieces at one or more relief points. An individual crew scheduling is then obtained by grouping one or more of these pieces together. The feasibility of
joining one piece with another depends not only on the end time of the first piece, relative to the start time of the second but also on the end location of the first piece relative to the start location of the second.

**Scheduling works at a fixed location.**

For this divide the work day into $T$ time intervals and specify a demand for workers $d_t$ associated with each time interval $t = 1, 2, 3, ..., T$. The worker scheduling problem is to find a set of worker schedules that cover all required works. It is assumed that workers are interchangeable and that any worker can be relieved at the end of any time period and that any worker can start at the beginning of any time period. To define a combined crew or vehicle scheduling problem the nature of the crew movements will be considered. Each line has one or more relief points which are stops along the line where one crew may relieve another. Thus a crew period of work on a single vehicle starts and ends at either a relief point or at the garage.

**Mathematical model**

The set of tasks with each task $i$ characterised by a start location $SL_i$, a start time $ST_i$ and end of location $EL_i$ and an end time $ET_i$. For any pair of location $L_1$ and $L_2$ we denote by $TM(L_1, L_2)$ the time to travel from $L_1$ and $L_2$ denotes the location of the depot. The node set $N$ consists of a nod representing each task together with a source node $s$ and the link node $t$. The arc set $A$ is obtained by inserting an arc from the task node $i$ to task node $j$ if it is feasible for a single vehicle to service both task. Further an arc is inserted from $s$ to each task node and from each task node to $t$. These arcs represents trips to and from the depot. Each $(s, t)$ path through this network represents a possible schedule for a single vehicle. The number of duties generated is equivalent to number of variables. The objective of the set partitioning is to select a subset of duties from all variables so as to minimise the cost of covering work.
Furthermore, it is assumed that the total work to be covered in the bus schedule can be expressed as a number of shorter increments of works. An increment of work is defined as work between two adjacent relief on a single block.

\[
\min z = \sum_{j=1}^{n} c_j x_j
\]

Subject to

\[
\sum_{j=1}^{n} a_{ij} x_j = 1 \quad (i = 1, 2, 3, \ldots, m)
\]

where

\[
\begin{align*}
x_j &= 1 \text{ if duty } j \text{ is retained in the solution} \\
&= 0 \text{ otherwise}
\end{align*}
\]

\[
c_j \text{ is the cost of duty } j \text{ taking in to account the time spent on a bus plus any allowances for such thing as over time.}
\]

\[
a_{ij} = 1 \text{ if trip } j \text{ covers crew } i, \quad 0 \text{ otherwise}
\]

It is clear that unless the problem to be solved are sufficiently small, in terms of number of buses and number of driver duties required or unless steps are taken some how to make them smaller than they were, that set partition would not be feasible because of the enormous number of rows and columns in the mathematical formulation. The original problem can be decomposed to a number of smaller problems, that set partitioning can be used to solve the sub problems and that the resulting solution can then be combined to form a total solution.

The set partitioning problem is defined as

\[
\min \{ \mathbf{e}^T \mathbf{x} : A \mathbf{x} = \mathbf{c}, x_j = 0 \text{ or } 1 \forall j \in N \}
\]

Where \( A \) is \( mn \) matrix of zeros and ones,

\( C \) is an arbitrary \( n \) vector

\( e = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1) \) is an \( m \) vector

and \( N = \{1, 2, 3, \ldots, n\} \)
If the rows of $A$ are associated with the elements of the set $M = \{1, 2, 3, \ldots, m\}$ and each column $a_j$ of $A$ with the subset $M_j$ of those $i \in M$ such that $a_{ij} = 1$.

A partial list of applications of set partitioning is crew scheduling, vehicle routing, information retrieval etc.

**Algorithm**

If $R_i$ is a null vector for any $i$, then no solution exists.

If $R_k$ is a unit vector with a one in column $t$ then $x_t = 1$ in every solution and all rows $R_i$ such that $a_{it} = 1$ may be deleted since they are covered by $A_t$. Also every column $A_p$, $p \neq t$ such that $a_{pt} = 1$ may be deleted, in order to overcover row $R_i$.

Arrange the columns $A_j$, $j \in P$ into $m$ lists as described above. Set $W = T = \emptyset$, $Z(w) = 0$, $V = Q$, and $Z = \infty$ ($Z$ = value of the best solution).

Let $V = Q - T$ and $i = \min \{i \mid i \in V\}$. Set an indicator that tells us to begin at the top of list $i$.

Begin at the indicated position in list $i$ and examine, in order of increasing cost the columns of the list. If we find a column $j$ such that $T \cap S_j = \emptyset$ and $Z(w) + C_j \leq Z$ goto 7.

There are no optimal solutions containing the columns in the current partial solution. If $W \neq \emptyset$ terminate. If $W = \emptyset$ let $k$ be the last element included in $w$. Set $w = w - \{k\}$. $Z(w) - C_k$ and $T = T - S_k$. Let $l = \text{number of the list in which column } k \text{ is stored}$ and set an indicator at the position below column $k$ in list $i$. Goto step 5.

Set $w = w \cup j$, $Z(w) = Z(w) + C_j$ and $T = T \cup S_j$. If $T = Q$ goto 8. otherwise goto 4.

A new best solution is found. Set $Z = Z(w)$ and save $w$. Goto step 6.
The given data are

\[
\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
4 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
5 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
6 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{array}
\]

c[j] 18 22 14 36 17 14 8 24 11 7

The data are organised as shown in the table

<table>
<thead>
<tr>
<th>Column</th>
<th>Sj</th>
<th>Cj</th>
<th>Field in list</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(5,6)</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>(3,5,6)</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>(1,4)</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>(2,3,4,6)</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>(1,2)</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>(2,6)</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>(5)</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>(3,4,6)</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>(3,5)</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>(4)</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>

Let \( P_1 = \emptyset, \ Q = (1,2,3,4,5,6), \ P = (1,2,3,4,5,6,7,8,9,10) \)
\( w = T = \emptyset, \ Z = \infty \)

step 2 \( V = (1,2,3,4,5,6) \) \( i = 1 \)
step 3 \( S_j = S_3 = (1,4) \)
step 4 \( w = 3 \) \( Z(w) = 14, \ T = (1,4) \)
step 2 \( V = (2,3,5,6) \) \( i = 2 \)
step 3 \( S_j = S_6 = (2,6) \)
step 4 \( w = (3,6) \) \( Z(w) = 28, \ T = (1,2,4,6) \)
step 2 \( V = (3,5) \) \( i = 3 \)
step 3 \( S_j = S_9 = (3,5) \)
step 5 \( w = (3,6,9) \) \( Z(w) = 42, \ T = (1,2,3,4,5,6) = Q \)

76
step 6  \( w(3,6,9) \ Z = 42 \)

step 4  \( w = (3,6) \ Z(w) = 28, \ T = (1,2,4,6), \ i = 3 \)

step 3  No \( S_j \) found

step 4  \( w = (3), \ Z(w) = 14, \ T = (1,4), \ i \ = \ 2 \)

step 3  No \( S_j \) found

step 4  \( w = \phi, \ Z(w) = 0, \ T = \phi, \ i = 1 \)

step 3  \( S_j = S_5 = (1,2) \)

step 5  \( w = (5), \ Z(w) = 17, \ T = (1,2) \)

step 2  \( V = (3,4,5,6), \ i = 3 \)

step 3  \( S_j = S_5 = (3,5) \)

step 5  \( w = (5,9), \ Z(w) = 31, \ T = (1,2,3,5) \)

step 2  \( V = (4,6), \ i = 4 \)

step 3  \( S_j = S_{10} = 4 \)

step 2  \( V = 6, \ i = 6 \)

step 3  No \( S_j \) found

step 4  \( w = (5,9), \ Z(w) = 31, \ T = (1,2,3,6), \ i = 4 \)

step 3  No \( S_j \) found

step 4  \( w = (5), \ Z(w) = 17, \ T = (1,2), \ i = 3 \)

step 3  \( S_j = S_2 = (3,5,6) \)

step 5  \( w = (5,2), \ Z(w) = 39, \ T = (1,2,3,5,6) \)

step 2  \( V = 4, \ i = 4 \)

step 3  No \( S_j \) found

step 4  \( w = (5), \ Z(w) = 17, \ T = (1,2), \ i = 3 \)

step 3  \( S_j = S_8 = (3,4,6) \)

step 5  \( w = (5,8), \ Z(w) = 41, \ T = (1,2,3,4,6) \)

step 2  \( V = (5), \ i = 5 \)

step 3  No \( S_j \) found

step 4  \( w = (5), \ Z(w) = 17, \ T = (1,2) \ i = 3 \)

step 3  No \( S_j \) found
step 4  \[ w = \phi Z(w) = 0, \ T = \phi, \ i = 1 \]
step 3  No \( S_j \) found
Step 4  Terminate. the only optimal solution is \( x_3 = x_6 = x_9 = 1 \),
all other \( x_j = 0 \), \( z = 42 \)

4.6  Optimizing Program Modules and software Tool

In our country planning and operational management of public transport has become increasingly complex, the ability to respond to changing demand for travel. The planning and scheduling problems faced by public transport is unimaginable. At the strategic level routing and frequency decisions are made in response to changing trends in demand. Clearly these decisions cannot be altered too often because of inconvenience to the travelling public. However it is undesirable for routes and frequencies to remain unchanged over a long periods. Manual scheduling is a skilled and time consuming job. To tackle the complexities of these problems at all levels a computer based decision support system has been developed to assist public transport authority. The computer based support system for vehicle scheduling and crew scheduling has been discussed.

Once the desired frequencies of service that should operate along each route through the day have been determined a set of time-tables be constructed and vehicles are scheduled to these time-tables. The main objective is to meet the desired service levels at minimum cost which is often interpreted as using the minimum number of vehicles. These are mainly concerned with providing a reliable service and avoiding an excessive amount of dead running.

The problem of crew scheduling may be stated as one of finding the set of crew duties of least total cost that states a given bus schedule. In practice, however the scheduler usually knows or has a good idea of the number of duties he is prepared to use in order to cover a bus schedule. In these cases it may be impossible to cover every trip of the bus schedule without minor adjustments being made to some of the times at which the trips are made. Such adjustments are usually preferable to and more cost effective than using
extra crew. Thus a better formulation is to obtain as near to a complete crew schedule as possible using a specified number of duties for a given bus schedule. A duty schedule is to be acceptable to both traffic management and crews. It must possess other characteristics such as minimum work time of a duty and times at which meal breaks may be taken. Thus in determining the validity of a duty it is often required to take the following factors into account.

- Start time
- length of work portions
- finish time of work portion
- duty spread
- finish time
- length of work for the whole duty

In the area of transportation planning and traffic control, development in Information Technology have presented great opportunities. Current developments in Computer Technology in relation to software systems making the computers easier to use and providing the user with greater access to relevant computer held data in the form of databases. The computer linked via a “local area network” in which user has to access to the data stored on the work station as well as his own data. In this case a distributed database is involved in which the relevant data for the application is distributed over a number of computers rather than residing in just one computer. The workstations of the type described above are being common. Such systems enable decision makers to obtain computer assistance in many of the decision areas in which they are involved. Thus greater ease of access to direct computing power and different databases together with software systems that recognise that the managers problem are inter linked, offers integrated support to decision making.
Data required for the system

Garages: This simply gives a correspondence between a two digit code for a garage and its name, for brevity in describing other items related to garages.

Places: This gives a correspondence between a three character code and the actual place name for certain points within the urban area which are frequently referred by a code rather than by full description. Such points are start point, terminal point, crew change point, fuel filling point and so on.

Route itinerary: This contains a list of the streets down which the bus must travel in order to make the outbound journey, together with a separate list for the return journey. It also contains information regarding which garages serve the route and crew change points.

Time table data: A route schedule is the kernel of a set of timetables, all of which have been created using same route itinerary and garage route instructions. It contains the timing details relating to a specific route itinerary. In particular it gives the times of the first and the last service to be run, and trip times in each direction for specified time phases for each time phase it also contains target figures for average headway, layover time at each terminus which are necessary for automatic creation of time tables.

Time table: A time table is a collection of running boards which go together to make up a complete service on a specific route for particular days. Each running board refers back to the relevant set of base data from which it was created. Time-tables for different routes are grouped together when crews swap from one route to another.

Running Board: A running board is a description of the work to be done by a specific bus. It shows the time that is required for the bus to leave and finally return to the garage and the departure and arrival times at each of the termini served throughout the day.
Function of the system

Automatic time-table creation and crewing is handled by mathematical models. It takes a set of requirements and produces its best result within those requirements.

Interactive time table creation or amendment

The system provides interactive screens for the creation and amendment of all of the base data associated with the time-table. Most of this base data will be used across a wide range of time-tables. The creation of a time-table can also be done interactively using the base data within the system.

Using the time-table editor, new time-tables can be drawn up or existing one be modified. The program allows graphic representation of the input data in the form of a time-table. The user is provided with tools to select any options available on the screen.

1. Delete a journey
2. Insert a journey
3. Re time a journey
4. Break a link between two Journeys
5. Form a link between two journeys
6. Create a new route variant
7. Create a new bus
8. Alter a garage allocation of a bus
9. Link a journey to garage
10. List all unlinked journeys
11. Alter a bus number
12. Display running board
13. Exit
Output:- A series of time-table analysis can be produced for any time-table or crew schedule

Duty list:- Shows the details of each duty in the time-table which includes the service time, paid time, and split duties.

Scheduling list:- Which lists each of the running boards in the timetable in bus number order and their departure time from the terminus.

Employees or Garage :- It will give all details about the crew pay particulars/ Garage information and so on.

4.7 Transit Management Behaviour Model

Transit management usually has relatively little information on the demand curve which faces for its services. It has information on the actual flow on its various routes. Information is available on the actual origin and destination pattern of traffic or in the other demand characteristics such as trip length distribution. It will be assumed that transit management has no control over the running time of buses from one end of the route to other. This time being determined by the transit vehicles acceleration and speed capabilities and the prevailing speed of traffic on the roads comprising the route. In this context, the only decision variable open to management is to plan how frequently to operate buses. First, buses would be operated with a frequency at least equal to the frequency considered acceptable for the transit service.

\[ f \geq F = \frac{1}{H} \]

where

\[ f = \text{frequency of bus departures in one direction, buses per hour} \]

\[ F = \text{maximum acceptable frequency} \]

\[ H = \text{maximum acceptable headway hour per bus} \]

since the volume of passenger could exceed that which can be accommodated in buses operated at the minimum frequency, the frequency also has to be greater than or equal to that which is required to accommodate the passenger flow during any particular period
\[ f \geq \frac{p}{q} \]

where \( p \) = passenger flow point peak load point on route, passenger per hour

\( q \) = capacity of bus

Assuming that management operates the minimum number of buses in order to meet these two service related criteria, the frequency of the buses are

\[ f = \max (F, \frac{p}{q}) \] that minimises the cost of operating the service, since the number of buses, operators and vehicle miles would be minimised for the given route conditions.

A further complication is that the rate of passenger flow may vary within any period. This would lead to non uniform headway if more than the minimum frequency service is required. If management policy permitted non uniform head-ways and if a constant headway is required, the headway would be adjusted for the peak passenger flow with in each schedule period. This would lead to an average load, less than the vehicle capacity and could be incorporated in the model by appropriate selection of the value of \( q \). It also should be noted that the number of bus trips made over the entire day must be an integer, thereby possibly requiring a slight adjustment in the frequency of operation in each period. Similarly variations in the passenger flow during any one period will be ignored.

One important aspect of the supply of service would be the travel time from Origin to Destination for any particular traveller including the waiting time as well as one vehicle time. Assuming uniform or random passenger arrivals at the origin stop and the constant head-ways the average waiting time would be one-half the head way

\[ w = (1/2)h = 60/(2f) \]

where \( h \) = bus headway minutes per bus
As the passenger traffic increases above that amount required to fill the minimum frequency of buses, travellers waiting time would decrease. Assume that the volume of traffic is sufficient to fill all the vehicles at the minimum frequency the average number of passenger past the peak load point will be independent of the traffic volume. Assuming the Origin-Destination pattern of traffic does not vary with volume, the same number of passengers will board each vehicle regardless of volume. Therefore it may be assumed that the number of stops and the dwell time is independent of volume and hence the travel time will be independent of volume. In this case travel time between any points i and j equals

which may not be made. Such an adjustment will not be considered \( t_{ij} = v_{ij} + \frac{1}{2} h \) where

\( v_{ij} = \) vehicle running time between stops i and j plus one-half the dwell time required for alighting at j minutes

\( t_{ij} = \) total travel time between stops i and j minutes

Two points regarding the \( v_{ij} \) term should be made. First in the case where the passenger traffic is less than sufficient to fill the minimum number of vehicle trips operated. (the minimum frequency), then the travel time presumably will be slightly less, reflecting a diminished value of \( v_{ij} \) due to fewer stops and dwell time for loading and unloading. Of course this assumes a timetable adjustment by the management which may not be made. Such an adjustment will not be considered further. The position of \( v_{ij} \) due to the time required for unloading at stop j is also likely to be very small. The perceived average travel time for any traveller between stop i and j will be

\[ t_{ij} = v_{ij} + w \frac{60}{(2f)} + 30 \frac{w}{\max (F,p/q)} \]

where

\( t_{ij} = \) total perceived travel time between i and j

\( w = \) relative weight of waiting to on board vehicle time

\( v_{ij} = \) vehicle running time from i to j
The Concept of Mobility Accessibility and Land use

Mobility means the possibility to move from one place to another constituting a freedom sought by all citizens, enabling them to maintain or expand choices in everyday life. It permits them to choose an employer or a work place not located in the vicinity of their home, to go to shopping where they want and where the prices are the best. The private car meets this concern perfectly well and this is the reason for its wide popularity.

Accessibility means ease of access, is a concern more closely related to the production and distribution of products or services. In trade and non trade sectors, players try to place their establishment in such a place to minimise transportation cost or to minimise the amount of time their clients spend travelling. At a time when the public transport was the main answer to the mobility demand of the people, the competition was responsible for the success of the town center for commercial and office establishments. More recently, the development of automobile mobility has re-oriented the search for accessibility by distributors of goods and services to more peripheral locations.

The combination of consumer’s mobility in private cars and the search for accessibility on the producers side therefore induced a process of de-localization of housing as well as the production and commercial functions which contributes to strengthening the dependence on cars for satisfying the need for mobility.

The effects of this behaviour of a greater search for mobility by the consumers and for a better accessibility by the producers combine to multiply themselves and lead to the congestion of road infrastructure.
4.8.1 URBAN MOBILITY AT DEAD LOCKS

Inhabitants of cities, particularly those in metropolitan cities, its surroundings and to a lesser extent, those of the rest of the country, feel that the transportation system which is at their disposal in order to travel around the city or to get the city imposes higher and higher restrictions on their mobility. These restrictions vary according to the case concerned.

Some have seen the time it took them to travel to work by car doubled in less than ten years due to the congestion. Moreover once they arrive close to their destination they cannot find a place to park their car.

Children of some people do not have effective public transport at their disposal to reach their school, and their parents are thus obliged to drive them to school and back. Finally people who do not have a car because of their age, a physical disability or insufficient revenues, which means those who are called "captive" users of public transport are more and more limited in their possibility to travel where they want and when they want because of their suppressor of services due to increased scarcity of public transport users.

THREATS TO PROSPERITY

The case of contracts and exchange of goods and services are the basis of the urban society prosperity. Any hindrance to these contracts and exchanges has a negative effect on this prosperity. At the same time the comparative advantages of housing in the cities are decreasing, which reinforces the tendency of better off inhabitants to look for a place to live in the surrounding area of the city where as the poorest inhabitants tend to accept being left in the centers. The rise in population also the cause of increase of private cars in road.

Restoring mobility by healing symptom of road traffic congestion, that is, trying to suppress the traffic bottlenecks, increasing the capacity of the main road networks and creating new parking lots is no longer realistic in the long term. Experience has shown that in any city where measures were taken to increase the fluidity
of traffic, the initial problem reappear after some years later in an even or more acute form. The two main dangers of the current trends are

1) The deterioration of the general accessibility of the city because this is an immediate danger for the source of its economic prosperity

2) The excess of automobile mobility in the city because the problem it creates for the land and environment is threat to the population.

With regard to these two dangers, the reactions that can be considered by the regional public authorities must be selective; these are

The selective improvement of accessibility in the public transport and the selective restriction of accessibility by private cars. The total geo-graphical area of the state is 38.85 lakh ha. The increase in population and the vehicle growth tend to grab the area utilised for agriculture. Land under non-agricultural use was 8.6% in 1998–1999 and has increased to 9.1% in 1999-2000. The following table shows that the land status and vehicle growth in Kerala

Land use pattern in Kerala (Area in ha.)

<table>
<thead>
<tr>
<th>Sl.no</th>
<th>classification</th>
<th>1998-99</th>
<th>1999-00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Actual</td>
<td>%</td>
</tr>
<tr>
<td>1</td>
<td>Total Area</td>
<td>3885497</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Forest</td>
<td>1081509</td>
<td>27.80</td>
</tr>
<tr>
<td>3</td>
<td>Non-agricultural uses</td>
<td>333822</td>
<td>8.6</td>
</tr>
<tr>
<td>4</td>
<td>Barren and un-cultivated</td>
<td>28341</td>
<td>0.70</td>
</tr>
<tr>
<td>5</td>
<td>Grazing land</td>
<td>682</td>
<td>0.02</td>
</tr>
<tr>
<td>6</td>
<td>Land under miscellaneous</td>
<td>20200</td>
<td>0.50</td>
</tr>
<tr>
<td>7</td>
<td>Cultivable waste</td>
<td>62710</td>
<td>1.60</td>
</tr>
<tr>
<td>8</td>
<td>Fallow other than current</td>
<td>31537</td>
<td>0.80</td>
</tr>
<tr>
<td>9</td>
<td>Current fallow</td>
<td>68022</td>
<td>1.80</td>
</tr>
<tr>
<td>10</td>
<td>Net area sown</td>
<td>2258674</td>
<td>58.10</td>
</tr>
<tr>
<td>11</td>
<td>Area sown more than once</td>
<td>657831</td>
<td>16.90</td>
</tr>
<tr>
<td>12</td>
<td>Total cropped area</td>
<td>2916505</td>
<td>75.10</td>
</tr>
<tr>
<td>13</td>
<td>Cropping intensities</td>
<td>129</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.8.1.1 (source:- Economic Review 2001- SIB, Pattom)
Category-wise growth of Motor vehicles in Kerala since 1980

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Goods vehicle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a Four Wheeler</td>
<td>20128</td>
<td>36699</td>
<td>51530</td>
<td>88180</td>
<td>135058</td>
<td>142168</td>
</tr>
<tr>
<td>b Three wheeler</td>
<td>993</td>
<td>4170</td>
<td>9576</td>
<td>12072</td>
<td>28385</td>
<td>31688</td>
</tr>
<tr>
<td>2 Buses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a Stage carriages</td>
<td>8705</td>
<td>12910</td>
<td>15056</td>
<td>19988</td>
<td>23537</td>
<td>25161</td>
</tr>
<tr>
<td>b Contract carriages</td>
<td>842</td>
<td>2324</td>
<td>5234</td>
<td>14874</td>
<td>35351</td>
<td>40520</td>
</tr>
<tr>
<td>3 Cars &amp; wagons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a Cars</td>
<td>54381</td>
<td>75731</td>
<td>116676</td>
<td>155150</td>
<td>257796</td>
<td>282996</td>
</tr>
<tr>
<td>b Station Wagens</td>
<td>196</td>
<td>507</td>
<td>849</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>c Taxi cars</td>
<td>17780</td>
<td>28189</td>
<td>37638</td>
<td>54681</td>
<td>71581</td>
<td>75628</td>
</tr>
<tr>
<td>d Jeeps</td>
<td>7023</td>
<td>12972</td>
<td>24351</td>
<td>37774</td>
<td>67497</td>
<td>69261</td>
</tr>
<tr>
<td>4 Three wheeler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a Auto-rickshaws</td>
<td>7397</td>
<td>24383</td>
<td>58165</td>
<td>103465</td>
<td>227895</td>
<td>248350</td>
</tr>
<tr>
<td>b Motorised cycle</td>
<td>38</td>
<td>34</td>
<td>62</td>
<td>77</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>c Rickshaws</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Two wheelers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a Motorised cycle</td>
<td>58</td>
<td>73</td>
<td>70</td>
<td>63</td>
<td>1124</td>
<td>1124</td>
</tr>
<tr>
<td>b Scooter/bike</td>
<td>50493</td>
<td>11629</td>
<td>248374</td>
<td>496873</td>
<td>1020797</td>
<td>1151735</td>
</tr>
<tr>
<td>6 Tractor Trailer</td>
<td>1864</td>
<td>2104</td>
<td>2661</td>
<td>3388</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Tractors</td>
<td>1892</td>
<td>3089</td>
<td>4115</td>
<td>5045</td>
<td>7782</td>
<td>8177</td>
</tr>
<tr>
<td>8 Tills</td>
<td>469</td>
<td>1118</td>
<td>1927</td>
<td>4626</td>
<td>4763</td>
<td>4763</td>
</tr>
<tr>
<td>9 Trailers</td>
<td>260</td>
<td>416</td>
<td>580</td>
<td>763</td>
<td>1506</td>
<td>1576</td>
</tr>
<tr>
<td>10 others</td>
<td>1735</td>
<td>2891</td>
<td>4190</td>
<td>8903</td>
<td>27107</td>
<td>28680</td>
</tr>
<tr>
<td>Total</td>
<td>174254</td>
<td>319259</td>
<td>581054</td>
<td>1005922</td>
<td>1910237</td>
<td>2111885</td>
</tr>
</tbody>
</table>

Table 4.8.1.2 (Source:- Economic Review –2001, SPB, Patlon)

From the tables [4.8.1.1 & 4.8.1.2] we can deduce that, to house the 2,111,885 vehicles needed an area of 42,237 ha.. The state has the road network of 1.141 lakh km. Further growth level poses a threat to grab the agricultural area. The agricultural area declined
year by year for inhabitation. All these parameters will affect prosperity. The following
suggestion have been made to overcome the land problem
1) Improve the accessibility of the periphery by developing suburban services of the
national railway.
2) Reduce parking possibilities on streets in the city centre.
3) Implement an effective system of restricting parking along roads to residents
   living in housing areas without garages.
4) Promote combined “Bicycle-public transport travel”
5) Restrictions for private cars in a limited level

4.8.2 Vehicle Routing and Scheduling problem on Networks

Designing vehicle routes is a problem which is often encountered. Often vehicle must
call at a certain number of nodes in the transportation network, or must go through
specifically determined by branches in the networks. Collecting garbage, mail, cleaning
streets, distributing news papers, scheduling plane crew and bus drivers for certain jobs
are daily problems encountered by traffic and transportation experts.

Depending on whether vehicles must go along certain branches or call at certain
nodes in the network, problems are differentiated into edge covering problems or node
covering problems respectively. These problems have been greatly studied in recent years
and one example “travelling sales man” (the best-known node covering problem) has
been the subject of hundreds of papers throughout the world.

In order to solve different variations of vehicle routing problems or crew
scheduling problems, diverse techniques are applied, including dynamic programming
and combinatorial programming (the branch and bound method). The heuristic
procedure is also used to solve many problems of this type. In the majority of cases, the
application of classical mathematical programming methods required a great deal of
computer work which rapidly increases with the increase in the number of nodes on the
transportation. For this reason many combinatorial problems are solved with heuristic
procedures.

Problems concerning vehicle routing, determining the optimal position for the
vehicle depot within transportation system crew planning belong to the class of so called
combinatorial problems, can be those dealing with sequences, assignments, choice
making or any combination of these problems.
For sequencing problems, there is usually a series of n elements whose objective functions reach an extreme value. This can be used to distribute n drivers onto n buses as well.

Classification of Vehicle routing and Scheduling on Transportation Network

Different versions of vehicle routing and scheduling problems on transportation network appear in all fields of transportation, depending on specific problem at hand. Well-organised vehicle routing or a well designed schedule can markedly contribute towards a decrease in transportation costs and increase the quality of transportation services.

Vehicle routing problems do not have time constraints as to when services in different nodes should start or finish contrary to this scheduling problems contain time fixed in advance within which service in each node must be completed.

In cases when a certain time interval is planned for performing services in each node, we usually speak of a combination vehicle routing and scheduling problem starting with specific characteristics which describe certain types of routing or scheduling problems.

1. Time to service in a specific node or on a specific branch.
   a) time to carry out service fixed in advance (scheduling problem).
   b) service in certain nodes must be carried out within a specific time interval (combined routing and scheduling problems)
   c) There are no specific demands regarding service in each node (vehicle routing problem).

2. Number of vehicle depots in the network
   a) there is only one depot in the network
   b) the network contains several depots

3. Size of vehicle fleet available
   a) the fleet contains only one vehicle
   b) the fleet contains several vehicles

4. Type of vehicles in the fleet
   a) all vehicles in the fleet are the same
   b) the fleet contains several vehicles
5. Nature of service demands
   a) deterministic demands appear in the network
   b) stochastic demands for service appear.

6. Location of service demands
   a) service demands appear in the networks' needs
   b) service demands appear in the networks' branches
   c) service demands appear in nodes and branches

7. Maximum allowed vehicle route length
   a) all vehicles in the fleet have the same maximum allowed route length
   b) some vehicles have different maximum allowed route length
   c) there are no constraints regarding the maximum allowed vehicle route length

8. Costs
   a) variable
   b) fixed

9. Operations carried out
   a) picking up
   b) delivering
   c) picking up and delivering

10. Objective functions on which optimisation is based
    a) minimising route costs
    b) minimising total fixed and variable costs
    c) minimising the number of vehicles needed to carry out transportation operation
Consider the graph $G(N,A)$ whose set of nodes $N$ can be divided into two subsets $S$ and $T$ so that $S \cup T = N$ and $S \cap T = \emptyset$. Decompose the acyclic oriented graph into chains. That is divide the set of nodes into sub set of nodes which do not have common element. The given graph can always be decomposed into number of chains each one made up of only one node. The figure 4.8.3.1 shows an acyclic oriented graph whose set of node contains nodes $x_1, x_2, x_3, \ldots, x_{20}$.

The decomposition of the graph is in figure 4.8.3.2. The graph is decomposed into chains. Three chains are made up of only one node. (chain $x_8$, chain $x_{17}$, chain $x_{19}$). An acyclic oriented graph can be decomposed into chains in several ways. It is clear that the larger the number the nodes included into the individual chains, the smaller the number of chains into which the graph is decomposed. The connection between determining the minimum number of vehicles needed to service of a given schedule on
the transportation network is determining the minimum number of chains into which an acyclic oriented graph can be decomposed. The fig [4.8.3.3] shows a space-time diagram with
14 trips to be carried out between cities A & B and cities B & C and Cities C & D and cities D & A. We can distribute the vehicles to carry out 14 trips in different ways. A vehicle can take trip 1 then trip 5 then trip 7 and finally trip 10.

In the network a branch is directed from node $x_i$ towards node $x_j$ only if trip $x_j$ can be made after trip $x_i$. $x_j$ can be made after trip $x_i$ if trip $x_i$ starts in the city where trip $x_i$ finishes and if the planned time trip $x_j$ is after the finishing time of the trip $x_i$. Since chains represent vehicle routes the minimum number of vehicles needed to service a given schedule on the transportation network equals the minimum number of chains into which the acyclic oriented graph can be decomposed with each node where the trips to be made.

Let us examine acyclic oriented graph $G(N, A)$. The number of chains into which the graph is decomposed with $|c|$. The chains are denoted respectively by $k = 1, 2, 3, \ldots |c|$. The number of nodes belonging to the $k^{th}$ chain is denoted $n_k$. The total number nodes in graph $G$ is denoted by $|N|$. Since every node belongs to only one chain.

We have $n_1 + n_2 + n_3 + \ldots + n_k = |N|$ and further

\[
|N| = \sum_{k=1}^{|c|} n_k = \sum_{k=1}^{|c|} n_k + (c - c)
\]

\[
|N| = \sum_{k=1}^{|c|} n_k - 1 + |c|
\]

The number of branches in every chain is 1 less than the number of nodes in the chain. Therefore if $n_k$ is the number of nodes in the chain $k$ then $n_k - 1$ is the number of branches in chain $k$. It is clear that $\sum_{k=1}^{|c|} n_k - 1$ is the number of branches belonging to the chains in to which the graph $G$ is decomposed. We denote this number with $|D|$. This means $|D| = \sum_{k=1}^{|c|} n_k - 1$ or $|N| = |D| + |c|$. Since the number of nodes $|N|$ of the graph $G$ is fixed we can minimise the number of chains $|c|$ into which the graph $G$ is decomposed by maximising the number of the branches $|D|$ which belong to the chains. We construct bipartite graph $G(S, T, A^*)$ which corresponds the graph $G(N, A)$ as shown in the figure [4.8.3.4].

94
The corresponding bipartite graph is given in the fig [4.8.3.5].

For example trip $x_5$ can be made after trip $x_4$ there is a branch in the corresponding bipartite graph $G(S,T;A')$ which joins node $s_4$ with $t_5$. We assume that the capacity of every branch in the bipartite graph $(s_i, t_j) \in A'$ equals 1. If branch $(s_i, t_j)$ from starting graph $G$ belongs to one of the chains into which the graph $G$ has been decomposed then we note that the flow with a value 1 goes through the corresponding branch on the bipartite graph. If branch $(s_i, t_j)$ is not part of any of graph $G''$ chains then we note that
there is no flow through the corresponding branch \((s_i, t_j)\) on the bipartite graph or that the flow value equals 0.

If the bipartite graph \(G(S, T, A^*)\) contains a flow along branch \((s_i, t_j)\) then branch \((x_i, x_j)\) of graph \(G(N, A)\) is part of a chain into which graph \(G\) has been decomposed. This means that the total number of branches belonging to graph \(G^*\) chains \(|D|\) equals the total number of flows going through the bipartite graph. By minimising \(|S|\) or maximising \(|D|\) we maximise the total flow through the bipartite graph \(G(S, T, A^*)\) keeping in mind that a flow with a maximum value of 1 can appear from every source \(s_i\) and a flow with a maximum value of 1 can arrive at every sink \(t_j\).

4.8.4 MULTI CRITERIA ANALYSIS

Many decision problems specially those arising in the infrastructure development of the transport sector today are complicated by the need to consider a range of uses, such on those relating to environment, quality of life sustainably of development, and by the participation of divergent interest groups. To reflect this majority of the transport infrastructure development problems has to deal with multiple objectives and methods which are designed to assist groups of decision makers. The evaluation process have to integrate the quantitative and qualitative aspects of transport infrastructure development. The general frame of the multi criteria analysis consists of the following steps.

Identification of

The policy maker(s)
Public elected officials
Private sector agencies rep.
Appointed government officials
Experts of financial institutions

The decision levels
Government (national)
Regional (local)
local (company level)

The time horizon of decision
Operative
Strategic
Political

The purpose of decisions
To find best solution

Resource allocation

Identification of the alternative courses of action
( variants for development )

Land use
community and neighbourhood for proximity to city centre
proportion of mixed land use
proportion of undeveloped land area.

Density of population
location of social institutions
location of neighbourhood boundaries

economic impacts
employment
income
business activity
residential activity
effects on property
regional and community plans
resource consumption

social impact
displacement of people
accessibility of facility and services
effects of terminals on neighbourhoods
special user groups

physical impacts
aesthetics and historic value
infrastructure

impact on the ecosystems
air quality (Ca, H2, No, sulphur oxides, particles)
noise
vibration
used land
public safety
dead
seriously injured
slightly injured
energy
assignment of value for each attribute to measure the performance of the alternatives on that attribute
determination of a weight for each attribute
taking a weighted average values assigned to that alternative
making a provisional decision
performing sensitivity analysis to see how robust the decision is changes in the figures supplied by the decision maker.
Over viewing the multi-criteria decision process it is also useful to describe a few basic definition and theoretical considerations.

In the analysis we implicitly make a number of assumptions about the decision makers preferences. These assumptions can be regarded as the axioms of the procedure, in that they represent a set of postulates which may be regarded as reasonable. If the decision maker accepts the axioms and if he or she is rational the he or she should accept the preference rankings. The generally considered axioms are

Decideability

: Ability to decide which of two options is to be preferred.
Transitivity means if a>b and b>c then a>c.
Summation if a>b and b>c then the strength of preference of a over c must be greater than the strength of a over b. Finite upper and lower bounds for value in assessing values we assume that the best option and the worst are not infinite.

In the multi-criteria evaluation model the decision making problem can be described as follows. There are n alternatives with m criteria. This type of decision situation contains ( one or ) more decision makers who are to evaluate and rank a finite number of alternatives with respect to a finite number of criteria
Let $A_1, A_2, \ldots$ denote the alternatives and $C_1, C_2, \ldots, C_m$ the criteria. Assume that the data related to the alternative are known. Let $a_{ij} \geq 0, \ C = 1, \ldots, m, \ j = 1, \ldots, n$ denote the value of $j^{th}$ alternative with respect to $i^{th}$ criterion.

Any assessment of transport infrastructure to be developed calls for a whole range of criteria. The multi model transport systems comprise a set of basic elements like:

- Infrastructure network (mode specific)
- Interface (stations, ports)
- Auxiliary (for operation and maintenance)
- Rolling stock, vehicles, fuels
- Human capital
- Information (information system and telematics including passenger information, booking, reservation, scheduling)
- Finance (availability revenue, subsidisation)

Given five different development alternatives $A_1, A_2, A_3, A_4, A_5$ evaluated according to four different criteria $C_1, C_2, C_3, C_4$. The weight of the criteria $i$ is $w_i$, where

$$\sum_{i=1}^{n} w_i = 1 \ \forall i$$

The scores for a five categories evaluation process are given in the tables 4.8.4.1 and 4.8.4.2.

### Table 4.8.4.1

<table>
<thead>
<tr>
<th></th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very good</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>Good</td>
<td>70</td>
<td>65</td>
<td>60</td>
<td>55</td>
</tr>
<tr>
<td>Medium</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Satisfactory</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
</tr>
<tr>
<td>Bad</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
</tr>
</tbody>
</table>

### Table 4.8.4.2

<table>
<thead>
<tr>
<th></th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>vg</td>
<td>90</td>
<td>m</td>
<td>50</td>
</tr>
<tr>
<td>$A_2$</td>
<td>m</td>
<td>50</td>
<td>v</td>
<td>80</td>
</tr>
<tr>
<td>$A_3$</td>
<td>s</td>
<td>30</td>
<td>g</td>
<td>65</td>
</tr>
<tr>
<td>$A_4$</td>
<td>g</td>
<td>70</td>
<td>s</td>
<td>35</td>
</tr>
<tr>
<td>$A_5$</td>
<td>b</td>
<td>10</td>
<td>b</td>
<td>30</td>
</tr>
</tbody>
</table>
Where \( \max | h_j - h_i | \) is the maximal difference of scores between the alternatives \( j \) and \( i \) considering all criteria and \( H = 90 - 10 = 80 \) = constant. Their values are given in the table 4.8.4.3

<table>
<thead>
<tr>
<th></th>
<th>A_1</th>
<th>A_2</th>
<th>A_3</th>
<th>A_4</th>
<th>A_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_1</td>
<td>C_{12}=60%</td>
<td>C_{13}=50%</td>
<td>C_{14}=100%</td>
<td>C_{15}=100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_{12}=37.5%</td>
<td>d_{13}=18.7%</td>
<td>d_{14}=0%</td>
<td>d_{15}=0%</td>
<td></td>
</tr>
<tr>
<td>A_2</td>
<td>C_{21}=40%</td>
<td>C_{23}=80%</td>
<td>C_{24}=40%</td>
<td>C_{25}=80%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_{21}=50%</td>
<td>d_{23}=37.5%</td>
<td>d_{24}=25%</td>
<td>d_{25}=25%</td>
<td></td>
</tr>
<tr>
<td>A_3</td>
<td>C_{31}=50%</td>
<td>C_{32}=20%</td>
<td>C_{34}=50%</td>
<td>C_{35}=90%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_{31}=75%</td>
<td>d_{32}=25%</td>
<td>d_{34}=50%</td>
<td>d_{35}=12.5%</td>
<td></td>
</tr>
<tr>
<td>A_4</td>
<td>C_{41}=10%</td>
<td>C_{42}=60%</td>
<td>C_{43}=50%</td>
<td>C_{45}=50%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_{41}=25%</td>
<td>d_{42}=54.2%</td>
<td>d_{43}=37.5%</td>
<td>d_{45}=18.7%</td>
<td></td>
</tr>
<tr>
<td>A_5</td>
<td>C_{51}=60%</td>
<td>C_{52}=20%</td>
<td>C_{53}=10%</td>
<td>C_{54}=60%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_{51}=100%</td>
<td>d_{52}=50%</td>
<td>d_{53}=25%</td>
<td>d_{54}=75%</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.8.4.3

The assortment graph for ranking the alternatives can be calculated starting from 100\% value of preference (P) and 0\% value of disqualification (Q)

\[ P=100\% / C_{14}=C_{15}=100\% \quad Q=0\% / d_{14}=d_{15}=0\% \]

At this level the order of the alternatives can be identified between \( A_1-A_4 \) and \( A_1 \mapsto A_5 \)

Decreasing the level of preference to the next discrete value of \( C_{ij} (P=90\%) \) and increasing the level of disqualification to the next discrete level of \( d_{ij} \) (Q=12.5\%) gives

\[ P=90\% / C_{35}=90\% \quad Q=12.5\% / d_{35}=12.5\% \]

And the order of alternatives can be identified between \( A_3-A_15 \). The next level of \( P \ & \ Q \) can be chosen like below

\[ P=80\% / C_{33}=80\% \quad Q=37.5\% / d_{33}=37.5\% \]

The order of the alternatives can be identified between \( A_3-A_5 \). To stop the ranking procedure at preference level of 100\% and disqualification level of 40\%
The order of alternatives can be identified between $A_1$-$A_2$. The final assortment graph is given in fig 4.8.4.1. The graph means that only $A_1$ and $A_4$ alternatives are included in the final rank with the order, first is $A_1$, the second is $A_4$. Others cannot be ranked at this level of preference and disqualification.

### 4.8.5 MATHEMATICAL MODEL

The weight $W_2 := 0$ be assigned to $k$th decision maker to the $c$th criterion by $A_1, A_2, \ldots, A_n$ the $n$ alternatives by $C_1, C_2, \ldots, C_m$ the $m$ criteria and by $D_1, D_2, \ldots, D_l$, the $l$ group members (i.e. decision makers). The procedure then includes the following steps.

The value $a_{ij}$ given by the $k$th decision maker $D_k$ for alternative $A_i$ the criteria $C_j$ is determined. The normalized linear combination is calculated at each simple sub tree $N'$

$$\mu^k_{ij} = \frac{\sum_{i \in N'} W_i^k a_{ij}}{\sum_{i \in N'} W_i^k}$$

j = 1, \ldots, n, k = 1, \ldots, l

Proceeding on the tree towards the roots weight on the higher level criteria are combined with values obtained from one level below.

To find individual score by the $K$th decision maker $D_k$ for $A_j$ will be the value assigned to the root and the alternative will be ranked in descending order.

Group ranking can be considered, let denote by $V(w)_{C_i}^k$ the voting power assigned to $D_k$ for his or her weighing on any criterion $C_i$ and by $V(q)_{C_i}^k$ the voting powers assign to $D_k$ for his or her criteria $C_i = 1, \ldots, n, k = 1, \ldots, l$. For calculating the group utility from the alternative $A_j$ the preference weight will be aggregated into group weights $W_i$ at each criteria by

$$\sum_{k=1}^{l} V(W)_{C_i}^k W_i^k$$
The group qualification \( Q_{ij} \) at each leaf criterion \( C_{ij} \) for each alternative \( A_j \) is given by

\[
Q_{ij} = \frac{\sum_{k=1}^{m} V(W)w^c}{\sum_{k=1}^{m} V(W)c^k} \quad i \in N, j = 1, \ldots, n
\]

### 4.8.5 Rail traffic control

Within rail traffic system (inter city, high speed rail networks), the real time control problems are essentially related to surveillance and safety issues. On the other hand for off line planning and scheduling problems (allocation of locomotives to trains, crew scheduling, time plan) combinatorial optimisation problems are mostly involved in the optimal utilisation of available infrastructure like bus crew scheduling.

An important safety related task within rail traffic systems is collision avoidance (on a line or at node). To this end, traditional measures that are based on robust Electro-mechanical devices implementing simple but efficient logical (boolean) functions are quite broadly utilised. Within modern systems the implementation of electronic devices (micro computers) with high redundancy architectures (to satisfy high reliability requirements) become increasingly common.

For surveillance of rail traffic from a central operation room, advanced telematic devices (radio transmission, satellite communications) tools are increasingly employed. Major surveillance task include

- Monitoring of the movement of each train in the network
- Verification of the proper functioning with respect to the time schedule
- Intervention in case of severe disturbances so as to re-normalise the traffic

The task of re-normalisation of traffic is fairly complex and largely manually executed as yet. Involved real time sub tasks include

- Prediction of the duration of an occurred incident that blocks a line.
- Routing of affected trains in the network, if necessary
- Suitable time schedule modification to address current abnormal situation.

The main goal of these actions is the traffic normalisation, i.e., the quick and smooth return to the initial time schedule. Automatic control and artificial intelligence methods may be adopted for a partial automation.
4.8 Conclusion:

The program first generates all the trips specified in the headway files and sorts them by terminal point and time. Arrival at terminal point are linked to subsequent departure from the same point, if the idle time between these two events does not exceed the time given in the run file for maximum durable lay over. Not all the arrivals and departures can be linked in this way; those remaining after this stage are examined to see whether buses can be moved between terminals to form links.

Preference is given to moving a bus to another terminal in the same point group to make a link. If this can not be done the program finds an unmatched arrival and an unmatched departure and looks for a route which either extends the arrival journey forward to the departure point or extends the departure journey backwards to the arrival point.

Failing this the program searches for a route which contains both the arrival and departure points. If such a route exists and there is a sufficient time then a new journey in service is inserted by the program. There may be several departures which could be linked in one of these ways to a particular arrival. If this is the case a departure is chosen according to the above order of priority, giving preference to an earlier departure at the same level of priority. The linking of an arrival and a departure by this set of rules does not take place and if there is a time for the bus to return to the garage and remains there for the specified minimum break period.

Following the matching of arrivals and departures, blocks of journeys have been formed. If there is more than one garage each block is examined to see whether it starts and finishes near the same garage. The blocks are linked to that garage which minimise the total running time from the garage to the start of the block, and from the end of the block to the garage. The garage linking journey are inserted live or dead according to the instructions in the run file. The schedule is now complete and the time tables have been created together with running boards, the crew relief time for each bus, and the other output documents requested by the user.