Dynamic Activity-Travel Networks: Framework for Integrated Travel Choice Models

Gitakrishnan Ramadurai (ramadg@rpi.edu)
Satish Ukkusuri (ukkuss@rpi.edu)

Rensselaer Polytechnic Institute

January 9, 2008

Abstract

Integrated choice models have several benefits over sequential models including consistent solutions, quicker convergence, and more realistic representation of behavior. Static travel choice models have been integrated using the concept of Supernetworks. However integrated dynamic transport models are less common. In this paper, activity location, time of participation, duration, and route choice decisions are jointly modeled in a single unified dynamic framework referred to as Activity-Travel Networks (ATNs). The framework is similar to the concept of Supernetworks where virtual links are added to augment the network to represent additional choice dimensions. However, solving such a multi-dimensional dynamic choice problem leads to combinatorially increasing choice dimensions. Therefore existing algorithms that depend on path enumeration such as route-swapping algorithm are difficult to implement even for moderately sized networks. We propose a novel extension of Algorithm B (Dial 2006) to dynamic networks, referred to as Algorithm B-Dynamic, that obviates path enumeration to solve for equilibrium in ATNs.

1 Introduction.

Urban transport modeling involves several dimensions of individual choice including activity participation, location, time of participation, duration, choice of mode, and route. Two critical characteristics in modeling urban transport are: i) each individual makes choices so as to maximize his/her benefit, however, ii) the choice environment is dynamic and interactive. Often the choice models are sequentially applied with feedback: initially, the choice environment is assumed fixed and the individual choices are determined. Subsequently, given the determined individual choices the choice environment is adjusted. If feedback is involved, the two steps are repeated until the individual choices and the resulting choice environment are in equilibrium. We also refer to this state as converged solution. This process of iteratively solving a sequence of models forms the basis of the four-step urban transportation modeling paradigm.

As opposed to the sequential procedure, several studies have explored integrated choice models particularly with respect to static transport models. Static travel choice models have been integrated using the concept of Supernetworks (Sheffi (1985); also referred to as Hypernetworks, Sheffi and Daganzo (1980)). Integrated choice models have several benefits over sequential models including consistent solutions, quicker convergence, and more realistic representation of behavior. However integrated dynamic transport models are less common. Few recent studies in this direction include Lam and Huang (2003) and Zhang et al. (2005).

In this paper, activity location, time of participation, duration, and route choice decisions are jointly modeled in a single unified dynamic framework referred to as Activity-Travel Networks (ATNs). The proposed simultaneous framework is motivated by the following considerations: (a) to capture activity demand-supply dynamics in addition to transportation demand-supply dynamics, and (b) to obtain consistent solutions across all dimensions of choice. The framework is similar to the concept of Supernetworks where virtual links are added to augment the network to represent additional choice dimensions. A major hurdle for extending the Supernetwork concept to dynamic networks considering activities is that the resulting multi-dimensional dynamic choice problem leads to combinatorially increasing choice dimensions. Therefore existing algorithms that depend on path enumeration such as route-swapping algorithm are difficult to implement even for moderately sized networks. We propose a novel extension of Algorithm B (Dial 2006) to dynamic networks, referred to as Algorithm B-Dynamic, that obviates path enumeration to solve for equilibrium in ATNs.

2 ATN Representation and Motivation

ATNs use a network representation where nodes are activity centers that are joined by travel links. Activities are represented by arcs that both originate and terminate in the same node (activity centers). Each activity arc is characterized by a unique activity type and an activity duration. An activity-travel sequence for an individual can be represented as a 'path' that includes both travel and activity arcs. All individuals at the beginning of the model start from 'home' and make a predefined set of activity stops and reach a fixed final destination (for example, home-shop-work where shop is an activity stop while work place is the final destination). The location of the final activity is assumed fixed; activity location is a decision dimension for the intermediate activity stops. This is not be a restrictive assumption when modeling commute behavior while for other tours the final destination may be assumed to be home. Time is discrete and the time horizon is divided into T equal sized, discrete time intervals. Durations of arc-traversal for travel arcs is always assumed to be a function of flow, while for activity arcs it is assumed fixed. Consistent with rational behavior assumption, each individual chooses the activity-travel sequence that provides the maximum generalized utility. DUE is defined as follows: "all individuals from an origin participating in the same set of activities have equal and maximum utility irrespective of their chosen route, departure time, activity location, and duration". In practice, obtaining a zero-tolerance DUE may require unusually long running times. Further, the DUE may not exist due to discretization of time. A more reasonable solution is ϵ-DUE (similar to the ϵ-UE in Dial (2006)). In ϵ-DUE, the route, departure time, activity location, and duration choice of all individuals from an origin participating in the same set of activities are such that the difference in utility between any two paths is at most ϵ units.

3 B-Dynamic Algorithm

The proposed B-Dynamic Algorithm is an extension of Dial's Algorithm B. We summarize Dial's algorithm first and extend it to obtain B-Dynamic algorithm.

In Algorithm B, the network is decomposed into acyclic sub-networks rooted at the origin. This acyclic sub-network, referred to as a 'bush', contains arcs that carry all, and only, flow from the given origin to a destination. In the B-Dynamic algorithm, we have the additional flow differentiating characteristic of activity sequence. Further, the arc flows are time-varying. Therefore the 'bush' in B-Dynamic algorithm will be derived from time-expanded network including activity arcs.

The basic principle of Algorithm B is to ensure the min- and max-cost paths for each origin specific bush are within the ϵ tolerance limit. This is achieved iteratively by equilibrating the current bush and updating it to include any new min-cost paths and equilibrating again till convergence is achieved. Equilibration of a bush, in turn, is achieved by computing the min- and max-cost paths, shifting flows between these two paths so that they are equilibrated, and repeating the process till all paths are equilibrated.

The overall structure of the process for the B-Dynamic Algorithm is exactly similar. We need to ensure that the min- and max-utility paths for each origin and each activity sequence combination specific bush are within the ϵ tolerance limit. However, the implementation details for each step is different.

B-Dynamic Algorithm

1 Initialization: For each origin, destination, and activity sequence combination create an initial feasible bush.

2 Equilibration: For each origin, destination, and activity sequence combination,

2a Construct dynamic min- and max-utility paths from corresponding bush.

2b If difference in cost between min- and max-utility paths is greater than ϵ, shift trips from min- to max-utility paths such that their cost difference is less than ϵ. Else, skip to [2d].

2c Re-compute travel delays for all travel arcs for all times and utilities for all activity arcs for all times. Return to [2a].

2d Check if the max-utility path on the entire network is greater than the min-utility path of the bush. If yes, augment the bush with new max-utility path. Return to [2a]. Else, continue to [3].

3 Termination: For each origin, destination, and activity sequence combination, check if the max-utility path on the entire network is lesser than the min-utility path of the bush. If yes, terminate. Else, return to [2].

4 Modified Dynamic Shortest Path Algorithm

The dynamic shortest path algorithm is required to compute the min- and max-utility paths is different from the traditional time-dependent shortest path (Ziliaskopoulos and Mahmassani 1993). First, the cost labels on the network are not the travel times. The cost is represented by the utility which is a function of the travel times (or durations in activity arcs). Second, the shortest (or max-utility) path must include certain activity arcs to satisfy the activity-sequence combination for each individual. Therefore, the shortest path labels at each node must keep track of the activity-sequence traversed in the current path. The modified dynamic shortest path algorithm is given below:

Given a network $G(N, A)$ where A includes both travel and activity arcs.

Notation:

$\lambda_i[t, a]$: Shortest-path label for node i at time-period t and activity-combination lexicon a

L: Set of all activity-combination lexicon.

Let us say we have two different activities Shop and Eat out; then the lexicon set

$L =$ None, Shop, Eat out, Shop+Eat out

SE: Scan eligible list. We have a 2-tuple consisting of (node i, time interval t) as opposed to just the node. This may significantly reduce the number of computations that need to be made.

FS_i: Forward star at node i.

AA_i: Set of activity arcs at node i. We store the id k of the activity arc.

l_k: Label of activity arc k, for example Shop, Eat out.

$u_k(t)$: Utility of participating in activity arc k when starting the activity at time t.

$d_k(t)$: duration of activity arc k.

$u_{ij}(t)$: (dis)utility of traveling on arc $i - j$ leaving arc i at time t.

$d_{ij}(t)$: duration of traversing arc $i - j$ leaving arc i at time t.

Usually, $u_{ij}(t) = -\alpha(t)d_{ij}(t)$, where α is the value of time.

Let $M = \max_{\forall k,t} u_k(t)$

then, let $u'_k(t) = M - u_k(t)$
and $u'_{ij}(t) = M - u_{ij}(t)$

Now the shortest-path obtained by using $[u'_k(t), u'_{ij}(t)]$ as the cost vectors will give us the maximum utility path. Further, all tranformed costs are clearly non-negative. We do not have to worry about negative cycles.

The modified TDSP is an extension of the traditional TDSP algorithm. For every (node i, time interval pair t) in the scan eligible list, we scan both travel arcs $((i, j) : j \in FS_i)$ as well as activity arcs $(k \in AA_i)$.

Step 1 Initialization
- $\lambda_i(t, l) = \infty \;\; \forall \;\; (i, t, l) \in (N, T, L)\backslash(origin, 0, None)$
- $\lambda_{origin}(0, None) = 0$
- Insert (origin, 0) into SE list.

Step 2 If SE is empty, then go to step 3.

```
Else, remove top (node i, time t) pair from SE list.
For each activity combination l ∈ L
  For each arc (i, j) ∈ FS_i
    If λ_j[t + d_ij[t], l] > λ_i[t, l] + u'_ij[t]
    Then, λ_j[t + d_ij[t], l] = λ_i[t, l] + u'_ij[t]
      PRED_j[t + d_ij[t], l] = [i, t]
      Insert (j, t + d_ij[t]) into SE list.
    Else, go to next node j
  End Loop
End Loop
For each activity combination lexicon l ∈ L
  For each activity arc k ∈ AA_i : l_k ∉ l
    If λ_i[t + d_k[t], l + l_k] > λ_i[t, l] + u'_k[t]
    Then, λ_i[t + d_k[t], l + l_k] = λ_i[t, l] + u'_k[t]
      PRED_i[t + d_k[t], l + l_k] = [i, t]
      Insert (i, t + d_k[t]) into SE list.
    Else, go to next arc
  End Loop
End Loop
```

Step 3 Stop

References

Dial, R.: 2006, A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration, *Transportation Science* **40**, 917–936.

Lam, W. H. and Huang, H.-J.: 2003, Combined activity/travel choice models: Time-dependent and dynamic versions, *Network and Spatial Economics* **3**, 323–347.

Sheffi, Y.: 1985, Urban transportation networks: Equilibrium analysis with mathematical programming methods, *Prentice-Hall Inc., Englewood Cliffs, NJ* .

Sheffi, Y. and Daganzo, C.: 1980, Computation of equilibrium over transportation networks: The case of disaggregate demand models, *Transportation Science* **14 (2)**, 155–173.

Zhang, X., Yang, H., Huang, H.-J. and Zhang, H. M.: 2005, Integrated scheduling of daily work activities and morningevening commutes with bottleneck congestion, *Transportation Research Part A* **39**, 41–60.

Ziliaskopoulos, A. and Mahmassani, H.: 1993, A time-dependent shortest path algorithm for real-time intelligent vehicle/highway systems., *Transportation Research Record* **1408**, 94–104.