TOPIC 16
TRAVEL SUPPLY-DEMAND
MODELLING

# PATH COMPOSITION AND MULTIPLE CHOICE IN A BIMODAL TRANSPORTATION NETWORK

MARIA GLORIA BATTISTA
Senior Researcher,
Finsiel S.p.A., Via A. Noale 200 - Rome, ITALY

MARIO LUCERTINI
Professor of Industrial Engineering,
Centro Volterra and Dip. Ingegneria Elettronica
Università di Roma "Tor Vergata"
Via della Ricerca Scientifica - Rome, ITALY

BRUNO SIMEONE
Professor of Optimization,
Dip. Statistica, Probabilità e Statistiche Applicate
Università di Roma "La Sapienza",
Piazzale A. Moro 5 - Rome, ITALY
and Centro Volterra Università di Roma "Tor Vergata"

## Abstract

The problem of defining the optimal network and the corresponding flows are classical problems in transportation system modelling. The path choices, the infrastructures and lines to be activated, the investment choices, correspond to decision problems that can often be formulated, in simplified version, in a format that can be faced by solving a sequence of shortest path problems.

## INTRODUCTION

Predicting accurate user flows in a transportation network and finding an optimal design of the network are classical problems in transportation system modelling (Gartner et al. 1976; Fisk, 1986; Papageorgiou, 1991). One crucial issue in developing these models is the determination of the set of all the paths from any given origin o to any given destination d that a user can reasonably choose. Early all-or-nothing proposals assumed that the whole user flow from o to d is funnelled along the shortest (or cheapest, or quickest) path. Subsequent stochastic network loading (SNL) models assumed that flows are spread over all paths from o to d according to a certain probability distribution. An intermediate approach consists in the preliminary determination of a restricted set P of "feasible" paths from o to d, with the stipulation that users can only choose paths from P. The choice of P reflects behaviour, service, and other aspects and in the context of random utility models corresponds to the choice set generation process (Manski, 1977).

A well-known instance of this approach is the SNL logit model of Dial (1971), where flows are restricted only to eligible paths. Leurent (1994) further requires that origin/destination (O/D) paths be efficient (a precise definition of eligible and efficient paths will be given in the next section). He states "we believe that our numerical experiment demonstrates above all that path-based equilibration algorithms are much more efficient than link-based algorithms", and also "... It is remarkable that, if paths are identified, more behavioural models are easier to solve mathematically". A similar remark applies to traffic assignment procedures (such as the ones described by GuÈlat et al. 1990 and by Crainic et al. 1990) based on the solution of multicommodity network flow problems.

Here we propose a fairly general strategy for finding a "good" set Π of feasible O/D paths. According to this strategy, which will be called path composition, all paths in Π are obtained through the concatenation (following suitable composition rules) of a small number of subpaths. All such subpaths share the property of being shortest paths from their own origin to their own destination. As a matter of fact, this path composition procedure has two main advantages: good realism (if the composition rules are well chosen, the resulting paths correspond to good approximations of real user choices) and polynomial complexity (the degree of the polynomial being the maximum number of subpaths in a final path). One can then rely on path composition to enhance the performance of traffic assignment or network design models.

The overall plan of this paper is the following. The first half deals with path generation: after discussing in the next section some desirable properties of a "good" set of feasible O/D paths, in the following section we introduce path composition as a general technique for obtaining a set with such properties. This technique is demonstrated through some examples. Algorithmic and complexity aspects are emphasized. In the second half of the paper we discuss the use of path composition within traffic assignment and network design models. We shall illustrate our approach on the following three specific problems: traffic assignment on a road transportation network (problem road) or on a collective road/rail transport network (problem rail), see the fourth section, network design for a road/rail integrated transportation system (problem network design), see the fifth section. The models are proposed for different requirements and for different economic, organizational and layout constraints. The complexity of the decision problems is discussed, and special cases are pointed out. The solution technique is presented in a unified framework, in terms of a combination of shortest paths on particular networks. For a survey of network optimization algorithms, see (Ahuja et al, 1993; Simeone et al. 1988.) Although our discussion is focused on bimodal (road-rail) networks, many procedures (eg path composition or the two network design heuristics) are still applicable, with obvious changes, in the context of single-mode or arbitrary multi-mode networks.

# HOW TO CHOOSE PATHS

## Feasible paths

The path that a user (passenger or car) could choose for a trip between an O/D pair must satisfy several constraints (Dafermos, 1972; Ben-Akiva et al. 1984; Ben-Akiva, Lerman, 1985). In particular, the time/cost of the path cannot be too far from the optimal one (a path with this property will be called efficient). Nevertheless, the number of paths satisfying such constraints is, in general, too large. In fact, assuming only the time/cost efficiency constraints (the so called set of e-optimal paths), the number of paths is, in general, an exponential function of the number of arcs of the transportation network. Such set of paths will contain several pairs of similar paths (ie paths differing by few arcs), which are not significant options from a modelling viewpoint.

To prevent these and other undesirable features, we shall impose on "feasible" paths some further requirements. Actually, we shall assume that a set of feasible paths (SFP) satisfies the following defining conditions, depending both on the structure of the problem (network, user behaviour,...) and the structure of the model (model purpose, approximations, complexity issues,...).

*Independence conditions*

- Given the transportation network (infrastructures, lines and rules), the SFP for a given O/D pair does not depend on the demand and on the SFP for the other pairs; therefore the overall SFP is the union of mutually disjoint SFP's, one for each O/D pair (we denote by SFP(o,d) the SFP between the O/D pair (o,d)).
- The flow on each path of SFP(o,d) will be computed afterwards, either on the ground of the (o,d) demand and the generalized cost of the path, or on the ground of the overall flow distribution.

*Rule-based conditions*

- Each path must satisfy the rules of the transportation system (junctions and transfers, monotonicity, constraints depending on fare regulations, sequence of modes,...).

*Efficiency conditions*

- Each path must be efficient, ie the generalized costs of the given path and of the optimal one should differ by less than a given percentage.

*Dissimilarity conditions*

- Any two distinct paths in the same SFP(o,d) must be significantly different, ie at least a given fraction of their length must use different arcs/nodes of the transportation network.

*Eligibility*

- Each path in SFP(o,d) must have the property that, for any arc (i,j) along the path, the distance d(o,i) should be smaller than d(o,j): the distance can be either the euclidean one or the length of a shortest path on the network.

*User behaviour conditions*

- Each path must correspond to real options selected by the user and must satisfy the constraints depending on the user behaviour.

*Complexity conditions*

- The number of paths in each set SFP(o,d) must be small enough and the running time required for path generation, flow simulation and optimal flow management must not grow too fast as the network size increases. In practice, one must be able to run these procedures, say, on a workstation; moreover, the organization and the information flow must be simple enough to allow for an efficient use of the system for design and management purposes.

These general defining conditions will be translated into modelling and algorithmic terms in the next section.

The SPF's can be used both for transportation system analysis and for designing good strategies for system improvement. The concept of feasible path is very general and could be adapted to several applications. Meaningful path types are the following:

*Access/exit path*

An access/exit path is a path on the road network either from the origin to a train station where a given line is taken (access path), or from a train station to the destination (exit path). The concept could be easily generalized to other bimodal transportation systems.

Let G be a bimodal road/rail network, (o,d) an O/D pair, tl a train line defined by the sequence of stop stations $\{s_1, s_2, \cdots, s_k\}$, $d_{ij}$ the generalized cost between two stations i and j on the line tl, $\pi'_i[\pi_i]$ the shortest path from o to $s_i[s_i$ to d] on the road subnetwork, $d'_i[d_i]$ the corresponding generalized cost.

Let us call the path an access path if: $\left(d'_r + d_{rk}\right) = \min_i \left\{d'_i + d_{ik}\right\}$.

Let us call the path an exit path if: $\left(d_{1r} + d_r\right) = \min_i \left\{d_{1i} + d_i\right\}$.

In other words, both the departure and the arrival stations are chosen so as to minimize the total generalized cost.

*Viable paths*

Let G be a bimodal road/rail network and (o,d) an O/D pair. A path $\pi$ from o to d may contain both road-arcs and rail-arcs. However, a user would hardly choose a path $\pi$ along which he takes a train, gets down at a certain station, goes on by car, and then takes another train. More formally, let us call a path $\pi$ *viable* if it has no more than one maximal rail-subpath, that is, if any two non incident rail-arcs are connected by a rail-path. Notice that: a viable path can be a road-only path; the road subpath of a road/rail path is not necessarily an access/exit path; moreover, in a shortest viable path the road access subpath is the shortest road path to reach the station and the road exit subpath is the shortest path from the station to the destination.

In the concept of viable path, train lines are not considered; each rail-arc has an average generalized cost and the cost of a sequence of rail-arcs is simply the sum of the arc costs. Therefore, in a shortest viable path, the rail-subpath is the shortest path between the two loading and unloading stations on the infrastructure rail subnetwork.

## Path composition

The basic idea of path composition is to divide each path into a given (small) number of subpaths, where each subpath is optimal (with respect to a given objective function). The number of nodes that are candidates for being a junction between two subpaths is relatively small and two paths belonging to the same SFP(o,d) must differ by at least a junction node. Path composition reflects one significant feature of user behaviour, namely, the decomposition of a goal (planning an O/D itinerary) into a sequence of subgoals (planning intermediate stages of the itinerary). A more formal way to present this procedure is the following.

*Network*

- A set P of O/D pairs.

- A set N of nodes and a set A of arcs.

  Let n =|N|, m = |A|, p = |P|.
- Each arc belongs to a unique mode.

Each node belongs to exactly one of the following types:
- O/D nodes. They are incident only to road-arcs and their total number is z;
- Road [rail] nodes. They are those infrastructure nodes that are incident only to road[ rail] arcs; let NR [NS] be the subset of road [rail] nodes (nr = |NR|, ns = |NS|).
- Mode-transfer nodes.
- Lines are pre-defined single-mode paths on the network. The nodes along a line may be loading/unloading stations, transfer stations or transit stations. Let L be the set of lines (l = |L|)

*Input*
- A network G(N,A) (either infrastructure- or line-based).
- A set of attributes for each arc or subset of arcs (mode definition, line definition, and so on).
- A set of subpath composition constraints.
- A weight for each arc (eg generalized cost).
- A set P of O/D pairs.
- A set of path composition rules.
- A set of path domination rules.

*Output*
- The sets {SFP(o,d)} for all O/D pairs (o,d).

We first give a broad description of the procedure, then we shall provide more details.

## Procedure PATHCOMP

*Pre-processing*
- Find for each O/D pair (o,d) the set I(o,d) of network nodes that are candidates to be junction nodes. For problem *road*, the junction nodes are a subset of NR and must be chosen on the ground of the criteria set out earlier. For problem *rail*, junction nodes are transfer nodes from one line to another.
- Let SPF(o,d) = $\varnothing$ $\forall (o,d) \in P$ .

*Double tree*
- Find for each origin node $o \in O$ the shortest path tree from o to all nodes in I satisfying the subpath composition constraints.
- Find for each destination node $d \in D$ the shortest path tree from all nodes in I to d satisfying the subpath composition constraints (this can be obviously avoided if the network is symmetric).

*Path composition*
- For each O/D pair (o,d) and for each node i in I(o,d), check whether the composite path satisfies the path composition rules and it is not dominated by a path already in SFP(o,d). If both test are successful then add the path to SFP(o,d) else continue.

To have a fairly complete outline of the procedure, we must qualify the constraints in the double tree procedure, and the path domination rules.

*Subpath composition constraints*

The set of junction nodes I may contain different types of nodes depending on the problem. For problem *road* only road nodes are included, for problem *rail* only rail stations are included. No mode transfer takes place at any node in I. The modal split among different transportation modes is computed a priori, on the ground of different criteria.

Given the O/D pair (o,d), if the node $i \in I$ is a road node, then the subpaths are simply the shortest paths from o to i and from i to d.

If $i \in I$ is a rail node, then the subpath from o to i is a multimodal shortest path formed by a road shortest subpath from o to a mode-transfer node s (departure or loading station) and a rail subpath on a feasible line from the loading station s to the junction node i (which is a transfer station).

The loading station s is chosen so that the overall path from o to i has minimum length.

The subpath from i to d is computed in the same way, by connecting a rail subpath to a road subpath.

*Path composition and domination rules for problem road*

The paths in the same SFP(o,d) are computed for non decreasing values of the objective function (eg generalized cost).

A path $\pi'$ from $o \in O$ to $d \in D$ through the junction node $i' \in I$, is dominated by a path $\pi$ from o to d through $i'' \in I$, already in SFP(o,d), ie with a generalized cost less or equal to $\pi'$, if $\pi$ contains a portion of $\pi'$ larger than a given threshold.

Given: an O/D pair (o,d), the set I(o,d), a set of feasible paths SFP(o,d) already computed and a node $k \in I(o,d)$ not contained in the paths of SFP(o,d), it is easy to devise an algorithm that in time verifies whether the (unique) path through k is feasible (and, therefore, must be considered for inclusion into SFP(o,d)) or not. The overall computation requires $O(n|I(o,d)|)$ time. To include the path in SFP(o,d) some additional conditions must be verified (efficiency, at most one entrance into the same highway).

The proposed approach can be easily generalized to multiattribute paths. In fact, if an arc is characterized by two or more weights (time, cost, generalized cost, etc.), we can define more sophisticated composition and domination rules with a small complexity increase.

*Path composition and domination rules for problem rail*

The main path composition rules for the rail case are the following:

- only one path with the same train path (with the shortest road-path);
- no more than two train lines (at most one transfer);
- given the two train lines, the loading/unloading station minimizes the path length;
- there is no transfer if the line reaches the destination;
- only eligible paths are allowed.

These rules lead to a complexity bounded by O(ns . n), since the number of transfer stations is O(ns) and the time for path composition given the access/exit paths and the two lines is O(n).

Also for the rail case the approach can be easily generalized to more sophisticated rules with a small complexity increase.

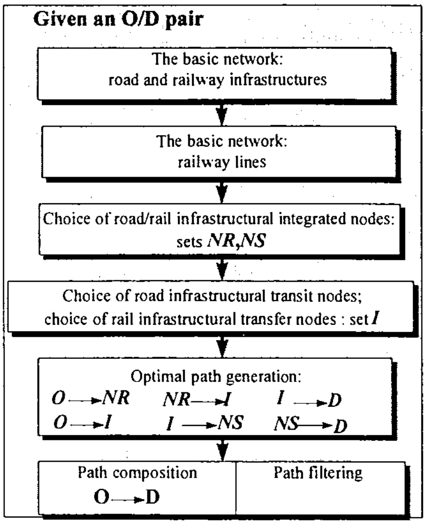


**Figure 1 Path composition framework**

To give one specific example, in the Italy main road/rail network the above parameters take the values: $z \cong 300$ $n \cong 1.000$ (250 rail nodes + 700 road nodes) $I \cong 900$ train lines.

*Size of the input*

The network contains O(n) arcs. In fact, G is a planar graph (therefore m = O(n)), the number of attributes for each arc is given and the number of bits needed for time, cost and generalized cost can be also assumed to be given.

The rail lines are I and the O/D pairs are p.

Moreover, the input should contain the information concerning the subpath constraints and the path composition constraints.

*Size of the output*

In the worst case, the size of the output is given by the number of O/D pairs (=p), times the maximum number of feasible paths for a given O/D pair $\left(\max_{(o,d)\in O/D} |I(o,d)| \leq n\right)$, times the maximum length of a path ($\leq n$).

Therefore the output size is $O(pn^2)$.

For realistic values of z and n, these complexity figures are large and become a crucial point of the procedure. Several techniques can be used in order to reduce the output. In particular, the number of feasible paths between a given O/D pair can be kept small by a filtering procedure (around 4÷10 for each O/D pair). Moreover, the average length of a path in practice is relatively small (< 20 arcs) and only a fraction of the O/D pairs are considered (≈ 10.000). The resulting size of the output for the Italian road/rail network is about 1.000.000 entries.

Complexity reduction techniques would affect most of the following procedures.

$$\text{Preprocessing}\begin{Bmatrix} \text{for the road subnetwork: } O(p \cdot nr) \\ \text{for the rail subnetwork: } O(I^2) \end{Bmatrix}$$

$$\text{Subpath composition}\begin{Bmatrix} \text{for road traffic: } O(z \cdot nr^2) \\ \text{for rail traffic: } O(z \cdot l \cdot nr^2) \end{Bmatrix}$$

$$\text{Path composition}\begin{Bmatrix} \text{for road traffic: } O(p \cdot nr^2) \\ \text{for rail traffic: } O(p \cdot l \cdot nr^2) \end{Bmatrix}$$

## TRAFFIC ASSIGNMENT

### General remarks

In the analysis of transportation systems, a basic module for simulating the system workload and behaviour is traffic assignment. System-based assignment, which is particularly meaningful in freight transportation, often leads to multicommodity flow models. On the other hand, user-based assignment often leads to nonlinear optimization or stochastic utility models. In either case, we shall argue that path composition is useful to enhance solution procedures in many practical situations. The subsequent flow assignment can be carried out either by a sequential approach (eg greedy), or by an equilibrium one (eg local search), or by global optimization (eg linear programming). Three examples are discussed below.

### Road traffic assignment

To face the complexity issues due to the large number of paths that a user can choose and to simplify the model, many analyses of the traffic flow assume that all users, moving between a given O/D pair, will choose the same optimal path (Wardrop, 1952) or the k shortest paths (Shier, 1976; Skiscim and Golden, 1989). This assumption does not correspond to the real pattern of the traffic flow. Moreover, the resulting model might be very sensitive to network changes, ie many routes may change as a result of a few minor changes in the network parameters, with major changes in the total flow on several arcs. A model assuming a probabilistic distribution of user choices among a set of efficient paths fits better the traffic flows measured on the floor (Dial, 1971; Crainic and Rousseau, 1986). The distribution of users could depend only on the generalized cost of the path (suitable combination of time and cost based on average values) or also on the overall traffic flow, to take into account delays due to the congestion on the arcs. The former assumption leads to a simpler and easier to use model (each O/D pair can be analyzed separately), whereas the latter leads to an equilibrium model which is more accurate but difficult to manage, often unstable and strongly dependent on the time schedule of the users on the network (congestion is not, in general, an aspect that can be calculated on the basis of average traffic, but requires peak time analyses) (Smith, 1979 and 1983; Patriksson, 1991). An interesting compromise between the two approaches could be based on decomposition techniques (Safwat and Magnanti, 1988; Larsson and Patriksson, 1992). In this paper we shall use, for the road network, the following:

- The overall road traffic is divided into:
  - road traffic for rail transportation
  - long distance commercial traffic
  - short distance commercial traffic
  - long distance passengers traffic
  - short distance passengers traffic.

- Such components are considered in a given sequence and for each component the network parameters are evaluated on the ground of the workload assignment of the previous components.
- Each component, but the last one, is assigned to the network on the basis of generalized cost for distributing the users among a set of feasible paths, which does not depend on the overall traffic.
- The last component is assigned on the ground of an equilibrium model, taking into account the overall traffic flow and a suitable set of feasible paths.

In particular, if the different sets of feasible paths are generated via the PATHCOMP procedure for problem *road*, described earlier the resulting paths of each SPF(o,d) have roughly the same generalized cost and they do not overlap too much. These features make the use of logit models particularly attractive.

Some critical points remain: the partition of users between the different modes (modal split between train and road); the choice of road flow components.

## Freight traffic on a rail network

The rail network transport model takes into account lines, ie services on the infrastructure network available for passengers, and road access/exit paths (Aashtiani, 1972). A *line* is a sequence of rail stations connected by rail subpaths. In this case a multicommodity flow model would produce, at the same time, all O/D paths carrying positive flows and the corresponding flows. If one restricts the set of O/D paths to the set of feasible paths generated via the PATHCOMP procedure for problem *rail*, described earlier, the resulting multicommodity model produces more realistic path flows. Moreover, in spite of the larger number of variables (path flows), the problem has a special structure which makes it amenable to efficient algorithms (Ahuja et al., 1993).

## Viable paths

We have already defined *viable paths*. We will now consider some traffic assignment problems on a bimodal network, when only viable O/D paths are allowed. There are 5 different types of viable paths, as shown in Figure 2.

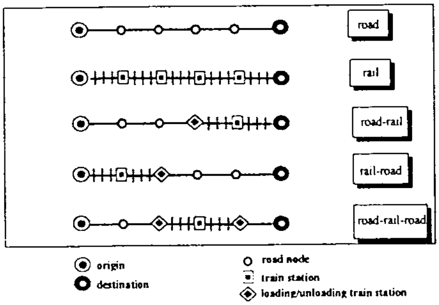


**Figure 2 The 5 types of viable path**

The simplest model to be considered is a deterministic user equilibrium one. Under the assumption of no congestion it is well known (Beckmann et al., 1956) that the solution to this model has an all-or-nothing pattern: the whole flow from any origin o to any destination d is conveyed along the shortest viable path.

Interestingly, finding shortest viable paths for all O/D pairs is of the same order of complexity as finding ordinary shortest paths for all O/D pairs. To see this, let G = (N,A) be a bimodal directed network, where $A = A_{road} \cup A_{rail}$, $A_{road}$ is the set of all road-arcs, $A_{rail}$ is the set of all rail-arcs. Define, for all k, h$\in$N,

| | | | | |
|---|---|---|---|---|
| road(k,h) | = length of a shortest | road | path from k to h | |
| rail(k,h) | = | " | rail | " |
| roadrail(k,h) | = | " | road-rail | " |
| railroad(k,h) | = | " | rail-road | " |
| roadrailroad(k,h) | = | " | road-rail-road | " |
| viable(k,h) | = | " | viable | " |

*Proposition 1*

One can compute viable(o,d), for all (o,d)$\in$P, in O(pn$^2$) time, that is, with the same order of complexity as computing the lengths road(o,d), rail(o,d) of the ordinary single-mode shortest paths from o to d, for all (o,d)$\in$P.

Proof: One has

$$\text{roadrail}(o,d) = \min_{h\in N} \{\text{road}(o,h) + \text{rail}(h,d)\}$$

$$\text{railroad}(o,d) = \min_{h\in N} \{\text{rail}(o,h) + \text{road}(h,d)\}$$

$$\text{roadrailroad}(o,d) = \min_{h\in N} \{\text{roadrail}(o,h) + \text{road}(h,d)\}$$

$$\text{viable}(o,d) = \min \left\{ \begin{array}{c} \text{road}(o,d),\ \text{rail}(o,d),\ \text{roadrail}(o,d), \\ \text{railroad}(o,d),\ \text{roadrailroad}(o,d) \end{array} \right\} \tag{1}$$

Clearly road(o,d) and rail(o,d), for all (o,d)$\in$P, can be computed in O(pn$^2$) by an ordinary shortest path algorithm. In view of (1), roadrail(o,d), railroad(o,d) and roadrailroad(o,d), for all (o,d)$\in$P, can be computed in extra O(pn) time starting from road(.,.) and rail(.,.). Finally, the function viable(o,d), for all(o,d)$\in$P, can be computed in O(p) time starting from the previous five functions road(.,.), ... , roadrailroad(.,.). Hence the proposition follows.||

Note: In actual computations, the set N in (1) may be replaced by a suitable $N' \subset N$ without loss of optimality.

Alternatively, probabilistic assignment procedures (eg a logit model) can be employed. When congestion phenomena cannot be ignored, stochastic user equilibrium (SUE) models can be used. In all such cases, for each O/D pair (o,d) a small set SFP(o,d) of viable paths is generated. This can be accomplished by a variant of the PATHCOMP procedure for problem road with an *ad hoc* composition rule at each intermediate node i $\in$ I(o,d) , so as to ensure that only viable paths are produced.

## A BILEVEL NETWORK DESIGN MODEL

### General remarks

The design phase involves several aspects. The infrastructure network design problem, in its simplest form, deals with the assignment of arc capacities, within a given budget, in order to maximize the overall system effectiveness. This leads to a bilevel decision problem, where the first level assigns the capacities and the second one assigns the flows, with two different objective functions: overall system effectiveness versus choice of user-efficient paths.

Other decision problems deal with line design problems, commercial traffic routing, the integration of different traffic flows, the integration of different transportation modes for the same stream of traffic, and so on. All these problems require O/D path computations where path composition, with a suitable definition of the SPF's, is a basic tool.

The choice of investment strategies, on a given set of network arcs, which minimize average travelling time has been analyzed by several authors (Dionne and Florian, 1979; Florian, 1986; Le Blanc and Boyce, 1986; Leurent, 1993) for both road and rail networks, with joint assignment of capacity and flows (Crainic and Rousseau, 1986; Winter, 1989; Patriksson 1990).

In the present section we shall describe a network design model embodying the notion of viable paths described in previous sections.

Given the bimodal road-rail network G = (N,A) defined earlier, we shall assume that investments are allowable on a subset of arcs $Q = Q_{road} \cup Q_{rail}$, where $Q_{road} \subseteq A_{road}$, $Q_{rail} \subseteq A_{rail}$. For each arc $j \in Q$, $n_j$ investment alternatives may be considered, from which one, and only one, alternative must be chosen. We shall assume that, for each $j \in Q$, alternative 1 corresponds to the null investment.

For each $j \in Q$, and for each investment alternative on j, investment costs

$$0 = c(j,1) < \ldots < c(j,k) < \ldots < c(j,n_j)$$

with corresponding arc transit times

$$t(j,1) > t(j,2) > \ldots > t(j,k) > \ldots > t(j,n_j)$$

are known.

Let $q = |Q|$. Let $r = n_1 + \ldots + n_q$ be the total number of alternatives and let $B = \{0,1\}$.

An *investment plan* can be identified by a binary vector

$$y=[y(1,1),\ldots,y(1,n_1);\ y(2,1),\ldots,\ y(2n_2);\ldots;\ y(q,1)),\ldots,\ y(q,n_q)] \in B^r$$

satisfying the *multiple-choice constraints*

$$\sum_{k=1}^{n_j} y(j,k) = 1 \qquad \forall j \in A, \tag{2}$$

where $y(j,k) = 1$ if the k-th investment alternative is chosen on arc j, and $y(j,k) = 0$ otherwise $(j \in Q, k = 1,\ldots,n_j)$

Let $Y \equiv \{y \in B^r\text{: } y \text{ satisfies } (2)\}$ be the set of all investment plans. Let b be the total budget available for investments. For each O/D pair $(o,d) \in P$, let the *demand* D(o,d) be known. The demand D(o,d) is equal to the estimated total number of users travelling from o to d along the arcs of the bimodal network G. We assume that the demand is *rigid*, that is, it does not depend on network upgrading. Furthermore, we shall assume that all users choose only viable paths and that there is *no congestion*: travelling times are independent of user flows.

Given an investment plan $y \in Y$, for each $(o,d) \in P$ let T(o,d;y) be the minimum travelling time (or generalized cost) along viable paths from o to d, conditional on the investment plan y.

For each $j \in Q$, due to the multiple-choice constraints (2), there will be a unique k such that $y(j,k)=1$. Then the transit time on arc j will be given by $t(j,k)$.

The mathematical formulation of the model is the following:

*Constraints*

*Multiple choice*: $\sum_{k=1}^{n_j} y(j,k) = 1,\ j \in Q;$ *Budget*: $\sum_{j \in Q} \sum_{k=1}^{n_j} c(j,k)y(j,k) = 1, \leq b$

*Binary variables*: $y(j,k) \in \{0,1\}\quad j \in Q;\ k = 1,\ldots,n_j.$

*Objective Function*

*User total transportation time*: $\sum_{(o,d) \in P} D(o,d)T(o,d;y).$

The above model is a bilevel optimization one. At the lower level, for any given feasible investment plan y one computes $T(o,d;y)$ for all $(o,d) \in P$. This is a traffic assignment problem along viable paths, which can be solved by one of the methods outlined earlier. At the upper level, one minimizes the total travelling time (or the total generalized cost):

$$F(y) = \sum_{(o,d) \in P} D(o,d)T(o,d;y)$$

over all feasible investment plans y.

## Derivatives and penalties

In order to describe heuristics for the upper level of the network design model discussed in the previous subsection, we need to introduce some definitions and notation. For simplicity, we assume that at the upper level the objective function is the total travelling time. The generalized cost case can be carried out with minor changes.

Suppose that the investment alternative k is chosen on arc $j \in Q$. If the alternative k+1 (upgrading) or k-1 (downgrading) is preferred, the total travelling time, as well as the investment cost and the transit time on arc j will accordingly change.

In order to quantify these changes, consider any given $y \in Y$. Let us associate with y, for each $j \in Q$, two modified investment plans $y_j^+$ and $y_j^-$ as follows.

There is a unique k such that $y(j,k) = 1$. Define $y_j^+$ to be the vector whose components are:

$$y_j^+(j,k) = 0$$

$$y_j^+(j,k+1) = 1$$

$$y_j^+(j',k') = y(j',k'),\qquad (j',k') \neq (j,k) \text{ or } (j,k+1)$$

Symmetrically, define $y_j^-$ to be the vector with components

$$y_j^-(j,k) = 0$$

$y_j^-(j,k-1) = 1$

$y_j^-(j',k') = y(j',k'), \qquad (j',k') \neq (j,k) \text{ or } (j,k-1)$

Notice that $y_j^+$ is undefined when $k = n_j$ and that $y_j^-$ is undefined when k=1.

When the investment plan changes from y to $y_j^+$[to $y_j^-$] we shall say that arc j is *upgraded* [downgraded].

Accordingly, one can define (boolean) derivatives of the total travelling time, of the investment cost on arc j, and of the transit time along arc j as displayed in Table 1.

**Table 1 Travelling time, transit time and investment cost**

| | Arc j downgraded | Arc j upgraded |
|---|---|---|
| Total travelling time | $\Delta_j^-(y) = F(y_j^-) - F(y) \geq 0$ | $\Delta_j^+(y) = F(y_j^+) - F(y) \leq 0$ |
| Arc transit time | $t_j^-(y) = t(j,k-1) - t(j,k) > 0$ | $t_j^+(y) = t(j,k+1) - t(j,k) < 0$ |
| Arc investment cost | $c_j^-(y) = c(j,k-1) - c(j,k) < 0$ | $c_j^+(y) = c(j,k+1) - c(j,k) > 0$ |

We shall also need a notion of up- and down-penalties.

Let $y \in Y$ be a given investment plan. For $j \in Q$, let $P_j$ be the set of all O/D pairs (o,d) such that the shortest viable path from o to d conditional on y, (as computed earlier), includes arc j.

The *flow* on arc j is then defined by

$$\varphi_j \equiv \varphi_j(y) = \sum_{(o,d) \in P_j} D(o,d)$$

The *up-penalty* $\pi_j^+$ is defined by

$$\pi_j^+ \equiv \pi_j^+(y) = t_j^+ \varphi_j$$

Similarly, the *down-penalty* $\pi_j^-$ is defined by

$$\pi_j^- \equiv \pi_j^-(y) = t_j^- \varphi_j$$

The use of the term "penalty" is justified by the following proposition.

*Proposition 2*

One has

$$\Delta_j^+(y) \leq \pi_j^+ \tag{3}$$

$$\Delta_j^-(y) \leq \pi_j^- \tag{4}$$

*Proof*: Suppose that arc j is upgraded. Then the total travelling time will decrease by an amount $\pi_j^+(y) = t_j^+ \varphi_j$ if no additional user chooses arc j after the upgrading, and by an even larger amount if some does. Similarly one proves (4).||

### Two heuristics

In this Section we are going to describe two heuristics for the upper optimization level of the above described network design model.

The first heuristic (DROP/ADD) may be viewed as a generalization of heuristic MQKM in (Dionne and Florian, 1979).

The DROP/ADD heuristic starts from the most expensive (infeasible) investment plan $y^0$ with cost $B_0$ (that is, $y^0(j,n_j) = 1$ for all $j \in Q$) and consists of two phases. In the DROP phase, only one arc at a time is downgraded: this is the arc j for which the benefit-to-cost ratio $\Delta_j^-/c_j^-$ is smallest. The DROP phase ends as soon as the current investment plan becomes feasible, ie, its cost B becomes $\leq b$. If $B<b$, an ADD phase follows in which, at each step, the arc j with largest ratio $\Delta_j^+/c_j^+$ is upgraded, as long as the current cost remains within the budget b. Here is a formal description of this heuristic.

### Drop/add heuristic

begin

y:= most expensive (infeasible) investment plan

B:= total cost of investment y

**DROP:** Until $B \leq b$

for each arc j do

if $y(j,1) = 1$ then $\rho(j) := +\infty$

else compute $c_j^-(y)$; compute (update) $\Delta_j^-(y)$; $\rho_j := \Delta_j^-(y)/c_j^-(y)$;

endif

endfor

let $\rho_{j*} \min_{j \in Q} \rho_j$; $B := B - c_{j*}^+(y)$; arc j* is downgraded;

repeat

**ADD:** While $B \leq b$

for each arc j do

if $y(j,n_j) = 1$ then $\rho_j := 0$

else compute $c_{j*}^+(y)$; compute (update) $\Delta_j^+(y)$; $\rho_j := \Delta_j^+(y)/c_j^+(y)$;

endif

endfor

let $\rho_{j*} \max_{j \in Q} \rho_j$; $B := B + c_{j*}^+(y)$; arc j* is upgraded;

endwhile

end

Note: Re-optimization techniques are used to update $\Delta_j^+(y)$ and $\Delta_j^-(y)$.

Dionne and Florian's heuristic MQKM is but a special case of DROP/ADD when

(i) there are only two alternatives, TO INVEST or NOT TO INVEST, for each arc;

(ii) the investment cost on each arc is proportional to the arc length;

(iii) the network is single-mode, and ordinary shortest O/D paths are considered (notice that they are viable by definition).

The above DROP/ADD heuristic has a greedy nature: arc downgrading decisions in the DROP phase and arc upgrading decisions in the ADD one are made sequentially, and once they have been made, there is no possibility to change them at a later stage. In order to alleviate this difficulty, we introduce a new heuristic where these decisions are made more globally. This heuristic starts in the same way and has the same overall structure as the first heuristic. The key difference is that, at each iteration, several arcs are downgraded or upgraded rather than a single arc.

At each iteration of the DROP phase, one finds the current set of arcs to be downgraded by solving a binary knapsack problem, whose variables $x_j$, $j \in Q$, are defined as follows: $x_j = 1$ if arc j is downgraded, else $x_j = 0$. The objective function, to be minimized, is the total down-penalty: this is an upper bound on the total decrease of travelling time in view of Proposition 2. The knapsack constraint ensures that the total saving is at least R: the amount R may change from one iteration to another and is chosen so as to be neither too small nor too large. On one hand, R should not exceed a fixed fraction s of the gap between the initial cost $B_0$ and the available budget b; on the other hand, R should not exceed a fixed fraction r of the current maximum possible saving C (ie the saving when all arcs are downgraded). Actually, one sets $R = \{\sigma(B_0 - b), \rho C\}$. The two parameters $\rho$ and $\sigma$ are chosen by the user and must be calibrated in advance. Along similar lines, in the ADD phase one solves a single knapsack problem to "fill-up" the budget by upgrading a suitable set of arcs. A formal description of the second heuristic follows.

## Penalty knapsack heuristic

begin

y:= most expensive (infeasible) plan; $B_0$:= total cost of investment y; B:= $B_0$;

**DROP:** Until $B \leq b$

compute total travelling cost $z(y) = \sum_{(o,d)\in P} D(o,d)T(o,d,y)$

for each $j \in Q$ compute flow$\varphi_j$; H:= Q;

for each $j \in Q$

if y(j,1) = 1 then H:=-{j}

else compute $t_j^-, c_j^-, \pi_j^- = \varphi_j, t_j^-$;

endif

endfor

$C := \sum_{j\in H} c_j^-$;

{C is the maximum possible current saving, ie all arcs are downgraded}

R:= min $\{\rho C, \sigma(B_0\text{-}b)\}$ ; {$\rho$ and $\sigma$ are two user-defined parameters}

Solve the linear binary knapsack problem:

$\min \sum_{j\in H} \pi_j^- x_j$ s.t. $\sum_{j\in H} c_j^- x_j \geq R$; $x_j = 0$ or $1, j\in H$

let $x^*$ be an optimal solution:

all arcs j such that $x_j^*$ are downgraded;

$B := B - \sum_{j \in H} c_j^- x_j^*$;

repeat

**ADD:** Compute z(y);

for each $j \in Q$ compute flow $\varphi_j$;

H:= Q;

for each arc $j \in Q$

if $y(j,n_j) = 1$ then H:=H-{j}

else compute $t_j^+(y), c_j^+(y); \pi_j^+ = \varphi_j, t_j^+$;

endif

endfor

Solve the linear binary knapsack problem:

$\min \sum_{j \in H} \pi_j^+ x_j$ s.t. $\sum_{j \in H} c_j^+ x_j \leq b - B$; $x_j = 0$ or $1, j \in H$

let $x^*$ be an optimal solution:

all arcs j such that $x_j^*$ are upgraded;

end

Our last result yields an upper bound on the total number of knapsack problems to be solved during the execution of the heuristic.

Let $s_{min} = \min\{c(j,k) - c(k-1,j) : j=1,\ldots,q; k=2,\ldots,n_j\}$ be the smallest possible saving obtainable by single-arc downgrading.

*Proposition 3*

The total number v of knapsack problems to be solved satisfies the inequality:

$$v \leq \left[1/\min\left\{\sigma, \frac{\rho s_{min}}{(B_0 - b)}\right\}\right] + 1. \tag{5}$$

Furthermore, if at the end of the DROP phase non-null investments are made on all arcs of Q, then the above inequality can be strengthened into

$$v \leq \left[1/\min\left\{\sigma, \frac{q\rho s_{min}}{(B_0 - b)}\right\}\right] + 1. \tag{6}$$

*Proof*

Within the DROP phase, each iteration results in a saving of at least R.
Since $R = \min\{\sigma(B_0 - b), \rho C\}$ and since

$$C \geq s_{min}, \tag{7}$$

the saving at each iteration is at least $\{\sigma(B_0 - b), \rho s_{min}\}$. On the other hand, the gap between the initial cost $B_0$ and the actual budget is $(B_0 - b)$. It follows that after at most

$$\left[\frac{B_0 - b}{\min\{\sigma(B_0 - b), \rho s_{min}\}}\right] = \left[\min\left\{\sigma \frac{\rho s_{min}}{(B_0 - b)}\right\}^{-1}\right]$$

iterations, the current cost B becomes $\leq b$ and then the DROP phase must stop. Since a single knapsack problem must be solved in the ADD phase, the inequality (5) follows. If at the end of the DROP phase the investment on every arc of Q is non-null, at each iteration inequality (7) can be strengthened into $C \geq q s_{min}$

Reasoning as above, one obtains (6). ||

Although inequality (5) is somewhat crude, one can use it to control v by properly adjusting the two parameters $\rho$ and $\sigma$. As expected, the larger are $\rho$ and $\sigma$., the smaller is v.

## ACKNOWLEDGMENTS

The authors are grateful to Professors M. Ben-Akiva. E. Cascetta, M. Florian, and A. Nuzzolo for their valuable comments. We also thank the referees: the presentation of our work has greatly benefited from their suggestions.

## REFERENCES

Aashtiani H.Z. (1972) The multi-modal traffic assignment problem. Ph.D. Thesis, Oper. Res. Center, MIT, Cambridge, Mass.

Ahuja R.K., T.L. Magnanti and J.B. Orlin (1993) *Network flows. Theory, algorithms, and applications*. Prentice Hall, Englewood Cliffs, New Jersey 07632.

Beckmann M., C.B. Mc Guire and C.B. Winsten (1956) *Studies in the economics of transportation*. Yale Univ. Press, New Haven, CT.

Ben-Akiva M., M.J. Bergman, A.J. Daly and R. Ramaswamy (1984) Modelling interurban route choice behaviour. *Symp. on Transpn. and Traffic Theory*, 299-330.

Ben-Akiva M. and S.R. Lerman (1985) *Discrete choice analysis: theory and application to travel demand.* Cambridge, Mass.: The MIT Press.

Crainic T.G., M. Florian, J. GuÈlat and M. Spiess (1990) Strategic Planning of freight transportation: STAN, an interactive graphic system. *Transpn. Res. Rec.*, 1283, 97-124.

Crainic T.G. and J.M. Rousseau (1986) Multicommodity multimode freight transportation: a general modelling and algorithmic framework for the service network design problem. *Transpn. Res.*, 20B, 225-242.

Dafermos S.C. (1972) The traffic assignment problem for multiclass-user transportation network. *Transpn. Sci.*, 6, 73-83.

Dial R.B. (1971) A probabilistic multi-path traffic assignment model which obviates path enumeration. *Transpn. Res.*, 5, 83-11.

Dionne R. and M. Florian (1979) Exact and approximate algorithms for optimal network design. *Networks*, Vol. 9, 37-59.

Fisk C.S. (1986) A conceptual framework for optimal transportation system planning with integrated supply and demand models. *Transpn. Sci.*, 20.

Florian M. (1986) Nonlinear cost network models in trasportation analysis. *Math. Progr.* Study 26, 167-196.

Gartner N. H., B.L. Golden and R.T. Wong (1976) Modelling and optimization for transportation system planning and operations. Working paper, MIT, Cambridge, Mass.

GuÈlat J., M. Florian and T.G. Crainic (1990) A multimode multiproduct network assignment model for strategic planning of freight flows. *Transpn. Sci.*, 24.

Larsson T. and M. Patriksson (1992) Simplicial decomposition with disaggregate representation for the traffic assignment problem. *Transpn. Sci.*, 26, 4-17.

Le Blanc L.J. and D.F. Boyce (1986) A bilevel programming algorithm for exact solution of the network design problem with user-optimal flows. *Transpn. Res.*, 20B, 259-265.

Leurent F. (1993) Cost versus time equilibrium over a network. Institut National de Recherche sur les Transports et leur SecuritÈ, Arcueil.

Leurent F. (1994) Some contributions to make the logit assignment model tractable. Institut National de Recherche sur les Transports et leur SecuritÈ, Arcueil.

Manski C., (1977) The structure of random utility models. *Theory and Decision*, 8, 229-254.

Papageorgiou M. (1991) *Traffic and transportation systems.* Amsterdam: Pergamon Press.

Patriksson M. (1990) The traffic assignment problem: theory and algorithms. Rep. LiTH-MAT-R-90-29, Linköping Inst. of Tech, Linköping.

Patriksson M. (1991) Algorithms for urban traffic network equilibria. Thesis, Linköping Studies in Science and Technology n∞ 263, Linköping.

Safwat K. and T.L. Magnanti (1988) A combined trip generation, trip distribution, modal split and trip assignment model. Transpn. Sci., 22, 14-30.

Shier D. R. (1976) Algorithms for finding the k shortest paths in a network. Paper presented at ORSA/TIMS Spring Meeting, 1976.

Simeone B., P. Toth, G. Gallo, F. Maffioli and S. Pallottino (Eds.) (1988) Fortran codes for network optimization. *Annals of Oper. Res.* 13.

Skiscim C.C. and B.L. Golden (1989) Solving k-shortest and constrained shortest path problems efficiently. *Annals of Operation Research.*, 20, 249-282.

Smith M.J. (1979) The existence, uniqueness and stability of traffic equilibria. *Transpn. Res.*, 13B, 295-304.

Smith M.J. (1983) The existence and calculation of traffic equilibria. *Transpn. Res.*, 17B, 291-303.

Wardrop J.G. (1952) Some theoretical aspects of road traffic research. *Proc. Inst. of Civil Eng.*, part II, 325-378.

Winter P.B. (1989) Topological network synthesis. in Simeone B. (ed.): Combinatorial optimization, Lecture Notes in Mathematics, Vol. 1403, Berlin: Springer.