Modelling Turns in Transit Network Design

Antonio Mauttone · Paula Riganti

Abstract We study the influence of turns over solutions of the Transit Network Design Problem (TNDP) in the context of bus-based systems. Turns affect both user and operator cost, by increasing travel time. We model explicitly the street network and we compute turning costs at a given node by considering the length of the external arc defined by the three nodes involved in the turn. This mechanism is embedded into the Pair Insertion Algorithm (PIA), which is applied to a real test case corresponding to a small city having 13 bus lines. The results show the relevance of modelling turns explicitly. Particularly, it is shown that hidden turning costs increase the real cost of solutions which are computed by ignoring turns. Also, we perform a comparison of PIA with state of the art optimization algorithms for the TNDP, to ensure that our conclusions are based on experiments conducted with a competitive algorithm. Finally, we apply the model to a scenario of a real city comprising more than 130 lines, to show that the computation of turning costs does not preclude the application of the optimization algorithm to larger instances.

Keywords: Transit network design · Turns · Heuristic · Large case
1 Introduction and motivation

Network design is at the core of transit system planning, due to its relevance in defining system performance (Ceder and Wilson, 1986). In classical bus systems, the transit network is determined by the routes defined over the street network. Bus based transit systems are relevant due to its flexibility, since transit routes are defined over the (usually dense) street network, and therefore they are not conditioned by the existence of specific infrastructure. Though high capacity transit systems (like underground, light rail and bus rapid transit) can achieve better performance in overall sense, classical bus systems are still relevant either for medium/small cities or for larger cities which require feeder services or connections between origins and destinations which are not reachable by other components of the system.

The Transit Network Design Problem (TNDP) has been studied since more than 50 years ago (Lampkin and Saalmans, 1967). Optimization models and algorithms have been proposed for several variants of the problem (Desaulniers and Hickman, 2007; Zanjirani et al., 2013). In general terms, the TNDP entails to find a set of routes over the street network in such a way that interests of both users and operators are fulfilled. To model properly the system performance from these different viewpoints, a central and common issue is the computation of vehicle travel time. This measure influences both the user cost (the time spent traveling on-board the vehicle) and the operator cost (which is directly proportional to vehicle operation and personnel cost).

Existing TNDP models usually consider simplified versions of the street network over which the routes are defined (Mandl, 1979; Mauttone and Urquhart, 2009). In such cases, the real street network is replaced by an abstract network which connects adjacent traffic zones of the study region. This is reasonable since it reduces the model size (and therefore the corresponding computational effort to handle it) and the level of detail (which may be suitable in strategic planning). However, by doing that, some relevant network characteristics may be lost, thus degrading the realism of the whole model. In this context, a very relevant aspect is the modelling of turns, i.e., a change in the direction of the vehicle, usually due to a change of street. Turns are present in many transit routes, either due to the original design or due to adjustments made to cover new nearby demand. Nevertheless, turns affect travel time in the overall sense, thus increasing the cost of both users and operators. Aggregated network models do not enable the treatment of turns explicitly, since they do not work with the real street network. In these cases, some models include parameters which control route directness. For example, Baaj and Mahmassani (1995) consider a circuity factor defined for each route as the ratio between the length of the route and the length of the shortest path over the street network between its endpoints. Though this measure does not account for turns explicitly, it is a reasonable proxy for identifying meandering routes. Direct routes are relevant mainly for trunk services, whereas for feeder routes, meandering patterns can be efficient. Therefore, in general terms we can say that turns are needed but they should be controlled.
The Pair Insertion Algorithm, PIA (Mauttone and Urquhart, 2009) makes intensive use of meandering patterns to generate transit routes which are efficient for both users and operators. Its main idea is to introduce deviations in existing routes to cover nearby demand, instead of creating new routes. This is somehow consistent with some criteria used in practice, where historical routes are adjusted to cover new demand for transit trips. However, the overhead in travel time due to deviations is not properly assessed in the original version of PIA since it uses an aggregated version of the underlying street network. The aim of this work is to study the influence of turns in the solutions obtained for the TNDP. Though we seek to obtain general conclusions about the subject, we study the problem by using a specific algorithm. We propose a new version of PIA which considers a detailed representation of the street network, thus enabling to compute turns explicitly.

To the best of our knowledge, there are not studies related to the TNDP which have addressed this research subject. Regarding existing literature, Zhao and Zeng (2008) model explicitly the street network of Miami-Dade, comprising 2,804 nodes and 4,300 arcs. Bagloee and Ceder (2011) build a detailed representation of the transit system of Chicago, comprising 13,487 nodes and 52,742 arcs. Both studies propose optimization algorithms to solve variants of the TNDP, however, none of them analyse the impact of turns over the solutions obtained.

In this work, we model explicitly the street network and we compute the cost of turns while building the transit routes. The turn cost is represented by the arc length of the angle defined by the corresponding three nodes in the network. This feature is embedded into the Pair Insertion Algorithm (Mauttone and Urquhart, 2009). After validating the model, we conduct a set of computational experiments aimed to show the influence of turns over the solutions (sets of routes) and the possible consequence of ignoring turns. We present both numerical and graphical (routes over the map) results. Although the primary goal of the study is to analyse the effect of turns over solutions of TNDP, we compare the results of our algorithm against state of the art results computed over benchmark instances in order to ensure that our conclusions are based on results produced by a competitive algorithm. Finally, we apply the new version of PIA to a large case, to ensure that the overhead introduced by the computation of turns does not preclude the application of the algorithm to such cases.

The article is structured as follows. After the introduction and motivation given in Sect. 1, we explain both network and optimization models for the TNDP in Sect. 2. The new version of PIA, including the computation of turns is presented in Sect. 3. The computational experiments and corresponding analysis are presented in Sect. 4, including the validation of the turning model, the analysis of the effects of turns, the comparison using benchmark instances and the application to a large case. Finally, Sect. 5 formulates conclusions and identifies future work.
2 Network and optimization models

In this section, we define the structure used to represent the street network, the routes defined over it and the trajectories of users over those routes.

2.1 Network model

We model the network of streets available for defining transit routes with a directed graph $G = (N, A)$. The set of nodes $N$ represents junctions and centroids. A junction is determined by the intersection of two streets whereas a centroid is an imaginary point which concentrates the demand leaving and entering to its corresponding zone. A partition of the study region into zones is assumed to be given. The set of arcs $A$ represents street sections and walk arcs. A street section connects two junction nodes whereas a walk arc is a fictitious walking path which connects a junction node with the centroid of a zone. We do not model explicitly the bus stops, therefore, we assume that every junction node is a potential stop, i.e. an access point to the route from the centroid. Both street and walk arcs are weighted by the travel time (bus and walking respectively), denoted as $c_a$ for $a \in A$. Moreover, an origin-destination matrix $D = \{d_{ij}\}$ is given, where each element $d_{ij}$ expresses the number of required trips from centroid $i$ to centroid $j$. The pair of nodes $(i, j)$ is known as OD-pair.

A transit line comprises both forward and backward routes which may be spatially separated, always converging at close nodes in their endpoints (Fig. 1a). A route is a sequence of junction nodes in $G$. For route $r$, let $N_r \subset N$ and $A_r \subset A$ be the sets of its nodes and arcs respectively, where $|N_r| = |A_r| + 1 = q$. The cost (travel time) of $r$ has two components, namely, arc cost $C_r$ and turn cost $C_u$. Thus, $C_r(r) = C_v(r) + C_u(r)$, where

$$C_v(r) = \sum_{a \in A_r} c_a \quad \text{(1)}$$

and

$$C_u(r) = \sum_{i=q-1}^{i=q} c_i \quad \text{(2)}$$

where $c_i$ is defined for node $i \in N_r$ as the time taken to perform the turn involving the sequence of nodes $i$, $i+1$, $i$ and $i+1$ of $r$.

For a given set of transit routes, the users corresponding to the demand given by origin-destination matrix $D$, define a network of trajectories. A trajectory is defined for an OD-pair as a directed path which connects its origin and destination nodes (Fig. 1b). The cost $C_r^D(ij)$ of the trajectory corresponding to OD-pair $(ij) \in D$ over a set of transit routes $R$ includes the walking time from the origin centroid and to the destination centroid, and the on-board travel time between the origin and destination stops, plus the cost of all turns of the route between these stops.
2.2 Optimization model

For a given street network and a given origin-destination matrix as defined in Sect. 2.1, we seek to find a set of routes $R$ which minimizes simultaneously the objectives of users and operators (Israeli and Ceder, 1995; Mauttone and Urquhart, 2009a). The cost of users is defined in terms of their trajectories, i.e.

$$UC(R) = \sum_{(i,j) \in D} C^*_R(i,j)$$

(3)

where $C^*_R(i,j)$ is the minimum cost trajectory of users traveling from centroid $i$ to centroid $j$ using routes of $R$. On the other hand, the cost of operators is defined as the total duration of routes, i.e.

$$OC(R) = \sum_{r \in R} C(r)$$

(4)

Moreover, unless a different specific hypothesis is stated, we assume that all trips given by $D$ must be satisfied directly, i.e., without transfers between routes.

![Diagram](image)

**Fig. 1** Underlying network model

### 3 Pair Insertion Algorithm considering turns

The Pair Insertion Algorithm, PIA (Mauttone, and Urquhart, 2009) is a greedy heuristic which proceeds by building shortest routes over $G$ between high demand OD-pairs and then inserting pairs of nodes taken from $D$ into existing routes. In this way, it seeks to minimize simultaneously the user total travel time $UC$ and the total route duration $OC$. The result is a set of routes $R$ defined over $G$. 
3.1 Working with directed networks

While the original version of PIA works with an undirected network (therefore, lines with identical forward and backward routes are assumed) the version introduced in this study builds routes over a directed street network. The pseudocode of the directed version of PIA (called DPIA) is shown in Fig. 2. The while loop executes while \( \text{totalDemand} = \sum_{(i,j) \in \mathcal{D}} d_{ij} \) is not yet covered by the routes of the solution under construction \( R \). The algorithm keeps a list of OD-pairs which demand is not yet covered and at each step of the loop, it selects the element \((i,j)\) with highest demand from that list. The routine getCandidateToExpand explores all possible insertions of nodes \( i \) and \( j \) into routes of \( R \), discarding results which exceed a maximum given deviation and selecting the insertion which produces the minimum incremental route cost. If there is at least one feasible insertion, the algorithm proceeds by modifying the route belonging to the set \( R \). If no expansion is feasible, the algorithm creates a new route by computing the shortest path in the street network from \( i \) to \( j \). Given that the street network is directed, the path from \( j \) to \( i \) is also computed, which can be spatially separated from its opposite counterpart. After a new route is created or an existing one is modified, the updateCoveredDemand routine checks whether additional OD-pairs are covered by the solution under construction \( R \).

3.2 Including turns

We observe that passing through a node always may involve a turn, particularly when changing streets, where a decrease in speed is experienced. Note that a turn involves three subsequent nodes of the network. Therefore, we model the turning cost as the time that would be required to traverse the external arc of the turn, centered on the middle node with a fixed radius (Fig. 3). Note that this is one possible way for approximating the turning cost, which computation is straightforward providing the coordinates of the nodes are available. A different approach is used by methods which compute driving directions, however, they involve larger networks with a

```
R \leftarrow \emptyset;
\text{while (coveredDemand}(R) < \text{totalDemand}) \text{ do }
\quad \text{ODpair} \leftarrow \text{pair with maximum demand not yet covered};
\quad r \leftarrow \text{getCandidateToExpand}(R, \text{ODpair});
\quad \text{if (expansion of } r \text{ is feasible)}
\quad \quad \text{expandCandidateRoute}(r, \text{ODpair});
\quad \quad \text{updateExpandedRoute}(R);
\quad \text{else}
\quad \quad r \leftarrow \text{shortest path between } O \text{ and } D;
\quad \quad R = R + \{r\};
\quad \text{end if;}
\quad \text{updateCoveredDemand}(R);
\text{end while;}
```

Fig. 2 Directed Pair Insertion Algorithm
corresponding increase of computational effort and of requirements of digital cartography preparation. Our model for estimating turning costs is based on the following rationale (see Fig. 3):

- Let \( p = 2\pi \rho \) be the perimeter of the circumference of radius \( \rho \) and let \( l = \theta \pi \rho / 180 \) be the length of arc \( \theta = \overline{ABC} = \arccos \left( \frac{a^2 + c^2 - b^2}{2ac} \right) \).

- Then, the length corresponding to the turn is \( p - l \).

- For a typical city we consider a radius \( \rho = 10 \) metres and to discard degenerate cases (e.g. arcs of 180 degrees) we consider that there is a turn only if the corresponding external arc is greater than 225 degrees.

Upon computing the measure explained above, we include a turning cost into the total route cost, which can be weighted by a parameter. This mechanism is embedded into the Directed Pair Insertion Algorithm at two specific points (highlighted in the pseudocode of Fig. 2), namely: (i) when computing shortest paths using the Dijkstra's algorithm, and (ii) when inserting nodes into existing routes. In both cases, whenever the route under construction is going to be modified by adding or connecting nodes, it is possible to identify the previous and next nodes, therefore the corresponding turning cost can be taken into account.

![Fig. 3 Turning model](image-url)
4 Experiments, results and analysis

We perform several computational experiments to study the behavior of the proposed turning model and the algorithm in which it is embedded. First, we validate the turning model by observing the results produced by the optimization algorithm for different penalizations of the turning cost. This is done by using a real test case related to a small city. Then, using the same case we study the influence of turns over solutions for the TNDP. Since we aim to formulate general conclusions based on results from an optimization model, we should ensure that our resolution algorithm produces good solutions in terms of the objective functions. This is studied by comparing our results against state of the art results computed over a set of benchmark instances of different sizes. Finally, we apply our resolution algorithm including the turning model to a large case, to ensure that the proposed methodology can be applied to such cases. All the experiments are conducted using a standard laptop computer.

4.1 Validation of the turning model

We apply DPIA including the turning model to a real case related to the city of Rivera, Uruguay, comprising 84 centroids, 380 OD-pairs and 13 transit routes (Mauttone and Urquhart, 2009). In that original study, we used an aggregated representation of the network. On the other hand, in this work we use a detailed representation based on the real street network, comprising 1,892 nodes and 9,784 arcs. The algorithm is run for different penalizing values for the turning cost.

Table 1 shows that the more we penalize turns, the higher is the number of routes $|R|$, which is an expectable result. In other words, for high values of the turning penalty, it is preferable a high number of direct routes instead of a small number of meandering routes. It is worth mentioning that using as much as possible realistic value for turn cost it is not a goal of this study. Instead, our goal is to model turns into an optimization algorithm for the TNDP and to observe the sensitivity of results to this new component of the model. Also, we can observe that as the turning penalty increases, so does the operator cost, since it includes penalized turns. The user cost also increases due to the same reason, even though the number of routes increases, which is convenient for the users. The last column reports execution time in seconds. The increase on that value is due to the need of handling a higher number of routes in the solution when we increase the turning penalty.

4.2 Influence of turns over solutions for the TNDP

Fig. 4 shows a possible consequence of ignoring turns when computing solutions for the TNDP. The green point represents user and operator cost of solution 1 from Table 1, which is obtained by running DPIA ignoring turns (only the $C_v$ component is used when computing travel time). The red point corresponds to solution 6, where turning costs are taken into account during route construction. The solutions (structure of
routes) are different, and so are the corresponding user and operator costs, as expectable. We observe that the red solution exhibits higher values for both number of routes and operator cost. User cost is also higher, however, its tendency is not monotone in Table 1. This is expectable because the effect of the penalization is opposite to the fact that a higher number of routes is more convenient for the users. The blue solution is exactly the same as the green one, but its user and operator costs are computed a posteriori by including turning costs $C_u$ penalized as in solution 6 (which were not considered when computing solution 1). This shows that if we ignore turning costs when computing a solution, hidden costs will then arise in the real setting due to turns. Clearly, it is preferable considering turning costs during the computation of the solution, which is evidenced by the domination in terms of both objectives, of the red solution over the blue one.

Table 1 Results of DPIA applied to Rivera for different penalizations of turns

| Solution | Turning penalty | $|R|$ | $UC$ | $OC$ | Time |
|----------|----------------|-----|------|------|------|
| 1        | 0.00           | 12  | 279.74 | 935.83 | 123  |
| 2        | 0.01           | 11  | 302.90 | 971.44 | 124  |
| 3        | 0.05           | 12  | 297.66 | 940.42 | 120  |
| 4        | 0.10           | 15  | 303.93 | 1012.75 | 118 |
| 5        | 0.30           | 18  | 323.32 | 1223.01 | 127 |
| 6        | 0.50           | 18  | 359.65 | 1325.18 | 127 |
| 7        | 0.70           | 20  | 345.18 | 1623.02 | 153 |
| 8        | 1.00           | 24  | 318.47 | 1715.26 | 147 |
| 9        | 5.00           | 66  | 440.97 | 5544.48 | 259 |
| 10       | 10.00          | 83  | 647.83 | 10488.00 | 313 |

Fig. 4 Consequence of ignoring turning costs
Fig. 5 and Fig. 6 show the routes of solutions 1 and 6 respectively. For brevity, we show only one direction of the routes. We can see that in general terms, the routes of solution 6 are more direct since they were computed by considering the turning cost. This solution comprises 18 routes, which is consistent with the size of the actual transit system of the city, which comprises 13 routes. On the other hand, the solution which ignores turns comprises 12 routes. The meandering pattern of this solution is due the pair insertion strategy of DPIA, which deviates existing routes to cover nearby demand. This tendency to deviate routes is smoothed by the penalization of turns in solution 6.

4.3 Comparison with state of the art results for the TNDP

Even though benchmarking of the proposed optimization algorithm DPIA is not the main purpose of this work, in order to ensure that our conclusions are based on results produced by a competitive optimization method, we compare our results against state of the art results. There is not a set of standard benchmark cases to compare optimization algorithms for the TNDP. The only benchmark case widely used in the literature is the one proposed by Mandl (1980), which is a small instance comprising 15 vertices and 21 edges. More recently, a set of cases has been available by Mumford (2013) including the instances Mumford0 to Mumford3, comprising 30, 70, 110 and 127 vertices, and 90, 210, 385 and 435 edges respectively. These cases model an aggregated street network, where any vertex can be either junction or centroid node. The edges are undirected, therefore, the routes are assumed to have identical forward and backward itineraries. The benchmark user cost \( bUC \) is computed as expression (3) divided by totalDemand. The benchmark operator cost \( bOC \) counts only one direction of each route.

We compare against results reported by John et al. (2013), computed over the benchmark instances Mumford0 to Mumford 3 using an evolutionary algorithm which produces an entire set of non-dominated solutions with respect to objectives \( bUC \) and \( bOC \) within a multi-objective framework (Ehrgott, 2005). Fig. 7 plots the results obtained by DPIA and two extremal non-dominated solutions reported in the reference study (Ref.) for each instance, namely, those with minimum \( bUC \) and \( bOC \).
Fig. 5 Routes of the solution without penalized turns
Fig. 6 Routes of the solution with penalized turns
The comparisons of Fig. 7 show that for one instance (Mumford 2), the solution of DPIA is non-dominated with respect to the reference solutions. In the remaining three instances the solution of DPIA is dominated, however, it is relatively close to the non-dominated set defined by the corresponding two reference solutions. It is worth to mention that the greedy deterministic nature of DPIA should be taken into account in the comparison. The evolutionary algorithm (EA) used to compute the reference solutions performs an extensive exploration of the search space. This is consistent with the execution time of the algorithms: for the Mumford3 instance EA took two days, while DPIA took 86 seconds. A more diversified search like the one introduced by the GRASP-TNDP algorithm (Mauttone and Urquhart, 2009a), which is based on the original PIA, would contribute to improve the results of our algorithm. Moreover, some differences in the parametrization of the algorithms makes difficult to conduct a fair comparison. For instance, the model solved by EA does not require to cover the whole demand with direct routes. On the other hand, the number of routes and the route duration are both constrained for EA, while they are free in DPIA.
Fig. 7 Comparison with state of the art results
Fig. 7 Comparison with state of the art results (cont.)
4.4 Application to a large case

The new version of the Pair Insertion Algorithm including directed routes and turning penalties is applied to a large case in order to determine the feasibility of application of the proposed methodology. We build a case based on data from Montevideo, capital city of Uruguay. The city has more than 1.5 million of inhabitants and its current transit system comprises around 140 bus lines. We build the street network based in open data provided by the municipality (IM, 2018) and we generate centroids and walking arcs based in a division which includes zones of different size, depending on population density. The resulting network has 29,222 nodes (from which 63 are centroids) and 62,142 directed arcs, see Fig. 8 for an example. Also, we generate a fictitious origin-destination matrix trying to simulate a realistic density. For doing that, for each pair \((i,j)\) of different centroids we generate its corresponding OD-pair with probability equal to 5%. Thus, we obtained 150 OD-pairs.

Table 2 reports measures as in Table 1. We can observe that the execution time is about 2.8 hours when we disable the penalization of turns. If we consider turning costs the execution time increases around 20%, which is still reasonable. Part of the computational overhead can be explained by the increase on the number of routes, which is around 34%. In general terms, the tendency of both user and operator cost when we penalize turns is similar to the one observed in Table 1 for the smaller case.

![Partial network of the large case](image)

**Fig. 8** Partial network of the large case

| Solution | Turning penalty | \(|R|\) | UC | OC | Time |
|----------|-----------------|------|----|----|------|
| 1        | 0.0             | 136  | 25,700 | 8,410 | 10658 |
| 2        | 0.5             | 182  | 34,332 | 14,007 | 12137 |
5 Conclusions and future work

We studied the effect of considering turns in the Transit Network Design Problem (TNDP). To do so, we proposed a new turning model based on the length of the arc defined by the corresponding three subsequent nodes in the network. Also, we proposed a new version of the Pair Insertion Algorithm, which considers direct routes and includes the turning model.

The computational experiments conducted with real data of a small city, show the relevance of considering turns when computing solutions for the TNDP. More specifically, it is shown empirically that if we ignore turns when computing routes, the cost of both users and operators increase when we translate the solutions to reality (i.e. when we include the turning costs a posteriori). This can be avoided by considering turns while computing the routes, which produces solutions which are better for both users and operators. A detailed calibration of the turning model (radius, maximum angle, penalty) should be conducted in order to simulate the real scenario more accurately.

We show the routes obtained over the map of the study region, which complements the quantitative information. By looking at the routes, we can note that some of them lack of realistic characteristics, as they are very short or they include unrealistic u-turns. This aspect is out of the scope of the study, which is focused on the effect of turns. However, current and future work include the development of a mechanism for controlling the spatial separation of forward and backward itineraries of the same route, as well as considering different street types (transit corridors, avenues, regular streets) with different travel speeds. All these features, along with turns, are intertwined concerning the definition of the route structure of a transit system, therefore, they should be modelled properly in order to obtain realistic solutions.

Though is not the main focus of this study, we conducted a performance comparison between the proposed algorithm and state of the art results over a set of benchmark cases for the TNDP. We show that our results are reasonable, even for one case our solution dominates the state of the art solutions. There is room to improve the accuracy of the proposed greedy algorithm, in the sense of its capability of producing good solutions in terms of the objective functions of the optimization model. A straightforward randomization of some parts and parameters of the algorithm have shown to be effective towards this goal. Also, more sophisticated mechanisms for exploring the search space can be included, using concepts of metaheuristics.

Finally, we have applied our proposed methodology to a large case, concluding that the overhead in running time due to considering turns does not precludes its application. The solutions obtained exhibit reasonable characteristics in comparison to the actual transit system of the corresponding city, particularly the number of routes.
Acknowledgements: This research was supported by Comisión Sectorial de Investigación Científica of Universidad de la República, Uruguay.

References


