URBAN BUS TRANSIT ROUTE NETWORK DESIGN USING GENETIC ALGORITHM

By S. B. Pattnaik,[1] S. Mohan,[2] and V. M. Tom[3]

(Reviewed by the Urban Transportation Division)

ABSTRACT: Urban bus route network design involves determining a route configuration with a set of transit routes and associated frequencies that achieves the desired objective. This can be formulated as an optimization problem of minimizing the overall cost (both the user's and the operator's) incurred. In this paper, the use of genetic algorithms (GAs), a search and optimization method based on natural genetics and selection, in solving the route network design problem is reported. The design is done in two phases. First, a set of candidate routes competing for the optimum solution is generated. Second, the optimum set is selected using a GA. The GA is solved by adopting the usual fixed string length coding scheme along with a new variable string length coding proposed in this study. The former assumes a solution route set size, and tries to find that many best routes from the candidate route set, using a GA. The route set size is varied iteratively to find the optimum solution. In the newly proposed variable string length coding method, the solution route set size and the set of solution routes are found simultaneously. The model is applied to a case study network, and results are presented.

INTRODUCTION

A great emphasis needs to be given to the urban public transport system, as the future demand for transport can be met only by public transport systems, especially in developing countries like India. Introduction of new modes and mass transit systems leads to bright prospects, but demands considerable infrastructure development and economic burden. Very often, bus transit continues to take the major share of public transport demand. The optimal design of the route network, essential for the quality of service offered, is subject to constraints. Also, the route network is the input for studies concerning the time tables, schedules, and regularity. Urban bus route network design involves selecting the routes and the associated frequencies. A review of the literature reveals various approaches for routing and scheduling of an urban bus transit system and the computational tools used. A system analysis approach has been attempted in the planning of urban bus networks by means of a transport assignment model (Chua 1984). Analytical models have used simplified networks to derive optimum relations for parameters of public transport system, e.g., headway and route spacing (Kochur and Hendrickson 1982). Models were developed assigning frequencies to a given set of routes by optimizing the travel time plus waiting time (Furth and Wilson 1981). Models were generated with a heuristic search method for determining routes and then assigning frequencies (Dubois et al. 1979). A three-stage procedure comprising trip assignment, routing, and scheduling was also suggested for route network design (Ceder and Wilson 1986). Nes et al. (1986) developed a model for simultaneous selection of routes and their frequencies.

A considerable amount of effort has gone into developing both analytical and numerical optimization techniques for route network design. The computational tools used by researchers for these problems were conventional mathematical programming (Steenbrink 1974). Baaj and Mahmassani (1995) presented a framework of a hybrid artificial intelligence/operations research (AI/OR) solution approach, which combines AI search concepts with vehicle routing heuristics and transit system analysis methods. In airline network design, which is similar to bus route network design, fuzzy logic was used to determine the route candidates to make flights, and fuzzy linear programming was used to determine flight frequencies on the routes (Dusan et al. 1994). The past decade witnessed an increasing interest in biologically motivated approaches like artificial neural networks and genetic algorithms for solving optimization problems. Chakroborty et al. (1995) used genetic algorithms in scheduling problems because of the inadequacy of the existing techniques and reported that genetic algorithm is an efficient tool for similar optimization problems arising in transportation systems.

The study reported in this paper focuses on urban bus route network design using a GA, which is a high-level simulation of a biologically motivated adaptive system, as the tool for selecting the optimal route network. Two methods based on the coding scheme are proposed for solving the problem by a genetic algorithm, namely, fixed string length coding and variable string length coding. In fixed string length coding, routes are considered as variables and the number of routes considered during genetic operation remains fixed. This demands an iteration for the number of routes to arrive at optimum configuration. But in the newly proposed variable string length coding, the number of routes and the set of routes are selected simultaneously, thus avoiding any iteration for the number of routes.

PROBLEM FORMULATION

Route network design (RND) is the single most important planning step in the urban bus transit planning process (Ceder and Wilson 1986). This is because the route structure designed will invariably affect both frequency setting and bus and crew scheduling, and also due to the fact that operators have the least flexibility in altering the routes. The important components of RND are identified as estimating demand, identification of objective function, constraints, passenger behavior, solution techniques, and computation time. Demand may be treated as fixed and independent of the service quality. Most of the models use this approach owing to its simplicity. On the other hand, demand can be a variable; i.e., the share of the

[1]Assoc. Prof., Dept. of Civ. Engrg., Indian Inst. of Technol., Madras, India.

[2]Assoc. Prof., Dept. of Civ. Engrg., Indian Inst. of Technol., Madras, India.

[3]Res. Scholar, Dept. of Civ. Engrg., Indian Inst. of Technol., Madras, India.

Note. Discussion open until January 1, 1999. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on March 10, 1997. This paper is part of the ***Journal of Transportation Engineering***, Vol. 124, No. 4, July/August, 1998. ©ASCE, ISSN 0733-947X/98/0004-0368–0375/$8.00 + $.50 per page. Paper No. 15319.

public transport demand can be estimated from the total trip matrix based on some service quality. The solution of RND is limited by constraints on the total operating cost or fleet size and constraints on passenger loads, which is a function of vehicle size, minimum service frequency, etc. Most of the formulations adopt a multiple path assignment of passengers with the assumption that a passenger will choose the first bus that arrives. This formulation is complex but realistic and accurate, unlike a single path assignment (Ceder and Wilson 1986).

In bus route network design, the primary aim is to determine a route configuration consisting of a set of transit bus routes and associated frequencies that achieves the maximum or minimum value of the desired objective, subject to the constraints. The objective function may be minimization of the overall cost measure covering both operator as well as user cost (Baaj and Mahmassani 1990). The user cost is a function of total travel time, whereas operator cost greatly depends on the bus kilometer required for a particular configuration. Feasibility constraints consist of the minimum and maximum available frequencies, maximum load factor, allowable fleet size, etc. In this study also, the objective function has been considered as minimization of the overall cost. Accordingly, the mathematical formulation of the model can be expressed as follows:

$$\text{Min: } Z = c_1 \sum_{i,j=1}^{n} d_{ij} \cdot t_{ij} + c_2 \sum_{k \in R_{SR}} f_k \cdot t_k \tag{1}$$

subject to the following constraints:

$$f_k \geq f_{\min} \; \forall \; k \in R_{SR} \quad \text{(frequency feasibility)} \tag{2}$$

$$l_k \leq l_{\max} \; \forall \; k \in R_{SR} \quad \text{(load factor constraint)} \tag{3}$$

where d_{ij} = travel demand from node i to j; t_{ij} = total travel time (the sum of in-vehicle travel time, waiting time, and transfer penalty) from node i to j; f_k = frequency of k^{th} route; t_k = round trip time of k^{th} route; $f_{\min}$ = minimum frequency of buses operating on any route; l_k = load factor on route k = $Q_k^{\max}/(f_k \cdot C)$; $l_{\max}$ = maximum allowable load factor; C = bus seating capacity; $Q_k^{\max}$ = maximum flow occurring on any link on route k; R_{SR} = set of routes in the solution; c_1 = factor to convert user travel time to user travel cost; c_2 = factor to convert bus kilometer to cost equivalent; and n = number of nodes in the network. The first term in the equation represents the total travel cost incurred by the user, and the second term represents the total operating cost of all the buses plying in the routes of the solution route set. Hence, the sum of both terms represents the overall travel cost. The preceding model was solved using the genetic algorithm approach and validated with a case study network. The basics of GAs is enumerated in the next section, followed by a discussion on the application of GAs for solving this model.

GENETIC ALGORITHMS

Genetic algorithms are search algorithms that are based on concepts of natural selection and natural genetics (Holland 1992). Genetic algorithms were developed to simulate some of the processes observed in natural evolution, a process that operates on chromosomes (organic devices for encoding the structure of living being). The genetic algorithm method differs from other search methods in that it searches among a population of points and works with a coding of parameters set, rather than the parameter values themselves. The transition scheme of the genetic algorithm is probabilistic, whereas traditional methods use gradient information (Goldberg 1989). Because of these features, genetic algorithms are being used as general purpose optimization algorithms. The working principle of GAs is illustrated in the form of a pseudocode as follows:

```
Randomly initialize the population
repeat
   find objective function value
   find fitness function value
   generate next population using operators
      reproduction
      crossover
      mutation
until convergence
```

The major steps involved are the generation of a population of solutions, finding the objective function and fitness function, and the application of genetic operators. An important characteristic of genetic algorithms is the coding of variables that describe the problem. The most common coding method is binary coding (Goldberg 1989). If the problem has more than one variable, a multivariable coding is constructed by concatenating as many single-variable coding as the number of variables in the problem. GAs process a number of solutions simultaneously. Hence, in the first step, a population having P individuals is generated by a pseudorandom generator whose individuals represent a feasible solution. This is a representation of the solution vector in a solution space and is called initial solution. This ensures the search is robust and unbiased, as it starts from a wide range of points in the solution space. In the next step, individual members of the population are evaluated to find the objective function value.

In the third step, the objective function is mapped onto a fitness function that computes a fitness value for each member of the population. Fitness function is used to allocate reproductive trails to the individuals in the population and thus acts as some measure of goodness to be maximized. This means that individuals with a higher fitness value will have a higher probability of being selected as candidates for further examination. The present work uses the following commonly adopted fitness function $F(i)$ to transform the objective function:

$$F(i) = V - \frac{O(i) \cdot P}{\sum_{i=1}^{P} O(i)} \tag{4}$$

where $O(i)$ = objective function value of ith individual; P = population size; and V = a large value to ensure nonnegative fitness values. The value of V adopted in this work is the maximum value of the second term of (4), so the fitness value corresponding to the maximum value of the objective function is zero. Finally, a new population is generated from the current population by performing three distinct genetic operations, namely, reproduction, crossover, and mutation.

Reproduction (or selection) is an operator that makes more copies of better individuals in a new population. Better individuals should be from the fittest individuals of the previous population; this is achieved by computing the reproduction count associated with each string and then removing or copying the string depending on the values of the reproduction count. Reproduction count is computed based on the fitness value by various methods; the present work uses stochastic sampling without replacement, because it is both superior to other schemes and widely used (Goldberg 1989).

In the crossover operation, a recombination process creates different individuals in the successive generation by combining material from two individuals of the previous generation. A crossover operator is used to recombine two strings to get a better string. Crossover is usually performed with a probability called crossover probability to preserve some of the good strings found previously. Crossover is done at the string level by randomly selecting two strings for crossover operation. Two types of crossover techniques are generally adopted; one-

J. Transp. Eng., 1998, 124(4): 368-375

site crossover and two-site crossover (Bartlett 1995). One-site crossover is more suitable when the string length is small, whereas two-site crossover is suitable for large strings. Hence, the present work adopts a two-site crossover. The underlying objective of crossover is to exchange information between strings to get a string that is possibly better than the parents.

Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It operates at the bit level, when the bits are being copied from the current string to the new string. Mutation operates with a probability, usually a quite small value, called the mutation probability. A coin-toss mechanism is employed; if a random number between 0 and 1 is less than the mutation probability, then the bit is inverted; zero becomes one and one becomes zero.

Application of these operators on the current population creates a new population. This new population is used to generate subsequent populations and so on, yielding solutions that are closer to the optimum solution. The values of the objective function of the individuals of the new population are again determined by decoding the strings. These values express the fitness of the solutions of the new generation. This completes one cycle of the genetic algorithm (called a generation). In each generation, if the solution is improved, it is stored as the best solution. This is repeated till convergence.

GA MODEL FOR ROUTE NETWORK DESIGN

Models for RND are rarely used in practice because of their complex structure and limited accessibility for the planners, as these models can only be used on a mainframe. Although there are several solution techniques, there exist cases such as a large network that are beyond the scope of analytical methods and present difficulties for numerical techniques. As a consequence, there is a continuing search for new and more robust optimization techniques capable of handling such problems. If a model is to be used as a tool in the design process, it should fulfill the following requirements: (1) it should be capable of handling several design problems from a short-range analysis, such as assigning frequencies and improving the existing network, to long-term decisions such as redesigning the entire route network; (2) it should give a good solution in a reasonable computing time; and (3) it should be easily accessible and comprehensible to the planner.

The model presented in this paper was developed with the objectives stated above. It consists of two phases. In the first phase, a set of potential routes called the candidate route set is generated by a candidate route set generation algorithm (CRGA). In the second phase, an optimum set of routes are selected from the candidate route set using a genetic algorithm. Due to the ability of the GA to provide a robust search as well as a near optimal solution in a reasonable time, this approach is used. The next section discusses CRGA, the first phase of the model.

Candidate Route Set Generation Algorithm (CRGA)

A set of candidate routes are generated in this phase, which will compete for the solution route set. The size of this set becomes enormous if one considers all possible routes that can be generated. CRGA is a heuristic design algorithm that will generate a set of routes which are guided by (1) a demand matrix; (2) the designer's knowledge to be implemented; and (3) the constraints for routes. The CRGA consists of the following steps:

- Step 1: Generate routes for every terminal node pair.
- Step 2: Generate a route by finding the shortest path between the origin and destination nodes for each pair identified in Step 1.
- Step 3: Check for minimum and maximum route length constraints. If the route satisfies the constraints, then the route is accepted as a candidate route.
- Step 4: Generate alternate routes by clamping every link of the shortest route generated in Step 2 successively and by finding the shortest path between the same origin and destination and then, release the link.
- Step 5: Check for each of the alternate routes whether it satisfies constraints such as (1) the existence of the route; (2) duplication of routes; (3) significant overlap with the shortest route (say, more than 75%); (4) maximum route length; and (5) maximum detour (1.5 times the shortest route). If the route satisfies, then the alternate route is accepted as a candidate route.
- Step 6: Rank all the routes based on a performance index and store them as the candidate route set.

In Step 1, the origin and destination of the routes are selected from the list of terminal nodes. The terminal nodes are decided by the designer, taking the network characteristics in consideration. In the next step, for the origin-destination pair, the shortest path is found using Dijakstra's shortest path algorithm (Teodorovic 1986). The shortest path is the one that takes minimum travel time among the various alternatives. Other criteria, like the shortest distance, can also be used for finding the shortest path, but travel time estimation is the widely accepted one. Very short routes as well as very long routes are avoided in Step 3 owing to operational difficulties, though the number of transfers may increase. In Step 4, a number of alternate routes were generated by successively clamping each link of the shortest route through inducing a large penalty for using that link. Step 5 checks the feasibility of these alternate routes. An alternate route thus generated will be selected only if the constraints are satisfied. The constraints are the existence of an alternate route, duplication of route, route overlap, maximum length of the route, and the detour with respect to the shortest route. If the alternate route generated satisfies all these constraints, it is added to the set of selected routes. The route set generated by CRGA is highly dictated by the set of terminal nodes, shortest path criteria, and the set of constraints used to check the feasibility of a generated route. Hence, the planner's knowledge is vital in this stage. The next section discusses how the RND can be represented as a GA problem.

Representation of Variables

To solve an RND problem using GAs, it must first be represented as a GA problem in which the GA operators can be applied. Hence, the representation scheme is the key issue, because it links the real-world problem to the GA problem as the GA manipulates the coded representation of the problem. The individual routes are considered as the variables. The value of the variable can be any performance index of an individual route, like cumulative demand satisfied or passenger kilometer of the individual route. The actual value of the performance index of the routes will be known only after selecting the routes, allocating the demand, and assigning frequencies. But the relative value of the performance index for individual routes are found out by assuming only that route as existing in the network and then assigning the frequency. This will yield only a relative value and may serve as a proxy for the potential of a route to satisfy the objective function. In applying GA operators, the routes are identified by route numbers and then ranked based on the value of the performance index.

The length of the substring varies with the desired precision

J. Transp. Eng., 1998, 124(4): 368-375

of the results; the longer the string length, the greater the accuracy. The relationship between string length and precision is

$$(x_i^u - x_i^l)10^\alpha \leq 2^\beta - 1 \tag{5}$$

where α = required precision; β = string length; and x_i^u and x_i^l = the upper and lower limits, respectively, which that variable can take. The value for each variable x_i is decoded using the following equation:

$$x_i = x_i^l + \frac{x_i^u - x_i^l}{2^\beta - 1} \sum_{j=0}^{\beta} \gamma_j 2^j \tag{6}$$

where γ_j = the binary digit in the jth position of the string.

Once the coding-decoding scheme of the variable is finalized, the implementation of the RND problem using a genetic algorithm is done in two methods, fixed string length coding and variable string length coding. Both methods are described in the following sections.

FIXED STRING LENGTH CODING (FSLC) METHOD

This section presents the implementation of the fixed string length coding (FSLC) method. The objective of the model is to find the optimum set of routes from the set of candidate routes. The strategy adopted in this method is to assume a size to the solution route set (i.e., the number of routes in a feasible solution). The GA will find the best routes among the candidate routes. However, the solution will correspond to the assumed route size. Hence, the evaluation may be carried out by varying the size of the route set over a range to find the optimal route set. The choice of this range is made based on the assumption that the size of the optimal route set lies in this range. If the range is narrow, then the computational time will be less. But the disadvantage is that how many routes will be in an optimum solution is not known. Thus, in the solution process the route set size has been varied in each iteration. The procedure is shown as follows in the form of a pseudocode:

```
Route set size = min. route set size
repeat
   formulate population for route set size
   randomly initialize the population
   repeat
      evaluate objective function
      update statistics if solution improved
      find fitness function
      generate new population
   until stopping criteria
   increment route set size
until route set size = max. route set size
```

For the purpose of explanation of this method, the coding of an example network with eight nodes and nine links is considered (Fig. 1). In the first phase, the candidate route set was generated by CRGA and is shown in Table 1. The set contains 28 routes, and the demand satisfied by the route was considered as the performance index used for ranking. To represent these 28 routes, the string length needed with precision 0.01 is 5, as per (5). In decoding a typical sub-string 10110, $\beta = 5$, $x_i^u = 27$, and $x_i^l = 0$; hence x_i obtained by (6) is 19.16, and route 19 is adopted as the route corresponding to the substring 10110. In the second phase, the GA will find the optimum set of routes from the 28 candidate routes generated in the first phase. The iteration for route set size can be varied from three to 12 routes with an increment of one route. Note that in this coding method, the number of routes in a feasible solution is fixed during the successive generation. These 28

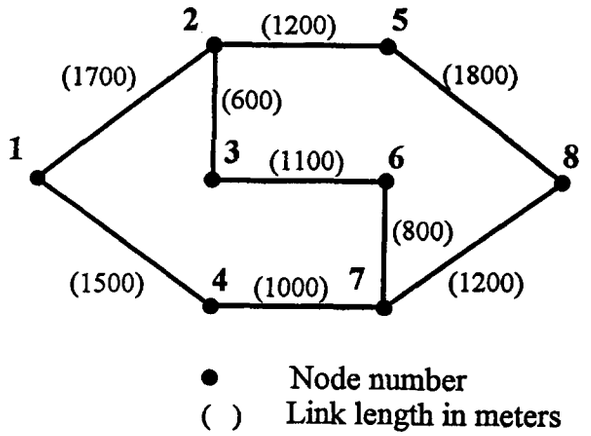
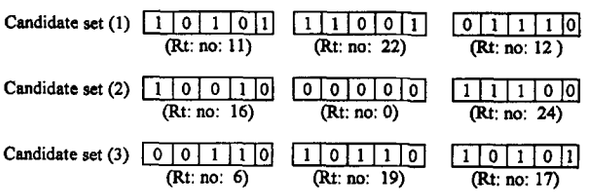


FIG. 1. Example Network with Eight Nodes and Nine Links

TABLE 1. Candidate Route Set Generated by CRGA for Example Network

Route number (1)	Shortest path (2)	Demand satisfied (3)
1	1-4-7-8	637
2	2-3-6-7	611
3	3-6-7-8	608
4	1-4-7-6	599
5	5-2-3-6	561
6	4-1-2-5	550
7	3-6-7-4	438
8	2-1-4	320
9	6-7-8	314
10	3-2-5	296
11	2-5-8	287
12	2-3-6	280
13	1-2-3	279
14	4-7-8	270
15	1-2-5	266
16	5-8-7	263
17	1-4-7	241
18	4-7-6	239
19	3-6-7	214
20	1-2	95
21	7-8	93
22	6-7	86
23	5-8	84
24	2-3	64
25	2-5	59
26	1-4	58
27	4-7	58
28	3-6	50

candidate routes will compete for the optimum set and let a feasible solution contains three routes. To start the iteration, the lower range of the route set size is adopted, i.e., with three routes for the example network. These three routes are represented by a string having three substrings. Each substring represents the route number in binary code as mentioned before. In this three-route system, three substrings representing the three routes are concatenated to form a string length of 15. This represents a solution vector in the solution space defined by the 28 candidate routes. A population of such solution vectors are generated randomly to form the initial solution. The coding of three individual candidate sets in a population for a route set size of three is shown as follows:



Once coding and decoding scheme is finalized and the initial population is generated, the genetic search can be started. The next step is to evaluate the population, i.e., to find the objective function value of each individual in the population. Note that each individual is a set of routes that tries to satisfy the demand subject to the constraints. This evaluation is done by a route evaluation module (REM), which is explained in the next section.

The objective function value obtained from REM is transformed into a fitness value using (4). In the next step, the genetic operators, namely, reproduction, crossover, and mutation, are applied to this initial population. Application of these operators results in a new population and is evaluated in the same way. If the solution improves over the current one, it is stored as the best solution. This completes one cycle (generation) and is repeated till convergence. This concludes the iteration for the route set size adopted. Then the size of the route set is incremented and successive iterations are made. This is repeated until the maximum route set size is reached. The best solution among the route set is adopted as the solution.

Route Evaluation Module (REM)

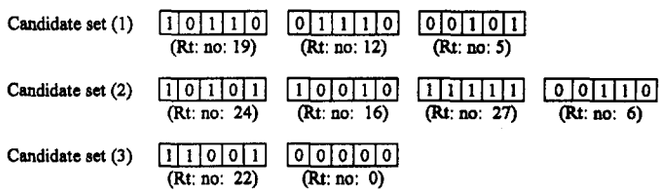
A vital component in the solution of the network design problem is a procedure to evaluate the objective function components in the mathematical formulation and other performance measures that are of concern to both the operator and user of a given network configuration. For this, the trip demand matrix is assigned to the network configuration defined by a set of feasible routes. Route choice in a transit network, especially in large urban areas with overlapping routes, could be the transfer avoidance and/or minimization among the competing routes as the primary choice criterion (Han and Wilson 1982). Baaj and Mahmassani (1990) incorporated this feature and developed a procedure to evaluate the objective function, and the same procedure of assignment and frequency setting is adopted in the present study. The REM computes the network statistics and finds the objective function value using (1). The objective function expresses the travel time in minutes of both passenger and buses in a single value. This total travel time will act as a proxy for the total cost. This is achieved by converting the duration of bus operation into the equivalent passenger travel time using constants c_1 and c_2 from (1). The values adopted for the present work are $c_1 = 1$ and $c_2 = 12$, based on a previous study (Bansal 1981). A maximum allowable frequency of 30 buses/hr, minimum allowable frequency of 1.5 buses/hr, maximum load factor of 1, transfer penalty of 10 min, and bus seating capacity of 60 are the other constants used in the present work. The next section describes a newly proposed alternate method of implementation called the variable string length coding (VSLC) method.

VARIABLE STRING LENGTH CODING (VSLC) METHOD

When the size of the network increases, the number of routes in the optimum network may also increase. Then the iteration for the route set size has to be carried out in a wide range, because it is not known how many routes constitute the optimal or near-optimal solution. This requires excessive computational time. Hence, variable string length coding is proposed where the variation for the solution route set size is incorporated in the coding itself. This is a modification of the constant string length coding, eliminating the constraint of iteration for the route set size. This proposed modified method is given as follows in the form of a pseudocode:

```
formulate variable string population
randomly initialize population
repeat
   evaluate objective function
   update statistics if solution improved
   find fitness function
   generate new population
until stopping criteria
```

When the population is initialized, the number of routes in each population is randomly selected, and then the routes in the individual population are initialized. In this method, the route set and number of routes are optimized simultaneously. Hence, there is no need to fix the route set size. Coding of three individual candidate sets in a population is shown as follows:



The other steps, e.g., finding the objective function and fitness function, reproduction, crossover, and mutation, are the same as in the previous method, except for the use of two additional new operators proposed in this study, namely the insertion and deletion operators. This is important in variable string length coding, as there is no iteration for the number of routes, and the final solution may contain a number of routes which were in the initial population. When the insertion operator is introduced as guided by an insertion probability, a new route is added to that individual population. The route is added by concatenating a substring whose value is initialized randomly. In the same way, the deletion operator will delete one route from the individual in the population. This will help the population to cover a wide range of route sets and with varying sizes of the route set. The insertion and deletion probabilities are kept the same in the present work because their effect is almost similar.

RESULTS AND ANALYSIS

To illustrate the application of the GA model, the transportation network of a part of the Madras Metropolitan City, South India, has been considered as a case study. The network is defined by 25 nodes and 39 links, as shown in Fig. 2. The node data contains the node number and its adjacent node numbers. The link data includes the length of the link and the average travel speed of buses in the links. The demand matrix for the network was not readily available in the desired format for the peak period. To demonstrate the performance of the model, a symmetrical demand matrix for a one-hour peak period was randomly generated to obtain meaningful input data. The sole purpose of this is not to evolve the exact design, but for illustration of the model. The set of candidate routes also forms the input data in which each route is defined by a route number, route length, potential demand satisfied, and the nodes of the route (Table 2).

Genetic Algorithm Parameters

Before the application of the model, the basic parameters of the genetic algorithm need to be estimated. The parameters that determine the performance of the GA are population size, crossover probability, mutation probability, insertion probability, deletion probability, and stopping criteria. To study the effect of population size, runs were made by varying the population size from 10 to 500, and the results are shown in Fig. 3(a). It can be seen from the figure that, as the population size

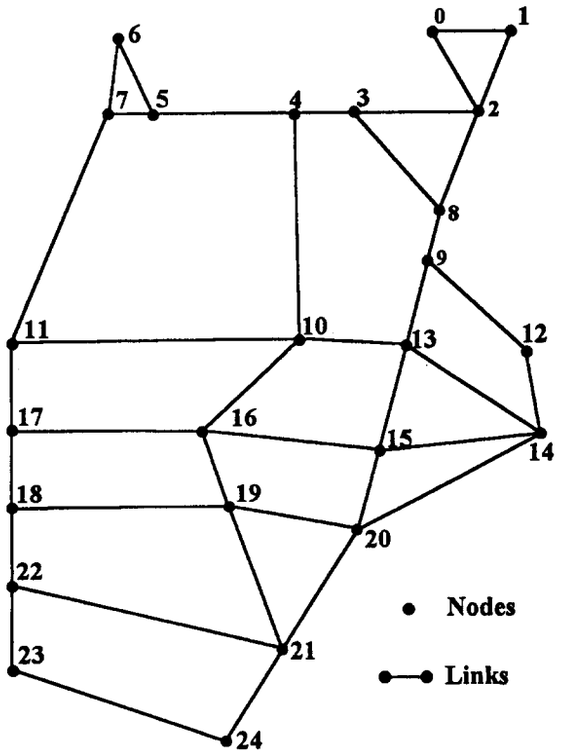


FIG. 2. Case Study Network with 25 Nodes and 39 Links

TABLE 2. Candidate Route Set Generated for Case Study Network

Route number (1)	Demand satisfied (2)	Distance (m) (3)	Nodes defining route (4)
0	21,224	14,600	0-2-3-8-9-13-10-16-19-18-22-23
1	21,162	13,900	0-1-2-8-9-13-10-16-19-18-22-23
2	21,162	14,300	1-0-2-8-9-13-10-16-19-18-22-23
3	20,666	14,200	1-2-3-8-9-13-10-16-19-18-22-23
4	17,202	13,600	2-3-8-9-13-10-16-19-18-22-23
5	17,080	12,200	0-2-3-8-9-13-10-16-19-18-22
...	...	...	...
171	5,594	5,400	1-2-8-9-13-10-16
172	5,572	5,600	2-8-9-13-10-16-19
173	5,404	6,500	2-8-9-13-15-16-19

increases, the convergence rate is increased. On the other hand, the computation time was also increased with an increased population size. The convergence rate is less beyond a population size of 50; hence, the value of population size is adopted as 50. It is to be noted that Goldberg (1989) also recommended a population size of 30 to 50.

Similarly, the effect of the crossover probability on the performance was found by varying the crossover probability values from 0.4 to 0.8; 0.6 was found to be the best [Fig. 3(b)]. This agrees with the values recommended by Koumousis and Georgiou (1994). The effect of the mutation probability was also found by varying its value from 0 to 0.1 and plotted in a semilog graph to show the effect. The results revealed that a value of 0.05 may be the best [Fig. 3(b)]. This also falls in the normal range recommended by Koumousis and Georgiou (1994). Runs were also made to study the effect of insertion and deletion operators on the convergence. Because they are newly proposed operators, their probability value was varied over a wide range (0.001–0.5). The results are shown in a semilog graph [Fig. 3(d)] to show the variation of the objective function value with the probability values. The results show that the objective function value was very low in the range of 0.001–0.01; the value of 0.001 is adopted in the present study,

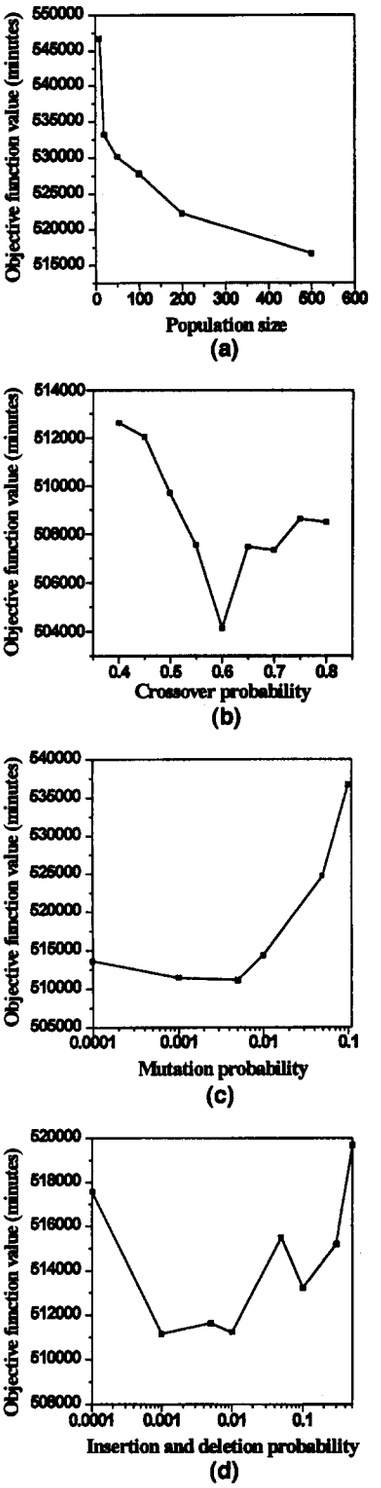


FIG. 3. Results of Sensitivity Analysis of GA Parameters: (a) Effect of Population Size; (b) Effect of Crossover Probability; (c) Effect of Mutation Probability; (d) Effect of Insertion and Deletion Probability

as the objective function value was obtained with this probability value.

Two stopping criteria are generally employed in GAs. The first one is the convergence of the objective function value, i.e., the generation will be stopped if the improvement in the solution is less than a specified value. The second one is the number of generation, which is fixed a priori; this approach was used in the present study. To determine the number of generations needed, trial runs were made with number of gen-

erations from 100 to 5000 for both FSLC and VSLC. The optimum value was obtained in less than 120 generations for FSLC and 250 generations for VSLC, beyond which no significant improvement occurred. It can be inferred that, for better performance, the number of iterations needs to be low, and the model may be applied a number of times by varying the random seed. Thus, the stopping criterion was the completion of 120 generations for FSLC and 250 generations for VSLC.

Application to Case Study Network

The two models, FSLC and VSLC, were applied to the case study network. The FSLC model was applied with iteration for a route set size ranging from 7 to 20, expecting the optimum solution in this range. The trend of the objective function variation for FSLC is given in Fig. 4(a). The solid line represents the best objective function value for the entire route set size range. The dashed line represents the optimal value of the objective function obtained when generation is made for each route set size. The solution improved initially when the route set size was increased, because more demand was assigned to routes having less unsatisfied demand. But after the optimum value was reached, with 16 routes, further improvement in the route set size did not result in an improvement in the objective function value, because of the under-

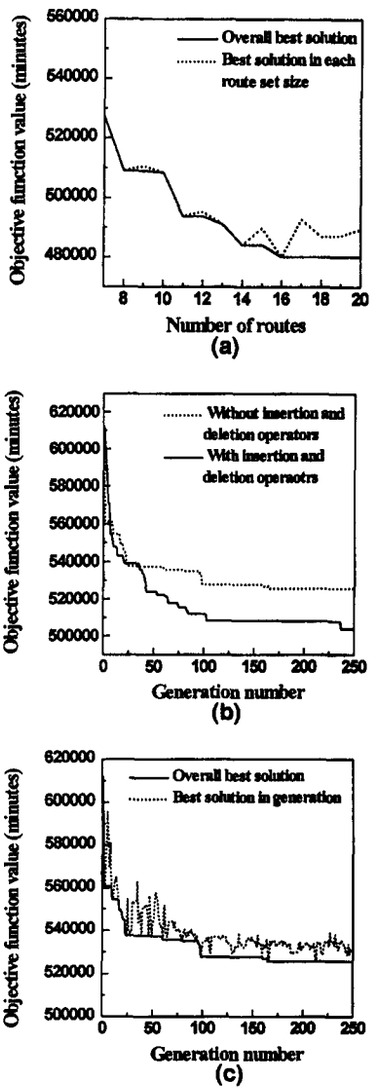


FIG. 4. Trend of Objective Function Convergence: (a) FSLC Model; (b) Effect of Insertion and Deletion Operators; (c) VSLC Model

TABLE 3. Values of Performance Measures with FSLC and VSLC

Descriptors (1)	FSLC (2)	VSLC (3)
Objective function value (min)	480,107	503,870
Number of routes	16	13
Demand satisfied with 0 transfer	12,925	11,377
Demand satisfied with 1 transfer	3,287	4,757
Total demand satisfied	16,213	16,135
Unsatisfied demand	0	78
Total in-vehicle travel time (min)	290,590	308,944
Total waiting time (min)	42,568	34,455
Total travel time (min)	333,158	343,400
Bus kilometer	12,245	12,982
Fleet size	102	108
Computation (CPU) time (min)	480	68

utilization of routes. This shows that the assumption for the route set size range proved valid. The performance measures of the solution route set are given in columns 1 and 2 of Table 3. The FSLC method resulted in a solution with 16 routes and 102 buses fully satisfying the demand. The solution route set, its associated frequencies, and the nodes in the route are given in columns 2–4, respectively, in Table 4.

The VSLC model was also applied to the case study network with and without the use of insertion and deletion operators. The trend of the objective function variation for both cases is given in Fig. 4(b). It can be inferred that the introduction of insertion and deletion operators improves the convergence and gives a better solution. Hence, the use of these operators is recommended in the VSLC model. The performance measures of the solution route set for the VSLC model are given in columns 1 and 3 of Table 3. The VSLC method gave a solution with 13 routes and 108 buses, leaving a demand of 78 unsatisfied. The solution route set, its associated frequencies, and the nodes in the route selected by VSLC are given in columns 5–7, respectively, of Table 4. The trend of the objective function variation for VSLC is shown in Fig. 4(c). The solid line represents the best objective function value, and the dashed line represents the best objective function value in each generation.

A comparison of the performance of the FSLC and VSLC models can be inferred from Table 3. FSLC gives a better solution in terms of low objective function value, unsatisfied demand, total travel time, bus kilometer, and fleet size. On the other hand, VSLC resulted in a smaller number of routes and less waiting time. It may be noted that the objective function value of VSLC is only 4.9% higher than that of FSLC, but the computation time of FSLC is approximately seven times that of VSLC. The high computation time for FSLC is attributed to the iteration carried out for route sizes from 7 to 20. Though FSLC gives better performance measures, these are marginal when compared to the significant computational time-saving of the VSLC model. Hence, the VSLC model can be adopted as a design procedure which gives a near-optimal solution with considerably less computation time. Nevertheless, the computation time of FSLC can be considerably reduced if the iteration range for the route set size is reduced, which requires a good knowledge of the network. FSLC is slightly superior to VSLC if computation time is not a constraint.

CONCLUSIONS

The urban bus route network design with conventional approaches poses considerable difficulties owing to the combinatorial nature of the problem. In this paper, genetic algorithms (GAs) were used in the route network design. The design is done in two phases: first, a set of candidate routes is generated;

TABLE 4. Solution Route Set Resulting from FSLC and VSLC

Method	FSLC			VSLC		
Solution number (1)	Route number (2)	Frequency (bus/hr) (3)	Nodes defining routes (4)	Route number (5)	Frequency (bus/hr) (6)	Nodes defining routes (7)
1	132	20	7-5-4-10-13-9-12	126	6	13-10-16-17-18-22-23
2	37	24	1-2-3-4-5-7-11-17-18	86	22	0-2-3-4-5-7-11-17
3	141	22	9-12-14-20-21-24-23	49	18	1-2-8-9-13-10-11-17-18
4	163	22	0-2-8-9-13-10-16	60	24	1-0-2-8-9-13-15-16-19
5	142	10	7-11-17-18-22-21-24	87	22	6-5-4-10-16-19-21-24
6	101	15	1-2-8-9-12-14-20-19	131	18	6-7-11-17-18-19-20
7	32	25	1-2-3-4-10-13-15-20-21	156	10	2-8-9-12-14-20-19
8	90	6	3-8-9-12-14-20-21-24	46	28	3-2-8-9-12-14-20-21-24
9	44	15	3-4-5-7-11-17-18-22-23	140	12	1-2-3-4-5-7-11
10	160	10	6-5-4-3-8-9-12	35	16	0-2-3-4-10-13-15-20-21
11	27	8	1-2-8-9-13-10-16-19-21-24	134	5	3-4-5-7-11-17-18
12	78	22	8-9-13-15-20-21-24-23	142	12	7-11-17-18-22-21-24
13	169	12	8-9-13-15-16-17-18	98	15	2-3-8-9-12-14-20-21
14	1	20	0-1-2-8-9-13-10-16-19-18-22-23	—	—	—
15	97	4	2-8-9-12-14-20-21-22	—	—	—
16	131	14	6-7-11-17-18-19-20	—	—	—

then the optimum route set is selected using a genetic algorithm. The coding of variables and the optimization process of the genetic algorithm enhanced the efficiency of the design problem. Two coding schemes are developed, namely, fixed string length coding and variable string length coding. Fixed string length coding is simple and gives a better solution, but it requires more computational time. On the other hand, variable string length coding can handle simultaneously selection of the route set size and the set of routes, but this requires complex coding. The model is applied to a case study network that is part of a real network, resulting in important findings. It is concluded that a combination of fixed and variable string length coding resulting in a hybrid scheme needs to be analyzed for its suitability to derive optimal solutions in route network design problems. The application of the model to a very large size network is in progress.

APPENDIX I. REFERENCES

Baaj, M. H., and Mahmassani, H. S. (1990). "TRUST: A LISP program for the analysis of transit route configurations." *Transp. Res. Rec. No. 1283*, 125–135.

Baaj, M. H.. and Mahmassani, H. S. (1995). "Hybrid route generation heuristic algorithm for the design of transit networks." *Transp. Res. Part C*, 3(1), 31–50.

Bansal, A. N. (1981). "Optimization of bus route network for fixed spatial distribution." *Scientific management of transport systems*, N. K. Jaiswal, ed., North Holland Publishing Company, Amsterdam, The Netherlands, 346–355.

Bartlett, G. (1995). "Genie: A first GA." *Practical handbook of genetic algorithms: applications*, Vol. 1, L. P. Chambers, ed., CRC Press, Boca Raton, Fla.

Ceder, A., and Wilson, N. H. M. (1986). "Bus network design." *Transp. Res. Part B*, 20(1), 331–344.

Chakroborty, P., Deb, K., and Subrahmanyam, P. S. (1995). "Optimal scheduling of urban transit systems using genetic algorithms." *J. Transp. Engrg.*, ASCE, 121(6), 544–553.

Chua, T. A. (1984). "The planning of urban bus routes and frequencies: a survey." *Transportation*, Amsterdam, The Netherlands, 12, 147–172.

Dubois, D., Bel, G., and Llibre, M. (1979). "A set of methods in transportation network synthesis and analysis." *J. Operation Res. Soc.* 30(9), 797–808.

Dusan, T., Malica, K., and Coran, P. (1994). "The potential use of fuzzy set theory in airline network design." *Transp. Res. Part B*, 28(2), 103–121.

Furth, P. G., and Wilson, N. H. M. (1981). "Setting frequencies for bus routes: theory and practice." *Transp. Res. Rec. No. 818*, 1–7.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Co., Reading, Mass.

Han, A. F., and Wilson, N. H. M. (1982). "The allocation of buses in heavily utilized networks with overlapping routes." *Transp. Res. Part B*, 16(3), 221–232.

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. 2nd Ed., MIT Press, Cambridge, Mass.

Kochur, G., and Hendrickson, C. (1982). "Demand of local bus service with demand equilibration. *Transp. Sci.*, 16(2), 149–170.

Koumousis, V. K., and Georgiou, P. G. (1994). "Genetic algorithms in discrete optimization of steel truss roofs." *J. Computing in Civil Engineering*, ASCE, 8(3), 309–325.

Nes, R., Hamerslag, R., and Immers, B. H. (1986). "Design of public transport networks." *Transp. Res. Rec. No. 1202*, 74–83.

Steenbrink, P. A. (1974). *Optimization of transport networks*. John Wiley & Sons, London, England.

Teodorovic, D. (1986). *Transportation networks: a quantitative approach*. Gordon and Breach Science Publishers, New York, N.Y.

APPENDIX II. NOTATION

The following symbols are used in this paper:

C = bus seating capacity;
c_1 = factor to convert user travel time to user travel cost;
c_2 = factor to convert bus kilometer to cost equivalent;
d_{ij} = travel demand from node i to j;
$F(i)$ = fitness value of individual i;
f_k = frequency of kth route;
$f_{\min}$ = minimum frequency of buses operating on any route;
l_k = load factor on route $k = Q_k^{\max}/(f_k \cdot C)$;
$l_{\max}$ = maximum allowable load factor;
n = number of nodes in network;
$O(i)$ = objective function value of individual i;
P = population size;
$Q_k^{\max}$ = maximum flow occurring on any link on route k;
R_{SR} = set of routes in the solution;
t_{ij} = total travel time from node i to j (sum of in-vehicle travel time, waiting time, and transfer penalty);
t_k = round trip time of kth route;
V = maximum fitness (large value);
x_i^l = lower bound of variable x_i;
x_i^u = upper bound of variable x_i;
α = length of substring;
β = precision needed in coding;
γ_j = binary digit in jth position of substring; and
δ_k = demand allocated to kth route.

J. Transp. Eng., 1998, 124(4): 368-375