Journal of Advanced Transportation, Vol. 25, No2, pp. 187 – 210

An AI-Based Approach for Transit Route System Planning and Design

M. Hadi Baaj and Hani S. Mahmassani

We present an AI-based solution approach to the transit network design problem (TNDP). Past approaches fall into three categories: optimization formulations of idealized situations, heuristic approaches, or practical guidelines and ad hoc procedures reflecting the professional judgement and practical experience of transit planners. We discuss the sources of complexity of the TNDP as well as the shortcomings of the previous approaches. This discussion motivates the need for AI search techniques that implement the existing designer's knowledge and expertise to achieve better solutions efficiently. Then we propose a hybrid solution approach that incorporates the knowledge and expertise of transit network planners and implements efficient search techniques using AI tools, algorithmic procedures developed by others, and modules for tools implemented in conventional languages. The three major components of the solution approach are presented, namely, the lisp-implemented route generation design algorithm (RGA), the analysis procedure TRUST (Transit Route Analyst), and the route improvement algorithm (RIA). An example illustration is included.

Introduction

The purpose of this paper is to describe an AI-based solution approach to the transit network design problem (TNDP). Such a problem has been studied by several authors in the past (see for example Lampkin and Saalmans, 1967; Rea, 1971; Silman, Barzily, and Passy, 1974; Mandl, 1979; Newell, 1979; Dubois, Bell et Llibre, 1979; Hasselstrom, 1981; and Ceder and Wilson, 1986). In the TNDP, one seeks to determine a configuration, consisting of a set of transit routes and associated frequencies, that achieves some desired objective, subject to the constraints of the problem. Mathematical formulations of the TNDP have been concerned primarily

M. Hadi Baaj is Assistant Professor in the Department of Civil Engineering at Arizona State University, Tempe, Arizona. Hani S. Mahmassani is Professor in the Department of Civil Engineering at the University of Texas at Austin, Austin, Texas.

with the minimization of an overall cost measure, generally a combination of user costs and operator costs. The former is often captured by the total travel time incurred by users in the network, while a proxy for operator costs is the total number of buses required for a particular configuration. Feasibility constraints may include, but are not limited to: 1) minimum operating frequencies on all or selected routes (policy headways, where applicable); 2) a maximum load factor on any bus route, and 3) a maximum allowable bus fleet size.

Most existing formulations can be viewed as variants of the following mathematical program:

$$\text{Minimize } \{c_1 [\sum_{j=1}^{n} \sum_{i=1}^{n} d_{ij} t_{ij}] + \sum_{\text{all } k \in SR} f_k T_k]\} \quad (1)$$

Subject to:

$$\text{frequency feasibility: } f_k \geq f_{min} \quad \text{for all } k \in SR \quad (2)$$

$$\text{load factor constraint: } LF_k = \frac{(Q_k)_{max}}{f_k CAP} \leq LF_{max}$$

$$\text{for all } k \in SR \quad (3)$$

$$\text{fleet size constraint: } \sum_{\text{all } k \in SR} N_k = [\sum_{\text{all } k \in SR} f_k T_k] < W$$

$$\text{for all } k \in SR \quad (4)$$

where

d_{ij} = demand between nodes i and j

t_{ij} = total travel time between i and j = $t_{invtt,ij} + t_{wt,ij} + t_{tt,ij}$

$t_{invtt,ij}$ = in-vehicle travel time between nodes i and j

$t_{Wt,ij}$ = waiting time incurred while traveling between nodes i and j

$t_{tt,ij}$ = transfer time incurred while traveling between nodes i and j

N_k = number of buses operating on route k; $N_k = f_k T_k$

f_k = frequency of buses operating on route k

f_{min} = minimum frequency of buses operating on any route

T_k = round trip time of route k

W = fleet size available for operation on the route network

LF_k = load factor of route k

$(Q_k)_{max}$ = maximum flow occurring on any link of route k

CAP = seating capacity of buses operating on the network's routes

SR = set of transit routes

c_1, c_2 = weights reflecting the relative importance of the two cost components

Note that by varying c_1 and c_2, one can generate different non-dominated configurations that achieve a different trade-off between user costs on one hand and operator costs (number of buses here) on the other. In practice, other important trade-offs need to be addressed in what is inherently a multiobjective problem. For example, should all demand be served, implying that some resources are allocated to low-density routes, or should service be more concentrated to provide high service levels on more productive routes, even if that means leaving some demand unmet? Similarly, a trade-off needs to be made between directness of service (no transfers) and coverage. Such considerations have not typically been included in mathematical programming formulations, explaining perhaps the generally low degree of acceptance of such formulations in practice.

The next section discusses the sources of complexity of the TNDP and the past solution approaches in addition to their shortcomings. Section 3 describes the role as well as the potential of AI search techniques to produce better and more efficient solutions. Section 4 presents the AI-based solution approach while sections 5, 6, and 7 discuss the three major components of the solution approach, namely, the lisp-implemented route generation design algorithm (RGA), the analysis procedure TRUST (Transit Route Analyst), and the route improvement algorithm (RIA). Section 8 presents an illustrative application to a small transit network reported by a previous author (Mandl, 1979). Section 9 presents some concluding remarks and future directions for research.

Sources of Complexity of the TNDP and Past Approaches

There are five main sources of complexity that generally preclude finding a unique optimal solution for the TNDP. The first relates to problem formulation: the difficulty of defining the decision variables and thereby expressing the components of the objective function. While the frequencies of buses do appear in the TNDP formulation, the number of routes and their nodal composition do not.

The second source of complexity results from the non-linearities and non-convexities exhibited by the TNDP formulations. Non-convexities are illustrated by the fact that a transit designer can deploy more buses in the transit network (thereby increasing the operator's costs) and still obtain a

higher total travel time (worse user costs). As pointed by Newell (1979), concavity is induced by the waiting time which occurs at entrance to the system or at transfer points. It is not a cost associated with the links of the transit network.

A third source is the combinatorial explosion arising from the discrete nature of the route design problem, and making the TNDP NP-hard. The complexity of the problem grows exponentially with the size of the transit network.

The fourth source results from the multiobjective nature of the TNDP. Most past approaches have considered reducing user costs and/or operator costs as their sole objective. In practice, important trade-offs among other conflicting objectives need to be addressed in what is inherently a multiobjective problem. The total demand satisfied and its components (the total demand satisfied directly, via one transfer, via two transfers, or unsatisfied) are examined against the total travel time and its components (the total travel time that is in-vehicle, waiting, or transferring) and against the fleet size required to operate the transit system.

The last source of complexity relates to the spatial layout of routes. It is difficult to formally characterize and incorporate in a formal procedure what constitutes a 'good' spatial layout of routes. This aspect has been to some extent addressed through design criteria such as route coverage, route duplication, route length, and directness of service (circuity).

The above sources of complexity of the TNDP preclude a formal exact optimization solution to the problem. Past approaches can be classified into three categories: optimization formulations of idealized situations, OR heuristic algorithms, and practical guidelines and ad hoc procedures.

Newell (1979) points out that most optimization formulations deal primarily with the choice of frequency of service on a predetermined route structure rather than with the combined problem of determining both the route structure and the schedule. Whereas the choice of service on an existing route structure is generally a convex optimization problem for which there are rapidly convergent algorithms, the choice of routes is generally a nonconvex (even concave) optimization problem (or an integer programming problem) for which no simple procedure exists short of direct comparison of local maxima. Hence, the selection of an optimal route structure for a network of realistic size is a very large combinatorial optimization problem. As a result, heuristic approaches have been proposed for the TNDP, with no guarantee of superior results.

Most of these heuristic approaches have uncoupled the TNDP's two main components: the routes' configuration and the frequencies of buses operating on these routes. Thus, after an initial 'good' set of routes is found, the corresponding bus frequencies would be determined. In some approaches, that set of routes is generated without considering the demand matrix and is then subjected to improvement. In other cases all route skele-

tons starting and ending with predetermined termini nodes are evaluated according to some heuristic function. The best skeletons would then be transformed to routes through some node selection and insertion strategy.

In addition to the OR heuristic approaches, there are guidelines and ad hoc procedures reflecting the professional judgement and practical experience of operations planners in the transit industry. NCHRP69 (1980) suggested service planning guidelines (in the form of rules of thumb) that include: service area and route coverage, route structure and spacing, route directness-simplicity, route length, and route duplication. Other guidelines with regard to service levels include: desirable minimum service frequencies and loading standards. These guidelines were based on interviews with transit agencies over a broad spectrum of U.S. and Canadian cities and emphasize practice rather than theory and short-range rather than long-range transit planning.

The principal shortcomings of the previous OR approaches include:

1) These approaches addressed a single-objective problem rather than the actual multiobjective one. Additional measures of service quality need to be evaluated, such as the various components of the total travel time, namely, in-vehicle time, waiting time, and transfer time. In addition, the percentages of the total demand that are satisfied directly, with 1 transfer, with 2 transfers, or unsatisfied (such is the case of demand originating or terminating at isolated nodes or demand that requires more than a pre-specified maximum number of transfers to be assigned), are important.

2) Most approaches did not rely sufficiently on the transit demand matrix for guidance in route layout. Different demand patterns normally require different route configurations; for example, where the demand matrix exhibits a radial pattern (the case where one row or column of the demand matrix dominates all other rows or columns respectively) the route configuration is expected to be radial too.

3) Most approaches did not seek to incorporate the professional judgement and practical experience of operations planners in the transit industry. This may explain why most approaches to the TNDP do not appear to have made much of an impact on practice.

4) Most approaches relied on procedures which may have ignored essential aspects of the problem. For example, procedures that computed the total travel time focused primarily on the total in-vehicle component without proper consideration of the waiting and transfer times involved. Similarly, assignment procedures relied on transit route choice models that did not consider important criteria, such as the number of transfers necessary to reach a trip's destination and the total travel time incurred on different alternative choices.

The practical guidelines and ad hoc procedures do not suffice on their own to produce good quality solutions to the TNDP, but their proper incorporation into the AI search techniques (along with the designer's own

knowledge of the transit network) would result in more acceptable and operationally implementable sets of routes. This leads us to the role of AI in the solution approach that we propose to the TNDP.

The Role of AI

To produce good solutions to the TNDP, it would be useful to incorporate the knowledge of experts as well as the judgement of experienced route planners-decision makers into the heuristic search algorithms. Knowledge could also be useful in reaching good solutions in a reasonable amount of time. Efficient solutions result from implementing knowledge to reduce the search space and render it computationally tractable. Sections 5 and 7 discuss in greater detail where knowledge can be deployed in the route generation design algorithm and the route improvement procedures, respectively.

Other advantages that may result from utilizing AI search techniques pertain to implementation benefits accrued as a result of representing the problem and carrying out search efficiently using the 'list' data structure of Lisp. Lisp is a fifth generation computer language that takes its name from **List Programming** (see Winston and Horn (1989) for a good reference on Lisp). The principal motivation for using Lisp (or, more generally, a fifth generation language) lies in the nature of the computational activity taking place in our solution approach, which consists of searching and screening paths in a graph. It has been common wisdom in transportation network applications to avoid any form of path enumeration. Thus, most existing assignment procedures are limited to shortest path constructs. However, other programming paradigms, and advances in computing hardware and software, can greatly facilitate some degree of path search and enumeration, which is justified by the added realism that it could allow into the resulting procedure. The present Lisp implementation is an attempt to explore these possibilities in the transit network design and analysis areas.

Lisp offers advantages over "conventional" languages such as Fortran, C, or Pascal both in terms of representation and search. The transit network data representation lends itself conveniently to the 'list' data structure representation of Lisp, which in turn supports the kind of path search strategies of interest in this application. This can be illustrated by the following:

a) The network connectivity can be conveniently represented in a descriptive language such as Lisp: to each network node, one associates a set (or, in Lisp, a list) of neighboring nodes as well as the trip time (cost) associated with the nodes. Thus, the list (2 ((1 11.4) (3 2.9)(6 8.0))) indicates that one can travel from node 2 to node 1 in 11.4 minutes, to node 3 in 2.9 minutes, and to node 6 in 8 minutes.

b) A route can be represented as a list of nodes. Thus route r25 is defined by the list of nodes (18 11 10 9 8 12 14).

c) The search techniques that are specific to the transit network design problem can be readily programmed in Lisp. In such techniques, a feasible path connecting two network nodes can be easily represented as a list. Thus, the list ((rl 9 16)(r8 16 21)) implies that one can travel from node 9 to node 21 by boarding route rl from node 9 to node 16 and route r8 from node 16 to node 21 (i.e. node 16 is a transfer node).

Taylor (1989) describes simple programs written in Prolog, another fifth generation language, to solve different route selection problems. His examples underscore the brevity of code as well as the relative ease of programming with fifth generation languages. At the basis of these programs are some general 'predicates' (Prolog meta-statements) that test for set membership, generate the intersection or union of any two lists as well as the compliment of one list in another, sort a list of objects according to some numerical property, or append a new element to a set. Such meta-statements define the necessary condition for the required solution, thus isolating the programmer from worrying about the elemental computing and house-keeping chores, as would be the case with conventional programming languages such as Fortran, Pascal, and C.

On the negative side, Lisp, like most higher-level languages, may experience relatively slower computational performance when it comes to mathematical computations (as opposed to symbolic manipulations). However, tests with our solution approach to date have shown reasonable execution times. Our attitude is that activities that are most efficiently and effectively handled by conventional languages (such as bookkeeping) would be programmed as such, while items such as the experts' design rules or path search strategies would be conveniently expressed in a symbolic language such as Lisp. Such use of multiple languages communicating in the execution of a particular program is an increasingly appealing approach to combine the advantages of Artificial Intelligence tools and standard scientific computing for the development of effective design procedures for engineering problems.

An AI-Based Solution Approach

The TNDP has procedural computational aspects, but also presents an opportunity to benefit from AI tools. We propose a hybrid solution approach that provides a framework to incorporate the knowledge and expertise of transit network planners and implements efficient search techniques using AI tools, algorithmic procedures developed by others (even if implemented in Lisp), and modules for tools implemented in conventional languages. There are three major components in the AI-based solution ap-

proach: a route generation design algorithm (RGA) that generates different sets of routes corresponding to different trade-offs; an analysis procedure (TRUST) that computes a whole array of network-level, route-level, and node-level descriptors as well as the frequencies of buses necessary on all routes to maintain their load factors under a prespecified maximum value; and a route improvement algorithm (RIA) that considers each set of routes and utilizes the result of the analysis package to generate an improved set of routes. Sections 5 and 6 discuss the first two components of the solution approach, while the third component is presented in section 7.

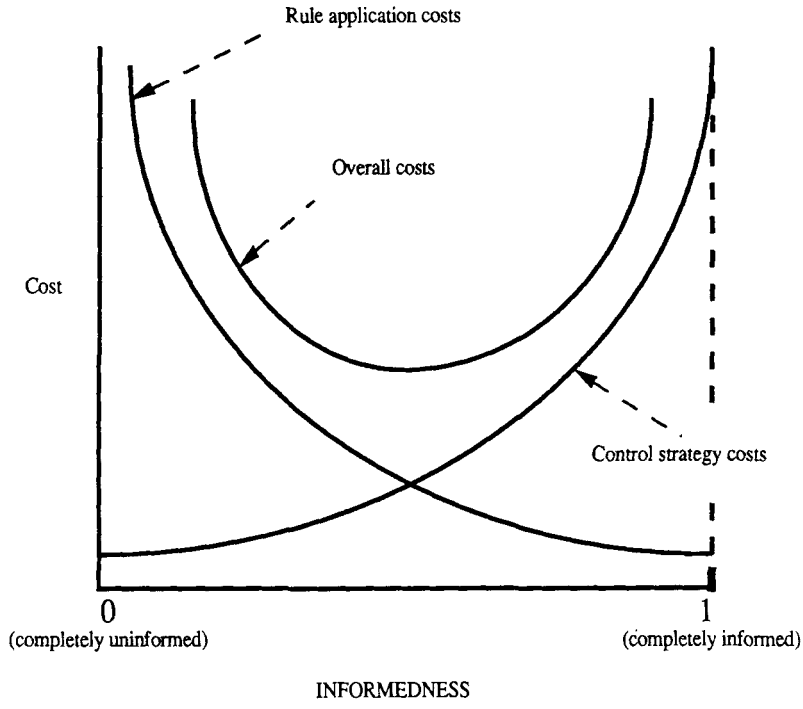


Figure 1: Computational Costs of an AI Production System

Fig. 1 (Nilsson, 1980) presents the principal computational trade-offs in the design of search techniques. The computational costs of any search algorithm can be separated into two major components: the rule application costs and control costs. A completely uninformed control system is characterized by a small control strategy cost because arbitrary rule selection need not depend on costly computations. However, such a strategy results in high rule application costs because it generally needs to try a large

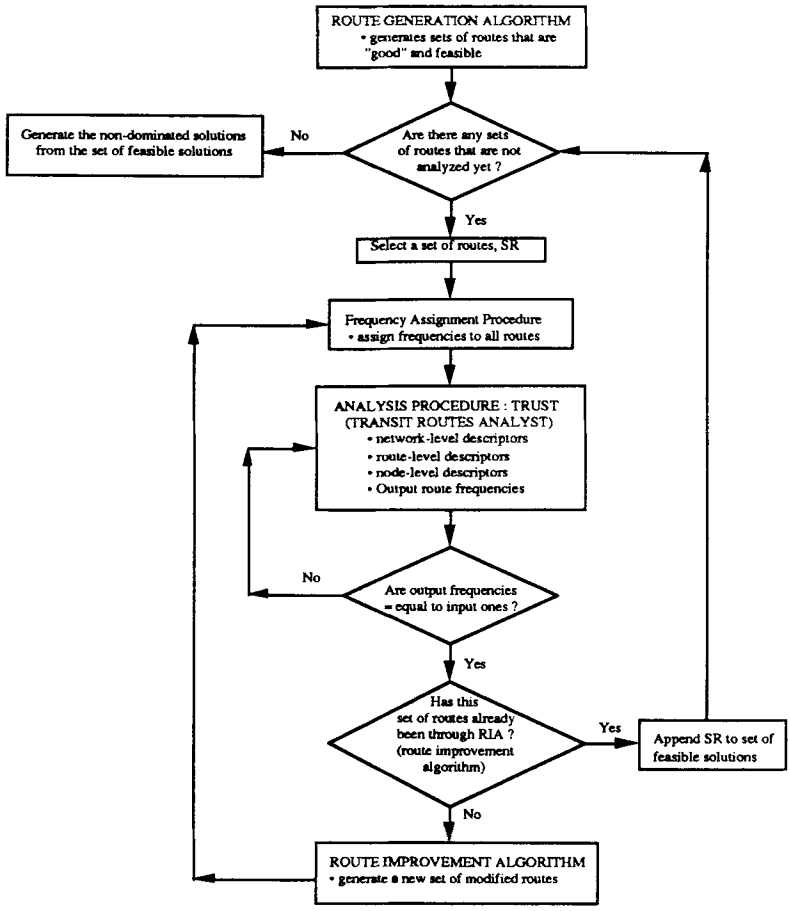


Figure 2: Solution Approach

number of rules to find a solution. To inform a control system completely about the problem typically involves a high-cost control strategy, in terms of the storage and computations required. However, such a completely-informed control strategy results in minimal rule application costs, for they guide the search directly to a solution.

The efficiency of an AI search technique is thus directly tied to achieving a proper balance between both computational cost components. This in turn relies on the 'informedness' or the amount of knowledge and information that the rule-selecting computations possess about the problem at hand. Optimum search efficiency is usually obtained from less than completely informed control strategies. Thus, all three major components of our solution approach shown in Fig. 2 constitute a 'testing ground' for selective application of knowledge.

Route Generation Algorithm (RGA)

This is a design algorithm that is: 1) heavily guided by the demand matrix, 2) allows the designer's knowledge to be implemented so as to reduce the search space, and 3) generates different sets of routes corresponding to different trade-offs among conflicting objectives. Fig. 3 shows the flow chart of RGA.

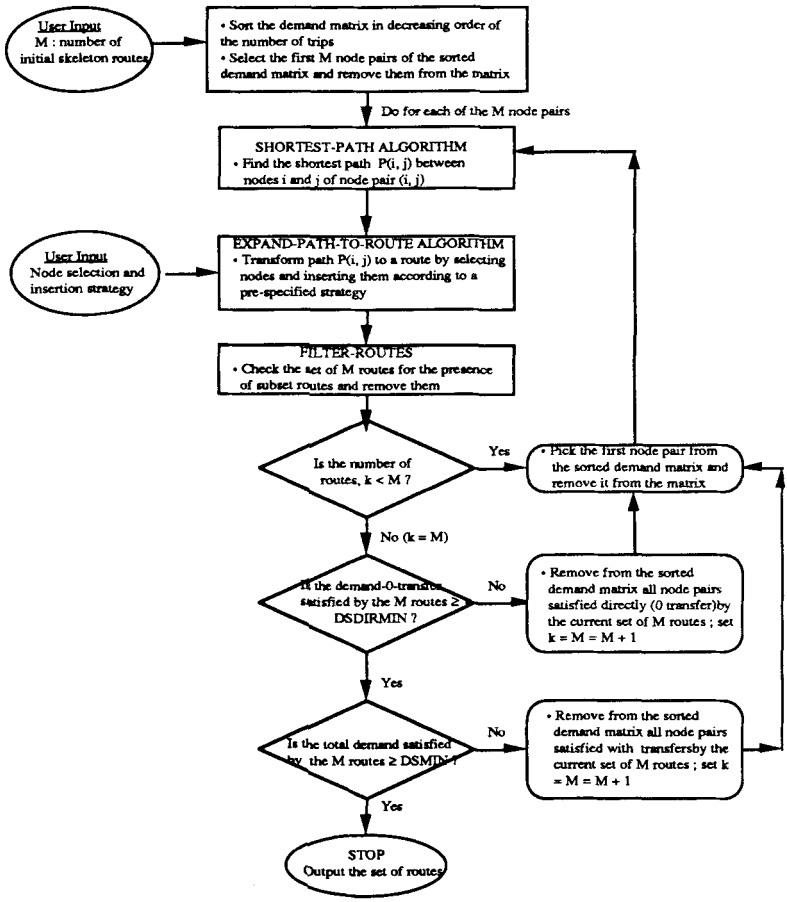


Figure 3: Route Generation Algorithm

RGA starts by sorting the demand matrix in decreasing order of the number of trips and selects the first M node pairs of the sorted demand matrix. The idea is to connect each of these M high demand node pairs

along either the shortest path or the next shortest path whose nodal composition is substantially different from that of the shortest path. A separate algorithm generates the shortest path and an alternate short path for each node pair. This task is performed only once, since the shortest paths utilize only the in-vehicle travel time link data and are therefore not affected by the transit route configuration.

RGA reads the shortest path of each of the M highest demand node pairs to produce M skeletons. Following a selected order of expansion each skeleton is transformed to a route via a chosen node selection and insertion strategy. Different node selection and insertion strategies result in different sets of routes. There are three strategies considered by RGA: 1) the maximum demand insertion strategy, 2) the maximum demand per minimum time insertion strategy, and 3) the maximum demand per minimum route length increase insertion strategy. The initial set of M routes is then examined to determine whether one route is substantially represented by one or more other routes. In the presently implemented version of the procedure, each route is examined to check whether it is completely contained in (i.e. is a subset of) another route. However, this condition may be relaxed to a check on the ratio of the number of nodes of a given route that are traversed by some other route to the route's total number of nodes. If that route is substantially represented by one or more other routes, then it may be discarded.

If there are no overlapping routes, RGA proceeds to the next test. Otherwise, those subset routes are dropped and RGA eliminates from the sorted demand matrix all node pairs whose demand is satisfied directly by the current set of routes. RGA reiterates by picking the first node pair in the remaining (and still sorted) demand matrix and selects one of its two short paths. The skeletons corresponding to the chosen short path are then expanded (in a selected order of expansion) to routes according to the applicable node selection and insertion strategy. Such iteration continues until M 'independent' routes are generated.

The next test is to check whether these M routes collectively satisfy *dsdirmin*, the prespecified minimum percentage of the total demand that is to be satisfied directly, i.e. without transfers. If they do not, then the node pairs satisfied directly on these M routes are eliminated from the sorted demand matrix and the resulting matrix's highest demand node pair is utilized to generate an additional route to the existing set of M routes. The demand that can be completely satisfied in connection with this additional route is computed and added to the demand which is satisfied by the previous set of routes. This iteration continues adding routes one by one until enough routes, joining high demand node pairs and meeting the minimum percentage of the total demand that has to be satisfied without transfers, are identified. It is to be noted that varying the minimum percentage of the demand satisfied directly results in different sets of routes that enable the

designer to address trade-offs between directness of service and coverage.

Once the previous iteration is completed, RGA proceeds to the third test. The demand satisfied with 1 transfer in the existing set of routes is computed and added to the demand satisfied directly. The approximated total satisfied demand is checked against *dsmin*, the prespecified minimum percentage of the total demand that has to be satisfied. Here, again, different values of the latter result in different sets of routes that enable the designer to address different trade-offs between the demand satisfied and coverage. If the total demand satisfied by the current set of routes exceeds the minimum total demand to be satisfied, then output the set of routes, otherwise, remove from the demand matrix all node pairs whose demand is satisfied with 1 transfer. Select the node pair with the highest demand from the sorted demand matrix and develop its routes. Continue this third set of iterations by adding routes one by one until a resulting set of routes that satisfies the third test is found. It is assumed that the total demand satisfied is approximated as the sum of the demand satisfied directly and the demand satisfied with 1 transfer. The demand satisfied with 2 transfers is not computed because RGA is primarily a design procedure and not a detailed analysis tool. Such an analysis tool (TRUST) assigns the demand to the transit network and computes the corresponding user and operator costs as well as the pertinent measures of system performance. It is described in greater detail in the next section.

The designer's knowledge can be implemented in several locations within RGA both to produce better quality solutions and to reduce the search space, thus improving the efficiency of search. The search space is affected by three basic factors: 1) the number (M), nodal specification, and order of expansion of the initial skeletons, 2) the strategy for node selection and insertion, and 3) the identification of termini nodes.

RGA's input data may be grouped under 5 categories:

1) Network: The number of bus transit nodes, the connectivity list specifying for each node its accessible neighboring nodes as well as the in-vehicle travel time (i.e. over the road network) to each of the neighboring nodes, the two different shortest paths for each node pair, the number of initial skeletons, and the set of terminal nodes.

2) Frequencies: The maximum frequency that can be allowed on any route. This is used to provide a route capacity constraint for the expansion routines.

3) Demand: A symmetric demand matrix representing the demand between each pair of nodes (the symmetry requirement is not essential, but has been used for convenience), *dsdirmin* (the minimum percentage of the total demand that has to be satisfied directly by the initial expanded skeletons), and *dsmin* (the minimum percentage of the total demand that has to be satisfied by the output set of routes).

4) Parameters: the transfer time (penalty) per transfer expressed in equivalent minutes of in-vehicle travel time, the bus seating capacity (assumed the same on all buses), the maximum load factor allowed by the planner on any transit route, the node-sharing factor *nsf* necessary to determine whether a node can be inserted or not, and the direct-capacity factor necessary to determine whether a route can be still expandable.

5) Insertion Rules and Cost Measures: The node selection and insertion heuristic and the minimum acceptable value for the demand satisfied per its cost of insertion (for each heuristic) that has to be exceeded by a candidate node to meet an economic feasibility constraint for insertion.

The Analysis Procedure Trust (Transit Route Analyst)

Once RGA generates different sets of routes, each set of routes is analyzed via TRUST. This is a procedure for the analysis and evaluation of alternative transit route network configurations consisting of a set of routes and associated frequencies. Its main function is to assign known demands between origin-destination pairs to the transit network, and compute a variety of performance measures reflecting the quality of service and costs experienced by the users, as well as the resources required by the operator. TRUST differs from existing assignment approaches in several respects, including: 1) the path choice mechanism that is the basis of the assignment procedure; 2) the use of Lisp, a so-called fifth-generation language associated with artificial intelligence applications, which greatly facilitates the implementation of the path search and enumeration inherent in the type of assignment procedure adopted; and 3) the broader range of performance measures and descriptors that are computed and displayed, particularly on the demand side. Fig. 4 shows the flow chart of TRUST.

The following five types of information are calculated:

1) The total travel time experienced by users in the network, and the respective percentages of in-vehicle travel time, waiting time, and transfer time (the latter reflecting a pre-specified time penalty considered to be equivalent to a transfer).

2) the total number of demand trips as well as the percentages of demand that are unsatisfied, or satisfied with 0, 1, or 2 transfers.

3) the number of trips originating at each node that could not be assigned, as well as the number of passenger trips that transfer at each node.

4) the link flows on each route's links and the route's maximum load factor.

5) the frequency and number of buses required on each route to maintain its load factor under LF_{max} and the resulting number of buses required for the whole network.

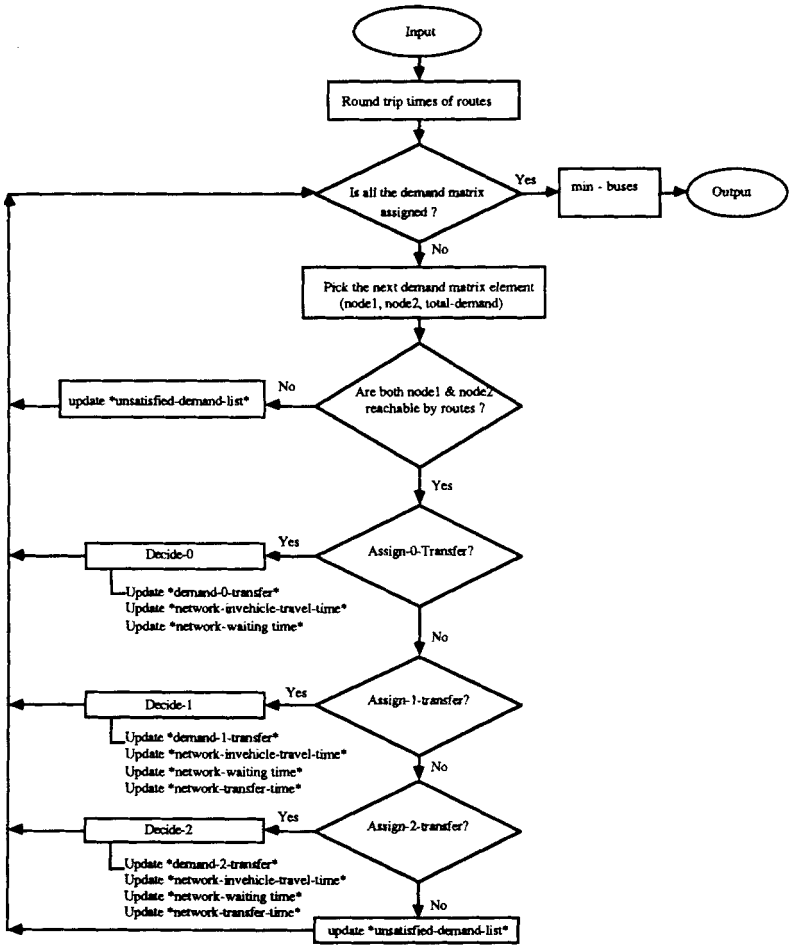


Figure 4: Flow Chart Illustration of TRUST (Transit Route Analyst)

If, for a given analysis run of a particular configuration, the output frequencies (i.e. the frequencies required to maintain all routes' load factors under a prespecified LF_{max}) are considered to be quite different from the input frequencies (say they differ by more than 5%), then the planner may reiterate the process by using the output frequencies of the previous run as the new input ones. Usually one iteration is sufficient to result in output frequencies that do not differ much (say by more than 10%) from input ones. In a separate paper (Baaj and Mahmassani, 1990) we present a detailed description of TRUST, including its application to the transit network of the city of Austin, Texas (metropolitan population approx. 500,000) with some simplifying assumptions.

A central component in the analysis of a given route configuration is a procedure to evaluate the objective function components (in math programming formulations) and other measures of effectiveness or service quality that are of concern to the operator. This requires the assignment of the passenger trip demand matrix to the set of routes that define any particular network configuration. As indicated by Speiss and Florian (1989), the transit assignment problem has been studied by several authors in the past, either as a separate problem (e.g. see Dial, 1967; Rapp et al, 1976) or as a subproblem of more complex models, such as transit network design (see e.g. Lampkin and Saalmans, 1967; Mandl, 1979; Hasselstrom, 1981), or multimodal network equilibrium (Florian and Speiss, 1981).

Dial's assignment is based on the least cost path. Lampkin and Saalmans assign a fraction of passengers to a path equalling the probability that the path's bus arrives before other buses. Mandl assumes that all the routes have buses operating with the same frequency. This implies that all passengers experience the same waiting time. Thus his assignment procedure is unrealistic. Hasselstrom regards the problem of estimating passenger paths as either one of eliminating links from a fine meshed network or one of reducing flow concentration by adding possible links to the "minimal spanning tree".

Most transit assignment algorithms are variants of procedures used for private car traffic on road networks (such as shortest path, stochastic multipath assignment) that are modified to reflect the waiting time phenomenon inherent to transit networks. However, path choice in a transit network, especially in large urban areas with overlapping routes, may not be well described by the assumptions of auto driver assignment. This was recognized by the transit route choice behavior model presented by Han and Wilson (1982), which allowed for a lexicographic strategy in the choice among competing routes, with transfer avoidance and/or minimization acting as the primary choice criterion. The main feature of their model (adopted by TRUST) is the consideration of the number of transfers as the most important criterion, and one which exercises preemptive priority over other considerations.

TRUST's transit route choice model considers two main criteria: the number of transfers necessary to reach the trip's destination and the trip times incurred on different alternative choices. The tripmaker is assumed to always attempt to reach his/her destination by following the path that involves the fewest possible number of transfers. In case of a tie, i.e., when there is more than one path with the same least number of transfers, additional criteria come to play. It is in the details of this aspect that our procedure differs from Han and Wilson's. When more than one path exists with the same number of (or no) transfers, the tripmaker's effective choice set is assumed to contain only those paths with respective travel times within a particular range. Where there is one or more alternatives whose

trip time is within a threshold of the minimum trip time, a "frequency-share" rule is applied. This is an allocation formula that reflects the relative frequencies of service on the alternative paths.

TRUST first starts by calling procedure "Assign-0-transfer?" to identify all paths, between i and j, that do not involve any transfers. Only if none are found will paths involving one (and only one transfer) be considered via procedure "Assign- 1 -transfer?". If it is not possible to assign the demand via 1-transfer paths, then "Assign-2-transfer?" determines whether or not there are paths involving two transfers. If these exist, then the demand is assigned in a manner similar to that of the 1-transfer case. Otherwise, if the demand cannot be assigned along paths involving a maximum of 2 transfers, then such demand is classified as unsatisfied. Hence, TRUST refrains from searching for paths that reach their destinations with 3 or more transfers, thereby circumventing an otherwise considerable amount of meaningless search and keeping its execution time within tolerable limits. It is assumed that a passenger will simply not consider (unless forced to, and this case represents very few trips) boarding the transit buses to accomplish a trip that requires 3 or more transfers .

In summary, TRUST checks each node pair (one element of the demand matrix) to determine whether its trips can be assigned with zero, one or two transfers. It appends the list ((i j) d_{ij}) to one of four possible demand lists: *UNSATISFIED-DEMAND-LIST*, *DEMAND-0-TRANSFER*, *DEMAND- 1 -TRANSFER*, and *DEMAND-2-TRANSFERS*. It also computes the contributions of this assignment to the network-wide measures of user costs: *NETWORK-INVEHICLE-TRAVEL-TIME*, *NETWORK-WAITING-TIME*, and *NETWORK-TRANSFER-TIME*. Once all elements of the demand matrix are assigned, the *NETWORK-TOTAL-TRAVEL-TIME* is calculated as the sum of the final values of the above three components. The program also considers each of the four demand lists above to compute the percentages of the total number of passenger trips that are unsatisfied, or satisfied via 0, 1, or 2 transfers. The program output includes each of the four demand lists, so that one can identify how the demand associated with a selected pair of nodes was assigned, as well as the final "list-of-link-flows" associated with each transit route. This information provides the principal measures of service quality and user costs that are of interest in the evaluation of a particular transit route network configuration. The other type of measures consists of the resources required by the operator, primarily the number of buses necessary to serve that particular configuration.

The Route Improvement Algorithm (RIA)

The route improvement algorithm (RIA) operates on the set of routes generated by the route generation algorithm. It identifies and checks for

possible improvement modifications which can be grouped into actions on the system coverage level and actions on the route structure level. The main action on the system coverage level is the discontinuation of service on routes that suffer from low ridership or are too short or both. The actions on the route structure level include joining of low ridership routes with other medium to high ridership routes, splitting of routes where desirable, and devising new combinations of routes through branch exchange in such a way that the total number of passengers transferring at their intersection nodes is reduced. Figure 5 shows the flow chart of RIA for one possible sequence of application of the improvement modules.

In this particular sequence of application of the improvement modules, RIA executes the service discontinuation procedure operating on the low ridership routes and then calls on TRUST to analyze the resulting modified transit network. If the transit planner determines that no further improvement is necessary, then RIA quits. Otherwise, RIA proceeds to execute the route splitting procedure followed by the branch exchange procedure to reduce transfers. A final call to TRUST is made to determine the characteristics of the resulting network, so that the transit planner can compare them with those of the set of routes obtained by discontinuing service on low ridership routes.

In parallel to these actions, RIA executes the route joining procedure on the original set of routes generated by RGA. If all the low ridership routes could be joined with medium to high ridership ones, then RIA proceeds with the resulting modified set of routes and calls the route splitting procedure followed by the branch exchange procedure. If not all routes could be joined, RIA calls an insertion procedure to insert nodes belonging to these low ridership routes (but not to remaining medium to high ridership routes) in the medium to high ridership routes and proceeds from there. A final call to TRUST again is made to generate a detailed analysis of the modified set of routes. Alternatively, the improvements can be applied individually, and in any order desired by the user. One focus of current research is to expand the system's capabilities through the addition of other improvement modules, allowing the user to specify which improvement modules to apply as well as the desired order of application.

Illustrative Application: Mandl's Benchmark Network

In this section we demonstrate the performance of our solution framework on a transit network abstracted from the Swiss context, and used as a case study by a previous author (Mandl, 1979). We present the Swiss network, Mandl's proposed solutions, and compare them with solutions generated by our approach.

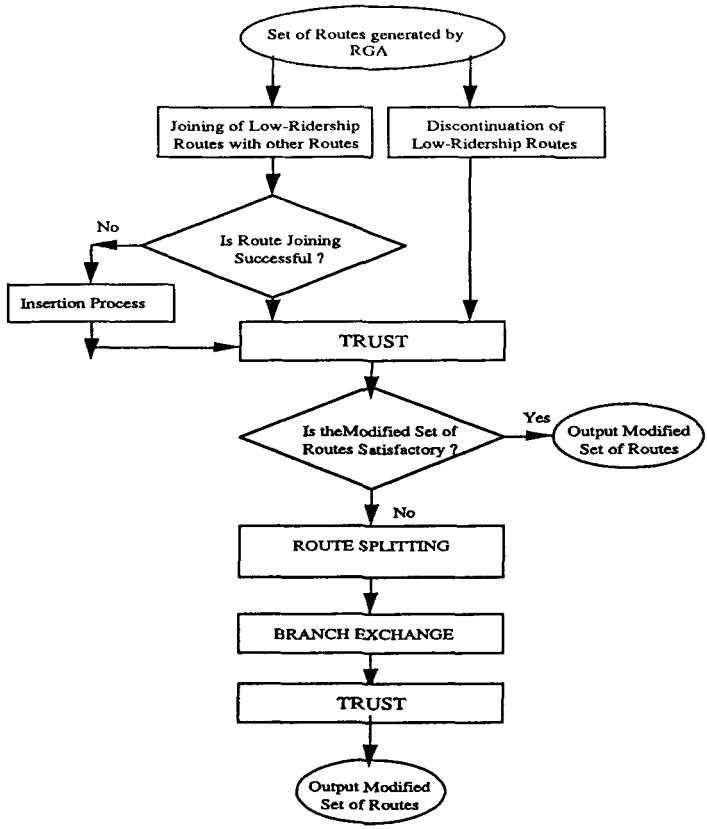
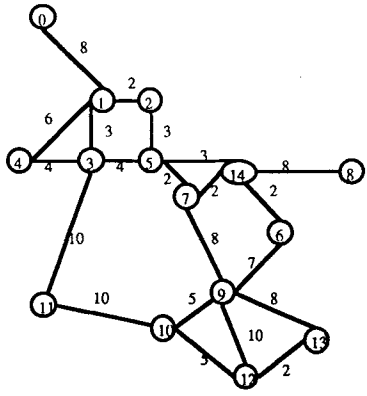


Figure 5: The Route Improvement Algorithm (RIA)

Figure 6 shows the data used for the case study. The arcs' in-vehicle travel times are shown in minutes. The transit demand matrix shown indicates the average number of passenger trips per day for each transit node pair. The initial set of routes (before improvement) generated by Mandl consisted of three routes: (10121397145210), (6145346), and (1135148). After his improvement algorithm is executed, the final set consisted of four routes: (0125791012), (4357146), (1135148), and (12139). This set of routes satisfied 100% of the total demand. Hence, RGA was executed with the minimum total demand satisfied set at 100%. Three runs were performed: in the first run, the minimum percentage of total demand satisfied directly was set at 50%, the shortest paths were used for the initial layout of skeletons, and the maximum demand insertion heuristic was employed as the node selection and insertion strategy. In the second run, the alternate short paths were used where feasible and the maximum demand



Transit Demand Matrix:

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	400	200	60	80	150	75	75	30	160	30	25	35	0	0
1	400	0	50	120	20	180	90	90	15	130	20	10	10	5	0
2	200	50	0	40	60	180	90	90	15	45	20	10	10	5	0
3	60	120	40	0	50	100	50	50	15	240	40	25	10	5	0
4	80	20	60	50	0	50	25	25	10	120	20	15	5	0	0
5	150	180	180	100	50	0	100	100	30	880	60	15	15	10	0
6	75	90	90	50	25	100	0	50	150	440	35	10	10	5	0
7	75	90	90	50	25	100	50	0	15	440	35	10	10	5	0
8	30	15	15	15	10	30	15	15	0	140	20	5	0	0	0
9	160	130	45	240	120	880	440	440	140	0	600	250	500	200	0
10	30	20	20	40	20	60	35	35	20	600	0	75	95	15	0
11	25	10	10	25	15	15	10	10	5	250	75	0	70	0	0
12	35	10	10	10	5	15	10	10	0	500	95	70	0	45	0
13	0	5	5	5	0	10	5	5	0	200	15	0	45	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6: Mandl's Network and Transit Demand Matrix

per minimum time insertion strategy was employed. In the third run, the shortest paths were used and the maximum demand heuristic was employed; however, the minimum percentage of total demand satisfied directly was increased to 70%. TRUST was used to analyze both sets of routes suggested by Mandl. In all runs, the bus seating capacity was selected as 40, the transfer penalty was set at 5 minutes of in-vehicle travel time, and a bus load factor of 1.25 was chosen. The results of the three design runs as well as the analysis of Mandl's networks are listed in Table 1.

All three solutions generated by RGA (even before improvement) dominated the final solution (after improvement) proposed by Mandl with the exception of the total waiting time component. From the user perspective, all three solutions had a higher percentage of the total demand satisfied directly (78.61, 79.96, and 80.99%) than Mandl's solution (69.94%). Consequently, the fraction of passengers transferring was less in the three

	% Directly % D-0t	%-1 Transfer % D-1t	%-2 Transfers % D-2t	Demand Un-satisfied % D-U	Total Travel Time TTT	In-Vehicle Time IVTT	Waiting Time WT	Transfer Time TT	# of Buses	CPU, Sec
SR1	78.61	21.39	0	0	205656	168076	20930	16650	89.3	2.65
SR2	79.96	20.04	0	0	210632	169101	25931	15600	76.9	3.08
SR3	80.99	19.01	0	0	217869	180350	27719	14800	82.2	2.13
Mandl Initial	68.85	31.15	0	0	250364	210920	15194	24250	116.0	--
Mandl Final	69.94	29.93	0.13	0	219044	177400	18144	23500	99.3	--

(IVTT) Theoretical Minimum = 155690

Table 1: Results of Testing RGA on Mandl's Network

runs (21.39, 20.04, and 19.04%) than in Mandl's (29.93%). From the transit operator perspective, all RGA's three runs required a smaller fleet size than that of Mandl's proposed network. The fleet size required varied from a maximum of 90% of that proposed by Mandl to a minimum of 75%. The levels of service were better in all three runs. The total travel time was at worst slightly less than that of Mandl (217869 vs. 219044) and in the best solution was 6.1% lower. The minimum theoretical possible value for the Swiss network's in-vehicle travel time was computed at 155690 (the case when all node pair demands are assigned to their shortest in-vehicle times paths). The first two runs were closer to that theoretical minimum than Mandl's, while the third solution was ~2% worse. Mandl's solutions had total waiting times that were better than those of RGA's three sets of routes, because the frequencies of service on Mandl's routes were higher. Overall, in comparison with Mandl's solution, RGA's second solution satisfied 14% more passengers directly with 25% fewer buses and 5% smaller total travel time.

The application to the above benchmark problem provides an example where RGA can generate better solutions than those proposed by Mandl's approach, which is one of the better procedures available for the TNDP. It also shows that different sets of routes usually result from different initial skeleton layouts as well as from different node selection and insertion strategies, hence the need to investigate in detail the performance of RGA under different starting solutions and different node selection and insertion strategies. The above network is not particularly suitable for this task because of its small size (only 15 nodes). Real world networks tend to be larger by an order of magnitude or more (in terms of number of nodes). Constraints in the search strategies are not all tested in a small network, where the search tends to converge to solutions rather rapidly. Such constraints as route length, route circuity, and route duplication may not affect search in small networks as prominently as in medium to large networks. Another important note concerns the nature of the demand matrix. In Mandl's network, 82% of the node pairs had non-zero demands. In more typical problems, transit demand matrices tend to be sparse, with fewer non-zero entries.

Concluding Remarks

An AI-based solution approach to the transit network design problem was presented, including a description of its three major components: 1) a route generation design algorithm (RGA), 2) an analysis procedure TRUST (Transit route analyst), and 3) a route improvement algorithm (RIA). The approach provides a framework for continuing development and enhancement through the inclusion of additional knowledge and al-

native algorithmic constructs. Ongoing and future research concentrates on continuing the development of additional route improvement modules, and on performing extensive computational testing of the overall procedure as well as of the individual components on networks reported in the literature as well as on those pertaining to real cities. The implementation of emerging and evolving knowledge into improved AI search techniques will contribute to providing good quality solutions to this complex problem with reasonable computational time and effort.

References

Baaj M. H. and Mahmassani H. S. (1990) TRUST: a lisp program for the analysis of transit route configurations, presented at the 69th Annual Meeting of the Transportation Research Board, Washington, D.C. (1990), and to be published in *Transportation Research Record* (1992).

Ceder A. and Wilson N.H. (1986) Bus network design. Transpn. Res.-B, **Vol. 20B**, No. 4, 331 - 344.

Dial R. B. (1967) Transit pathfinder algorithm. *Highway Res. Rec.* **205**, 67 - 85.

Dubois D., Bell G., and Llibre M. (1979) A set of methods in transportation network synthesis and analysis.J. *Oper. Res. Soc.* **30**, 797 - 808.

Florian M. and Speiss H. (1983) On two mode choice/assignment models. *Transpn. Sci.* **17**, 32 - 47.

Han A. F. and Wilson N. H. M (1982) The allocation of buses in heavily utilized networks with overlapping routes. *Transpn. Res.-B*, **Vol. 16B**, No. 3, 221- 232.

Hasselstrom D. (1981) Public transportation planning— A mathematical programming approach. Ph.D.thesis, Department of Business Administration, University of Gothenburg, Sweden.

Janarthanan N. and Schneider J. B. (1989) Development of an expert system to assist in the interactive graphic transit system design process. *Transportation Research Record* **1187**, PP. 30 - 46, TRB, National Research Council, Washington, D. C..

Lampkin W.and Saalmans P. D. (1967) The design of routes, service frequencies and schedules for a municipal bus undertaking: A case study. *Oper. Res. Quart.* **18**, 375 - 397.

Mandl C. E. (1979) Evaluation and optimization of Urban Public Transportation Networks. Presented at the 3rd European Congress on Operations Research Amsterdam, Netherlands.

Newell G. (1979) Some issues related to the optimal design of bus routes. *Transpn. Sci.* **13**, 20 - 35.

Nilsson Nils J. (1980) *Principles of Artificial Intelligence.* Tioga Publishing Company, Palo Alto, Califomia.

NCHRP69 (1980) Bus Route and Schedule Planning Guidelines. Transportation Research Board, National Research Council, Washington, D.C..

Rapp M. H., Mattenberger P., Piguet S. and Robert-Grandpierre A. (1976) Interactive graphic system for transit route optimization. Transpn. Res. Rec. **619.**

Rea J. C. (1971) Designing urban transit systems: an approach to the route-technology selection problem. PB 204881, University of Washington, Seattle, WA.

Silman L. A., Barzily Z., and Passy U. (1974) Planning the route system for urban buses. *Comp. Oper. Res.* **1**, 201- 211.

Speiss H. and Florian M. (1989) Optimal strategies: a new assignment model for transit networks. *Transpn. Res.-B,* **Vol. 23B**, No. 2, 83 -102.

Taylor M. P. A. (1989) Knowledge-based systems for transport network analysis: a fifth generation perspective on transport network problems. (to be published). Department of Civil Engineering, Monash University, Clayton, Victoria 3168, Australia.

Winston P. H. and Horn B. K. P. (1989) *Lisp* . 3rd edition. Addison-Wesley Publishing Company.