

# CHAPTER 1

## INTRODUCTION

Many problems of engineering and scientific importance can be related to the general problem of finding a **shortest path** through a network or graph. Examples of such problems include routing within transportation networks, managing telephone traffic, layout of printed circuit boards, and mechanical theorem-proving and problem-solving. We can say that the shortest path problem lies at the heart of network flow programming. Some might mistakenly consider a shortest path problem as a procedure to find a shortest *distance* from one point to another in a network. However, in actuality, the shortest path problem involves finding a path connecting two specified nodes that minimizes the sum of the *branch values* on the path. There is no reason that these branch values need represent only *distances*, even indirectly, or for that matter in general, be nonnegative. For example, the branches might correspond to *activities* of some kind, where the value associated with each branch is the *cost* of that activity. The problem would then be to find a sequence of activities that accomplishes a specified requirement while minimizing the total *cost* involved. Another alternative is that the value associated with each branch might be the *time* required for that activity. The problem would then involve finding a sequence of activities that accomplishes the required task while minimizing the total *time* involved. Furthermore, in many algorithmic settings such as in column generation procedures, shortest path subproblems arise having mixed-sign reduced cost coefficients. Thus many important applications of shortest path problems have nothing to do with minimal distance traversals in the usual sense of the word.

In real-world situations, there are other practical considerations that arise in the context of shortest path problems such as *time-dependent* delays and *multi-modal* traversals. These extensions are significant from a practical standpoint since they more realistically represent the underlying physical scenario. Accordingly, this research draws

on such *extended* shortest path problem concepts while studying a practical dynamic, time-based *transportation* routing problem.

We begin our discussion below by describing some basic ideas and definitions of different variants of shortest path problems. Following this, we present a brief overview of TRANSIMS (Transportation Analysis Simulation System, and in particular, of its Route Planner Module, which is the focal point of this study.

**The time-independent shortest path problem**, often referred to simply as the shortest path problem, is the most basic of all shortest path problems. For this problem, we wish to find the best way to traverse a network from an origin to a destination as inexpensively (or as fast) as possible, given a fixed cost (or length) associated with each arc connecting any pair of nodes within the network. The time-independent descriptor connotes that the cost associated with the arc is not dependent on the time of day during which this arc is traversed. As we shall see in our later discussions, some of the costs in a network can be negative, but the network is assumed to contain no negative circuits, i.e., the sum of the costs on arcs comprising any circuit in the network is nonnegative. Without this assumption, the shortest path procedure could get trapped in a negative cost circuit within the network, repeatedly traversing this circuit an infinite number of times, with the cost decreasing in each loop around the circuit. The shortest path problem is usually modeled as a minimization linear program, and is solved more effectively via its dual, as discussed later in Chapter 2.

**The label-constrained shortest path problem** extends the standard shortest path problem by additionally specifying a label (e.g. designating a travel mode) to each link in the network, and then restricting the selected path to conform to some stipulated string of admissible label-constrained sequence of links. The problem involves finding a path that accomplishes the required travel plan while *minimizing* some value attributes, and also conforming with an admissible sequence of link labels. For this research, the label string contains a list of admissible *travel mode* sequences that could be used in the specified order along the path from the origin to the destination. For example, given a mode string

“walk-car-walk” for a trip from home to the office, any admissible travel pattern would start via a “walk” mode from home to the “car”, use the “car” as a transportation vehicle, and finally “walk” to the office. The label-constrained shortest path problem is relatively new, and has only recently been introduced by Barrett, et al. [1998] in the context of TRANSIMS.

For **the time-dependent shortest path problem**, unlike as in the time-*independent* shortest path problem, the cost/length/travel time on each arc connecting some node  $A$  to node  $B$  in the network depends on the time of arrival at node  $A$ , such as for driving times in congested networks. Note that a departure from node  $A$  occurs instantly upon arrival, there being no delay or waiting permitted in departures from any of the nodes in the network. Such problems are often solved using an extension of Dijkstra’s shortest path algorithm applied to an expanded static time-space network in which the link travel times are effectively time-invariant. Again, when some travel time functions in the network can take negative values, it is assumed that no negative circuits exist, i.e., the sum of the costs on arcs comprising any circuit in the network provides a *nonnegative* value for each instant of time.

**TRANSIMS**, which is an acronym for the **TR**ansportation **AN**alysis and **SIM**ulation **S**ystem developed by the Los Alamos National Laboratory and sponsored by the U.S. Department of Transportation (DOT) and the Environmental Protection Agency (EPA), is an integrated transportation and air quality forecasting program designed to provide transportation planners accurate and complete information on traffic impacts, congestion, and pollution. TRANSIMS is mainly composed of five modules (written in C++ on Linux): (1) a Population Synthesizer Module that generates synthetic households from census data, (2) an Activity Generator Module that assigns a set of activities, along with their priorities, travel modes, and vehicle preferences for each household member of the synthetic household for a 24-hour period, **(3) a Route Planner Module, which is the focus of this thesis, that develops a time-dependent label-constrained shortest path for each trip executed by a traveler in the system**, (4) a Traffic Microsimulator Module that considers individual vehicle interactions and predicts the transportation system

performance, and (5) an Emission Estimator Module that estimates motor vehicle emissions using the vehicle information obtained from the Traffic Microsimulator Module. Figure 1 details the inputs and outputs of these modules, along with their interactions.

The technique adopted by TRANSIMS to identify a suitable travel route for any user within the Route Planner Module is a variant of Dijkstra's procedure for finding shortest paths, which is suitably modified to accommodate time-dependent travel times and label sequence constraints. The underlying problem is referred to as a Time-Dependent Label-Constrained Shortest Path Problem, and is unique to TRANSIMS applications. In the present study, we extend this algorithm by developing the following features:

- (a) We recommend a method to work implicitly with a certain composition graph  $G^*$  that combines the transportation network with the admissible label-sequence graph. This graph  $G^*$  captures all *possible* paths for a given single trip starting from the origin node and ending at the destination node, while conforming with *the admissible mode string*. Hence, this graph  $G^*$  represents traversal in the underlying multimodal transportation network. It is constructed via using the transportation network and a transition graph  $G_L$ , which is a graph that shows all the possible sequences of the travel modes by which we can reach the destination node from the starting node.
- (b) We use more modern partitioned shortest path algorithmic schemes to implement the time-dependent label-constrained procedure. The partitioned shortest path algorithm is a special efficient label-correcting procedure that partitions the set of nodes in a network into two subsets: NOW and NEXT, in order to sequence their consideration while revising shortest path estimates during the iterative solution procedure.

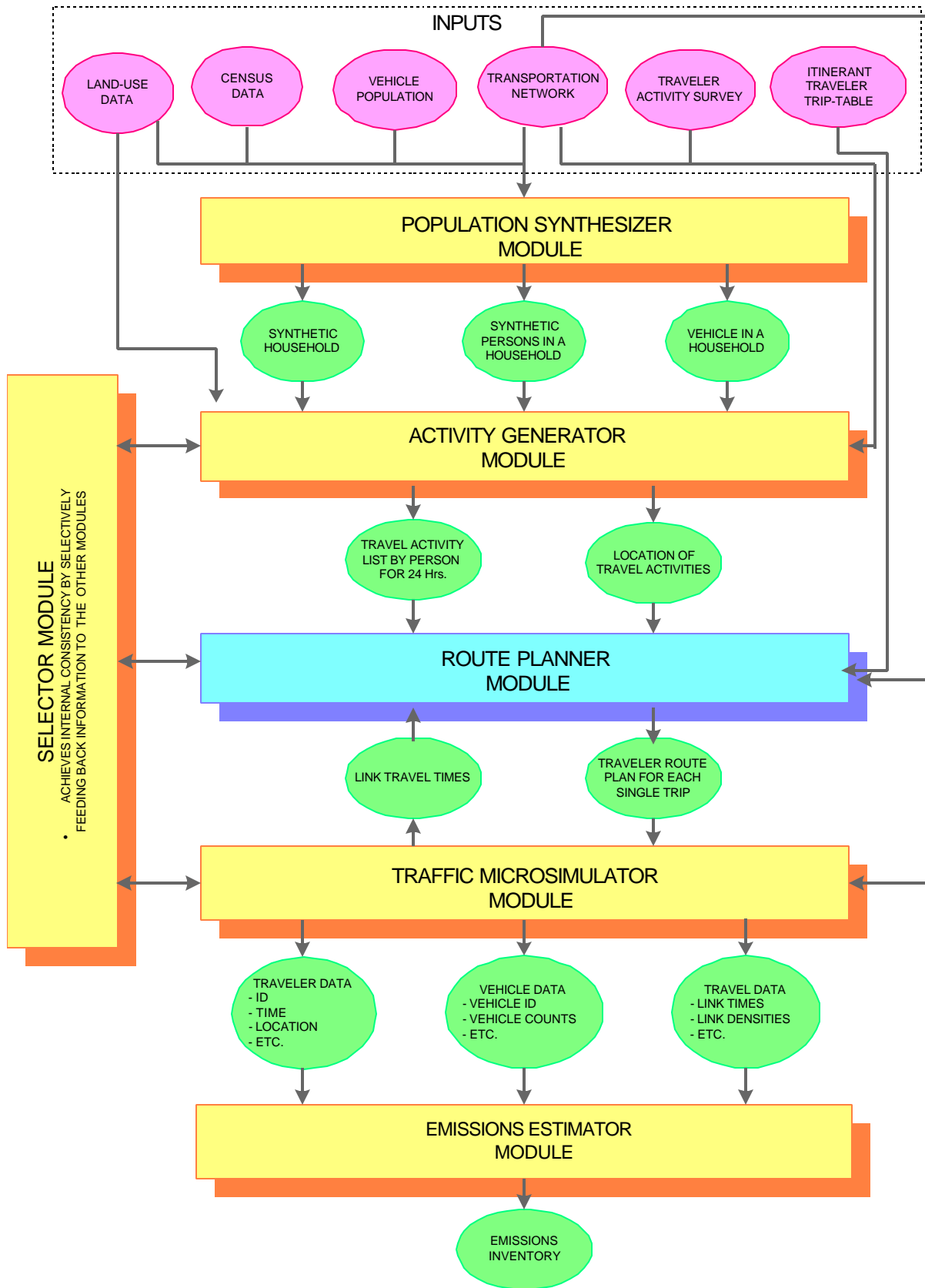


Figure 1: TRANSIMS framework.

- (c) We introduce the notion of curtailing search based on an indicator of progress and projected travel times to complete the trip. While such a consideration could be devised to retain exact optimality of the solution process, it is far more effective to develop a heuristic strategy for obtaining near optimal paths. The idea of the proposed heuristic is to bias the search more pointedly towards the *destination*. The Euclidean heuristic developed examines far fewer nodes and links than does the original exact algorithm.

In the following chapter, we present a literature review of the foregoing three shortest path problem variants, and provide a brief description of TRANSIMS. In Chapter 3, we describe the time-independent label-constrained shortest path problem in greater details, which are suitably modified to accommodate label sequence constraints and time-independent travel times to TRANSIMS network. In Chapter 4, which is in contrast with the former chapter of constant travel times, we present a procedure to solve a shortest path when the link delays are time-dependent functions. The next chapter, Chapter 5, is concerned with the Route Planner Module, a principal module in the TRANSIMS framework, which develops a time-dependent label-constrained shortest path for each trip executed by a traveler in the system. Finally, Chapter 6 delineates the tasks to be accomplished to complete this thesis.