Many-to-many demand responsive transportation systems consist of one or more multiple occupant vehicles which take passengers from their origins to their destinations within a service area. They differ from many-to-one systems in that all origin-destination pairs are served without requiring a transfer. Many-to-many transportation systems can be divided into two main categories on the basis of whether or not ride sharing is permitted. In conventional taxicab systems, once a vehicle picks up a customer, it proceeds non-stop to the customer's destination. In this case, at all times each vehicle has at most one request on board (a request is a group of passengers traveling together from a common origin to a common destination) and the maximum total number of request-miles per hour that can be accommodated (i.e. the capacity of the system) cannot exceed the maximum achievable number of vehicle-miles per hour with the available vehicle fleet. Thus, the capacity of a taxicab system is severely limited by the number of vehicles in operation. In dial-a-bus systems, in order to circumvent this undesirable feature of taxicab systems, vehicles are allowed to deviate from their direct route to serve other passengers and the emphasis is on building efficient tours to increase vehicle productivity. This strategy increases riding times but also increases average occupancy and productivity of the vehicles, and hence decreases average waiting times.

Models of demand responsive transportation systems have also been studied—models of taxicab operations have been reported in the literature (Fargier and Cohen, 1972; Gerrard, 1974)—a satisfactory analytical model of dial-a-bus operations has yet to be developed. This is perhaps because dial-a-bus routing algorithms have become so complicated (especially when they try to accommodate customers with different preferences and/or advance service requests) that they seem to defy mathematical modelling. One such algorithm is described by Wilson et al. (1971). At present, simulation models (Heathington et al. 1968; Wilson et al. 1971) or empirical models based on simulated or real data (Arilarga and Medville, 1974; Flusberg and Wilson, 1976) seem to be the only alternatives open to the planner. However, the former are expensive, time consuming, and not suitable for parametric analysis, and the latter require careful calibration and are limited in range.

This paper develops a simple analytic model to predict average waiting and riding times of many-to-many demand responsive transportation systems. In order to model different operating strategies corresponding to the operations of taxicab and dial-a-bus systems, three different routing algorithms are studied applying the same modelling technique used by Daganzo et al. (1977) for the many-to-one case. The resulting model is simple enough to facilitate parametric analysis of these systems and model results agree well with the results of simulation. Section 1 develops asymptotic formulae with a deterministic model, Section 2 investigates some refinements of these formulas, and Section 3 discusses the results obtained.

1. THE DETERMINISTIC MODEL

A many-to-many dial-a-bus system can be visualized as a two-stage queueing network. Fig. 1, where buses perform service on requests. A typical request is processed as follows: when the service request is made, it joins a queue of requests awaiting pick-up in stage one. The requests move from stage one to stage two when a bus arrives at the customer's origin and picks up the customer. During stage two, the customer is in the queue...
of passengers riding on a given bus. The request leaves stage two when he is delivered at his destination.

We analyze the characteristics of a dial-a-bus system with $M$ buses over an approximately circular (or square) service area, $A$, with a request arrival rate, $\lambda$. We further assume that the origins and destinations of requests are uniformly and independently distributed over the service area; although this assumption is not true for large metropolitan areas (see Blumenfeld, 1977), it has been used in all previous studies because it makes modelling easier and may not be unreasonable for smaller areas where many-to-many dial-a-bus systems typically operate.

In order to build a model of system behavior, one must also make some assumptions regarding the routing algorithm; as was done by Daganzo et al. (1977), we will assume that when a bus has a choice of stops, it selects the next nearest one in order to increase the number of stops per unit time (i.e. the productivity). This simple routing strategy can be regarded as an approximation to manual routing strategies used in real life systems (the many-to-one analytic model of Daganzo et al. reproduced the results of the manually dispatched Ann Arbor system extremely well by assuming a next nearest point routing strategy) and will enable us to model many-to-many dial-a-bus systems in a simple way.

Use of the next nearest point strategy, however, does not confine us to studying a single routing algorithm. One can obtain different algorithms by defining the set of stops from which the choice can be made. The following three algorithms are studied in this paper:

(i) Algorithm I.
After each stop, the bus is routed to the nearest feasible point (whether an origin or a destination of one of the passengers in the bus).

(ii) Algorithm II.
The bus alternates pick-ups and deliveries (always selecting the closest pick-up and the closest feasible destination).

(iii) Algorithm III.
The bus collects a fixed number of passengers and then delivers them.

Of course, other algorithms could also be analyzed with the methodology presented in the paper, but these three algorithms seem general enough to approximate most others.

In the remainder of this section, we assume that the arrival process is continuous, deterministic, and constant, that travel distances between stops are a deterministic function of the number of stops the bus can select from, and that the boarding and alighting times ($b_1$ and $b_2$, respectively) are also deterministic.

We will also use the following result from geometric probability (see Kendall and Moran, 1963) for the average distance, $d_n$, between a random point and the nearest of $n$ independent random points in an area, $A$:

$$ \lim_{n \to \infty} d_n = \frac{1}{2} \sqrt{\left( \frac{A}{n} \right)} \tag{1} $$

Since $d_1 = 0.51\sqrt{A}$ for a circle (Kendall and Moran, 1963) and $d_1 = 0.52\sqrt{A}$ for a square (Daganzo, 1975) one can safely assume that

$$ d_n \approx 0.50\sqrt{\left( \frac{A}{n} \right)} \tag{2} $$

for circular or quasi-circular shapes (i.e. areas that overlap heavily with a circle with the same area; say, by 80% or more). Several Monte-Carlo experiments were performed and eqn (2) was found to predict the true distance within a 1% error margin in circular areas.
quasi-circular areas with 90% overlap with a circle (e.g. a square), the error margin is about 3%.

In practical situations, eqn (2) must be corrected to account for the non-euclidean metric of real street networks and we introduce the travel factor, \( r \), to capture the circuity of the network (\( r \) is defined as the ratio of the average over the network distance of a trip to the average airline distance; it takes a value of 1.27 for 2-directional grid networks in circular areas—Fairthorne, 1963). Since for square areas with 2-directional grid networks, \( r \) is also approx. 1.27, independent of the orientation of the network, and for both squares and circles, the value of \( r \) for homogeneous networks (i.e. grid-like networks, as opposed to radially symmetric networks, in which the length of road per unit area and the general orientation of the network remain constant in the study area) is approximately independent of trip distance (Daganzo, 1975), it is safe to assume that for quasi-circular areas with homogeneous networks:

\[
d_\text{eq} = \frac{r}{2} \sqrt{\frac{A}{n}}.
\]

The rest of this paper revolves around this formula.

1.1 Algorithm I

Since with this algorithm the routing strategy of the bus is to go to the closest trip end regardless of whether it is an origin or a destination, in the steady state a bus should be equally likely to select an origin or a destination (otherwise the number of passengers on the bus would tend to either increase or decrease). For this to happen, the number of requests waiting, \( N \), must equal the number of requests in the bus under consideration, \( n \):

\[
n = N. \tag{4}
\]

Equation (4) guarantees the equilibrium for the stage two queues. The equilibrium of the stage one queue is achieved when the arrival rate, \( \lambda \), equals the rate at which buses pick up requests (the rate at which any one given bus picks up requests, \( \lambda_b \), times the number of buses \( M \)):

\[
\lambda = \lambda_b \cdot M, \tag{5}
\]

where \( \lambda_b \) is the inverse of the time between two successive bus pick-up stops; \( \lambda_b \) is given for this algorithm by:

\[
\lambda_b = \left( b_1 + b_2 + 2 \cdot d_{\text{eq}} \cdot r \right)^{-1}, \tag{6}
\]

where \( b_1 \) and \( b_2 \) are, respectively, the boarding and alighting times and \( v \) is the average bus speed excluding stops for passengers. We note that eqn (6) implies that buses are always able to pick up the customers they select (i.e. buses do not compete for waiting requests) and that since in addition we are neglecting the stochastic phenomena of the problem, the equation will be unreliable for small values of \( n + N \). Section 2 explores these two issues in more detail.

Substituting eqns (4) and (5) into (6), one gets

\[
M = \frac{b_1 + b_2 + 2d_{\text{eq}}}{v};
\]

and since \( d_{\text{eq}} = 0.5r \sqrt{\left( \frac{A}{2N} \right)} \)

\[
M = b_1 + b_2 + \frac{r}{v} \frac{A}{1 \sqrt{(2N)}}.
\]

Equation (7) can be solved for \( N \) to yield

\[
N = \frac{1}{2} \left( \frac{\frac{r}{v} \sqrt{(2N)}}{\frac{M}{\lambda} - (b_1 + b_2)} \right)^2 = \frac{r^2}{2v^2} \left( \frac{M}{\lambda} - (b_1 + b_2) \right)^2. \tag{8}
\]

The result of eqn (8) has a physical meaning only if \( M/\lambda > (b_1 + b_2) \) since otherwise the required time between pick-ups for equilibrium, \( M/\lambda \), would be smaller than the combination of a boarding and alighting time. Thus, \( \lambda_{\text{max}} = M(b_1 + b_2) \) represents the theoretical capacity of the system under algorithm I.

Equation (8) enables us to obtain waiting times, \( w \), riding times, \( u \), and the total time in the system, \( t \). One has:

\[
w = \frac{N}{\lambda}, \quad u = \frac{n}{\lambda_b} = \frac{NM}{\lambda}
\]

and

\[
t = \lambda(M + 1) - \frac{r^2}{2v^2} \left( M - \lambda(b_1 + b_2) \right)^2. \tag{9}
\]

Equation (9) cannot be accurate for uncongested systems since \( t \to 0 \) as \( \lambda \to 0 \). This happens because, as was mentioned before, eqn (6) is not accurate for small values of \( n + N \) and because stochastic queueing phenomena were neglected. On the other hand, the predictions of eqn (9) should be good as \( M/\lambda > (b_1 + b_2) \) since in that case the system is so crowded that the deterministic elements of the problem (i.e. the fact that one can serve people at a faster rate when the queues are long) play a more dominant role than the stochastic fluctuations. This remark is substantiated in Section 2 where a formula more accurate than eqn (9) is obtained.

We now derive asymptotic equations similar to eqn (9) for algorithms II and III.

1.2 Algorithm II

For this algorithm the bus keeps approximately a constant number of passengers on board at all times by alternating pick-ups and deliveries. We denote by \( n_i \) the number of passengers in bus \( i \) after a pick-up (which for this algorithm is a decision variable) and by \( \lambda_{b(i)} \) the rate at which bus \( i \) picks up customers. For a busy system with all queues having at least one customer \( \lambda_{b(i)} \) equals the inverse of the sum of times for a pick-up trip and a delivery trip:

\[
\lambda_{b(i)} = \left( b_1 + b_2 + \frac{d_{\text{eq}} + d_{\text{eq}}}{v} \right)^{-1}. \tag{10a}
\]
where as with algorithm I, we have assumed that buses do not have to compete for pick-up of waiting requests. Using eqn (3), we can write for the equilibrium of the stage one queue:

\[
\lambda = \sum_{i=1}^{\infty} \lambda_i = \sum_{i=1}^{\infty} \left[ b_1 + b_2 + \left( \frac{1}{\sqrt{N}} + \frac{1}{\sqrt{n_i}} \right) 0.5 \frac{\sqrt{A}}{v} \right]^{-1}.
\]

It is possible to show (using Lagrange multipliers and straightforward calculus minimization techniques) that for a given average number of persons in the buses, \( n \),

\[
\left( n = \frac{\sum_{i=1}^{\infty} n_i}{M} \right)
\]

the values of \( n \), that minimize \( N \) (the solution of the above equation), and consequently minimize the total time spent by a customer in the system, are

\[
n_i = n, \quad i = 1, 2 \ldots M.
\]

That is, the total delay is minimized when the number of customers in each bus is equal. This simplifies the equilibrium equation which becomes:

\[
\lambda = M \left[ b_1 + b_2 + \left( \frac{1}{\sqrt{N}} + \frac{1}{\sqrt{n}} \right) 0.5 \frac{\sqrt{A}}{v} \right]^{-1}
\]

and yields:

\[
N = \left( \frac{M \lambda - (b_1 + b_2 - 1)}{0.5 \sqrt{A} \lambda v} \right)^2.
\]

Note that as with algorithm I, the solution has physical meaning only for \( \lambda < \lambda_{\text{max}} \) (the theoretical capacity) and that in this case, \( \lambda_{\text{max}} \) is the value of \( \lambda \) for which the r.h.s. of eqn (11) becomes \( \infty \). Thus,

\[
\lambda_{\text{max}} = M \left( b_1 + b_2 + 0.5 \frac{\sqrt{A}}{v} \right)^{-1}
\]

and the system cannot have a steady state if more than \( M \) requests arrive in the combination of a boarding time, an alighting time and the travel time for one delivery.

The total time spent in the system by a request is

\[
t = M(n - 0.5) + N
\]

where the numerator of eqn (12) is the approximate number of customers in the system. The value of \( n \) that minimizes eqn (12) subject to eqn (11) is the optimal number of passengers we should keep on each bus; that is:

\[
\lambda = (1 + M^{-1/3}) \left[ 0.5 \frac{\sqrt{(A/v)}}{\lambda} \left( \frac{M}{\lambda} - (b_1 + b_2) \right) \right]^{-2}.
\]

Note that with \( n = \lambda \) the system is under capacity as long as \( \lambda < M(b_1 + b_2) \) since in this case the quantity in brackets on the r.h.s. of eqn (11) is always positive. Substituting \( \lambda \) for \( n \) in eqn (11), we get that \( \hat{N} = M^{2/3} \lambda \) and entering with these values into eqn (12) we get for the optimal delay, \( \hat{t} \):

\[
\hat{t} = \frac{\hat{N}((M + M^{2/3}) M)}{2} = \frac{M^{1/3}}{(1 + M^{-1/3})^3} \cdot \lambda M \frac{1}{b_1 + b_2 + 0.5 \frac{\sqrt{A}}{v} \cdot (M - \lambda (b_1 + b_2) - 2 M/2)}.
\]

It is interesting to compare algorithms I and II. The first noticeable difference is that algorithm II has a longer waiting-to-riding time ratio; \( \hat{t}/\hat{t} = N/(t^\ast - M) = M^{-1/3} \) as opposed to \( w/u = M^{-1} \) for algorithm I. This is a result of forcing buses to deliver customers even though a pick-up might be more efficient.

We also note that, if we neglect the (relatively small in congested systems) \( M/2 \lambda \) term in eqn (14), it has the same functional form as eqn (9) and that the ratio of the total times in both cases is:

\[
\frac{t}{t^\ast} = \frac{2(M + 1)}{M(1 + M^{-1/3})^3}.
\]

This equation is plotted in Fig. 2 where it can be seen that in congested systems, algorithm II provides a lower total time in the system for large numbers of buses. This is a reasonable result since for large numbers of buses, \( M \), the riding time with algorithm I is very large compared to the waiting time and it makes sense to trade off some waiting for riding time. This is accomplished with algorithm II by forcing buses to go on a delivery trip every other time.

Before discussing the last algorithm, we note that as with algorithm I, eqn (14) is, strictly speaking, only asymptotically valid, and that for \( n = 1 \) this algorithm reproduces the operations of a taxicab company (eqn (12a) would be adequate in this case) where the waiting requests are served on the basis of proximity to idle taxicabs. It should also be noted that with this algorithm one can select \( n \) to obtain a desired riding-to-waiting time ratio, or to minimize a linear combination of them.
1.3 Algorithm III

This algorithm, or others similar to it, can be used when it is important to reduce the variance of the riding times. In this mode of operation, which is very similar to operating strategies for many-to-one systems (see Daganzo, Hendrickson and Wilson, 1977), buses are assumed to go through a cycle consisting of a collection phase, where exactly \( n \) passengers are picked up, and a distribution phase where the passengers picked up are taken to their destinations. Note that for \( n = 1 \), algorithms II and III coincide.

It will be convenient to develop expressions for the length of the cycle, \( C \), the collection phase, \( G \), and the distribution phase, \( R \), as a function of the number of passengers picked up, \( n \). Using similar arguments as Daganzo et al. (1977), we get:

\[
R = nb_2 + \frac{0.5rv}{v}\left[(\sqrt{2+4n}) - 1.45\right],
\]
(16a)

\[
G = n\left[b_1 + \frac{0.5rv}{v}\left(\frac{1}{\sqrt{N}}\right)\right]
\]
(16b)

and

\[
C = n(b_1 + b_2) + \frac{0.5rv}{v}\left(\frac{n}{\sqrt{N}} + \sqrt{2+4n} - 1.45\right).
\]
(16c)

If in the distribution phase buses use optimum traveling salesmen type tours, the quantity in parenthesis in eqn (16a) could be replaced by \((a(\sqrt{2+4n}) - \sqrt{6}) + 1\), where, according to Eilon et al. (1971), \( a = 0.8 \) for optimally built tours. Note that tours resulting from the next nearest point strategy, \( a = 1 \), are not much longer than the optimal tour.

The rate at which passengers are picked up depends on the cycle time for each one of the buses and we have that for equilibrium:

\[
\lambda = \sum_{i=1}^{N} \frac{n_i}{C_i},
\]
(17)

where \( n_i \) is the number of passengers picked up by the \( i \)-th bus and \( C_i \) is the corresponding cycle length.

It is possible to show in the same way as was done for algorithm II that the best level of service is achieved when all buses pick up the same number of requests, \( n_i = n \); \( i = 1, 2 \ldots N \). In such case eqn (17) becomes:

\[
\lambda = \frac{Mn}{C},
\]
(17a)

which combined with eqn (16c) yields:

\[
N = \left[\frac{M\lambda - (b_1 + b_2)}{0.5rv/A} - \frac{\sqrt{2+4n} - 1.45}{n}\right]^2.
\]
(18)

The capacity in this case is

\[
\lambda_{\text{max}} = M \cdot \left[b_1 + b_2 + \frac{0.5rv}{v}\left(\sqrt{2+4n} - 1.45\right)\right].
\]

The waiting time is:

\[
\bar{w} = \frac{N}{\lambda}.
\]
(19a)

For computation of riding times, it is assumed that the rate at which buses collect (distribute) passengers remains constant throughout the collection (distribution) phase. Although this is not strictly true for the next-nearest point strategy (pick-up and delivery rates tend to be higher at the beginning of their respective phases with this strategy), it is sufficiently accurate for our purposes, especially since the same phenomenon may not occur with strategies that use other tour building methods. During a typical cycle, thus, the number of passengers in the bus increases linearly to \( n \) and then decreases linearly to 0. This results in a total (for all passengers) riding time per cycle equal to \( nC/2 \) and in an average riding time per passenger equal to half a cycle:

\[
\bar{c} = \frac{C}{2}.
\]
(19b)

The total time in the system is

\[
t = \frac{N + C}{\lambda} + \frac{C}{2}.
\]
(20)

where \( N \) is given by eqn (18). In this case, as happened with algorithms II and with the many-to-one case studied by Daganzo et al., there is an optimal value of \( n(\bar{c}) \) that minimizes \( t \). Although a simple closed form solution for \( \bar{c} \) could not be found, \( \bar{c} \) can be easily obtained in all cases by trial and error (say, by a Fibonacci search on eqn (20)).

Although it is difficult to compare algorithm I with algorithm III in a general way (a closed form solution for \( \bar{c} \) could not be obtained), we note that algorithm III will tend to behave similarly to algorithm II (both of them control the number of passengers on the bus) and therefore both may be advantageous in similar instances.
Algorithm III, however, will in general be less efficient than algorithm II (unless optimally built traveling salesman tours are built for the former) because for equal riding times algorithm II results in lower waiting times. This can be seen by remembering that riding times are determined by the average number of persons on the bus and that therefore if algorithm III uses *n*-person tours algorithm II must keep \( n/2 \) persons on the bus to yield the same riding times. Since an *n*-person tour takes longer than distributing one person, with an *n/2* person backlog, *n* times, algorithm II can process requests more rapidly. It thus results in shorter waiting times. On the other hand, it must be noted that algorithm III will tend to provide more reliable riding times as the maximum ride can never exceed \( C \). The reader can check that in a circular area with a radius of \( R \) miles with \( A = 0.157R^2 \) req./min., \( r = 1.27 \), \( v = 0.25 \) mi./min., \( b_1 = b_2 = 0.5 \) min., for the two different situations:

(A) \( R = 3 \) miles, \( M = 20 \) buses
(B) \( R = 7 \) miles, \( M = 200 \) buses

one gets the results in Table 1. Note that in case (A) algorithms I and II are about equal but in case (B) as was to be expected algorithm II is better than algorithms I and III.

<table>
<thead>
<tr>
<th>Case</th>
<th>Algorithm A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>31 min</td>
<td>83 min</td>
</tr>
<tr>
<td>II</td>
<td>31 min</td>
<td>53 min</td>
</tr>
<tr>
<td>III</td>
<td>37 min</td>
<td>66 min</td>
</tr>
</tbody>
</table>

### 2. Fine Tuning

In this section, we relax some of the assumptions in section 1 in an attempt to extend the asymptotic formulae to uncongested systems. Although the bulk of the discussion will concentrate on algorithm I, algorithms II and III could be similarly treated.

We model the request arrival process as a time-homogeneous Poisson process and the service rates as mutually independent negative exponential variables independent of the arrival process. Although a formal justification of these assumptions is impossible, it is clear that they are more realistic than the ones in Section 1. Furthermore, if the results obtained with the new assumptions match the results obtained in Section 1 over a certain range, it would be safe to infer that (at least over that range) these measures of dial-a-bus systems performance are fairly insensitive to the arrival and service processes and that the formulae given in Section 1 apply.

We now proceed to analyze algorithm I.

### 2.1 Algorithm I

Since under this algorithm buses go to the nearest stop regardless of whether it is an origin or a destination, we define the state of the system as the total number of customers in the system, \( X = N + \sum_{i=1}^{M} n_i \), and model \( X(t) \) as a birth and death process where a birth occurs when a request arrives to the system and a death when a departure takes place. Although rigorously speaking, \( X(t) \) is not Markovian, the statistical dependencies neglected by assuming that it is (e.g. dependency among successive service times, non-forgetful behavior of the time in between stops, etc.) cannot logically have the same importance as including random arrivals and random service times in the model.

The transition rate matrix for the Markov process is:

\[
P = \begin{bmatrix}
-\lambda & \lambda & 0 & 0 & \\
\mu_1 & -(\lambda + \mu_1) & \lambda & 0 & \\
0 & \mu_2 & -(\lambda + \mu_2) & \lambda & \\
0 & 0 & \mu_3 & -(\lambda + \mu_3) & \\
\end{bmatrix}
\]

where \( \lambda \) is the request arrival rate and \( \mu_i \) is the rate at which customers are served from the system when \( X = i \).

The problem will be solved once we obtain a reasonable expression for \( \mu_i \). Since eqn (6) is not reasonable when the system is uncongested, we develop an expression for \( \mu_i \) that will be reasonable for small values of \( i \) and will approach \( M \cdot \lambda_b \) for large values of \( i \).

We let \( z \) denote the expected fraction of empty buses (the customer in service is not counted); assuming that customers are randomly distributed among buses and home (and according to the binomial probability law) one has:

\[
z = \left( \frac{M}{M+1} \right)^{i-1} i = 1, 2, 3 \ldots
\]

Equation (21) can be regarded as a function that indicates how idle the system is; thus, for a busy system:

\[
\mu_i = (M \cdot \lambda_b) = M \left[ b_1 + b_2 + \frac{\sqrt{A}}{v} \cdot \frac{1}{\sqrt{\frac{2i}{M+1}}} \right]^{-1}
\]

and for an idle system

\[
\mu_i = \left[ b_1 + b_2 + 0.5 \frac{\sqrt{A}}{v} \cdot \left(1 + \frac{1}{\sqrt{M}}\right) \right]^{-1} \quad \text{if } z = 0 \quad (22a)
\]

Thus, as \( i \) goes from 1 to \( \infty \), \( \mu_i \) should go from the r.h.s. of eqn (22b) to the r.h.s. of eqn (22a). Equation (23), below, gives such an expression.

\[
\mu_i = \left[ \frac{M(1-z)}{b_1 + b_2 + \frac{\sqrt{A}}{v} \sqrt{\frac{M+1}{2i}}} \right]^{-1} + \frac{i \lambda^2}{b_1 + b_2 + 0.5 \frac{\sqrt{A}}{v} \left(1 + \frac{1}{\sqrt{Mz}}\right)} \quad (23)
\]

The first term of this equation represents the rate at
which the $M(1 - z)$ busy buses serve requests and the second term the rate at which the $Mz$ idle buses serve requests. In this second term, the denominator is the service time for a request which is picked up by an idle bus when there are $M - z$ idle buses; the numerator, $i \cdot z^2$, is an approximation for the average number of empty buses engaged in collection when there are $i$ awaiting requests and a fraction $z$ of the buses is empty. The numerator expression was obtained assuming that requests and individual buses are matched randomly. However, other expressions with similar characteristics (i.e. vanishing when $i = 0$ and $i \gg M$) could serve just as well. It would be reasonable for instance to use $iz^2$ instead of $iz^2$ with $\beta$ to be calibrated. This was not done, however, as eqn (23) matched well simulated results (see later results).

One can obtain the steady state probability vector, $p_i = Pr\{X = i\}$, by getting:

$$p_i = p_0 \prod_{j=1}^{i} \left( \frac{\mu_j}{\lambda_i} \right)$$  \hspace{1cm} (24a)

with

$$p_0 = \left[ 1 + \sum_{i=1}^{\infty} \left( \prod_{j=1}^{i} \left( \frac{\lambda_j}{\mu_j} \right) \right) \right]^{-1}.$$  \hspace{1cm} (24b)

The series in eqn (24) converges if $\lim (\lambda_j/\mu_j) < 1$; this condition is equivalent to $\lambda < \lambda_{\text{max}} = M(b_1 + b_2)$, the same as in the deterministic case.

The steady state probabilities as well as $E(X)$ and $\text{var}(X)$ can be easily calculated numerically. The mean $E(X)$ can be approximately calculated without resorting to the computer by finding the value of $i$ for which $\lambda = \lambda_i$. Of course, the solution to such equation, $\hat{X}$, (as opposed to the value obtained numerically, $\hat{X}$) can be regarded as coming from a deterministic model like the one in Section 1.1, only replacing eqn (6) by eqn (23).† Although obtaining $\hat{X}$ seems an involved proposition, one can find $\hat{X}$ very efficiently with standard search methods. For instance, the secant method usually needs less than five evaluations of eqn (23) and less than a minute with a pocket calculator. Figure 3 displays $\hat{X}$, $\hat{X}$ and the results of the model in Section 1.1 for $M = 1$, $\lambda = 0.5$ requests/min and $b_1 + b_2 = 30$ sec. Note that $\hat{X}$ and $\hat{X}$ coincided for all values tested and that they asymptotically approach the solution from Section 1.1 as the system becomes congested. Consequently, as was mentioned before, the stochastic elements (such as the randomness in the arrival and service processes) play a secondary role in the determination of queue lengths and $\hat{X}$ can be used confidently in practical instances.

As a final test of the model, the values of $\hat{X}$ obtained from eqn (23) were compared with the simulated data of Wilson et al. (1971). They simulated the performance of a many-to-many dial-a-bus system in a square area with a square grid network with requests arriving at random in the way described in this paper (Poisson in time, uniform and independent in space). The routing algorithm used is quite involved (it has a complicated set of rules to ensure that no customer waits or rides longer than a specified time; see Wilson et al. (1971) for a description) but is similar to algorithm I (and to a lesser extent algorithm II) because it attempts to build the most efficient tours at all times and does not divide the time of a bus into collection and distribution phases.

Figures 4 and 5, which correspond to Figs. 5-13 and 5-14 in Wilson et al.’s report, summarize the results. The slight overprediction of the model in congested conditions (when the time in the system/direct driving time ratio is much larger than one) was to be expected in

---

†Since in this case $i$ is a continuous variable that ranges from 0 to $\infty$, we define $z = 1$ if $i < 1$. Systems with $X < 1$, however, are rarely encountered in practice.
light of the inefficiencies built into Wilson's algorithm to ensure a fair service to everyone.

The peculiarities of Wilson's algorithm can be captured by replacing the \( \sqrt{(1 + M)/2I} \) term in eqn (23) by \( [(1 + M)/2I]^\alpha \); with \( \alpha \) to be calibrated. An excellent fit (see Figs. 4 and 5), was obtained for \( \alpha = 0.45 \), which means that for that particular algorithm, buses select requests for service according to a rule such that:

\[
d_r \approx 0.5 \left( \frac{A}{n} \right)^{0.45}
\]

which is slightly less efficient than eqn (2).

Although a finely tuned model of algorithm I that gives separate values for waiting and riding times could have been easily derived, such a model was not considered useful because waiting to riding time ratios are very sensitive to the routing strategy used and real systems differ substantially from one another. On the other hand, as shown by Wilson and Hendrickson (1977) in their evaluation of different models with data from several operating systems, the total time is not as sensitive to the routing strategy and one can use the finely tuned version of algorithm I for most design purposes. Algorithms II and III can be more effective predictors of waiting to riding time ratios as both of these algorithms have one degree of freedom (the number of people in the buses is an exogeneous variable) which can be adjusted to reflect the conditions of the system being modelled. A systematic way of doing that prior to observing the operation of the system is, of course, a further research issue.

2.2 Algorithms II and III

The other algorithms could also be similarly analyzed; for instance, one can model in both cases the number of persons waiting at home, \( N(t) \), as birth and death processes with death rates carefully defined for small values of \( N \) (one must take into account in these cases that the buses may not be able to keep their prescribed number of customers on board). In this case the death rate would approach eqn (22b) for \( N = 1 \), and \( M \cdot \lambda_b \) (eqns (10b) and (17)) for large values of \( N \). However, the evidence so far presented indicates that for moderate demand levels the formulae in Section 1.1 are accurate; as a matter of fact, they should be far more accurate than the formula for algorithm I, since with algorithms II and III the riding time is accurately given and riding time is the most significant part of total delay.

3. CONCLUSIONS

This paper has presented a simple model of many-to-many dial-a-bus systems that makes it possible to predict quickly and accurately average total time in the system. With it, it is possible to design and evaluate planned dial-a-bus systems without resorting to expensive (and not always available) computer simulation programs.

Section 1 discusses the modelling framework and presents asymptotic formulae for three different algorithms; the methodology can be applied to other algorithms as well. Section 2 concerns itself with the validity of the approach in Section 1 for uncongested systems. It is shown that the formulae in Section 1 can be modified to capture stochastic phenomena and that such modified formulae reproduce simulated results satisfactorily for both congested and uncongested conditions. It is also mentioned that the formulas for algorithms II and III may not have to be corrected.

Further research in analytical modelling of many-to-many dial-a-bus systems should include development of models of reliability. In addition it would also be desirable to calibrate the models in this paper to achieve greater accuracy (especially in the separate prediction of waiting and riding times) and to explore the effects of correlated origins and destinations on the capacity and level-of-service that can be provided by the system (this issue, of course, is only relevant for large service areas). Perhaps more importantly, though, the formulae should be extended to highly irregular service areas. Research in this last area is presently underway (Hendrickson, 1977).

Finally, both the results in this paper and those reported by Daganzo et al. (1977) can be embedded into a larger project aimed at evaluating the relative merits of different public transportation schemes (e.g. many-to-many, many-to-one as a feeder to scheduled line haul, closely spaced line-haul, etc. . .) and at improving the methodology for selecting among alternate public transportation systems.

Acknowledgements—The author of this paper benefitted from interaction with Donald E. Cleveland of the University of Michigan during the early part of this research and Nigel H. M. Wilson of the Massachusetts Institute of Technology in the latter stage. Professor Wilson's remarks on the last draft of this paper were especially helpful.

REFERENCES


