Proactive Empty Vehicle Redistribution for Personal Rapid Transit and Taxis

Abstract

This is my testing. This is my abstract. This is my abstract. This is my abstract. I can still edit? I can edit in RT mode... This is my abstract. Testing. again. Here are some edits with the auto preview. It's still reasonably fast. Too bad about the flash. Woot is not a word.

$$\alpha x^2 + \beta x + \gamma = 0$$

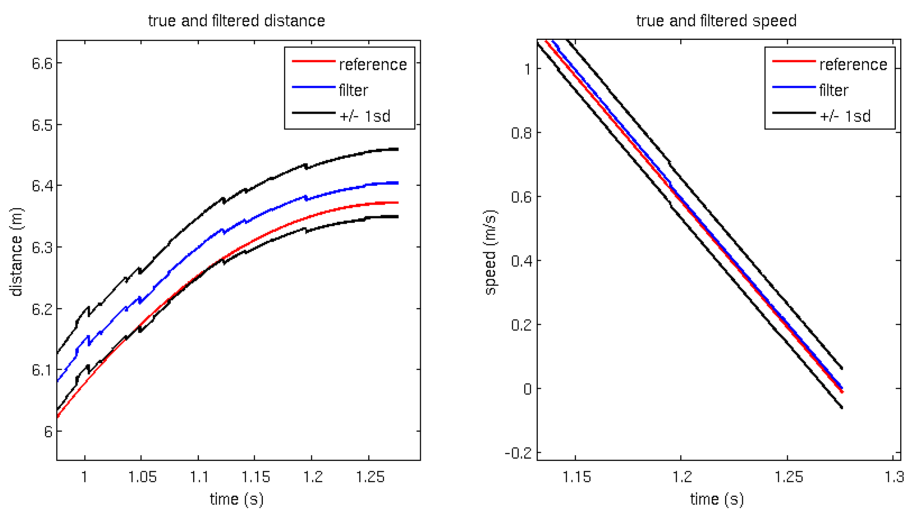
This is my abstract. This is my abstract. This is my abstract. This is my abstract. This is my abstract. This is my abstractThis is my abstract. This is my abstract. $a^2 + b^2 = c^2$This is my abstract. This is my abstract. This is my abstract. This is my abstract. This is my abstract. Here is some more text.

1 Introduction

?

Personal Rapid Transit (PRT) is a new urban transport mode. It will use small, computer-guided vehicles to carry individuals and small groups between pairs of stations on a dedicated network of guideways. The vehicles will operate on-demand and provide direct service from origin station to destination station, much like conventional taxis...

2 Testing....



Worship Songs

1 *Doxology*

Revelation 5:13

Louis Bourgeois and Thomas Ken

```
G             D     Em     Bm
Praise God, from Whom all
     Em D    G
     blessings flow;
G                  D   Em   C    G D
Praise Him, all creatures here below;
Em    D     G  D      G  C      D
Praise Him a - bove, ye heav'nly
     Em
     host;
G          Em  D    Am
Praise Father, Son, and
     G/B G/C  D  G
     Ho    -  ly Ghost.
C  G
A - men.
```


2.1 Blah

2.1

The world's first PRT system is in the final stages of commissioning at London's Heathrow Airport (ULTra PRT, 2010). It is a "last mile" circulator with three stations and twenty-one driverless vehicles (Figure 1), connecting a business car park 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 and create new possibilities for urban planning. In order for PRT to play this role, it must provide a high-quality service, which means low passenger waiting times, low travel times and high levels of safety and comfort.

Figure 1: Photograph of PRT vehicle and at-grade station at London Heathrow Airport. PRT vehicles, stations and infrastructure are smaller than typical Automated People Mover and urban rail systems. Vehicle length, width and height are 3.7m, 1.4m and 1.8m, respectively. Photo courtesy of ULTra PRT Ltd

The focus of this paper is on methods for moving empty vehicles to provide low passenger waiting times. Deciding which vehicles to move and where to move them is known as the empty vehicle redistribution (EVR) problem. It is assumed that passengers request immediate service (they do not book ahead) from their origin station to their chosen destination station. A central control system can move empty vehicles either *reactively*, in response to a request that has just been received, or *proactively*, in anticipation of future requests. 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, because a passenger's waiting time is often equal to the travel time of the empty vehicle assigned to serve him (Lees-Miller et al., 2010). Clearly, proactive movement of empty vehicles in anticipation of future arrivals can reduce waiting times, but there is a risk of unnecessary empty vehicle travel.

This paper proposes two EVR heuristics that can move vehicles proactively, and it develops methods for assessing the performance of EVR heuristics absolutely, both in terms of throughput and in terms of passenger waiting times. The proposed heuristics are evaluated using these methods and in simulation, and they provide lower waiting times than other algorithms in the literature. The results show that proactive movement of empty vehicles significantly reduces mean passenger waiting times, typically with a modest increase in empty vehicle travel.

The basic PRT system model is adapted from the urban taxi model of Bell and Wong (2005) (section 3). When a request is received, the vehicle that minimizes the request's waiting time is immediately assigned to serve the request. This is a reactive algorithm, and it assumes no knowledge of future requests. In order to improve on this, it is assumed that, while future requests are not known with certainty, the average request rates between each pair of stations are known from historical data, in the form of an origin-destination demand matrix. In vehicle routing terminology, this makes the EVR problem *dynamic* (because requests are received while the system is operating) and *stochastic* (because statistical information about future requests is available).

However, it proves useful to first consider two deterministic versions of the EVR problem. In the *fluid limit* problem, the variables are long run average flows of vehicles, rather than individual vehicle movements. Analysis in the fluid limit gives a way of benchmarking EVR algorithms in terms of throughput (section 4). In the *static* problem, all future requests are assumed to be known in advance. An optimal solution to the static problem provides a benchmark in terms of passenger waiting time (section 5).

The two proposed heuristics are extensions of the basic model to allow proactive movements. The Sampling and Voting (SV) heuristic works by generating an ensemble of possible sequences of future requests from the demand matrix over a given finite horizon. Each sequence, together with the current state of the system, defines an instance of the static EVR problem whose (approximate) solution suggests a plan of empty vehicle movements. Features of these plans that are common across the ensemble are then extracted to determine which empty vehicles should actually be moved (section 6). The Dynamic Transportation Problem (DTP) heuristic works by attempting to maintain a given *target* number of vehicles inbound to each station. The problem of satisfying the targets with minimum empty vehicle movement is a classical transportation problem (section 7). The SV algorithm was first introduced in Lees-Miller and Wilson (2011), but the presentation here is different; the DTP algorithm has not been described previously......

(TODO MORE ON RELATED WORK...) PRT has much in common with a conventional hackney service, and the principle of proactively moving empty vehicles is applicable to conventional taxis. Moreover, the PRT system model used to implement the algorithms described in this paper is exactly the urban taxi model of Bell and Wong (2005), with zones replaced by PRT stations. So, these conclusions may also be of interest to taxi operators. On proactively moving empty taxis: Rank Homing (Horn, 2002): special case of TP. Random roaming (Lee et al., 2004), or in macroeconomic model (taxi). Alshamsi (2009) – maybe – must review notes. Seow et al. (2010) mention it as future work. The Li (2006) thesis. The static problem is to decide how to move empty vehicles when all requests are known in advance. Whereas the dynamic problem models a pure hackney service, the static problem models a pure livery service, in which every request is ‘phoned in’ before any vehicles are moved.

3 The Model

The three main factors that determine passenger waiting times are congestion on the guideway, congestion in stations and the availability and location of empty vehicles. For very large systems with many vehicles, congestion effects are often significant, but most 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.
3. The requested average flows of occupied vehicles between stations are known, in the form of an origin-destination matrix derived from historical data, and these averages are steady (they do not change with time).

The relevant input data are then as follows. Let S be the set of stations. For each pair of stations i and j, let T_{ij} be the travel time from i to j, in minutes, with $T_{ij} = 0$ if $i = j$. Similarly, let D_{ij} be the number of occupied vehicle trips per minute from i to j, with $D_{ij} = 0$ if $i = j$.

The PRT system model used in this paper is based on the urban taxi model of Bell and Wong (2005). Let K be the set of vehicles, and let n_K be the fleet size ($n_K = |K|$). Each vehicle $k \in K$ has a planned route, which consists of a list of stations that it must visit in order. Each pair of adjacent stations defines a *trip*, during which the vehicle may be occupied or empty. A vehicle's route changes over time: completed trips are deleted from the head, and new occupied or empty vehicle trips are appended to the tail as they are assigned. If a vehicle completes all of the trips in its list, it becomes *idle* at the last station on its route. For simplicity,

Figure 2: Illustration of the BWNN algorithm. There are four off-line stations (labeled *A* - *D*) in a ring and two vehicles (labeled *1* and *2*). Traffic flow is counter-clockwise. (*a*) Vehicle *1* is initially moving to station *B*, and vehicle *2* is idle at station *A*. (*b*) When a request for travel from *C* to *D* is received, vehicle *1* is assigned, because it gives a smaller waiting time than vehicle *2*. The new request is appended to vehicle *1*'s request list; this requires an empty vehicle trip (dashed line) from *B* to *C* and an occupied trip (solid line) from *C* to *D*. Note that vehicle *1* stops at station *B* and station *C* (filled circles), but it does not become idle at either station, because it has not finished with its request list. (*c*) However, vehicle *1* does become idle at *D*, because no further requests are assigned to it. (*d*) When another request is received from *C* to *A*, vehicle *2* is assigned, and it begins an empty trip to *C*. Vehicle *2* was idle at *A*, so it could have moved to *C* proactively, if the request from *C* had been anticipated; this would have reduced the passenger's waiting time.

it is assumed that the lists are not reordered, so for each vehicle $k \in K$, it is enough to know the last station, d_k, that vehicle k was assigned to visit, and the time, a_k, at which the vehicle will arrive (or has already arrived) at d_k. With this notation, an idle vehicle is one whose a_k is in the past.

When a new *request* for travel is received, a vehicle is immediately assigned to serve it. Each request, r, has associated with it an origin station i_r, a destination station j_r and the time e_r at which the system receives the request. It is assumed that each request is for immediate travel from its origin station, so the *waiting time* of the request is the delay between e_r and when the assigned vehicle *picks up* the request at i_r.

The following heuristic, here called Bell and Wong nearest neighbours (BWNN), is used to decide which vehicle to assign. When a request r is received at time e_r, BWNN immediately assigns the vehicle

$$k^* = \operatorname*{argmin}_{k \in K} \left[\max\{0, a_k - e_r\} + T(d_k, i_r)\right] \tag{1}$$

where the travel times T_{ij} are written $T(i,j)$ here for readability. The first term, $\max\{0, a_k - e_r\}$, is the earliest time that the vehicle can start a new trip; note that the vehicle cannot begin its trip in the past, before e_r. The second term, $T(d_k, i_r)$, is the required empty vehicle trip time; if k is already going to the request's origin station ($d_k = i_r$), then the empty vehicle trip time is zero, because no empty trip is required. Figure 2 gives an illustrated example.

The BWNN algorithm is *reactive*, because it moves vehicles only in response to requests. This tends to result in idle vehicles accumulating at stations with a net inflow of occupied vehicles, which in turn causes long waiting times at stations with net a outflow of occupied vehicles. The following sections motivate and describe heuristics for extending this basic reactive model to move idle vehicles proactively.

4 The Fluid Limit EVR Problem

The *fluid limit* deals with long run average flows of vehicles, rather than individual vehicle trips. In particular, the aim here is to find the flows of empty vehicles, X_{ij}, that are needed to balance the given flows D_{ij} of occupied vehicles. The result is a classical transportation problem that is well-known in the PRT literature (Anderson, 1978) and can also be derived from the urban taxi economics model of Yang et al. (2002). The main output of this fluid limit analysis is a measure of the theoretical maximum throughput of the system, which provides a benchmark against which EVR algorithms can be compared (Lees-Miller et al., 2010).

The transportation problem in the fluid limit is formulated as follows. Let $s_i = \sum_j (D_{ji} - D_{ij})$ be the occupied vehicle flow *surplus* at station i, and partition the stations into $S^+ = \{i \in S : s_i \geq 0\}$ and $S^- = \{i \in S : s_i < 0\}$. The stations in S^+ have a net inflow of occupied vehicles on average, and the stations in S^- have a net outflow on average. The flows of empty vehicles (X_{ij}) required to balance the total inflows and outflows can then be found from the following transportation problem:

$$\min \sum_{i\in S^+} \sum_{j\in S^-} T_{ij} X_{ij} \tag{2}$$

$$\text{s.t.} \sum_{j\in S^-} X_{ij} = s_i \qquad \forall\, i \in S^+ \tag{3}$$

$$\sum_{i\in S^+} X_{ij} = -s_j \qquad \forall\, j \in S^- \tag{4}$$

$$X_{ij} \geq 0 \qquad \forall\, i \in S^+,\ j \in S^-$$

The objective (2) is to minimise the average number of moving empty vehicles (minutes, times vehicles per minute, gives vehicles), and constraints (3, 4) ensure that the empty vehicle flows balance the inflows and outflows of occupied vehicle flows. This problem can be solved efficiently using standard techniques (Bertsimas and Tsitsiklis, 1997).

An important use of this analysis is to define the *intensity* of the demand (Lees-Miller et al., 2010), which is the total number of vehicles (occupied and empty) required, according to the fluid limit analysis, divided by the number of vehicles actually available, n_K. In particular, the objective (2) gives the number of empty vehicles required, and the number of occupied vehicles required can be computed similarly to give the intensity of the demand as

$$\rho = \frac{1}{n_K}\left(n_X^* + \sum_{i,j} T_{ij} D_{ij}\right) \tag{5}$$

where n_X^* is the optimal objective value (2). Intensity one corresponds to maximum throughput. When $\rho > 1$, requests are arriving faster than the system can serve them, because it does not have enough vehicles. This means that both the number of passengers waiting and their waiting times will grow indefinitely, so long as the demand is held constant. When $\rho < 1$, the system may (depending on how efficiently it uses empty vehicles) be able to keep up with demand. So, while this fluid limit analysis does not give a direct measure of passenger waiting times, the intensity of the demand is an important factor. The term intensity is motivated by similar definitions in the theory of queueing systems.

The fluid limit problem provides a benchmark for throughput, but it does not provide much information about waiting times; this requires a more detailed model, such as the one in the next section.

5 The Static EVR Problem

In the static problem, it is assumed that all requests are known in advance instead of being revealed while the system is operating. This problem is of interest because an optimal solution for an instance of the static problem provides a benchmark for waiting times, and also because the static problem motivates the SV method for the dynamic problem (section 6).

The static EVR problem can be formulated as a *multivehicle truckload pickup and delivery problem with time windows*, which is a well-studied (Yang et al., 2004) vehicle routing problem. An instance of the static EVR problem requires that the initial state of the vehicles and all requests over a given finite horizon be known. For simplicity, let the current time be time zero, adjust the vehicles' a_k times accordingly, and set $a_k = 0$ for vehicles that are currently idle. Let R be the set of requests with e_r within the horizon. All of the i_r, j_r and e_r are known at time zero, and e_r is the earliest time that request r can be picked up.

To translate the static EVR problem into a vehicle routing problem, we construct an auxiliary *vehicle-request graph*, G, with one node for each vehicle, one node for each request, and a *sink* node, s. The routing problem is to find the best routes in G that serve all of the requests. Each route must start from a vehicle node, go through zero or more request nodes, and end at the sink node. More formally, the node set of G is $K \cup R \cup \{s\}$, and the edge set is $\{(u, v) : u \in K \cup R, v \in R \cup \{s\}, u \neq v\}$. The routing objective is to minimize mean request (passenger) waiting time, which is known as a *minimum latency* objective. The requirement that request r

be picked up only after e_r is a *one-sided time window* constraint. The edge costs encode the required travel times in the PRT network, with

$$c_{uv} = \begin{cases} a_u + T(d_u, i_v) & u \in K,\ v \in R \\ T(i_u, j_u) + T(j_u, i_v) & u, v \in R,\ u \neq v \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The cost (6) of an edge from vehicle k to request r includes the time required for k to finish serving previously assigned requests (if any) plus the required empty vehicle trip time (if any). Similarly, when both u and v are requests, the cost includes the occupied travel time $T(i_u, j_u)$ for request u and any empty travel time $T(j_u, i_v)$ required in order to serve v directly after u.

The static EVR problem is NP-hard, because the corresponding routing problem on G generalises the minimum latency version of the asymmetric travelling salesman path problem (ATSPP), which is NP-hard (Nagarajan and Ravi, 2008). In particular, the routing problem is a minimum latency ATSPP when there is one vehicle and $e_r = 0$ for all requests. Small instances can be solved exactly using standard techniques, but very large instances must be solved in order to benchmark EVR algorithms, because waiting times must be averaged over thousands of requests. These large instances can only be solved approximately, at present.

In this paper, large instances of the static EVR problem are solved using the following static nearest neighbours (SNN) heuristic (Lees-Miller and Wilson, 2011). For each request r, in ascending order by e_r, SNN chooses the vehicle

$$k^* = \operatorname*{argmin}_{k \in K} \max\{0,\ a_k + T(d_k, i_r) - e_r\} \tag{7}$$

that minimizes the waiting time for request r. SNN is similar to BWNN (1), and in fact for vehicles with $a_k \geq e_r$, the objective values in (1) and (7) are the same. However, if a vehicle has $a_k < e_r$, SNN allows the vehicle to start its empty trip before e_r, whereas BWNN does not. In this sense, SNN moves empty vehicles proactively. If there are several vehicles that minimize (7), one is chosen using the tie-breaking rules in Lees-Miller and Wilson (2011). The vehicle state for k^* (a_{k^*} and d_{k^*}) is then updated accordingly, and the next request is considered. Testing.

The solutions produced by SNN are not provably optimal, so other methods may produce lower mean waiting times for a given instance; in this sense, the solutions are not, strictly speaking, benchmarks. However, nearest neighbour heuristics have been found to produce high quality solutions to other minimum latency routing problems (Fischetti et al., 1993, Larsen et al., 2004, Swihart and Papastavrou, 1999). This is particularly true when the fleet size is large, because each vehicle's route tends to be short, and fleet sizes in PRT are large relative to those usually studied in the vehicle routing literaturearst (the case study systems in section 8 have 200 vehicles). The intensity of the demand is also important in determining the difficulty of a given instance; when the intensity of the demand is low, there are long delays between requests, so most requests will simply be served in the order in which they are received. In fact, the SNN heuristic finds solutions with zero waiting time (which are clearly optimal) when the intensity of the demand is below about 80% on the case study systems.

Another use for the static problem is in the proposed Sampling and Voting heuristic, below. Here are some new words.

6 New EVR Heuristic: Sampling and Voting (SV)

The sampling and voting (SV) heuristic moves idle vehicles proactively by generating an ensemble of static problems, solving them approximately using SNN, and extracting common features from the solutions in order to decide which idle vehicles should actually be moved.

When a new request is received at time t, SV assigns a vehicle using the BWNN algorithm. Immediately after a vehicle has been assigned to serve the request, SV may then move idle vehicles proactively. To decide which idle vehicles to move, an ensemble of n_E possible sequences of n_R future requests each is generated from the demand matrix. Each sequence in the ensemble, together with the current state of the system, defines an instance of the static

EVR problem. Each of these instances is solved approximately using the SNN algorithm, and each resulting solution prescribes a sequence of empty vehicle trips, which constitutes 'advice' on which idle vehicles the system should actually move. However, because each solution is computed for a (probably) different sequence of requests, they may offer conflicting advice. To determine which action should actually be taken, a voting system is used. The system adopted here is that at most one idle vehicle at each station may be moved. So, each solution casts one vote on the best destination (as defined below) for an idle vehicle at each station i with idle vehicles; note that it may vote for i, which means that it votes not to move any idle vehicles from i at this decision point. If the destination with the most votes is not i, an idle vehicle at i is selected (breaking ties on minimum vehicle index) and moved.

For each station i with idle vehicles, the destination to vote for are determined as follows. If all vehicles now idle at i were used for requests from i, vote for i. Otherwise, if a vehicle now idle at i was moved empty to another station, j, vote for j (for the first such trip). Otherwise, if any vehicle was moved empty from i to another station, j, vote for j (for the first such trip). Otherwise, vote for i. These voting rules are discussed in more detail in Lees-Miller and Wilson (2011)....

7 New EVR Heuristic: Dynamic Transportation Problem (DTP)

The dynamic transportation problem (DTP) heuristic requires a *target* for the number of vehicles that should be inbound to each station at any given time. If the number of vehicles inbound to a station is below its target, idle vehicles should be moved there; conversely, if a station has inbound vehicles in excess of its target, some of its idle vehicles may be moved elsewhere. The problem of moving idle vehicles to meet targets with minimum empty vehicle running time is a classical transportation problem.

The idea of maintaining a target number of vehicles at each station is common in the PRT literature (Andréasson, 1998, Anderson, 1998). The decision on which vehicles to move is typically made according to rules that can be viewed as approximation algorithms for the transportation problem. For example, when a station has a deficit, it may call the nearest idle vehicle at a station with a deficit not greater than its own (Andréasson, 1998). The targets may be treated as parameters to be tuned, or they may be set adaptively based on the history of past empty vehicle movements (Andréasson, 1998). In this paper, the targets are treated as parameters to be set using meta-heuristics, and the results are compared with a heuristic based on these existing algorithms (section 8).

More formally, for each station i, let θ_i be the target number of inbound vehicles at station i. Let t be the current time. Let b_i be the number of vehicles that are inbound to i (that is, $d_k = i$), and let l_i be the number of vehicles that are idle at i (that is, $d_k = i$ and $a_k \leq t$); note that $b_i \geq l_i \geq 0$. We can now define

$$u_i = \min\{b_i - \theta_i, l_i\}$$

as the *surplus* of vehicles at station i. If $u_i > 0$, station i has a surplus of inbound vehicles, but only l_i of these are currently idle at i, so at most l_i vehicles can be moved now. If $u_i < 0$, station i has a *deficit* of inbound vehicles. In general, the surpluses and deficits need not balance, so we introduce an extra dummy node q with $u_q = -\sum_i u_i$. Let $S' = S \cup \{q\}$ and partition S' into sets $S_t^+ = \{i \in S' \ : \ u_i \geq 0\}$ and $S_t^- = \{i \in S' \ : \ u_i < 0\}$. Note that this partition may change over time. For each $i \in S_t^+$ and each $j \in S_t^-$, let x_{ij} be the number of vehicles to send from node i to node j, which is to be solved for. If i and j are both stations, then $x_{ij} > 0$ means that x_{ij} vehicles which are currently idle at i are to move to station j. If either $i = q$ or $j = q$, then no vehicles are actually moved, regardless of the value of x_{ij}; in other words, a decision to move idle vehicles from or to the dummy node means that they should be left where they are until the next decision point. The costs for sending vehicles to and from the dummy node are zero (define $T_{iq} = T_{qi} = 0$), because none of these vehicles are actually moved.

The transportation problem to be solved has the same form as that in the fluid limit (2), but the variables are x_{ij} instead of X_{ij}, the surpluses are u_i instead of s_i, and the stations (and q) are partitioned into S_t^+ and S_t^- instead of S^+ and S^-. When the targets θ_i are integers, there is an optimal solution in which all of the x_{ij} are integer, because it is a special case of the minimum cost network flow problem and the u_i are integers (Bertsimas and Tsitsiklis, 1997, ch. 7). The problem is solved exactly with the integer RELAX IV code (Bertsekas and Tseng, 1988).

The DTP heuristic is embedded in the PRT system model (section 3) as follows. When a request is received, a vehicle is assigned using BWNN (1), as usual. Immediately after this, the DTP transportation problem is formulated and solved to decide on proactive empty vehicle trips. The transportation problem is also solved every time a vehicle becomes idle.....

8 Results

In this section, the proposed algorithms are evaluated in simulations on two case study systems. The main simulation outputs are passenger waiting times and empty vehicle use. The steady state distributions of these outputs are estimated by running long simulations with the demand matrix held constant for each run. Passenger requests from station i to station j are generated from a Poisson process with rate D_{ij}. For convenience, passenger arrival and travel times are rounded to the nearest integer second.

The input data used here are the 'Grid' and 'Corby' networks and their corresponding demand matrices from Lees-Miller et al. (2010) (for the Grid network, the demand matrix with dispersion parameter $\theta = 0.01$ is used). The fleet size is set at $n_K = 200$ vehicles. Intensity one corresponds to a total demand of 1414 requests/hour for the Corby system and 2035 requests/hour for the Grid system.

The targets for the DTP algorithm are chosen using simulated annealing, as implemented in the GNU Science Library (Galassi et al., 2003). An initial estimate of the levels targets is obtained from the fluid limit problem (2), namely

$$\hat{\theta}_i = \left(\sum_{\substack{j \\ i \neq j}} (D_{ji} + X^*_{ji}) T_{ji} \right) \left(\sum_{\substack{j \\ i \neq j}} \frac{D_{ij}}{D_{ij} + X^*_{ij}} \right).$$

The first factor is the number of vehicles that are expected to be inbound to station i on average, and the second factor is the fraction of vehicles that leave station i occupied. The rationale for the second factor is that if most of the vehicles leaving station i are empty, on average, then the station should not attempt to retain very many vehicles. Neighbouring solutions were generated by adding -1, 0 or 1 with equal probability to each target. The initial temperature was set to 10; the temperature decay factor was 1.01; the final temperature was 0.1; 10 evaluations were performed at each temperature; the Boltzmann constant was set to 1. Each energy evaluation was a simulation with 20000 requests. Two trials were performed for each point Figure 3, for a total of around 9200 energy evaluations per point.

For comparison, another EVR algorithm, here called the Surplus / Deficit (SD) algorithm, is also evaluated. It is an algorithm for the dynamic EVR problem that moves vehicles proactively. The general approach in SD is similar to several other published EVR algorithms Anderson (1998); it is most similar to that of Andréasson (1998). Each station i has an associated call time τ_i, which is the cumulative average of all previous empty vehicle trip times to that station. The surplus of vehicles at station i is the number of inbound vehicles minus the expected number of requests over the call time, namely $\tau_i \sum_j D_{ij}$. When a new request is received, a vehicle is assigned using BWNN. Immediately afterward, SD may move idle vehicles proactively, as follows. For each station i with idle vehicles, in descending order by number of idle vehicles, if the surplus of vehicles at i is greater than or equal to one, an idle vehicle at i is sent to the nearest station with surplus less than zero (if any). Additionally, when a vehicle becomes idle at station i, the above actions are taken for station i only.

Figure 3(*a*) compares the mean waiting times observed for the five heuristics on the Corby system. An important observation is that waiting times increase rapidly as intensity approaches one, regardless of which EVR algorithm is used, as is expected based on the definition of intensity (5). In practice, we are most interested in the system's performance at around 70% to 90% intensity, because in this range the system is well-utilized, but acceptably low passenger waiting times may still be obtained. At intensity 0.8, for example, mean waiting times are 355s for BWNN, 41s for SD, 20s for DTP, 15s for SV. By moving vehicles proactively, SV reduces

Figure 3: Mean passenger waiting times (*a*, *b*) and vehicle fleet utilization (*c*, *d*) for the Grid and Corby networks for the five heuristics. The SV and DTP heuristics move idle vehicles in anticipation of future requests, which reduces waiting times significantly below the BWNN baseline and below the SD heuristic from the literature. The SNN algorithm operates with perfect information about future requests in order to estimate how much further waiting times might be reduced. Here there are $n_E = 50$ sequences, each with $n_R = 300$ requests for SV. Each point is averaged over 10 independent runs of 50,000 simulated passengers each.
arst

mean waiting times by 96% from the BWNN baseline. The relative reduction decreases as intensity increases, however, and in fact the SV, DTP and SD algorithms become increasingly similar to the BWNN algorithm at higher intensities, because there are fewer idle vehicles to redistribute.

Figure 3(*c*) shows that the reduction in passenger waiting times comes from a modest increase in the average number of moving empty vehicles, or equivalently in empty vehicle travel time. The largest increase occurs at intensity 0.91, and this is from 47 concurrently moving empty vehicles with BWNN to 51 with SV (out of 200 vehicles). With perfect information about future arrivals, SNN finds routes with average waiting times less than those for the dynamic case, as expected; at intensity 0.8, the mean waiting time for SNN is 3s. Only mean waiting times are reported, but the ranking of the four algorithms is the same at the 90th percentile of the waiting distribution; at intensity 0.8, 90% of passengers wait less than 51s with SV and less than 106s with SD.

Results for the Grid network are qualitatively similar, as shown in Figure 3(*b*, *d*). However, it is notable that waiting times diverge at around intensity 0.95, which is below the theoretical maximum (intensity one). It thus appears that nearest neighbor algorithms of the type studied here do not deliver maximum possible throughput; it is not yet known whether there is any practical algorithm that does.

The results and conclusions presented here are consistent with simulations that have been conducted on nine case study systems in total, with between 15 and 60 stations, between 50 and 600 vehicles, and total demand at intensity one between 360 and 5050 requests/hour. However, it is possible to construct pathological systems in which the SV and DTP heuristics perform poorly; details of these will be documented in (FORTHCOMING THESIS).

9 Conclusions

Future work: rerouting, MDP approach....

References

J. E. Anderson. *Transit systems theory*. Lexington Books, 1978. ISBN 9780669019025. URL http://www.worldcat.org/isbn/9780669019025.

J. E. Anderson. Control of personal rapid transit systems. *Journal of Advanced Transportation*, 32(1):57–74, 1998.

Ingmar Andréasson. Quasi-Optimum redistribution of empty PRT vehicles. In William J. Sproule, Edward S. Neumann, and Stanford W. Lynch, editors, *Automated People Movers VI: Creative Access for Major Activity Centers*, pages 541–550, Reston, VA, April 1998. American Society of Civil Engineers.

M. G. H. Bell and K. I. Wong. A rolling horizon approach to the optimal dispatching of taxis. In Hani S. Mahmassani, editor, *Transportation and traffic theory : flow, dynamics and human interaction : proceedings of the 16th International Symposium on Transportation and Traffic Theory*, pages 629–648, Amsterdam, July 2005. Elsevier.

Dimitri P. Bertsekas and Paul Tseng. Relaxation methods for minimum cost ordinary and generalized network flow problems. *Operations Research*, 36(1):93–114, 1988. ISSN 0030364X. doi: 10.2307/171381. URL http://dx.doi.org/10.2307/171381.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997. ISBN 1886529191. URL http://portal.acm.org/citation.cfm?id=548834.

P. H. Bly and P. Teychenne. Three financial and Socio-Economic assessments of a personal rapid transit system. In *Proceedings of the Tenth International Conference on Automated People Movers*, pages 39+. American Society of Civil Engineers, May 2005. URL http://link.aip.org/link/?ASC/174/39.

Matteo Fischetti, Gilbert Laporte, and Silvano Martello. The delivery man problem and cumulative matroids. *Operations Research*, 41(6):1055–1064, 1993. ISSN 0030364X. doi: 10.2307/171600. URL http://dx.doi.org/10.2307/171600.

Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice Rossi. *Gnu Scientific Library: Reference Manual*. Network Theory Ltd., February 2003.

Allan Larsen, Oli B. G. Madsen, and Marius M. Solomon. The a priori dynamic traveling salesman problem with time windows. *Transportation Science*, 38(4):459–472, 2004. ISSN 1526-5447. doi: 10.1287/trsc.1030.0070. URL http://dx.doi.org/10.1287/trsc.1030.0070.

John D. Lees-Miller and R. Eddie Wilson. Sampling for personal rapid transit empty vehicle redistribution. *Transportation Research Record: Journal of the Transportation Research Board*, 2011.

John D. Lees-Miller, John C. Hammersley, and R. Eddie Wilson. 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, December 2010. doi: 10.3141/2146-10. URL http://dx.doi.org/10.3141/2146-10.

Viswanath Nagarajan and R. Ravi. The directed minimum latency problem. In A. Goel, K. Jansen, J. D. P. Rolim, and R. Rubinfeld, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques.*, volume 5171 of *Lecture Notes in Computer Science*, pages 193–206. Springer, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.7480.

Michael R. Swihart and Jason D. Papastavrou. A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *European Journal of Operational Research*, 114(3):447–464, May 1999. doi: 10.1016/S0377-2217(98)00260-4. URL http://dx.doi.org/10.1016/S0377-2217(98)00260-4.

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

H. Yang, S. Wong, and K. Wong. Demand-supply equilibrium of taxi services in a network under competition and regulation. *Transportation Research Part B: Methodological*, 36(9): 799–819, November 2002. ISSN 01912615. doi: 10.1016/S0191-2615(01)00031-5. URL http://dx.doi.org/10.1016/S0191-2615(01)00031-5.

Jian Yang, Patrick Jaillet, and Hani Mahmassani. Real-Time multivehicle truckload pickup and delivery problems. *Transportation Science*, 38(2):135–148, May 2004. doi: 10.1287/trsc.1030.0068. URL http://dx.doi.org/10.1287/trsc.1030.0068.