Application of A Route Expansion Algorithm for Transit Routes Design in Grid Networks

Iran Khanzad\textsuperscript{1}, Amirali Zarrinmehr\textsuperscript{2}, Seyedehsan Seyedabrisghami\textsuperscript{3}, Mahmoud Saffarzadeh\textsuperscript{4}

\textit{Received: 01. 02. 2016  Accepted: 17.05.2016}

Abstract

Establishing a network of transit routes with satisfactory demand coverage is one of the main goals of transit agencies in moving towards a sustainable urban development. A primary concern in obtaining such a network is reducing operational costs. This paper deals with the problem of minimizing construction costs in a grid transportation network while satisfying a certain level of demand coverage. An algorithm is proposed following the general idea of “constructive algorithms” in related literature. The proposed algorithm, in an iterative approach, selects an origin-destination with maximum demand, generates a basic shortest-path route, and attempts to improve it through a route expansion process. The paper reports the scenarios and further details of the algorithm considered for expanding a transit route in a grid network. A random 6×10 grid network is applied to report the results. The results support that application of the proposed algorithm notably reduces the operational costs for various amounts of demand coverage.

\textbf{Keywords:} Transit routes, grid transportation network, demand coverage, operational costs

Corresponding author E-mail: saffar_m@modares.ac.ir

1. MSc Grad., Department of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran
2. Ph.D Candidate, Department of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran
3. Assistant Professor, Department of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran
4. Professor, Department of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran
1. Introduction and Background

In many modern cities, urban transportation is confronted with serious challenges such as traffic congestion, air pollution, and parking shortages. These problems have been recognized as the impacts of excessive use of private modes of transportation. To resolve such issues and move towards a sustainable transportation, it is essential to design and implement an affordable public transportation system, namely transit system, offering high-quality services to the network users [Guihaire and Hao, 2008; Kepaptsoglou and Karlaftis, 2009].

Designing a complete transit system poses a very large problem which is often divided in smaller tractable sub-problems to solve [Cancela, Mauttone, and Urquhart, 2015]. Consequently, decision making in transit planning has been categorized in different levels from long-term decisions on urban infrastructures to short-term ones such as vehicle/crew scheduling [Guihaire and Hao, 2008]. In one of the most fundamental levels of decision making, Transit Routes Design (TRD) is concerned with determining an optimal set of transit routes in a public transportation system, e.g. bus or rail networks. Since almost all decisions in transit planning are conditioned to the configuration of routes, TRD plays a significant role in the process of generating an efficient transit system [Ceder and Wilson, 1986].

TRD is a hard problem to solve [Schöbel, 2012]. At least five sources of complexity for this problem have been mentioned in the study of Baaj and Mahmassani [1991] as follows: (1) Difficulty in defining decision variables and objective function; (2) Non-linearities and non-convexities exhibited by formulations; (3) Combinatorial explosion of the problem; (4) Multi-objective nature of the problem; and (5) Characterizing a “good” spatial layout for the routes. Due to such difficulties arising in TRD problem, most of the studies have adopted heuristic or meta-heuristic solution approaches to tackle the problem. A categorization of these solution approaches can be found in review papers such as Kepaptsoglou and Karlaftis [2009] or Farahani et al [2013].

Various objective functions have been applied so far to address the problem of TRD. The difference in objective functions mainly stems from the different levels of decisions in transit planning. Most review papers, such as Farahani et al [2013] or Ibarra-Rojas et al [2015], generally categorize these decisions in three levels: (1) Strategic decisions, e.g. designing or expanding transit routes; (2) Tactical decisions, e.g. allocating exclusive bus lanes or service frequency determination; and (3) Operational decisions, e.g. scheduling problems. Since TRD falls into the category of strategic planning problems, it is natural to ignore many unnecessary details and take into account only the main modeling aspects of the problem.

Two important features of TRD problem at strategic level are transit operational costs and users’ benefits. To model these features of the problem, many authors have used the total duration (i.e. length/travel-time) of transit routes and direct coverage, respectively. The total duration of transit routes is a representative of the operational costs in a sense that construction costs and transit fleet size are directly proportional to it [Mauttone and Urquhart, 2009]. Also, direct coverage stands for the percentage of the travel demand which can be satisfied with no transfers between transit routes and is a representative of users’ benefit. This is supported by the observation that as soon as the number of transfers increase the transit users switch to private modes of transportation [Stern, 1996; Guihaire and Hao, 2008].

Generating transit routes in TRD often consists of building shortest-paths and then improving them by adding further network links [Mandl, 1980; Kepaptsoglou and Karlaftis, 2009]. Among various studies over TRD problem in which a certain level of demand coverage is ensured and the total duration of routes is minimized, there are only a few algorithms that build the routes set from scratch [Mauttone and Urquhart, 2009]. These algorithms, also known as “constructive” algorithms, have been the topic of research in Baaj and Mahmassani [1995] and Mauttone and Urquhart [2009].
The general idea of constructive algorithms in TRD is to generate basic shortest-path routes between node-pairs and improve the coverage of routes by marginally modifying their layout. In the study of Baaj and Mahmassani [1995] a route generation process is proposed in which node-pairs with maximum demand values are iteratively selected and a transit route is generated between the selected nodes. To cover more transit trips, the authors take the advantage of the network nodes that are close to the generated routes. To this end, they propose a “node insertion” procedure in which transit nodes at distance of 1 link from generated routes are examined to improve the routes layout. Mauttone and Urquhart [2009] also proposed a new constructive algorithm based on the inter-zonal nature of demand in the network. Instead of inserting a single node in a transit route, they proposed to insert a pairs of nodes (in form of a path) into the set of routes. This approach was reported to significantly reduce the total duration of transit routes in the problem.

The above-mentioned constructive algorithms have been suggested as general-purpose approaches in the literature. Designing constructive algorithms and investigating their performances over special networks, to the best of our knowledge, have not been reported. Different attributes are recognized in the literature for transportation networks with special structures, such as grid or radial networks [Badia, Estrada and Robusté, 2014; Snellen, Borgers, and Timmermans, 2002]. As Figure 1 illustrates, the outcome of node insertion in a grid network may be different from a general transportation network. It can be observed from this figure that, for example, after insertion of the node $n$ in the transit route, the route directness (between nodes $i$ and $j$) and duration in the grid network may turn to be more degraded than the general network.

In this study, we present a detailed description of a constructive algorithm and corresponding results for grid transportation networks. The general scheme of the algorithm is based on the idea of node insertion in constructive algorithm [Baaj and Mahmassani, 1995] whereas the details are designed for a grid network. The paper investigates whether or not, and to what extent, the proposed constructive algorithm can improve the layout of shortest-paths in a grid network. In the following sections, the problem is described in section 2 and the solution algorithm is presented in section 3. The functionality of the algorithm and numerical results are discussed in sections 4 and 5 respectively. The paper is finally conclude in section 5.

2. Description of the Problem

The problem of Transit Routes Design (TRD) deals with determining the set of transit routes in an urban transportation network. The set of routes must meet the requirements of both network users and operators [Mauttone and Urquhart, 2009]. To model the benefits of network users, many authors have used the demand coverage offered by transit routes (see for example Zhao and Zeng [2006] or Farahani et al [2013]). The demand coverage is defined in this paper as the percentage of commuters that can make their trips by transit routes with no transfers. Also, the total length of transit routes has been widely applied in the literature as an index for the operating costs. The two mentioned measures, namely demand coverage and total length of transit routes, are applied in this paper to address the requirements of network users and operators, respectively.

There are two main approaches in TRD literature dealing with demand coverage and operational costs. Some studies consider the operational costs within a constraint of the problem and attempt to maximize the coverage [Kermanshahi, Shafahi, and Bagherian, 2015], while others take the coverage as a constraint and minimize the operational costs [Mauttone and Urquhart, 2009]. The latter approach is adopted in this paper to solve the TRD problem. We assume that a grid-type network of highways and the corresponding matrix of travel demand are available. The demand matrix and also link travel-times are assumed to be fixed in this study. It is intended to minimize the operational costs while a minimum demand coverage must be satisfied.
To describe the problem in terms of a mathematical model, let us define:

**GS**: the global set of all feasible transit routes between origin-destinations in the entire network;

**R**: the set of selected transit routes from GS;

**r**: a transit route from **R**;

**t(r)**: the travel time (i.e. length) associated with the transit route **r** (minutes);

**Cov(R)**: demand coverage for the set of routes **R** (%);

**Cov\(_{\text{min}}\)**: the minimum demand coverage to be satisfied by selected routes (%).

Based on these definitions, a global formulation for TRD problem may be written as follows:

\[
\begin{align*}
\text{Min} & \quad \sum_{r \in R} t(r) \\
R & \subseteq GS \\
\text{Cov}(R) & \geq \text{Cov}_{\text{min}}
\end{align*}
\]  

(1) \hspace{1cm} (2) \hspace{1cm} (3)

In the above formulation, the objective function (1) aims at minimizing the total travel time of transit routes, namely **R**. Constraint (2) states that the set of selected routes is a subset of the global set of all feasible routes in the network and constraint (3) ensures that demand coverage will not be lower than a predefined value i.e. the minimum coverage.

In the formulation (1)-(3), the exponential number of alternative sets of transit routes, **R**, renders the problem computationally intractable. This fact may be better observed through a simple analysis in Figure 2.

Figure 2 indicates that the number of alternative subsets, **R**, to be selected in a transportation network with **p** pairs of origin-destinations and **q** feasible routes on average for each pair is in the order of \(2^{pq}\). For example, considering a transportation network with 10 nodes, which may be looked upon as a toy example in comparison with real-world instances, the number of origin-destinations is \(p=10\times9=90\).
Given that only $q=2$ feasible candidate routes exist on average for each origin-destination, there will be the number of $|GS|=90\times2=180$ routes to be selected at the entire network. Now, the total number of subsets that may be selected from these feasible routes is clearly of the order $2^{180}$ which is an extremely-large number. Such an order is clearly a hindrance to the exact solution of the problem (1)-(3). Application of non-exact (e.g. heuristic) solutions algorithms, as a result, is inevitable to tackle this problem.

3. The Proposed Algorithm

This section presents a heuristic algorithm to solve the TRD problem. The general idea of the algorithm is based on the route generation algorithm proposed by Baaj and Mahmassani [1995]. However, the focus is put on designing a detailed algorithm for a grid transportation network as one of the regular structures used in modern cities [Badia, Estrada and Robusté, 2014]. Further, since there are many details arising at the stage of implementing the algorithm, some most important details of the algorithm will be reported.

The flowchart of Figure 3 sketches the main components of the proposed algorithm. The notations required beforehand are as follows: $N$: the set of all nodes in the network; $P$: the set of all origin-destinations in the network, $P = N \times N$; $R$: the set of all transit routes generated as the algorithm proceeds; $r$: a candidate transit route; $t_{ij}^*$: the shortest-path travel-time between nodes $i$ and $j$ (minutes), $i,j \in N$; $t_{ij}(r)$: the travel-time between $i$ and $j$ within route $r$ (minutes), $i,j \in N$, $r \in R$; $\omega$: an upper bound for the ratio of $t_{ij}(r)/t_{ij}^*$, $i,j \in N$, $r \in R$; $\text{dis}(n,r)$: the minimum number of links intervening between node $n$ and route $r$, $n \in N$, $r \in R$; $\text{Adj}(r)$: the set of nodes, such as $n$, for which $\text{dis}(n,r)=1$, $r \in R$; $\text{Cov}(n,r)$: the total amount of travel demand between node $n$ and all nodes in route $r$ (passengers per hour), $n \in N$, $r \in R$; $S(r,n)$: the set of all candidate expansions of route $r$ covering node $n$, $n \in N$, $r \in R$; $S^a(r,n)$: the set of all acceptable candidate expansions of route $r$ covering node $n$, $S^a(r,n) \subseteq S(r,n)$, $n \in N$, $r \in R$.
Application of A Route Expansion Algorithm for Transit Routes Design in Grid Networks

Common($r$, $R$): the number of common links between a new transit route $r$ and previously generated routes $R$;

$\lambda$: a maximum value for Common($n$, $r$);

The algorithm, as shown in Figure 3, builds the set of routes $R$ from scratch in an iterative approach. Following the study of Baaj and Mahmassani [1995], in each iteration, the algorithm generates a shortest-path route to cover a new origin-destination with maximum demand. Then, in a route expansion loop which is designed for grid networks, the algorithm attempts to further cover the most promising nodes that are close to the generated route. The process of route generation finishes after the minimum coverage constraint is satisfied during the iterations.

In what follows, the main components of the algorithm are discussed in more details.

**Initial Settings:** The algorithm starts with an empty set of transit routes and initially sets the corresponding coverage to zero.

**Minimum Coverage Condition:** This condition checks whether the set of routes built so far satisfies the minimum demand coverage or not. If the minimum demand was covered, the algorithm is terminated by printing the routes’ information in the output. Otherwise, an iteration is started to generate a new route.

**Building a Basic Transit Route:** A basic transit route is built by picking up the origin-destination with maximum demand and generating the associated shortest-path. The set of nodes that are adjacent to the route $r$, namely Adj($r$), is also generated. These nodes serve as candidate nodes to be covered along with the route $r$.

**Route Expansion Loop:** The loop of route expansion is a main part of the algorithm in which the transit route $r$ is iteratively expanded and updated until the list of candidate nodes to be covered becomes empty. In this loop, first, a candidate node from Adj($r$) is selected. Then, alternative route expansions are built to cover the selected node, and finally, the route is updated.

**Finding a Candidate Node:** In a greedy approach, the algorithm selects the node $n$ with maximum value of Cov($n$, $r$) in each iteration of route expansion loop. It is obvious that covering such node is more likely to increase demand coverage.

To define alternative expansions of route $r$ covering the node $n$, i.e. set $S(r, n)$, let denote the node adjacent to $n$ in route $r$ by $n_i$. Moreover, let $n_{i-1}$ and $n_{i-2}$ be the two nodes of route $r$ lying just before node $n_i$ and also $n_{i+1}$ and $n_{i+2}$ be the two nodes just after that, as depicted in Figure 4 (a). Based on these definitions, the algorithm generates alternative sub-routes between nodes $\{n_{i-2}, n_{i-1}, n_i\}$ and $\{n_i, n_{i+1}, n_{i+2}\}$ to cover the node $n$.

To make sure that the sub-routes generated in the route expansion loop will offer new transit configurations, first, the existing transit sub-routes are removed from the network. Then, shortest-paths including node $n$ are used to generate alternative sub-routes between pairs of $(n_{i-1}, n_i)$, $(n_{i-2}, n_i)$, $(n_i, n_{i+1})$, $(n_i, n_{i+2})$, $(n_{i-1}, n_{i+1})$, $(n_{i-2}, n_{i+2})$, $(n_{i-2}, n_{i+1})$, $(n_{i-1}, n_{i+2})$. Figure 4 shows a simple topology of a transit route in a grid network and alternative route expansions therein.

Two constraints are also considered for alternative route expansions, which are defined as route acceptability conditions in the flowchart of Figure 3. The first condition ensures that an alternative route expansion, temporarily named as $r_{temp}$, does not violate the minimum or maximum lengths as predefined standards of transit routes. The second condition states that the travel-time between each two nodes of $r_{temp}$ must be not more than $\omega$-times the shortest travel-time between those nodes. The latter condition ensures a certain level of route directness between any pairs of transit nodes in a transit route.

**Update the Route and Adjacent Nodes:** Among alternative route expansion (including the first route itself), it is desirable to select the candidate route with maximum demand coverage and minimum length. Since these two goals may be conflicting with one another, the measure $\text{Cov}(r_{temp})\text{V}(r_{temp})$ is used in the algorithm to select the final route expansion.
Iran Khanzad, Amirali Zarrinmehr, Seyedehsan Seyedabrishami, Mahmoud Saafrazadeh

Figure 3. The proposed constructive algorithm
Application of A Route Expansion Algorithm for Transit Routes Design in Grid Networks

Therefore, the best route expansion will be determined by the maximum $\text{Cov}(r_{\text{temp}})/t(r_{\text{temp}})$ ratio for all alternative routes.

After adopting the expanded route, the value of $\text{dis}(n,r)$ may turn greater than 1 for some nodes of $\text{Adj}(r)$. It is worth to mention that these nodes are removed from $\text{Adj}(r)$ after each update. This prevents the algorithm from putting much effort.
on nodes that are far from the current underlying transit route.

**Final Acceptability Condition:** To disperse the routes configuration in the network, each generated route $r$ is supposed to have no more than $\lambda$ common links with the set of previously established routes. This is shown as a final acceptability condition after the route expansion loop. If the newly generated route satisfies this constraint, it will be added to the routes set $R$. Otherwise, a new iteration is started by the algorithm.

**Adding the New Route to the Routes Set:** After expanding and finalizing the new transit route, $r$, the routes set is updated by including $r$. Also, all node-pairs lying within the route $r$ are removed from $P$ as covered origin-destinations. The coverage of the routes set is finally updated before a new iteration for routes generation is started.

### 4. Illustrative Example

To explore the functionality of the algorithm, a single iteration of the algorithm over an illustrative example is discussed in this section. A grid network with 12 nodes (3×4 nodes) is considered as depicted in Figure 5. The values written next to the links are considered as their associated travel times in minutes.

Table 1 also illustrates the values of the travel demand from node $i$ to node $j$, namely $D_{ij}$ ($1 \leq i,j \leq 12$). Since the functionality of the algorithm is based on undirected travel demand between nodes, namely $D(i,j) = D_{ij} + D_{ji}$, we simply consider a symmetric pattern for demand matrix in Table 1. It is assumed that design parameters are $T_{min}=15$ (minutes), $T_{max}=25$ (minutes), $\omega=2$, and $\lambda=4$.

The algorithm starts with $R=\emptyset$ at the stage of initial settings. Since demand coverage initially equals to zero, the algorithm goes to the step of building a basic transit route. At this step, a pair of nodes with maximum demand, $D(i,j)$, must be selected. There are three alternatives for this selection: $(1,11)$, $(6,7)$, and $(7,11)$. Let assume that the algorithm picks up the first origin-destination which is $(1,11)$. Then, a shortest-path is built between 1 and 11 as $r = \{1-5-6-10-11\}$. This shortest-path (as shown in Figure 6(a)) serves as a basic transit route prior to the route expansion loop. Further, the set of vertices adjacent to route $r$ is built as $\text{Adj}(r) = \{2,7,9,12\}$.

![Figure 5. The 3×4 grid network used as illustrative example](image)
Application of A Route Expansion Algorithm for Transit Routes Design in Grid Networks

Table 1. The demand matrix for the illustrative example (in passengers per hour)

<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>0</td>
<td>1000</td>
<td>1450</td>
<td>0</td>
<td>700</td>
<td>600</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>1500</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>0</td>
<td>1200</td>
<td>850</td>
<td>0</td>
<td>900</td>
<td>700</td>
<td>250</td>
<td>150</td>
<td>100</td>
<td>1000</td>
<td>1300</td>
</tr>
<tr>
<td>3</td>
<td>1450</td>
<td>1200</td>
<td>0</td>
<td>750</td>
<td>600</td>
<td>0</td>
<td>400</td>
<td>0</td>
<td>0</td>
<td>1210</td>
<td>1300</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>850</td>
<td>750</td>
<td>0</td>
<td>900</td>
<td>100</td>
<td>100</td>
<td>550</td>
<td>700</td>
<td>0</td>
<td>1200</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>700</td>
<td>0</td>
<td>600</td>
<td>900</td>
<td>0</td>
<td>650</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>1000</td>
<td>700</td>
<td>650</td>
</tr>
<tr>
<td>6</td>
<td>600</td>
<td>900</td>
<td>0</td>
<td>100</td>
<td>650</td>
<td>0</td>
<td>1500</td>
<td>0</td>
<td>1350</td>
<td>0</td>
<td>1200</td>
<td>150</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>700</td>
<td>400</td>
<td>100</td>
<td>0</td>
<td>1500</td>
<td>0</td>
<td>500</td>
<td>730</td>
<td>690</td>
<td>1500</td>
<td>600</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>250</td>
<td>0</td>
<td>550</td>
<td>0</td>
<td>1350</td>
<td>500</td>
<td>0</td>
<td>1490</td>
<td>1470</td>
<td>1150</td>
<td>500</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>150</td>
<td>0</td>
<td>700</td>
<td>400</td>
<td>0</td>
<td>730</td>
<td>1490</td>
<td>0</td>
<td>990</td>
<td>800</td>
<td>1200</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>100</td>
<td>1210</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>690</td>
<td>1470</td>
<td>990</td>
<td>0</td>
<td>0</td>
<td>700</td>
</tr>
<tr>
<td>11</td>
<td>1500</td>
<td>1000</td>
<td>1300</td>
<td>1200</td>
<td>700</td>
<td>1200</td>
<td>1500</td>
<td>1150</td>
<td>800</td>
<td>0</td>
<td>0</td>
<td>650</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>1300</td>
<td>0</td>
<td>0</td>
<td>650</td>
<td>150</td>
<td>600</td>
<td>500</td>
<td>1200</td>
<td>700</td>
<td>650</td>
<td>0</td>
</tr>
</tbody>
</table>

In the route expansion loop, since the set of \( \text{Adj}(r) \) is not empty, the algorithm starts by finding the node \( n \) with maximum \( \text{Cov}(n,r) \). The values of \( \text{Cov}(n,r) \) are 60000, 73800, 45800, and 43000 (passengers per hour) for nodes 2, 7, 9, and 12, respectively. As a result, the node \( n=7 \) is picked up as the candidate node to be inserted into the route \( r \), considering either node 6 or 11 as the adjacent node to \( n=7 \), there are totally 4 different choices (including the basic route \( r \)) for inserting the node 7 into \( r=\{1-5-6-10-11\} \). These choices are listed in Table 2 as well as the corresponding calculations to select the expanded route. As illustrated in Table 2, the maximum value of \( \frac{\text{Cov}(r_{\text{exp}})}{t(r_{\text{exp}})} \) ratio corresponds to the route expansion \( r_{\text{exp}}=\{1-5-6-7-11\} \). Therefore, the route \( r=\{1-5-6-10-11\} \) between 1 and 11 is updated to \( \{1-5-6-7-11\} \) as shown in Figure 6 (b).

The set of \( \text{Adj}(r) \) is also updated by removing node 7. However, the remaining nodes of \( \text{Adj}(r) \) i.e. 2, 9, and 12 will still stay at 1-link distance from the updated route. These nodes will be kept in \( \text{Adj}(r) \) for the next iteration of route expansion loop.

Table 2. Alternative choices and calculations for expanding the route \( r=\{1-5-6-10-11\} \) in the illustrative example

<table>
<thead>
<tr>
<th>Alternative choices for the expanded route, ( r_{\text{temp}} )</th>
<th>( t(r_{\text{temp}}) ) (minutes)</th>
<th>Acceptability</th>
<th>( \text{Cov}(r_{\text{temp}}) ) (passengers per hour)</th>
<th>( \frac{\text{Cov}(r_{\text{temp}})}{t(r_{\text{temp}})} ) ('passengers per hour' per 'minute')</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Transit Route ( {1-5-6-10-11} )</td>
<td>15</td>
<td>✔️</td>
<td>12700</td>
<td>847</td>
</tr>
<tr>
<td>Inserting between nodes 6 and 11 ( {1-5-6-7-11} )</td>
<td>19</td>
<td>✔️</td>
<td>16700</td>
<td>879</td>
</tr>
<tr>
<td>Inserting between nodes 1 and 6 ( {1-2-3-7-6-10-11} )</td>
<td>31</td>
<td>✗</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Inserting between nodes 1 and 11 ( {1-2-6-7-11} )</td>
<td>23</td>
<td>✔️</td>
<td>19800</td>
<td>861</td>
</tr>
</tbody>
</table>

International Journal of Transportation Engineering, 188 Vol.4/ No.3/ Winter 2017
For brevity, the remaining iterations of route expansion loop are not reported here. However, it is worth mentioning that all other route expansion choices during the subsequent iterations will be either infeasible or worse than the expanded route \{1-5-6-7-11\}. Consequently, the route expansion loop terminates by the route \( r = \{1-5-6-7-11\} \) which leads to the value of \( 100 \times \text{Cov}(r)/(\text{total travel demand}) = 100 \times 16700/80360 = 20.8 \% \) as demand coverage. A notable share of the demand (i.e. more than one fifth of the total demand), therefore, gets covered at the first iteration of the algorithm.

5. Numerical Results

This section reports numerical results of the algorithm on a 60-nodes grid network. The general topology of nodes/links is shown in Figure 7. The network is used for illustrative purposes and is not a real one. Following Nie’s “VNET” software [Nie, 2016], random link travel-times and demand matrix are adopted for this grid network. The travel demand in the entire network is 333000 passengers per hour. Design parameters are considered to be \( T_{\min} = 15 \), \( T_{\max} = 50 \), \( \omega = 2 \), and \( \lambda = 12 \) for this section. Further details, however, are referred to Appendix A at the end of this paper.

Considering 10 (\%) as the minimum demand coverage, for example, the algorithm results in 5 transit routes which are illustrated in Table 3. The general configuration of these routes is also sketched in Figure 8. The objective function (i.e. total travel time) and demand coverage associated with these 5 routes are 154.2 (minutes) and 10.2 (\%), respectively.

It is interesting also to see how the idea of route expansion can help improving basic shortest-paths between origin-destinations. Two design scenarios are considered to do so. The first scenario involves creating shortest-path routes with no further expansions. All details of this scenario are the same as the algorithm in this paper unless the route expansion loop is omitted. The second scenario is also the algorithm with route expansion as discussed in this paper. Figure 9 shows the results of both scenarios for the 60-nodes grid network.

It can be observed from Figure 9 that, for all levels of demand coverage, the operational costs corresponding to expanded routes are less than those of shortest-path routes. Figure 10 shows the amount of improvements obtained after expansion of the routes, i.e. the vertical gap between the two curves in Figure 9. Note that the values of demand coverage for the curves of Figure 9 are not the same. Therefore, a linear interpolation has been applied in Figure 10 to calculate the gap between these curves.
Application of A Route Expansion Algorithm for Transit Routes Design in Grid Networks

Figure 7. The general topology of the 6×10 grid network

Table 3. The information of 5 routes generated by the algorithm to satisfy demand coverage of 10 (%)

<table>
<thead>
<tr>
<th>Generated Route, r</th>
<th>T(r) (minutes)</th>
<th>Covered Demand (Passengers per hour)</th>
<th>Cov(r) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1 {47-46-45-44-33-23-22-12-2-1}</td>
<td>39.4</td>
<td>11964</td>
<td>3.6</td>
</tr>
<tr>
<td>r2 {27-17-16-15-14-13-3}</td>
<td>23.2</td>
<td>4578</td>
<td>1.4</td>
</tr>
<tr>
<td>r3 {54-44-34-24-23-13-3-4}</td>
<td>28.1</td>
<td>5454</td>
<td>1.6</td>
</tr>
<tr>
<td>r4 {12-22-23-24-25-26-16-6}</td>
<td>29.3</td>
<td>4020</td>
<td>1.2</td>
</tr>
<tr>
<td>r5 {54-44-34-35-25-15-16-17-18-8}</td>
<td>34.3</td>
<td>7914</td>
<td>2.4</td>
</tr>
</tbody>
</table>

Figure 8. The general configuration of transit routes generated to satisfy demand coverage of 10 (%)
The results of the proposed algorithm are in accordance with the previous studies, e.g. Baaj and Mahmassani [1995], in a sense that the objective function increases more sharply as the level of demand coverage is raised. The reason behind this observation is that a main part of the demand matrix gets covered in the first iterations of the algorithm when origin-destinations with the heaviest demand are selected. As a result, covering further origin-destinations will become more costly as the algorithm proceeds.
It is also worthy to note that the gap between the two curves, as depicted in Figure 10, has an overall increasing trend. In other words, applying the idea of expanded routes saves more operational costs in high levels of the demand coverage than it did in low levels. This observation can be justified by the fact that, as the number of routes increases, there is more room for the route expansion algorithm to cover further high-demand nodes along with the routes. However, a more general judgment on this observation would require deeper studies accounting for the demand pattern, length of routes, etc.

6. Concluding Remarks

This paper presented a constructive algorithm with detailed descriptions for designing transit routes in a grid network. Alternative route expansion scenarios were considered to improve the route layout. The results over a 60-nodes random grid network supported that the algorithm is capable of improving (i.e. reducing) the total duration of transit routes in different levels of direct demand coverage.

We applied a grid network with random attributes, in this paper, to report the results. However, to obtain more realistic results and make more general judgments, a research direction is to apply the algorithm on several real transportation networks with grid structure. Design and implementation of constructive algorithms for other special networks, e.g. radial networks, and reporting the results in comparison with this study would further extend the findings of this paper. Also, developing constructive algorithms based on the concepts other than node insertion (e.g. “pair-node insertion” in the study of Mauttone and Urquhart [2009]) and comparing the results with this study may be another interesting topic for future research.

7. Acknowledgement

The authors would like to thank the anonymous referees of the International Journal of Transportation Engineering (IJTE) for many insightful and constructive comments that enhanced the presentation of this paper.

8. References


Appendix A. Travel time and demand information for the 6×10 grid network

Given that T(i-j) is the travel time of link i-j in the network, the travel times (in minutes) associated with 104 links of the 6×10 grid network are as follows:

T(1-3)=3.2, T(1-12)=4.5,
T(2-3)=3.2, T(2-5)=4.6, T(2-14)=5.8,
T(3-5)=4.6, T(3-7)=3.8, T(3-16)=3.7,
T(4-7)=3.8, T(4-9)=4.5, T(4-18)=5.1,
T(5-9)=4.5, T(5-11)=4.0, T(5-20)=4.8,
T(6-11)=4.0, T(6-13)=4.5, T(6-22)=4.3,
T(7-13)=4.5, T(7-15)=5.5, T(7-24)=3.5,
T(8-15)=5.5, T(8-17)=4.1, T(8-26)=3.7,
T(9-17)=4.1, T(9-19)=4.9, T(9-28)=5.1,
T(10-19)=4.9, T(10-30)=6.0,
T(11-12)=4.5, T(11-23)=4.7, T(11-32)=3.8,
T(12-14)=5.8, T(12-23)=4.7, T(12-25)=5.3, T(12-34)=3.4,
T(13-16)=3.7, T(13-25)=5.3, T(13-27)=3.6, T(13-36)=4.5,
T(14-18)=5.1, T(14-27)=3.6, T(14-29)=3.8, T(14-38)=5.7,
T(15-20)=4.8, T(15-29)=3.8, T(15-31)=3.2, T(15-40)=3.5,
T(16-22)=4.3, T(16-31)=3.2, T(16-33)=4.1, T(16-42)=5.1,
T(17-24)=3.5, T(17-33)=4.1, T(17-35)=3.1, T(17-44)=4.8,
T(18-26)=3.7, T(18-35)=3.1, T(18-37)=5.4, T(18-46)=4.6,
T(19-28)=5.1, T(19-37)=5.4, T(19-39)=4.4, T(19-48)=3.1,
T(20-30)=6.0, T(20-39)=4.4, T(20-50)=3.2,
T(21-32)=3.8, T(21-43)=5.4, T(21-52)=5.6,
T(22-34)=3.4, T(22-43)=5.4, T(22-45)=3.1, T(22-54)=5.7,
T(23-36)=4.5, T(23-45)=3.1, T(23-47)=4.1, T(23-56)=3.3,
T(24-38)=5.7, T(24-47)=4.1, T(24-49)=6.0, T(24-58)=4.3,
T(25-40)=3.5, T(25-49)=6.0, T(25-51)=3.3, T(25-60)=5.5,
T(26-42)=5.1, T(26-51)=3.3, T(26-53)=5.0,
T(27-44)=4.8, T(27-53)=5.0, T(27-55)=3.3,
T(28-46)=4.6, T(28-55)=3.3, T(28-57)=5.3,
T(29-48)=3.1, T(29-57)=5.3, T(29-59)=3.3,
T(30-50)=3.2, T(30-59)=3.3, T(31-52)=5.6,
T(32-54)=5.7,
T(33-56)=3.3,
T(34-58)=4.3,
and T(35-60)=5.5.
Also, Table A illustrates travel demand values (in passengers per hour) between all origin-destinations of the network.
Table A. Travel demand matrix for the 6×10 grid network

<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>177</td>
<td>39</td>
<td>138</td>
<td>18</td>
<td>96</td>
<td>21</td>
<td>90</td>
<td>102</td>
<td>84</td>
<td>167</td>
</tr>
<tr>
<td>2</td>
<td>177</td>
<td>0</td>
<td>90</td>
<td>108</td>
<td>60</td>
<td>126</td>
<td>153</td>
<td>27</td>
<td>105</td>
<td>9</td>
<td>63</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
<td>90</td>
<td>0</td>
<td>177</td>
<td>24</td>
<td>102</td>
<td>72</td>
<td>54</td>
<td>24</td>
<td>144</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>177</td>
<td>24</td>
<td>0</td>
<td>126</td>
<td>153</td>
<td>60</td>
<td>84</td>
<td>18</td>
<td>22</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>180</td>
<td>60</td>
<td>24</td>
<td>153</td>
<td>0</td>
<td>24</td>
<td>156</td>
<td>141</td>
<td>150</td>
<td>11</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>96</td>
<td>126</td>
<td>150</td>
<td>126</td>
<td>24</td>
<td>0</td>
<td>111</td>
<td>19</td>
<td>34</td>
<td>180</td>
<td>126</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td>153</td>
<td>72</td>
<td>166</td>
<td>111</td>
<td>1</td>
<td>0</td>
<td>156</td>
<td>27</td>
<td>72</td>
<td>54</td>
</tr>
<tr>
<td>8</td>
<td>90</td>
<td>27</td>
<td>54</td>
<td>144</td>
<td>141</td>
<td>9</td>
<td>156</td>
<td>0</td>
<td>19</td>
<td>34</td>
<td>180</td>
</tr>
<tr>
<td>9</td>
<td>102</td>
<td>105</td>
<td>24</td>
<td>96</td>
<td>150</td>
<td>117</td>
<td>27</td>
<td>99</td>
<td>0</td>
<td>87</td>
<td>94</td>
</tr>
<tr>
<td>10</td>
<td>84</td>
<td>9</td>
<td>144</td>
<td>177</td>
<td>118</td>
<td>72</td>
<td>129</td>
<td>87</td>
<td>1</td>
<td>85</td>
<td>162</td>
</tr>
<tr>
<td>11</td>
<td>84</td>
<td>9</td>
<td>144</td>
<td>177</td>
<td>118</td>
<td>72</td>
<td>129</td>
<td>87</td>
<td>1</td>
<td>85</td>
<td>162</td>
</tr>
</tbody>
</table>

195 International Journal of Transportation Engineering,
Vol.4/No.3/Winter 2017
Application of A Route Expansion Algorithm for Transit Routes Design in Grid Networks

Table A. Travel demand matrix for the 6×10 grid network (continued)