Annals of Operations Research manuscript No.
(will be inserted by the editor)

Minimizing Average Passenger Waiting Time in Personal Rapid Transit Systems

John D. Lees-Miller

Received: date / Accepted: date

Abstract Personal Rapid Transit (PRT) is an emerging urban transport mode. A PRT system operates much like a conventional hackney taxi system, except that the vehicles are driven by computer (no human driver) between stations in a dedicated network of guideways. The world's first two PRT systems began operating in 2010 and 2011. In both PRT and taxi systems, passengers request immediate service; they do not book ahead. Perfect information about future requests is therefore not available, but statistical information about future requests is available from historical data. If the system does not use this statistical information to position empty vehicles in anticipation of future requests, long passenger waiting times result, which makes the system less attractive to passengers, but using it gives rise to a difficult stochastic optimisation problem. This paper develops three lower bounds on achievable mean passenger waiting time, one based on queuing theory, one based on the static problem, in which it is assumed that perfect information is available, and one based on a Markov Decision Process model. An evaluation of these lower bounds, together with a practical heuristic developed previously, in simulation shows that these lower bounds can often be nearly attained, particularly when the fleet size is large. The results also show that low waiting times and high utilisation can be simultaneously obtained when the fleet size is large, which suggests important economies of scale.

Keywords Personal Rapid Transit · Empty Vehicle Redistribution · Taxi · Queuing Model · Markov Decision Process · Waiting Time

1 Introduction

Personal Rapid Transit (PRT) is an emerging urban transport mode. It uses small, computer-guided vehicles to carry individuals and small groups between pairs of stations on a dedicated network of guideways. The vehicles operate on-demand and provide direct service from origin station to destination station. Two PRT systems are now operational, one at Masdar in Abu Dhabi (2getthere 2011), and one at Heathrow Airport in London (ULTra

John D. Lees-Miller
Transportation Research Group
University of Southampton
E-mail: J.Lees-Miller@soton.ac.uk

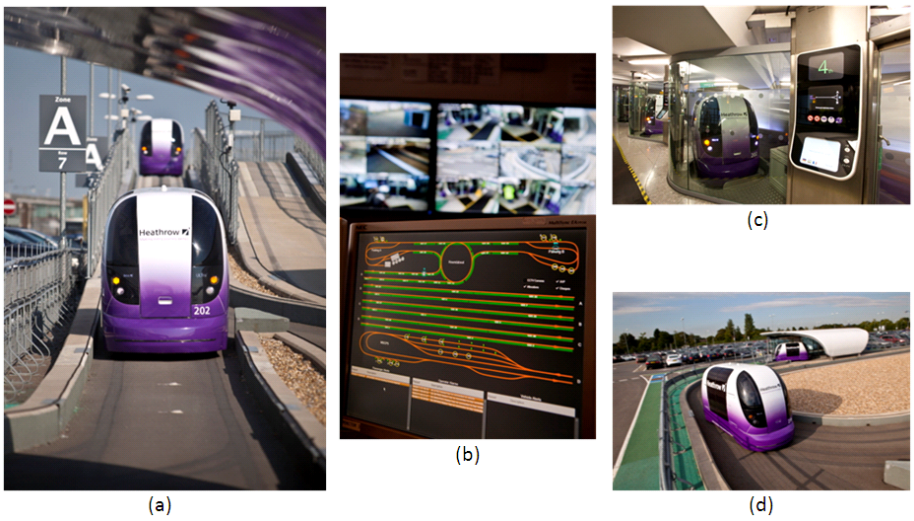


Fig. 1 Heathrow Pod driverless vehicles and guideway (a), control room (b), vehicle and bay in the Terminal 5 station (c) and vehicles and station in Business Parking (d). PRT vehicles, stations and infrastructure are smaller than typical Automated People Mover and urban rail systems. Each vehicle seats four passengers with luggage. Photographs courtesy of ULTra PRT Ltd, 2011.

PRT 2010). The 'Heathrow Pod' PRT system is a last mile circulator with three stations and twenty-one driverless vehicles (Fig. 1), that connects Business Parking with Terminal 5. Many other recently proposed PRT systems provide connections between train stations, bus stations or off-site car parks and a wide range of destinations (Bly and Teychenne 2005). Used in this way, PRT can increase the efficacy of other public transport modes. In order for PRT to do this, it must provide a high-quality service, which means low passenger waiting times, low travel times and high levels of safety and comfort.

The focus of this paper is on passenger waiting times. PRT systems operate much like conventional taxi systems. Passengers request immediate service (that is, they do not book ahead) from their origin station to their chosen destination station. A central control system can move empty vehicles reactively, in response to a request that has just been received, and proactively, in anticipation of future requests. The empty vehicle redistribution (EVR) problem is to decide which vehicles to move and where to move them. Without proactive movements, empty vehicles tend to wait idle at stations where there is a net inflow of occupied vehicles. This leads to long passenger waiting times at stations where there is a net outflow of occupied vehicles (Lees-Miller et al. 2010). Proactive empty vehicle movements, which must be based on statistical knowledge of passenger demand, are therefore required in order to provide low waiting times. Several practical and fairly effective heuristics have been developed that make proactive empty vehicle movements (Lees-Miller and Wilson 2011, 2012). Lower bounds on mean passenger waiting times are desirable in order to assess these heuristics in absolute terms. They may also provide insight into the basic performance characteristics of demand responsive transportation systems that use small vehicles, including PRT and taxis, in a manner that is independent of the details of the algorithms used to operate them.

This paper develops three lower bounds on mean passenger waiting times using different methods. The first is based on a queuing model (Sect. 4) that is constructed to have the

same capacity (as defined in Sect. 3) as the PRT system that it models. The second lower bound is obtained from a *static* version of the EVR problem, in which perfect information is available about future requests (Sect. 5). This is a type of vehicle routing problem (VRP) (Toth and Vigo 2002), and it is NP-hard. It is not presently possible to obtain provably optimal solutions for usefully large instances, but a simple constructive heuristic, here called Static Nearest Neighbours (SNN), produces good solutions (Lees-Miller and Wilson 2012) that give an estimate of this lower bound. The third lower bound is provided by a Markov Decision Process (MDP) model (Sect. 6). MDPs are a formalism for modelling discrete time control problems that involve uncertainty. For small systems, on which the MDP model can be solved exactly, it is possible to obtain a provably optimal solution to the EVR problem for a given system and demand. The three lower bounds, along with a practical heuristic called Sampling and Voting (SV) from Lees-Miller and Wilson (2011), are then evaluated on the Heathrow Pod system (Sect. 7).

The EVR problem in PRT is a kind of dynamic VRP. The majority of the VRP literature is concerned with static problems, in which all of the requests are known in advance, but significant work has been done on extending VRP methods to fully or partially dynamic settings, in which all or some requests are received as the system is operating; see Berbeglia et al. (2007, 2010) for recent surveys. The most closely related work found in the VRP literature is that of Yang et al. (2004), Bent and Van Hentenryck (2004) and Hvattum et al. (2006). The SV method considered in Sect. 7 is based on ideas from these papers. Concepts from queuing theory have been applied to related fully dynamic VRPs (Bertsimas and Levi 1996; Swihart and Papastavrou 1999). However, none of these is an exact match for the EVR problem.

PRT has much in common with a conventional taxi service, and the principle of proactive empty vehicle movement also applies to conventional taxis. The use of GPS-enabled smart phones to make short-notice taxi bookings is a recent development (e.g., `www.uber.com`) that is blurring the line between conventional street-hail (hackney) taxis, for which passengers generally do not book ahead, and private hire vehicles, for which passengers must book ahead. The lower bounds developed here might also be of interest for such services, especially in view of the fact that they are currently premium services with users who have a high value of time. Most existing work on taxi dispatch focuses on reactive algorithms (Horn 2002; Bell and Wong 2005; Seow et al. 2010) and related operational challenges such as travel time estimation (Lee et al. 2004) and the handling of both advanced bookings and immediate requests (Horn 2002; Wang et al. 2009). Approaches to proactive movement include random roaming (Lee et al. 2004), the 'go to hotspot' heuristics of Li (2006), and the 'rank homing' heuristics of Horn (2002). Similar problems that involve the repositioning of idle vehicles arise in the control of automated guided vehicle (AGV) systems (Vis 2006) and elevators (Wesselowski and Cassandras 2006).

2 Assumptions and Notation

The three main factors that determine passenger waiting times are congestion on the guideway, congestion in stations and the availability and locations of empty vehicles. For very large systems with many vehicles, congestion effects will often be significant, but most PRT systems proposed for the near and medium term will operate well below the congested limit. This motivates the following simplifying assumptions.

1. Congestion on the guideway is ignored, so vehicles take quickest paths, and the travel times between stations are constants.

2. Congestion in stations is ignored; any delays in stations are constants included in the travel times.

Under these assumptions, the relevant characteristics of a PRT network are the number of stations and the quickest path times between those stations. Let S denote the set of stations, and let n_S denote the number of stations. For each pair of distinct stations i and j in S, let t_{ij} denote the quickest path *trip time* from i to j, in minutes, and define constants $t_{ij} = 0$ when $i = j$. Taken together, these t_{ij} form a *trip time matrix*. It is further assumed that the trip times satisfy the strict triangle inequality,

$$t_{ih} + t_{hj} > t_{ij} \tag{1}$$

for any three distinct stations i, j and h in S. That is, even if station h is 'on the way' from station i to station j, there is a time penalty if the vehicle stops at h, due to acceleration and deceleration, passenger loading and unloading, or congestion in the station.

The unit of passenger demand will be the *request*. Each request results in a single occupied vehicle trip, which may be for an individual passenger or a small party of passengers travelling together by choice. This ignores *ride sharing*, in which parties can combine to share a vehicle, which can significantly increase capacity (Lees-Miller et al. 2009). Each request has associated with it an origin station, a destination station and the time at which the system receives the request. It is assumed that every request is for immediate travel from its origin station, so the *waiting time* of a request is the delay between when the system receives it and when a vehicle picks up the request.

It is assumed that requests for travel from station i to station j are received according to a Poisson process with rate d_{ij} in requests per minute, where $d_{ij} = 0$ if $i = j$ (no recreational trips). The Poisson processes for pairs of stations are assumed to be independent and stationary (that is, the d_{ij} do not vary with time). Taken together, these d_{ij} form a *demand matrix*. It is assumed that the demand matrix is known from historical data.

The capacity of a vehicle is defined to be one request. It is assumed that the number of vehicles in the fleet is fixed, and that the vehicle fleet is homogeneous (that is, vehicles are interchangeable). The set of vehicles will be denoted K, and n_K will denote the fleet size.

3 Capacity and the Fluid Limit

The waiting times that can be achieved for a given demand on a given system depend strongly on how close the demand is to the system's theoretical maximum capacity. Under the above assumptions, there is a well-known (Anderson 1978) method for computing the capacity of a PRT system. This method can also be derived from the urban taxi economics model of Yang et al. (2002). The method works at a macroscopic level, where the variables are long run average *flows* of vehicles between stations. The flows of occupied vehicles are given by the demand matrix. We want to determine the empty vehicle flows required to balance these occupied vehicle flows, and short empty trips are preferred. Let x_{ij} be the flow of empty vehicles from station i to station j, with $x_{ij} = 0$ if $i = j$. This gives a minimum cost network flow problem:

$$\min \sum_{i,j \in S} t_{ij} x_{ij} \tag{2}$$

$$\text{s.t.} \sum_{j \in S} (d_{ij} + x_{ij}) = \sum_{j \in S} (d_{ji} + x_{ji}) \qquad \forall\, i \in S \tag{3}$$

$$x_{ij} \geq 0 \qquad \forall\, i \in S,\ j \in S \tag{4}$$

The objective function (2) is the number of vehicles that will, on average, be required to sustain the flows of empty vehicles (minutes times vehicles per minute gives vehicles). Constraints (3) are flow conservation constraints that ensure that the total number of vehicles remains constant. This problem can be solved with standard techniques (Bertsimas and Tsitsiklis 1997).

An optimal solution, x^*_{ij}, to the problem (2) defines the long run average flows of empty vehicles between each pair of stations. It also determines the minimum number of vehicles required to serve the demand, which is

$$n^*_K = \sum_{i,j \in S} t_{ij}(d_{ij} + x^*_{ij}). \tag{5}$$

The ratio

$$\rho = n^*_K / n_K \tag{6}$$

is then a measure of how close the demand is to the maximum theoretical capacity of the system. It therefore acts like a *utilisation* or demand *intensity* factor in queuing theory: when $\rho > 1$, requests are being received faster than the system can serve them, because it does not have enough vehicles, so the queues of waiting passengers, and their waiting times, grow without bound. This connection with queuing theory will be made more concrete in the next section.

4 An $M/G/s$ Queuing Model

The aim here is to use the results of the fluid limit problem (2) to estimate passenger waiting times. To this end, we will model the whole PRT system as a multi-server queuing system in which the 'customers' are passengers and the 'servers' are vehicles. The key feature of this queuing system is that its demand intensity is the same as that for the PRT system that it models. It will be seen that the queuing system is an exact model of the corresponding PRT system when all requests originate from a single station, but when there are multiple origin stations, it tends to underestimate achievable passenger waiting times.

Formally, the model is an $M/G/s$ queue, in Kendall notation (Adan and Resing 2002). The M, for Markovian, means that requests are received according to a Poisson process. The G refers to a General service time distribution, a particular instance of which will be defined below. The s denotes the number of servers, which is the fleet size, n_K.

The mean rate for the customer arrival process is the total request rate for the whole system, $\sum_{i,j} d_{ij}$. The main task is to define the service time distribution. The service time for a single request is taken to be the sum of the occupied and empty trip times required to serve the request. Each request requires one occupied vehicle trip, with origin and destination as determined by the request, and zero or more empty vehicle trips to the origin of the next request that happens to be assigned to the same vehicle. The sequence of empty trips can be viewed as a random walk through the network. However, we will see that if an empty vehicle moves according to the optimal empty vehicle flows x^*_{ij}, then this random walk consists of at most one empty trip. Each request therefore involves three stations: the request's origin and destination, and the empty trip's destination, which may be same as the request's destination, if there is no empty vehicle trip. Let the random variables I, J and H denote these three stations, respectively. To simplify notation, let

$$d_{i\sigma} = \sum_{j \in S} d_{ij}, \quad d_{\sigma i} = \sum_{j \in S} d_{ji}, \quad \text{and} \quad d_{\sigma\sigma} = \sum_{i,j \in S} d_{ij}$$

denote the row, column and total sums of the demand matrix, and define $x^*_{i\sigma}$ and $x^*_{\sigma i}$ similarly. The demand matrix directly defines the joint distribution of I and J, namely

$$\Pr(I = i, J = j) = \frac{d_{ij}}{d_{\sigma\sigma}}. \tag{7}$$

To define the distribution of the empty trip's destination, H, conditional on the occupied trip's destination, J, we proceed as follows. Consider a vehicle that has just completed an occupied trip to station $j \in S$; it may now make an empty trip to another station, or it may stay at j. Define

$$p_{jh} = \frac{x^*_{jh}}{d_{j\sigma} + x^*_{j\sigma}} \tag{8}$$

as the probability that the vehicle's next trip is to station $h \in S$, when $h \neq j$. In (8), the numerator is the empty flow out of station j, and the denominator is the total (occupied plus empty) flow out of station j. The probability that the vehicle stays at j is then set to

$$p_{jj} = 1 - \sum_{\substack{h \in S \\ j \neq h}} p_{jh} \tag{9}$$

so that the p_{jh} sum to one. We then have $\Pr(H = h | J = j) = p_{jh}$.

The reason that there is at most one empty vehicle trip when the empty vehicles move according to the probabilities p_{jh} is that, because the trip times satisfy the strict triangle inequality (1), any optimal solution to problem (2) must satisfy

$$x^*_{\sigma i} > 0 \implies x^*_{i\sigma} = 0 \tag{10}$$

for all stations i. That is, it is never optimal (in the fluid limit) for a station to have empty vehicle flow both in and out; it would always be better to route the flow directly to its eventual destination. The number of empty trips in the random walk is therefore zero if $x^*_{jh} = 0$ for all $h \neq j$, since this implies $p_{jh} = 0$ for all $h \neq j$, and $p_{jj} = 1$. Otherwise, we have $x^*_{\sigma h} > 0$, and (10) implies that $x^*_{h\sigma} = 0$, so there is no empty flow out of h, and the vehicle's random walk ends there, after one trip.

It should also be noted that the denominator in (8) is zero if and only if a station has no demand in or out (that is, $d_{i\sigma} = d_{\sigma i} = 0$), so stations with no demand must be excluded from the queuing model. This does not affect the result, because the strict triangle inequality for trip times also ensures that any such station also has $x^*_{i\sigma} = x^*_{\sigma i} = 0$ in any optimal solution, so empty vehicles moving according to the optimal flows never visit stations with no demand.

We can now define the service time of a request as a random variable

$$\tau = t_{IJ} + t_{JH} \tag{11}$$

with distribution defined by the joint distribution of I, J and H. The relevant figure for computing the demand intensity for the queuing system is the expected service time, $\mathrm{E}[\tau]$, which is given by $\mathrm{E}[t_{IJ}] + \mathrm{E}[t_{JH}]$. The expected occupied trip time is

$$\begin{aligned} \mathrm{E}[t_{IJ}] &= \sum_{i \in S} \sum_{j \in S} t_{ij} \Pr(I = i, J = j) \\ &= \frac{\sum_{i,j \in S} t_{ij} d_{ij}}{d_{\sigma\sigma}} \end{aligned} \tag{12}$$

using (7). For the expected empty trip time, we need the joint distribution of J and H, which is

$$\begin{aligned}\Pr(J=j,H=h) &= \Pr(H=h|J=j)\Pr(J=j)\\ &= \Pr(H=h|J=j)\left(\sum_{i\in S}\Pr(I=i,J=j)\right)\\ &= \frac{p_{jh}d_{\sigma j}}{d_{\sigma\sigma}}\end{aligned}$$

by summing out the possible origin stations, I. The expected empty trip time is then

$$\begin{aligned}\mathrm{E}[t_{JH}] &= \sum_{j\in S}\sum_{h\in S} t_{jh}\frac{d_{\sigma j}p_{jh}}{d_{\sigma\sigma}} \qquad (13)\\ &= {d_{\sigma\sigma}}^{-1}\sum_{j\in S} d_{\sigma j}\left(\sum_{\substack{h\in S\\ h\neq j}} t_{jh}p_{jh} + t_{jj}p_{jj}\right)\\ &= {d_{\sigma\sigma}}^{-1}\sum_{j\in S} d_{\sigma j}\sum_{\substack{h\in S\\ h\neq j}} t_{jh}\left(\frac{x_{jh}}{d_{j\sigma}+x^*_{j\sigma}}\right)\\ &= {d_{\sigma\sigma}}^{-1}\sum_{j\in S}\left(\frac{d_{\sigma j}}{d_{j\sigma}+x^*_{j\sigma}}\right)\sum_{\substack{h\in S\\ h\neq j}} t_{jh}x^*_{jh}\\ &= \frac{\sum_{j,h\in S} t_{jh}x^*_{jh}}{d_{\sigma\sigma}}\end{aligned}$$

where the last equality holds because for each j, either $x^*_{\sigma j}=0$, in which case

$$\frac{d_{\sigma j}}{d_{j\sigma}+x^*_{j\sigma}} = \frac{d_{\sigma j}+x^*_{\sigma j}}{d_{j\sigma}+x^*_{j\sigma}} = 1$$

by flow conservation (3), or $x^*_{\sigma j}>0$, in which case $x^*_{jh}=0$ for all h, by (10). So, in summary, the expected service time is

$$\mathrm{E}[\tau] = \frac{\sum_{i,j\in S} t_{ij}(d_{ij}+x^*_{ij})}{d_{\sigma\sigma}} = \frac{n^*_K}{d_{\sigma\sigma}} \qquad (14)$$

where n^*_K is the fleet size estimate (6) from the fluid limit.

The intensity of the demand on an $M/G/s$ queuing model is $\rho^{M/G/s}=\lambda/(\mu s)$ where λ is the mean customer arrival rate, μ is the mean service rate (per server), and s is the number of servers. As with the intensity measure defined for the fluid limit in (6), $\rho^{M/G/s}>1$ means that customers are arriving faster than the system can serve them. For the model developed here, $\lambda = d_{\sigma\sigma}$, $\mu = 1/\mathrm{E}[\tau]$, and $s=n_K$, so this condition is equivalent to

$$\rho^{M/G/s} = \frac{\lambda}{\mu s} = \frac{d_{\sigma\sigma}}{(d_{\sigma\sigma}/n^*_K)n_K} = \frac{n^*_K}{n_K}$$

which is the same as the intensity (6) computed for the fluid limit.

The queuing model defined here therefore matches the PRT system that it models in at least one important way: the waiting times diverge as the intensity of the demand approaches

one. The main limitation of this queuing model is that, while it accurately represents the total empty vehicle travel required according to the fluid limit calculation, it does not represent the case where a vehicle becomes idle at the 'wrong' station for the current request. That is, there may be an idle vehicle (and so a free server in the system), but it may not be located at the same station as the passenger, in which case the queuing model predicts a zero wait, but the passenger must actually wait for the vehicle to move empty to his origin station. When all of the demand originates at a single station (and empty vehicles are proactively moved back to that station), the queuing model is exact, because vehicles never become idle at the wrong station. When there are multiple origins, the waiting times predicted by the queuing model may not be achievable in practice.

To estimate the mean waiting times for the queuing model, we use simulation, because no exact analytical results are known for the mean waiting time in an $M/G/s$ queue when $s > 1$. When $s = 1$, the Pollaczek-Khinchine formula gives a closed-form expression for the mean waiting time, but no such closed form is known for the multi-server case (Adan and Resing 2002). A large literature is available on approximations for the $M/G/s$ queue; see Boxma et al. (1979) and, for a recent survey, Gupta et al. (2010). Closed form results are known for the $M/M/s$ queue (Adan and Resing 2002), in which the service time distribution is Markovian. However, Markovian service times with a sufficiently large mean tend to be over-dispersed relative to the distributions obtained from (7), (8) and (9) for τ. There are efficient numerical schemes for the $M/D/s$ queue (Tijms 2006), in which service times are deterministic, which is the case for some special networks and demands.

Whereas the queuing model defined in this section includes an implicit assumption that empty vehicles do not become idle at the 'wrong' stations, the next section describes how vehicle routes could be planned to avoid this, if perfect information about future requests were available.

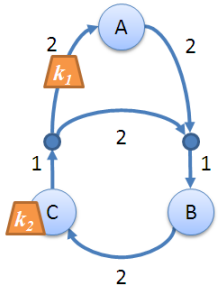
5 The Static Problem

The static EVR problem, in which all of the requests to be served are known in advance, will be formulated on a *vehicle-request graph* that is constructed from the PRT system. We have so far described the EVR problem in terms of vehicles moving between stations. In what follows, however, it will be more convenient to think about vehicles moving between requests. Let the vehicle-request graph, G, be a directed, weighted graph with node set $K \cup R \cup \{s\}$, where R is a set of request nodes, K is a set of vehicle nodes, and s is a special *vehicle sink node*. Each vehicle begins at its vehicle node, visits zero or more request nodes, and then terminates at the sink node. The sink node is just a technicality; it does not correspond to anything physical. The arc set is

$$E = \{(u,v) : u \in K \cup R, v \in R \cup \{s\}, u \neq v\}$$

and the arc weights are delays, defined to encode the structure of the original PRT network, as follows. For readability, we will write the trip times t_{ij} as $t(i,j)$. For each request $u \in R$, let i_u, j_u and e_u denote its origin station, destination station and time of receipt, respectively. We will assume that the first request is received at time 0. The vehicles may initially be idle at stations, or they may still be serving passengers that arrived in the past, before the set of requests being considered now. For each vehicle $u \in K$, define constants j_u and e_u so that vehicle u first becomes idle at station j_u at time e_u; if the vehicle is initially idle, then $e_u = 0$.

(a) The PRT network and initial vehicle positions.



(b) The requests to be served.

r_0	A → B at 0.0min (now)
r_1	C → B at 0.3min
r_2	C → A at 1.2min

(c) The resulting vehicle-request graph.

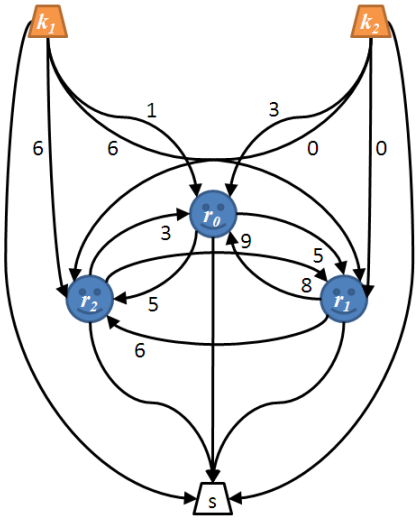


Fig. 2 An example of a vehicle-request graph. The PRT network (a) includes stations, merges and diverges, but only quickest-path times are important for the vehicle-request graph. The arc weights are travel times, in minutes. There are two vehicles, k_1 and k_2; k_1 will arrive in station A in one minute, and k_2 is idle at station C. A complete list of requests (b) is provided, including times of receipt, origins and destinations; here there are three requests. The resulting vehicle-request graph (c) has a node for each vehicle and each request, and a sink node, s. Arc weights correspond to vehicle travel times. Suppose that, for example, vehicle k_1 serves requests r_0, r_1 and r_2, in that order. First, it finishes its current trip to station A, which takes 1 minute; then it arrives at A, which is r_0's origin. So, the arc (k_1, r_0) has weight 1. It then serves r_0 by moving from A to B, which takes 3 minutes, and moves empty for 2 minutes from r_0's destination, B, to r_1's origin, C; so, arc (r_0, r_1) has weight $2+3=5$. This repeats for r_2, and then the vehicle terminates at the sink node.

The arc weights

$$w_{uv} = \begin{cases} e_u + t(j_u, i_v) & u \in K,\ v \in R \\ t(i_u, j_u) + t(j_u, i_v) & u, v \in R,\ u \neq v \\ 0 & u \in K \cup R,\ v = s \end{cases}$$

are defined to include both occupied travel time (e_u if $u \in K$ or $t(i_u, j_u)$ if $u \in R$) and empty travel time. Note that if $j_u = i_v$, then there is no empty vehicle trip required, and $t(j_u, i_v) = 0$. Also, since the $t(i, j)$ satisfy the triangle inequality (1), so do the arc weights. An example construction is given in Fig. 2.

The problem can then be stated using the following arc flow formulation. For each arc $(u, v) \in E$, let x_{uv} be a binary variable; $x_{uv} = 1$ signifies that a vehicle traverses arc (u, v), and $x_{uv} = 0$ signifies that no vehicle traverses that arc. For example, if u and v are request nodes, then $x_{uv} = 1$ means that a vehicle serves request u and then proceeds to pick up request v. For each request u, let the variable t_u be the time at which a vehicle picks up that request; this is the time at which the vehicle departs from the request's origin station. It is also convenient

to define constants $t_u = 0$ for $u \in K$. The Static EVR problem is then:

$$\min \sum_{u \in R} t_u \tag{15}$$

$$\text{s.t.} \sum_{v:(u,v)\in E} x_{uv} = 1 \qquad \forall\, u \in K \cup R \tag{16}$$

$$\sum_{u:(u,v)\in E} x_{uv} = 1 \qquad \forall\, v \in R \tag{17}$$

$$\sum_{u:(u,v)\in E} x_{us} = n_K \tag{18}$$

$$x_{uv} = 1 \implies t_u + w_{uv} \leq t_v \qquad \forall\, (u,v) \in E, v \neq s \tag{19}$$

$$e_u \leq t_u \qquad \forall\, u \in R \tag{20}$$

$$t_u \leq t_v \qquad \forall\, (u,v) \in E_{\text{FCFS}} \tag{21}$$

$$x_{uv} \in \{0,1\} \qquad \forall\, (u,v) \in E \tag{22}$$

Request u's waiting time is $t_u - e_u$, so the objective (15) is equivalent to minimising the total waiting time, since the e_u are constants, and hence the average waiting time, since all requests are served. Constraints (16), (17) and (18) ensure that each request is served by exactly one vehicle, and that each vehicle terminates at the sink node. (Constraint (18) is actually redundant.) The vehicle trips can be determined by starting from each vehicle node and following the arcs (u,v) for which $x_{uv} = 1$, through zero or requests, to the sink node. Constraints (19) ensure that pickup times are updated according to the vehicle flows. These are conditional constraints: if a vehicle serves request u and then request v (that is, if $x_{uv} = 1$), the vehicle must serve u and then move from his destination to v's origin before picking up v. Constraints (20) are *one-sided time window constraints* that ensure that request u is not picked up before it is received, at e_u. Constraints (21) are *precedence constraints* that ensure that requests with the same origin station are served in first-come-first-served order, with

$$E_{\text{FCFS}} = \{(u,v) : u \in R,\ v \in R,\ i_u = i_v,\ e_u < e_v\}.$$

The static EVR problem is NP-hard, because it generalises the minimum latency version of the asymmetric travelling salesman path problem, which is NP-hard (Nagarajan and Ravi 2008). So far, some small instances (10–20 requests) have been solved exactly with mixed integer linear programming using the technique of van Eijl (1995) to linearise the conditional constraints (19). However, here we are interested mainly in average waiting times over large numbers (thousands) of requests, for which exact solution is not currently feasible, so we rely on a heuristic. The heuristic used here is a simple nearest-neighbour method, in which the requests are considered in ascending order by e_r, and the vehicle that minimises the request's waiting time is assigned, subject to a number of tie-breaking rules; see Lees-Miller and Wilson (2011) for details.

The waiting times that can be attained on the static problem are not generally attainable in practice, because the PRT system does not have the benefit of perfect information about future requests — it has only statistical information. The model developed in the next section takes this uncertainty about future requests into account.

6 The MDP Model

MDPs are a formalism for modelling discrete time control problems in which the outcome is at least partially due to random chance. At each time step, the process is in a state s, and

the decision maker takes an action a, that is valid in this state. The process then moves to a new state, s', for the next time step, with a given probability $\Pr(s,a,s')$. Each state has an associated reward, $R(s)$, which the decision maker receives upon entering state s. The aim is to find a corresponding *policy* that the decision maker can follow to choose which action to take in each state; in its simplest form, a policy is a table that maps states to actions. An *optimal* policy is one that maximises the (discounted) sum of the rewards that the decision maker receives over time.

This section shows how to define an MDP for a given scenario — that is, for a given trip time matrix, demand matrix and fleet size. In particular, we will define the states, the actions that are valid in those states, the transition probabilities that relate states and actions, and the structure of the rewards. The rewards are defined so that maximum total reward corresponds to minimum total passenger waiting time. An optimal policy is thus an optimal empty vehicle redistribution strategy for the corresponding scenario (subject to the assumptions in Section 2 and the simplifications below).

A range of standard techniques are available to find good policies. For small MDPs, provably optimal policies can be obtained by dynamic programming (Puterman 2005), which is the method used in this paper. For larger systems, which generate larger MDPs on which dynamic programming is computationally infeasible, algorithms such as approximate dynamic programming (Bertsekas and Tsitsiklis 1996; Powell 2007), reinforcement learning (Sutton and Barto 1999) and receding horizon control (Wesselowski and Cassandras 2006) can produce high-quality policies.

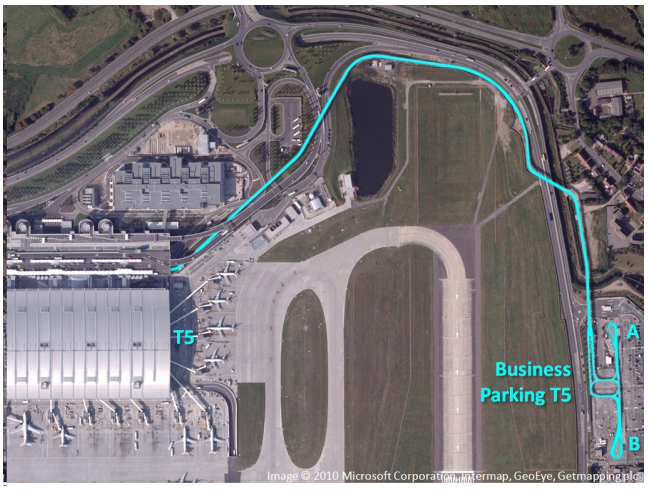
The relevant system state consists of (i) the queue of requests at each station, and (ii) the position of each vehicle. Here we will represent only the queue length; this effectively means that passengers do not tell the system their destination until a vehicle picks them up. This is sometimes the case in practice, but if destinations are known, as is usually the case in PRT, it is a somewhat pessimistic assumption at high demand. To represent vehicle positions efficiently, we make two observations. The first is that vehicles are interchangeable, and the second is that it is not necessary to know where a vehicle is coming from, but only where it is going to and how long it will be until it gets there. These imply that it is sufficient to know only the number of vehicles with each possible (discretised) 'time until arrival' at each station. The system state is then as follows.

Time is taken to be discrete with unit time steps, with all t_{ij} integer. At time $t = 0, 1, \ldots$, let q_i be the number of requests at station i, and let b_{ik} be the number of vehicles that are k time steps away from station i, for $k = 0, \ldots, t_i^{\max} - 1$, where $t_i^{\max} = \max_j t_{ji}$ is the duration of the longest trip to station i. Here, b_{i0} is the number of vehicles that are either idle at station i or will become idle before time $t+1$ (unless they are used by passengers or sent away empty). Every vehicle is counted in exactly one b_{ik}, so $\sum_{i,k} b_{ik} = n_K$ in any valid state. The system state is then a vector with components for all of the q_i and b_{ik}. Fig. 3 gives an example of how to encode the state of the system in this way.

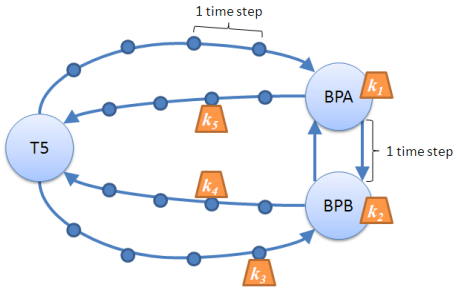
The action taken at time t determines the empty vehicle movements that the system will make over the course of the time step. It can in principle send any of the b_{i0} vehicles at station i empty to any other station. However, if there are passengers queuing at station i, it is undesirable to have the system move their vehicles away empty. So, queued requests are given priority over empty vehicle movements by taking

$$\bar{a}_i = \max\{0, b_{i0} - q_i\}$$

as the number of vehicles available to be moved empty from station i. An action is then specified by a matrix $\mathbf{A}$ with non-negative integer entries a_{ij} that give the number of empty



(a)



(b)

i	b_{i0}	b_{i1}	b_{i2}	b_{i3}	b_{i4}
BPA	1	0	0	0	0
BPB	1	1	0	0	0
T5	0	0	0	2	0

(c)

Fig. 3 (a) The guideway alignment of the Heathrow Pod Personal Rapid Transit system. The system has three stations, T5 (Terminal 5), BPA (Business Parking A) and BPB. (b) Illustration of the MDP state space for this system. The trip time between BPA and BPB (or back) is roughly one minute, and the trip time from T5 to either car park station (or back) is roughly five minutes, so we choose one minute as the time step for the MDP model. The time steps are marked by filled circles. In this example, there are five vehicles, $k_1,\ldots,k_5$. (c) The 'time until arrival' representation for the vehicle positions in (b). Note that k_4 and k_5 are both three time steps from T5, so the corresponding entry in the table is 2, and also that the sum of the entries is equal to the fleet size.

vehicles to move from station i to station j. The sum of the off-diagonal entries in each row, $a_i = \sum_{j,j\neq i} a_{ij}$, must satisfy $a_i \leq \bar{a}_i$ for any valid action.

We now turn to the description of all possible successor states and the corresponding transition probabilities. The randomness in the system is from the passenger requests. Let the random variable N_i be the number of requests in $[t,t+1)$ at station i; it is assumed that the N_i has a Poisson distribution with rate parameter $d_{i\sigma}$. Some of these N_i requests may be served by vehicles in this time step, and the rest will be added to the queue at station i. This means that each state technically has an infinite number of successor states, and that the state space is infinite, because N_i is unbounded above. In order to use dynamic programming, the state space must be made finite, so we truncate the queues at an arbitrary positive length, $q_{\max}$, beyond which requests are discarded. The largest number of new requests at i that

must be considered is then

$$\bar{N}_i = q_{\max} - q_i + b_{i0} - a_i$$

since there are q_i requests in the queue already, and up to $b_{i0} - a_i$ vehicles are available to serve (new or queued) requests this time step.

Similarly, the total number of requests at i, either new or queued, that the system serves in the current time step is

$$V_i = \min\{q_i + N_i, b_{i0} - a_i\}.$$

Each of these requests has a destination, which is also random. Let the random variable M_{ij} be the number of requests from station i to station j, and let the random vector $\mathbf{M}_i = (M_{i,1}, \ldots, M_{i,n_S})$ denote the numbers of trips from i to all destinations; $\mathbf{M}_i$ then has a multinomial distribution for V_i events, where the probability of the event corresponding to a request to station j is $d_{ij}/d_{i\sigma}$.

We can now define the next state in terms of these random variables. The number of queued requests at time $t+1$ will be

$$q_i' = q_i + N_i - V_i$$

and the 'times until arrival' at time $t+1$ will be

$$b_{ik}' = \begin{cases} \theta_{ik} + b_{i,k+1} + b_{ik} - \sum_j (M_{ij} + a_{ij}), & k = 0 \\ \theta_{ik} + b_{i,k+1}, & k = 1, \ldots, t_i^{\max} - 2 \\ \theta_{ik}, & k = t_i^{\max} - 1 \end{cases}.$$

where θ_{ik} is the number of vehicles (occupied or empty) that start new trips to station i that are $k+1$ time steps long; it is given by

$$\theta_{ik} = \sum_j [\mathrm{I}(t_{ji} = k+1)(M_{ji} + a_{ji})]$$

where $\mathrm{I}(\cdot)$ is an indicator function that is 1 if its condition is satisfied and 0 otherwise.

Successor states s' are generated by enumerating all possible numbers of requests, N_i, from 0 to $\bar{N}_i$, and, for each of these, enumerating all feasible values for the M_{ij}. The transition probability $\Pr(s, a, s')$ for each such s' is determined by the probability mass functions of the random variables N_i and $\mathbf{M}_i$. Let n_i and $\mathbf{m}_i$ be the particular values of N_i and $\mathbf{M}_i$ that yield the successor state s'. The random variables are all mutually independent, so

$$\Pr(s, a, s') = \prod_{i \in S} f(n_i) \prod_{i \in S} g(\mathbf{m}_i)$$

where

$$f(n_i) = \begin{cases} \Pr(N_i = n_i), & n_i < \bar{N}_i \\ \Pr(N_i \geq n_i), & n_i = \bar{N}_i \end{cases}$$

are the Poisson probabilities and

$$g(\mathbf{m}_i) = \frac{V_i!}{\prod_{j \in S} m_{ij}!} \prod_{j \in S} \left(\frac{d_{ij}}{d_{i\sigma}} \right)^{m_{ij}}$$

is the probability mass function for the multinomial distribution.

Finally, the reward, $R(s)$, associated with each state is chosen to be the negative sum of the queue lengths at all stations. If a passenger is left waiting for several time steps, the

accumulated negative reward is the passenger's waiting time. It is also possible to penalise empty vehicle movements by defining rewards $R(s,a)$ that depend on the action, but here we will focus only on waiting times.

Now that the MDP is fully specified, small instances can be solved using policy iteration (Puterman 2005), which computes a value, $V(s)$, and an associated action, $\pi(s)$ for each state. The values are initialised to 0 for all states. The algorithm then proceeds in two steps. The first step is to find a policy that is greedy with respect to the current value function, namely

$$\pi(s) = \arg\max_a \sum_{s'} \Pr(s,a,s') \left(R(s) + \gamma V(s')\right) \tag{23}$$

where $\gamma \in (0,1)$ is a discount factor that reduces the value of rewards received far in the future and is required to ensure convergence. The second step is to refine the value function estimates for this policy, according to

$$V(s) = \sum_{s'} \Pr(s,\pi(s),s') \left(R(s) + \gamma V(s')\right). \tag{24}$$

That is, the value of state s is set to the expected value of the immediate reward, $R(s)$, plus the discounted expected value of its successor states, assuming that we follow the current policy. Equations (24) are linear, so they can be solved explicitly for moderately large state spaces, or an iterative scheme can be used for larger state spaces. We then repeat the first step to find a new greedy policy with respect the refined value function. Because the rewards are bounded (in $[-q_{\max} n_S, 0]$) and the state and action spaces are finite, equations (23) and (24) are guaranteed to converge to an optimal policy (Puterman 2005).

7 Results

This section compares the waiting times predicted by the three methods developed in this paper on the Heathrow Pod example from Fig. 3. To indicate what is achievable in practice, waiting times obtained from the 'Sampling and Voting' (SV) heuristic of Lees-Miller and Wilson (2011) are also given. SV is a practical heuristic for the EVR problem that moves empty vehicles proactively. It is based on ideas from the Dynamic Vehicle Routing Problem (DVRP) literature. At each decision point, SV generates an ensemble of possible sequences of future requests from the demand matrix. Each of these sequences, together with the current state of the system, defines an instance of the static EVR problem (Sect. 5). Each instance is solved approximately using the SNN heuristic, and common features of the solutions are identified and implemented using a voting system; the idea is that if a particular empty vehicle trip occurs frequently in the ensemble, it is probably a good trip to make in the actual system. Here SV generates an ensemble of ten sequences, each containing up to fifty requests ($n_E = 10$ and $n_R = 50$, in the notation of that paper).

The small size of the example network used here is due to limited scalability of the dynamic programming method currently used to solve the MDPs. Table 1 shows the number of states used to model the system for various fleet sizes and maximum queue lengths; dynamic programming quickly becomes infeasible beyond about six vehicles, even when $q_{\max} = 1$. However, the Heathrow Pod system is sufficient to illustrate one of the main issues in proactive empty vehicle redistribution: when there is demand from both Business Parking (BP) stations to T5, the system has to choose which BP station to send an empty vehicle to, and it is penalised if it chooses the wrong one, because it takes one time step for the vehicle to move to the correct station. The SNN and $M/G/s$ methods have been evaluated

n_K \ $q_{\max}$	1	2	3	4	5
1	60	135	240	375	540
2	480	1,080	1,920	3,000	4,320
3	2,720	6,120	10,880	17,000	24,480
4	12,240	27,540	48,960	76,500	
5	46,512	104,652	186,048		
6	155,040				

Table 1 Size of the MDP state space with varying fleet size, n_K, and maximum queue length, $q_{\max}$. Here $t_i^{\max} = 5$ time steps for all stations, and the demand is tidal from both BP stations to T5.

previously on larger systems with tens of stations and hundreds of vehicles (Lees-Miller and Wilson 2011; Lees-Miller 2011).

To evaluate an MDP policy fairly, it is necessary to evaluate it without an arbitrary limit on the queue lengths (that is, with $q_{\max} = \infty$). It is also helpful to use smaller time steps, because waiting times less than one time step cannot be known exactly. The evaluation here is therefore based on using the policy computed via dynamic programming to control a simulation of the MDP model; this simulation does not require the whole state space or the transition probabilities to be explicitly represented, so there is no requirement that the state space be small (or finite). However, it is necessary to project from the simulated state space into the state space for which the policy is defined. This projection involves three steps.

1. Truncate the queues from the simulation at $q_{\max}$.
2. Round the 'times until arrival' from the simulation up to the next integer multiple of the MDP time step and truncate at $t_i^{\max}$. Here the simulation uses one second time steps, whereas the MDP model uses one minute time steps.
3. If the queue for station i was truncated, the action recommended by the policy may not be valid, because more idle vehicles may be used by queued requests, which leaves fewer vehicles available for empty trips. The approach taken here is to zero the row i of the action matrix, $\mathbf{A}$, so that no empty vehicle trips will be started from station i.

Fig. 4 shows the mean waiting times obtained by the four methods on the Heathrow Pod system for up to three vehicles. Three demand distributions are considered; in all three, all requests are from BP to T5, but they differ in the split between the two BP stations, BPA and BPB. For each split, the total demand is scaled to produce a family of demands with the same spatial distribution but different intensities (Sect. 3). For example, when there are three vehicles, intensity 1 corresponds to 18 requests per hour (larger capacities clearly require more vehicles). In this and the other figures, the mean waiting time reported for each intensity is taken over five runs of 0.5 million requests each for the $M/G/s$ and SNN methods and five runs of 0.1 million requests each for the MDP and SV methods.

Panels 4(a, d, and g) show the case where all requests are from BPB to T5. Empty vehicle redistribution is trivial in this case, because the optimal policy is clearly to send all idle vehicles from T5 to BPB, and all four methods give the same waiting times.

Panels 4(c, f, and i) show the case where the demand is split evenly between BPA and BPB. In this case, the $M/G/s$ and SNN methods agree, but the MDP and SV methods produce longer waiting times. For example, for a single vehicle (panel (c)) at intensity 0.1, $M/G/s$ and SNN predict a mean waiting time of 33s, but SV and MDP give 65s. The gap of approximately 30s reflects the fact that the optimal policy, if one does not know which BP station the next request will come from, is to send the vehicle to one of the stations arbitrarily, say BPB, because the next request is equally likely to be from either one. Half of

the time, the next request will be from BPB, and that request will have zero waiting time; but, the other half of the time, the next request will be from BPA, and that request will have a waiting time of 60s. The expected waiting time penalty for this uncertainty in where the next request will come from is therefore 30s, in this case. When there are multiple vehicles, this penalty is reduced, as indicated by the MDP results in panels (f) and (i), which show a smaller gap. Waiting times for SV are significantly higher than for the optimal MDP policy when $n_K = 2$ or 3, but the gap for SV is smaller for larger fleet sizes.

Fig. 5 shows the effect of increasing fleet size on the waiting times given by the optimal MDP policy. The shape of the waiting time vs. intensity curve changes significantly as the fleet size increases. It becomes flatter at low intensities, and then it increases more sharply as it approaches maximum theoretical capacity, at intensity one. For example, when demand is split evenly between BPA and BPB (panel (c)), the mean waiting time at intensity 0.6 is 200s with two vehicles but 51s with five vehicles. One explanation for this is that as the fleet size increases, the system becomes more like an 'insensitive' $M/G/\infty$ queuing system (Adan and Resing 2002), in which there are an infinite number of servers, and customer waiting times are zero for all demand intensities less than one.

Fig. 6 shows the results from the $M/G/s$ and SV methods for fleet sizes up to twenty vehicles, for which it is not currently feasible to find an optimal MDP policy. The trend observed in Fig. 5 toward flatter waiting time vs intensity curves continues. For example, when $n_K = 20$ the mean waiting time achieved at intensity 0.6 by SV is roughly 6s, compared with 75s for $n_K = 5$. It is also notable that the gap between the results actually achieved by SV and those predicted by the $M/G/s$ method decreases as the fleet size increases. These effects suggest an economy of scale in providing low passenger waiting times in PRT and taxi systems: as the fleet size increases, low passenger waiting times can be achieved even when the system is operating at high utilisation.

8 Conclusions

Three lower bounds were developed, one from an $M/G/s$ queuing model, one based on approximately solving a static optimisation problem with a heuristic called SNN, and one using a formal Markov Decision Process (MDP) model. An evaluation of these bounds shows that the $M/G/s$ accurately predicts the average waiting times produced by the SNN heuristic, which are in this case optimal. The optimal MDP policy, which explicitly considers the inherent uncertainty about future requests, gives longer mean waiting times than the $M/G/s$ and SNN bounds, particularly when the fleet size is small.

For larger fleet sizes, the gap between waiting times achievable using only statistical information and those achievable with perfect information is reduced. Larger fleet sizes also allow the system to operate at higher intensity (higher utilisation) while still providing low passenger waiting times, other things being equal. This result suggests important economies of scale.

There is much scope for future work. The cost of empty vehicle movement was not included; the MDP model is amenable to this, and the SNN heuristic could be modified to include travel costs. Travel time variability was not considered. Uncertain travel times can be easily incorporated into the MDP model, but, particularly in a PRT system, the empty vehicle redistribution policy affects congestion on the network, which in turn affects travel times; accurately representing this appears to require a more detailed model that does not discard the network topology, as we did here by representing the network only by its quickest path trip times. A promising line of work is the application of approximate dynamic

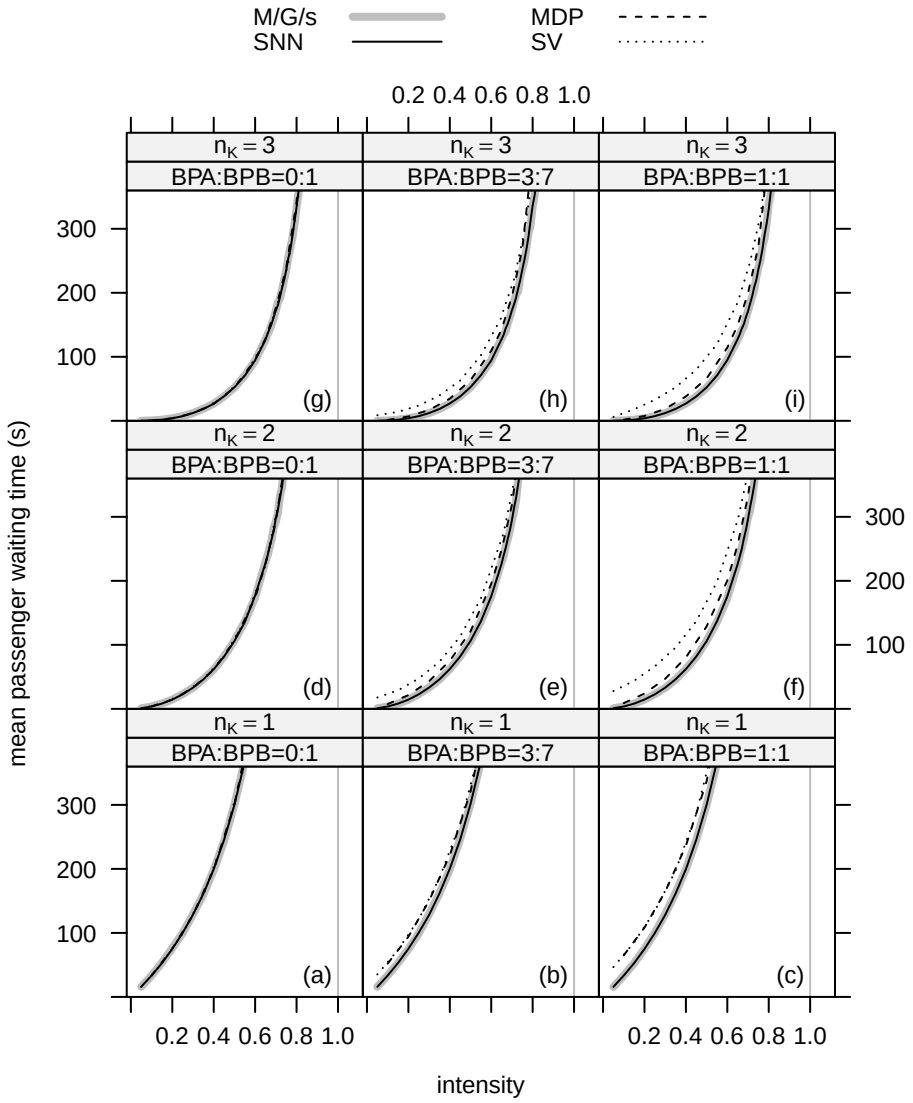


Fig. 4 Mean passenger waiting times obtained by the four methods. All four methods agree in the trivial case, when the demand is from one origin (a, d, g). The $M/G/s$ and SNN results agree for all cases. The gap between the optimal MDP policy and the $M/G/s$ and SNN bounds in (c) is due to the fact that only statistical information is available about future requests; this gap decreases as the number of vehicles increases (f, i).

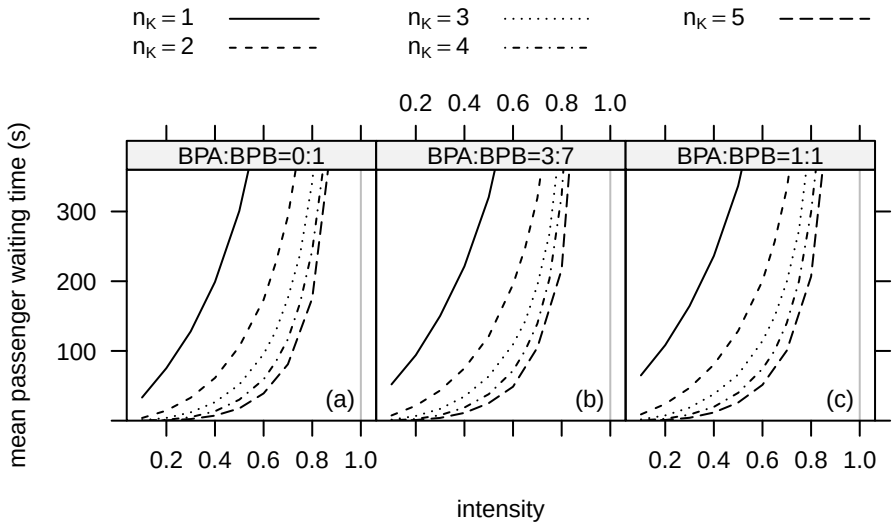


Fig. 5 Mean passenger waiting times for the optimal MDP policy for varying fleet sizes. As the fleet size increases, the curves become flatter at low intensity. In these results, $q_{\max} = 4,4,4,3$ and 1 for $n_K = 1,2,3,4$ and 5, respectively.

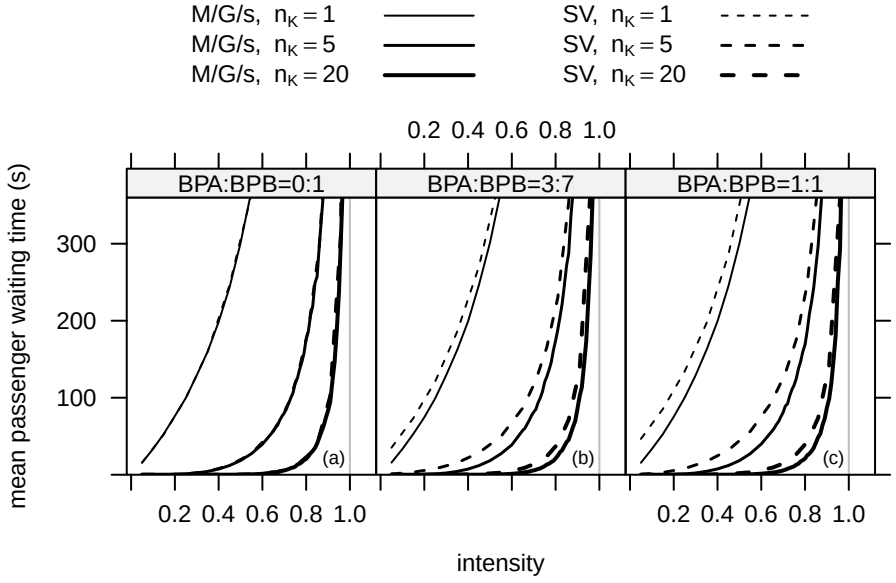


Fig. 6 Mean passenger waiting times for the $M/G/s$ and SV methods for varying fleet sizes. For larger fleet sizes, the system can operate at high intensity (high utilisation) and still provide low waiting times. The gap between the waiting times that heuristic can provide in practice and the $M/G/s$ lower bound diminishes also decreases as the fleet size increases.

programming methods to the MDP model developed here; this has the potential to produce effective algorithms for the EVR problem that are both practical and principled.

References

2getthere (2011) Masdar operations. Accessed 11 May, 2011, URL http://www.2getthere.eu/?p=128

Adan I, Resing J (2002) Queueing theory. Course Notes, Eindhoven University of Technology

Anderson JE (1978) Transit systems theory. Lexington Books

Bell MGH, Wong KI (2005) A rolling horizon approach to the optimal dispatching of taxis. In: Mahmassani HS (ed) Transportation and traffic theory : flow, dynamics and human interaction : proceedings of the 16th International Symposium on Transportation and Traffic Theory, Elsevier, Amsterdam, pp 629–648

Bent RW, Van Hentenryck P (2004) Scenario-Based planning for partially dynamic vehicle routing with stochastic customers. Operations Research 52(6):977–987

Berbeglia G, Cordeau JF, Gribkovskaia I, Laporte G (2007) Static pickup and delivery problems: a classification scheme and survey. TOP 15(1):1–31

Berbeglia G, Cordeau JF, Laporte G (2010) Dynamic pickup and delivery problems. European Journal of Operational Research 202(1):8–15

Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming. Athena Scientific

Bertsimas D, Tsitsiklis J (1997) Introduction to Linear Optimization. Athena Scientific

Bertsimas DJ, Levi DS (1996) A new generation of vehicle routing research: Robust algorithms, addressing uncertainty. Operations Research 44(2):286–304

Bly PH, Teychenne P (2005) Three Financial and Socio-Economic Assessments of a Personal Rapid Transit System. In: Proceedings of the Tenth International Conference on Automated People Movers, American Society of Civil Engineers, p 39, URL http://link.aip.org/link/?ASC/174/39

Boxma OJ, Cohen JW, Huffels N (1979) Approximations of the mean waiting time in an M/G/s queueing system. Operations Research 27(6)

van Eijl CA (1995) A polyhedral approach to the delivery man problem. Tech. Rep. COSOR 95-19, Eindhoven University of Technology

Gupta V, Harchol-Balter M, Dai J, Zwart B (2010) On the inapproximability of M/G/k: why two moments of job size distribution are not enough. Queueing Systems 64(1):5–48

Horn M (2002) Fleet scheduling and dispatching for demand-responsive passenger services. Transportation Research Part C: Emerging Technologies 10(1):35–63

Hvattum LM, Løkketangen A, Laporte G (2006) Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transportation Science 40(4):421–438

Lee DH, Wang H, Cheu R, Teo S (2004) Taxi dispatch system based on current demands and Real-Time traffic conditions. Transportation Research Record: Journal of the Transportation Research Board 1882:193–200

Lees-Miller JD (2011) Empty vehicle redistribution for personal rapid transit. PhD thesis, University of Bristol

Lees-Miller JD, Wilson RE (2011) Sampling for personal rapid transit empty vehicle redistribution. Transportation Research Record: Journal of the Transportation Research Board

Lees-Miller JD, Wilson RE (2012) Proactive empty vehicle redistribution for personal rapid transit and taxis. Transportation Planning and Technology

Lees-Miller JD, Hammersley J, Davenport N (2009) Ride sharing in personal rapid transit capacity planning. In: Griebenow RR (ed) Automated People Movers 2009, American Society of Civil Engineers, Reston, VA, pp 321–332

Lees-Miller JD, Hammersley JC, Wilson RE (2010) Theoretical maximum capacity as benchmark for empty vehicle redistribution in personal rapid transit. Transportation Research Record: Journal of the Transportation Research Board 2146:76–83

Li S (2006) Multi-attribute taxi logistics optimization. Master's thesis, Massachusetts Institute of Technology

Nagarajan V, Ravi R (2008) The directed minimum latency problem. In: Goel A, Jansen K, Rolim JDP, Rubinfeld R (eds) Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques., Lecture Notes in Computer Science, vol 5171, Springer, pp 193–206

Powell WB (2007) Approximate dynamic programming : solving the curses of dimensionality. Wiley-Interscience

Puterman ML (2005) Markov decision processes : discrete stochastic dynamic programming. Wiley-Interscience

Seow KT, Dang NH, Lee DH (2010) A collaborative multiagent Taxi-Dispatch system. IEEE Transactions on Automation Science and Engineering 7(3):607–616

Sutton RS, Barto AG (1999) Reinforcement learning : an introduction. MIT Press

Swihart MR, Papastavrou JD (1999) A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. European Journal of Operational Research 114(3):447–464

Tijms H (2006) New and old results for the M/D/c queue. AEU - International Journal of Electronics and Communications 60(2):125–130

Toth P, Vigo D (2002) The vehicle routing problem. SIAM monographs on discrete mathematics and applications, Society for Industrial and Applied Mathematics

ULTra PRT (2010) ULTra at london heathrow airport. Accessed July 31, URL `http://www.ultraprt.com/applications/existing-systems/heathrow/`

Vis IFA (2006) Survey of research in the design and control of automated guided vehicle systems. European Journal of Operational Research 170(3):677–709

Wang H, Lee DH, Cheu R (2009) PDPTW based taxi dispatch modeling for booking service. In: Fifth International Conference on Natural Computation, IEEE, pp 242–247

Wesselowski K, Cassandras CG (2006) The elevator dispatching problem: Hybrid system modeling and receding horizon control. In: Cassandras C, Giua A, Seatzu C, Zaytoon J (eds) Analysis and Design of Hybrid Systems 2006: a Proceedings volume from the 2nd IFAC Conference, Elsevier, pp 136–141

Yang H, Wong S, Wong K (2002) Demand-supply equilibrium of taxi services in a network under competition and regulation. Transportation Research Part B: Methodological 36(9):799–819

Yang J, Jaillet P, Mahmassani H (2004) Real-Time multivehicle truckload pickup and delivery problems. Transportation Science 38(2):135–148