Optimal generation of fast transit corridors in a city

Aman Sharma* and Raman Parti
Department of Civil Engineering, National Institute of Technology, Hamirpur 177 005, India

This study proposes a design methodology to generate optimal fast transit corridors in a city (Lucknow, Uttar Pradesh, India) integrated with a GIS platform. Population density distribution throughout the city was used for identification of nodes. Origin–destination (OD) distance matrix was generated between the nodes using Open Route Service. Centrality model consisting of connectivity and global integration centrality was used to generate an O–D demand matrix. Pre-defined number of clusters was generated to determine terminals using clustering algorithms. The optimal number of clusters was selected with an objective function to minimize the ‘total commuter time’ of the network. Ant colony optimization algorithm was used to generate fast transit corridors between the selected terminals that led to the generation of five such corridors for the study area.

Keywords. Ant colony optimization, bus terminals, clustering, fast transit corridors.

PUBLIC transportation is considered as the backbone for social and economic structure of a city. The rapid growth in population and continuous expansion of cities lead to additional commuter demand requiring new bus routes. The already existing bus transit network, usually designed according to destination-oriented approach in the cities, needs a redesign in view of the demand for new bus routes. This leads to an inefficient and complex bus transit network, as introduction of additional routes necessitates their overlapping in certain corridors due to high demand. This affects the load factor of buses on certain routes. The problem can be alleviated using direction-oriented bus transit network similar to a hub and spoke system, where fast corridors can be utilized for catering to demand between terminals of considerable high demand. The nodes with lesser demand can be catered within an influence area of terminal already connected through a fast corridor to another terminal. Introducing direction-oriented fast corridors would lead to a bus transit network that is easy to manage, handle and operate for large cities. In this study, we develop a methodology for designing routes as the potential fast corridors in a direction-oriented bus transit network for Lucknow, Uttar Pradesh, a fast-growing city in India. The design methodology proposed includes generation of origin–destination (OD) demand matrix based on centrality model, selection of optimal number of terminals using clustering algorithms and generation of fast corridors using an optimization technique.

Literature review

Transit routing is considered as a non-deterministic polynomial time (NP) hard problem and near optimum solutions can be obtained using metaheuristics methods. Newell discussed some of the issues related to the design of minimum-cost bus routes serving a multiple origin–destination trip distribution. Jayasinghe et al. suggested a design methodology using network centrality values to generate a model for urban areas of developing nations to predict attraction values of any node in the transit network. Fan and Machemehl used genetic algorithm to solve a bus transit network design problem with variable transit demand. Yang et al. presented a parallel ant colony algorithm to optimize the bus transit network of Dalian City, China, which resulted in significant reduction in travel times. Kechagiopoulos and Beligiannis developed a particle swarm optimization-based algorithm to provide an efficient solution for the urban transit routing problem and compared it on the basis of Mandl’s benchmark problem. They concluded that the proposed algorithm is comparable to existing techniques. Buba and Lee proposed a differential evolution algorithm for solving the urban transit routing problem with the objective function of minimizing average travel time of all served passengers. Daganzo reported that the bus rapid transit (BRT) had equivalent performance with metro, except in the case of a large city with low demand; in some cases it can even outperform the Metro if the city has enough corridors to run the BRT. A review of the literature reveals that various studies have used clustering techniques to determine the locations of hubs in a hub and spoke network. O’Kelly used a clustering approach to locate hubs in a planar hub problem, where he clustered a system of cities to locate a given set of hubs. Huang et al. used clustering to select hubs from a rail network for a multimodal transit network design. Parti et al. selected an optimal number of hubs for New Delhi; a large city in India using self-organizing maps clustering algorithm for selecting the optimal number of hubs for the planning of...
a bus transit network. A study of the literature clearly indicates that the centrality model can be used to determine nodes, clustering algorithms can be helpful in identification of terminal selection and soft computing optimization techniques can invariably be utilized in the generation of fast corridors.

**Study methodology**

A study methodology has been evolved using clustering techniques and Ant Colony Optimization (ACO) algorithm to generate fast corridors for public transit network in a developing city (Figure 1). Population density distribution throughout the city is used for identification of nodes, which are then given a unique ID number. It is assumed that all the transit demand from an area would come from its nearest node. The demand generated by any node is estimated and used to create the O–D demand matrix. The O–D distance matrix is generated using the Open Route Service (ORS) matrixV2 application programming interface (API). Every cell of the distance matrix represents the distance between the row and column numbered node of that cell. The maximum size of this matrix can be $56 \times 56$ due to limitation of the API. In network analysis, centrality value of a node or vertices defines the importance of that particular node in the network. Jayasinghe *et al.*\(^4\) derived the relation of trip attraction of a node in an urban area of a developing nation with centrality values of that node. Further demand between all node pairs is calculated using the gravity model. O–D demand matrix is generated using the gravity model of trip distribution. The trip attraction values were obtained using network centrality values, namely connectivity and global integration centrality. The connectivity centrality is a measure of direct connections with the given node and is represented as $C_C^i = k$, where $C_C^i$ is the connectivity centrality of node $i$ and $k$ is the number of direct connections to node $i$. The global integration centrality is a measure of how far a given node is from all other nodes in the graph\(^13\) and is given by $C_G^i = (\sum d_{ik})^{-1}$, where $C_G^i$ is the global integration centrality of node $i$ and $d_{ik}$ is the shortest distance between nodes $i$ and $k$. The model generated by Jayasinghe *et al.*\(^4\) to estimate trip attraction values in terms of connectivity and global

![Diagram of Study Methodology](image-url)
integration centrality has been used to determine the trip density as follows

\[ \exp(TAD) = 18.33 + 1.87 \exp(C_{Gi}) + 0.98 \exp (C^C_i) \]

where TAD is the trip attraction density.

The O–D demand matrix of the city is generated using the trip attraction data obtained from eq. (1) in the gravity model of trip distribution.

Selection of an optimum number of terminals

The nodes are divided into a predefined number of clusters using clustering technique. There are two types of clustering algorithms – soft clustering and hard clustering. An algorithm from both types was selected to determine which type of clustering would lead to better results, namely fuzzy C-means and hierarchical clustering algorithms\(^{14}\). Fuzzy C-means is a soft-type clustering algorithm which assigns a probability value to a node, which indicates the strength of belongingness of that node to a cluster. The probability value is assigned on the basis of the distance between the centre of the cluster and the particular node. A membership vector is developed which assigns a probability (0–1) that means how close a node is to the mean of a cluster, i.e., cluster centre. The total sum of membership of any node should be 1. In this soft clustering algorithm, a node can belong to multiple clusters depending on its membership values\(^ {15–17}\). Hierarchical clustering is a hard-type clustering algorithm, and is a flexible and deterministic method for clustering data objects. Hierarchical methods can be classified into agglomerative and divisive. Agglomerative methods begin by taking single-data objects to the bottom level and continue to merge two clusters at a time to build a bottom-up hierarchy of clusters. On the other hand, divisive methods generate a top-down hierarchy of clusters. This study uses the agglomerative or bottom-up technique. In this, the bottom represents all data to be clustered. This diagram of clusters is also called a dendrogram. Every level in the dendrogram represents a set of clusters. The nearest cluster sets are merged at each level and then the dissimilarity matrix is updated accordingly. This process is continued until the last group is obtained which contains all data objects in a single cluster\(^ {18–20}\).

Both the clustering algorithms are used for the generation of clusters and the stop within each cluster with highest demand is designated as its terminal. The optimal number of clusters is obtained with an objective of minimizing the ‘total commuter travel time’ of the entire network. For any trip, the origin and destination nodes for any commuter may lie within the same cluster or in different clusters of the transit network. If both origin and destination nodes lie within the same cluster, then the commuter can travel from origin to destination directly. If the origin and destination nodes lie in different clusters, then the commuter has to pass through the terminals of the origin and destination nodes to reach the destination (Figure 2). In a transit network, the travel time between any node pair \((i, j)\) has the following components

(i) Intra-terminal in-vehicle travel time.
(ii) Inter-terminal in-vehicle travel time.
(iii) Waiting time at the node(s).
(iv) Waiting time at the terminal(s).

Intra-terminal in-vehicle time is the shortest possible travel time from nodes \(i\) and \(j\) to their respective terminals. Inter-terminal in-vehicle time is the shortest possible travel time from the terminal of node \(i\) to that of node \(j\). The waiting time at various nodes and terminals will depend on the operating characteristics of the transit system. Selection of the optimum number of clusters is done with an objective to minimize the ‘total commuter time’ for the overall transit network of the city. The objective function can be represented as follows

\[ \sum_{i=1}^{n} \sum_{j=1}^{n} \text{dem}_{ij} (t_{ih_i} + t_{h_ij} + t_{h_j} + t_{wh_i} + t_{wh_j}) \]

where \(\text{dem}_{ij}\) is the demand between node pairs \((i, j)\), \(n\) the total number of nodes, \(t_{ih_i}, t_{h_j}\) the intra-terminal in-vehicle time, \(t_{h_ij}\) the inter-terminal in-vehicle time, and \(t_{wh_i}, t_{wh_j}\) are the waiting time at nodes and terminals respectively.

Sensitivity analysis can be done by varying these parameters to obtain the ‘total commuter time’ for the entire transit network.

Generation of an optimal number of fast corridors

Fast corridors were generated using the ACO algorithm that is based on the seeking behaviour of ants, as proposed by Dorigo and Di Caro\(^ {21}\). It imitates the behaviour of real ants to solve optimization problems. Ants use a
probability rule to select a solution to a problem, called tour. The probability of choosing a path depends on the amount of pheromone present and quality of the edge\(^7,22–25\). The probability is given as

\[
P_{i,j} = \frac{\tau_i^\alpha \eta_j^\beta}{\sum_k \tau_k^\alpha \eta_k^\beta},
\]

where \(P_{i,j}\) is the probability of choosing node \(j\) from node \(i\).

In eq. (3), \(\tau_{i,j}\) is the pheromone of edge \(i, j\) and is given by

\[
\tau_{i,j} = (1 - \rho)\tau_{i,j} + \sum_{k=1}^{n} \tau_{k,j},
\]

where \(\rho\) is the evaporation rate [0–1] and \(n\) the total number of nodes.

In eq. (3), \(\eta_{i,j}\) is the quality of the edge \(i, j\) and is given by

\[
\eta_{i,j} = 1/d_{i,j},
\]

where \(d_{i,j}\) is the length of edge \(i, j\).

In eq. (3), \(\alpha\) is the pheromone influence and \(\beta\) the heuristic influence.

Routes are generated between the terminals of different clusters as fast transit corridors with the following constraints:

(i) The individual route must have sufficient length.
(ii) No backtracking of the routes.
(iii) Overlapping distance of any two routes should be within certain limits.

**Study area – Lucknow**

Lucknow the capital city of Uttar Pradesh, a densely populated old city with an old and destination-oriented bus transit network was selected for the study. A total of 56 nodes were selected throughout the city to develop the routes. The nodal coverage was according to the population distribution of the city. Data for the existing bus transit system were obtained from the Lucknow Metropolitan Transport Service and population data of the city were obtained from the Population Census of India 2011 (source: http://www.censusindia.gov.in/), which was then forecasted for year 2021 using the arithmetic progression method. The demand generated from each node was estimated using the existing bus service data and population estimate.

The O–D distance matrix between all the selected nodes was generated using open route service matrixV2 API, resulting in a matrix of dimension 56 × 56. The time of travel between nodes was calculated assuming an average bus speed of 30 kmph.

The O–D demand matrix for all the 56 nodes was generated using the model of Jayasinghe *et al.*\(^4\). All the 56 nodes were divided into a predefined number of clusters (4–8) using two different clustering techniques, namely hierarchical clustering and fuzzy C-means clustering. Figure 3 shows the dendrogram generated by agglomerative hierarchical clustering. Figure 4 shows the clusters generated using fuzzy C-means algorithm. In Figure 4, squares show the cluster centre, dots show the nodes in the network, and the individual colours of nodes show the cluster to which they belong.

As the number of terminals increases, the influence area of all terminals will decrease, resulting in decrease in intra-terminal distances. This also leads to covering of the same geographical areas by more terminals by bringing them closer to each other; hence the inter-terminal distance will also decrease. However, the increase in the number of terminals will increase the number of transfers required for travel and result in an increase in waiting time. To take these effects into account, sensitivity analysis by varying waiting time at the origin/transfer nodes and at the terminals was done and the number of clusters

![Figure 3. Agglomerative hierarchical clusters dendrogram (X-axis shows node numbers; Y-axis is a measure of closeness of clusters).](image)

![Figure 4. Fuzzy C-means clusters.](image)
Table 1. Total network commuter time (million seconds)

<table>
<thead>
<tr>
<th>Waiting time (sec)</th>
<th>Terminal 300</th>
<th>300</th>
<th>600</th>
<th>900</th>
<th>600</th>
<th>900</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer node</td>
<td>No. of terminals</td>
<td>600</td>
<td>900</td>
<td>600</td>
<td>900</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Method-I: Hierarchical clustering technique</td>
<td>325.7164</td>
<td>368.9851</td>
<td>347.3508</td>
<td>390.6195</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>324.3200*</td>
<td>368.4671*</td>
<td>346.3935*</td>
<td>390.5406*</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>326.5925</td>
<td>370.0246</td>
<td>348.6977</td>
<td>392.1274</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>329.4258</td>
<td>373.6945</td>
<td>351.5602</td>
<td>395.8288</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Method-II: Fuzzy C-means clustering technique</td>
<td>347.5031</td>
<td>395.9453</td>
<td>371.7242</td>
<td>420.1664</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>337.5694*</td>
<td>386.0116*</td>
<td>361.7905*</td>
<td>410.2327*</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>393.0837</td>
<td>441.5259</td>
<td>417.3048</td>
<td>465.7470</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>351.1629</td>
<td>399.6051</td>
<td>375.3840</td>
<td>423.8262</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>358.5461</td>
<td>406.9883</td>
<td>382.7672</td>
<td>431.2094</td>
<td></td>
</tr>
</tbody>
</table>

*Optimum values.

with minimum total commuter time was used for further analysis. The following waiting times are used for sensitivity analysis:

- Waiting time at origin/transfer nodes: As the frequency of operation of buses at origin/transfer nodes will be comparatively low, trials were made for 600 and 900 sec.
- Waiting time at terminals: The frequency of operation of buses at terminals will be relatively high, hence trials were made for 300 and 600 sec.

Table 1 shows the total network commuter time in million seconds using hierarchical and fuzzy C-means clustering algorithm. An assessment of both of the algorithms shows that the minimum commuter time for the whole network is for the number of clusters = 5, which is considered to be the optimum number of clusters for the entire network.

Figure 5 a and b shows the variation of total network commuter time for hierarchical and fuzzy C-means clustering algorithms with respect to changing number of terminals respectively. From the figures it can be observed that the fuzzy C-means technique has more variation in total network commuter time for different number of terminals compared to the hierarchical method, but the overall optimum value for total commuter time is obtained through the hierarchical method of clustering. Hence, the hierarchical clustering technique is a better alternative to determine the terminals for the network. The nodes within these five clusters with maximum demand were designated as terminals (Table 2) and routes generated between these terminals using ACO algorithm with the number of ants as 50 and iterations/ant as 200, using $\alpha = 1$, $\beta = 3$ and $\rho = 0.8$. The number of ants and iterations were chosen to ensure a reasonable running time of the program. The values of $\alpha$ and $\beta$ were selected based on previous studies. Pheromone update rate was kept at 0.8 to avoid convergence of results early on.

Five corridors were generated between the selected terminals of different clusters with the constraints that they must have a minimum length of 10 km and
maximum length of 30 km, with no backtracking and maximum overlapping distance between any two transit routes to be within 50%. A fast corridor will also allow heavy vehicles to pass through it. This condition was satisfied using ‘driving-hgv’ constraint in the ORS API. Table 3 shows the list of corridors generated between the selected terminals along with the nodes through which the corridors will pass. ArcGIS from ESRI was used to integrate the input data in the form of shape files of the study area (Lucknow) using Python’s library (OSMnx), which uses OpenStreetMap’s API28. Figure 6 displays the network with all fast transit corridors generated on a GIS platform. The dots in the map represent the nodes in the network and the numbered circles represent the terminals selected for the network. The labels on terminals represent the node IDs.

Table 2. Optimal number of terminals

<table>
<thead>
<tr>
<th>Terminal</th>
<th>Node ID</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>Sharda Nagar</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>Lalkuanwa</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>Kharika</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>Kakori</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>Engineering College</td>
</tr>
</tbody>
</table>

Table 3. Fast corridors generated

<table>
<thead>
<tr>
<th>Route ID</th>
<th>Length (km)</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22.80</td>
<td>25,22,6,5,4,3,40,2</td>
</tr>
<tr>
<td>2</td>
<td>25.20</td>
<td>2,50,55,31</td>
</tr>
<tr>
<td>3</td>
<td>13.51</td>
<td>9,26,8,48,47,11</td>
</tr>
<tr>
<td>4</td>
<td>12.10</td>
<td>9,8,12,54,31</td>
</tr>
<tr>
<td>5</td>
<td>26.24</td>
<td>25,52,53,20,11</td>
</tr>
</tbody>
</table>

Figure 6. Fast corridors generated.

Results and discussion

In this study, a design methodology to generate optimal fast transit corridors in a city has been developed that is then integrated with a GIS platform. Population density distribution throughout the city was used for identification of nodes. The O–D distance matrix was generated between the nodes using ORS. Centrality model consisting of connectivity and global integration centrality was used to generate the O–D demand matrix as input parameters for the design methodology. Pre-defined number of clusters was generated for the nodes using fuzzy C-means and hierarchical clustering algorithms. The hierarchical clustering was found to be superior, indicating that hard assignment clustering would lead to better solutions. The optimal number of clusters was selected with an objective function to minimize the total commuter time of the network, which resulted in a total of five nodes selected as terminals. ACO algorithm was used to generate fast transit corridors between the selected terminals using few constraints that led to the generation of five fast transit corridors for the study area of Lucknow, which were then displayed on a GIS platform.

Conclusion and limitations of work

This study helped develop a transit network with potential corridors for fast transit using metaheuristics. The OpenRouteService matrix and distance APIs can be used to create distance matrix and plot routes respectively, with few restrictions. During the selection of optimum number of terminals, it was observed that the hard clustering method outperformed soft clustering in determining the optimum, and hence was a better alternative. The routes between the selected terminals can be generated using ACO algorithm with some constraints. The study will help policy makers in identifying corridors which can be used in a rapid transit system network. The designed network could help in further determining the scheduling and fleet size of a bus service that can utilized.

Limitations

In this study, due to resource restrictions, comparison between the existing service and the proposed network was not done. The network consisted of total 56 nodes due to the restriction of the API used. Only one clustering algorithm from each assignment type was compared. It also currently lacks benchmarking with the literature, and the parameters for ACO were selected randomly within the desirable range without comparison using various values and their effect on the solution.

Future scope

Detailed spatial and temporal data can be collected in future so that comparison between the existing service
and the proposed network can be done. More APIs and methods could be used to further increase the network size. More clustering techniques can be involved in the terminal selection process to further justify the selection of hard or soft clustering algorithms. The methodology can be benchmarked by the literature and model networks to check the relative performance and results. More variations in constraints could be done to achieve alternate optimum solutions which can be compared further. Optimum-problem specific ACO parameters can also be determined instead of acceptable values within the optimum range.


Received 16 March 2020; revised accepted 25 February 2021
doi: 10.18852/cs/v120/i9/1500-1506