Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 196–203

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

Enumerating minimum path decompositions to support route choice set generation

Irith Ben-Arroyo Hartman[a,*], Luk Knapen[b], Tom Bellemans[b]

[a]*Caesarea Rothschild Institute for Interdisciplinary Applications of Computer Science, University of Haifa, Haifa 31905, Israel*
[b]*Transportation Research Institute (IMOB) Hasselt University, Diepenbeek, Belgium*

Abstract

This paper concerns the structure of movements as were recorded by GPS traces and converted to routes by map matching. Each route in a transportation network corresponds to a collection of directed paths or cycles in a digraph. When considering only directed paths, corresponding to utilitarian trips, the path is not necessarily a shortest path between its origin and destination, and can be split up into a small number of segments, each of which is a shortest or least cost path. Two consecutive segments are separated by *split vertices*. Split vertices act as intermediate destinations in the mind of travellers who try to hop between them using minimum cost paths. Hence they provide useful information to build route choice models. In this paper we identify and enumerate all possible decompositions of a path into a minimum number of shortest segments. This gives us an indication of the importance of split vertices occurring in particular sets of revealed routes that belong either to a single traveller or to a specific group. The proposed technique allows for automatic extraction of frequently used intermediate destinations (way-points) from *revealed preference* data.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: clique cover, proper interval graph, indifference graph, transportation network, route choice set generation, GPS traces

1. Introduction

We begin with a short background from transportation science and graph theory in order to motivate our problem. We model a transportation map by a directed graph. where the vertices denote junctions or points of interest such as a petrol station, shop, restaurant, rest area, or any point where drivers may choose to stop. Each edge of the graph corresponds to a road segment and represents a set of lanes having the same direction (forward or backward). Travelers move between locations on the *geometries* of source and destination segments. Transportation scientists are interested in modelling route choice behaviour in order to forecast and simulate travellers' decisions under different conditions and resources of information. See Bovy[3] for a review on route choice set generation and selection. The *branch and bound* technique by Prato[11,8] generates candidate routes to populate the choice set and removes the ones that do

* Corresponding author. Tel.: +972-4-8288370 ;
E-mail address: irith.hartman@gmail.com

Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.325

not meet specific constraints (e.g. number of traffic lights, left turns, and others). We proposed in [10] an additional criterion for route choice set, which is the minimum number of shortest subroutes which constitute a given route. In [10] it was claimed that when a traveller considers a choice of route between an origin and a destination, he/she does not necessarily choose the quickest/fastest/cheapest route, but rather a route which is a concatenation of a *small* number of shortest (cheapest) routes. The vertices connecting these shortest routes may have a special significance for the traveller, they are intermediate destinations which can be quantified by the use frequency of these vertices in the set of all paths chosen by the travellers. The intermediate destinations support the route choice modeling. Finding these vertices, and all possible ways of breaking up a path into a minimum number of shortest paths relates to the problem of finding some minimum clique covers in an indifference graph - as will be shown in the following sections.

2. Definitions and Basics

We begin with some basic standard definitions from graph-theory. We use definitions and notations as in [1]. We will then continue to new definitions related to the applications in transportation networks.

Let $G = (V, E)$ be a directed graph with vertex set V and edge set E. The vertices correspond to *nodes* in a road network, and the edges correspond to *links* in the network. Each edge e has a non-negative *cost* $c(e)$ which is the effort (e.g. time or money) required to traverse the link in the network. For a subgraph $H \subseteq G$, let $V(H)$ and $E(H)$ denote the sets of vertices and edges of H, respectively.

A *walk* is a sequence of vertices $P = (v_0, v_1, \ldots, v_l)$, not necessarily distinct, where $(v_i, v_{i+1}) \in E(G)$ for all $i = 0, 1, \ldots, l-1$. Vertices v_0 and v_l are called *initial* and *terminal* vertices, respectively, of P, and vertices $v_1, \ldots, v_{l-1}$ are called *internal vertices* of P. The walk P is said to be *connecting* v_0 and v_l, and it is also denoted by $P(v_0, v_l)$. A walk $Q(v_0, v_l)$, is *internally-disjoint* from P if all the internal vertices of Q, are distinct from the vertices in P.

A *path* is a walk where all its vertices are distinct. For a path $P = (v_0, v_1, \ldots, v_l)$, any subsequence of vertices $v_i, v_{i+1}, \ldots, v_j$, where $0 \le i \le j \le l$ is a *subpath* of P, and is denoted by $P(v_i, v_j)$. The *length* of a path, is the number of edges in it (i.e. l), the *size* of a path, denoted by $|P|$, is the number of vertices in it (i.e. l+1), and the *cost* of a path, denoted by $c(P)$ is the sum of the costs of its edges. A path $P(v_0, v_l)$ is a *least cost path* between v_0 and v_l, if there exists no other path connecting v_0 and v_l of lower cost.

We remark that if $c(e) = 1$ for all $e \in E$ then the cost of a path coincides with its size. We assume that the vertex traversal cost is zero. A single edge (u, v), being a path connecting between u and v, may be least cost, or not. If it is not a least cost path connecting between u and v, then it is called a *non-least-cost-edge*.

It is easy to see that if P is a least cost path, then any subpath of P is also a least cost path.

The converse of this statement is false since it is possible that all the subpaths of $P(v_0, \ldots, v_l)$ (except P itself) are least cost paths, but P is not a least cost path connecting v_0 and v_l and there is another least cost path Q connecting v_0 and v_l. This fact motivated the following definition, as in [10].

Definition 2.1 (P- shortcut, minimal shortcut, fork and join vertices, bypassed vertex set). *Let $P = (v_0, v_1, \ldots, v_l)$ be a given path. A $P(v_i, v_j)$-*shortcut *(or for brevity, P -* shortcut*, or* shortcut*), is a path $Q(v_i, v_j)$, internally- disjoint from P, where $v_i, v_j \in V(P)$, such that $c(Q(v_i, v_j)) < c(P(v_i, v_j))$. The vertices v_i and v_j are called* fork *and* join *of the shortcut, respectively, and the internal vertices of P between the fork and the join (i.e. $v_{i+1}, \ldots, v_{j-1}$) are called Q-*bypassed vertex set*, or* bypassed vertex set *and denoted by $B(Q)$. A shortcut Q is* minimal *if $B(Q)$ does not contain $B(Q')$ where Q' is another shortcut to P. (See Figure 1).*

We emphasize that $B(Q)$ contains consecutive vertices on P. Therefore, it can be marked by the fork and join of a shortcut Q, which are the vertices preceding, and following the set $B(Q)$, respectively.

Clearly, a least cost path cannot have any shortcuts.

Definition 2.2 (Basic Path Component (BPC), path splitting, splitVertex). *Given a path P, a subpath of P is called a* Basic Path Component*, or for short, a BPC, if it is either a least cost path connecting its endpoints, or P is a single non-least-cost-edge. A* path splitting *of P is a partition of P into subpaths each of which is a basic path component. A* splitVertex *is a vertex separating two consecutive BPC in a path splitting, and is denoted by v_i^s.*

We remark that there may be many ways to split a path, for example, the trivial partition into edges $(v_0, v_1), (v_1, v_2), \ldots, (v_{l-1}, v_l)$ is an example of such a partition. We are interested in finding a path splitting with a