A HYBRID FIXED AND FLEXIBLE TRANSPORTATION SERVICE:
DESCRIPTION, VIABILITY,
FORMULATION, OPTIMIZATION AND HEURISTIC

by

Luca Quadrifoglio

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(INDUSTRIAL AND SYSTEMS ENGINEERING)

August 2005

Table of Contents

List of Tables

List of Figures

Abstract

Most transportation systems fall into two broad categories: fixed route systems that are cost efficient but lack of flexibility and demand responsive transit (DRT) systems which are flexible but costly. The Mobility Allowance Shuttle Transit (MAST) service is a new concept in transportation that merges the flexibility of DRT systems with the low cost operability of fixed route bus systems. In a MAST system vehicles follow a base fixed route composed by a few mandatory checkpoints conveniently located at major connection points; given an appropriate slack time, vehicles are allowed to deviate from the fixed path within a proper service area to pick up and drop off passengers at their desired locations.

The purpose of this research is to address the gap in the research community by studying this hybrid fixed and flexible type of service providing insights of its challenges and foreseeing its performance for utilization in large scale as an alternative to conventional public services.

The system is defined as viable if the longitudinal velocity along the primary direction of the service is higher than a minimum threshold value to maintain the service attractive to customers. By using continuous approximations we develop a relationship between this velocity and the demand to assess the viability and aid in the setting of the main parameters of the service.

For the static operating scenario we provide a mathematical formulation of the MAST system as an integer linear program and we aim to find the optimal schedule. Because of the combinatorial nature of the problem we develop a set of valid inequalities to increase the lower optimality bound and efficiently speed up the search for the optimal solution.

For the dynamic operating scenario we develop a customized insertion scheduling algorithm, which includes control parameters to prevent the "wild" consumption of the slack time and significantly improve the performance of the algorithm. A comparison vs. conventional fixed route systems shows that MAST services are competitive with conventional ones and perform better under certain demand distributions.

1 Introduction

Transit is one of the vital service sectors of the present and the future US economy and it holds tremendous social significance. Transit systems are essential for preserving and revitalizing the nation's cities by minimizing congestion, urban sprawl, central city decline, and air pollution. Owing to their inherent ease of routing and low capital costs, bus transit systems in particular are integral to meeting the growing transportation requirements. However, today's urban transit systems are at a crossroads. On the one hand, demands on transit agencies for improved and extended services are increasing. Yet on the other, there is little public support for increases in fares or subsidies. Therefore, transit agencies are currently seeking ways to improve service flexibility in a cost efficient manner.

Most bus transit systems fall into two broad categories: fixed-route and demand responsive transit (DRT) systems. Fixed-route systems are typically more cost efficient because of the predetermined schedule, the large loading capacity of the vehicles and the consolidation of many passenger trips onto a single vehicle (ridesharing). However, the general public considers them to be inconvenient because of their lack of flexibility since either the locations of pick-up and/or drop-off points or the service's schedule do not match the individual rider's desires. Moreover, the total trip time is perceived as being too long and, for longer trips, there is often a need for transfers between vehicles. DRT systems instead provide much of the desired flexibility with a door-to-door type of service but they are much more costly to deploy and therefore largely limited to specialized operations such as taxicab, shuttle vans or Dial-a-Ride services mandated under the Americans with Disabilities Act (paratransit DRT). The National Transit Summaries and Trends (NTST) report for 2002 indicates that the average cost per passenger trip for DRT systems is $20.8 with fares ranging from $2-3. By way of contrast, the average cost per trip for fixed-route lines is $2.4 with fares being roughly 25% of the cost.

The Mobility Allowance Shuttle Transit (MAST) system is an innovative concept that merges the flexibility of DRT systems with the low cost operability of fixed-route bus systems. A MAST service has a base fixed-route that covers a specific geographic zone, with one or more mandatory checkpoints conveniently located at major connection points or high density demand zones; the innovative twist is that, given an appropriate slack time, buses are allowed to deviate from the fixed path to pick up and drop off passengers at their desired locations. The only restriction on flexibility is that the deviations must lie within a service area designed around the base fixed-route. Customers make a reservation in order to add their desired pick-up and/or drop-off stops in the schedule of the service. The MAST system works under a dynamic environment since the majority of the requests occur while the vehicle is on duty. Passengers willing to use the service as a regular fixed-route line traveling between checkpoints can do so without the need of going through a booking process.

Such a system already exists in a reduced and simplified scale. The Metropolitan Transit Authority (MTA) of Los Angeles County introduced MAST as part of its feeder line 646. Line 646 transports passengers between a large business hub in the San Pedro area of Los Angeles County and a nearby bus terminal. The area served by Line 646 is located close to the Los Angeles harbor and is one of the County's busiest commercial hubs, consisting of several warehouses, factories and offices. However, for safety reasons, employees of local firms working on night shifts have been finding it extremely inconvenient to walk to and wait at a bus stop. Therefore, Line 646 offers a MAST *nightline* service. During daytime, this line serves as a fixed-route service. During nighttime, the line changes to a MAST service and allows specific deviations of half a mile from either side of the fixed-route. Customers may call in to be picked up, or may ask the operator to be dropped off at their desired locations if within the service area.

The demand of line 646 is currently low enough to allow the bus operator to make all the decisions concerning accepting/rejecting requests and routing the vehicle. Clearly, in case of heavier demand in a potential daytime MAST operation and several requests for deviations, the operator would not be able to handle this task efficiently by him/herself and would need help from the recent developments in communication and computation technologies that allow real-time information about pick-up/drop-off requests and buses status to be used to re-route the vehicles dynamically by means of a scheduling algorithm.

While DRT systems focus strictly on point-to-point transport services, the hybrid characteristic of the MAST service adds additional and significant time and space constraints to the problem mainly due to the need of having the vehicles arrive at the checkpoints on or before their scheduled departure time. This is because checkpoints typically represent major transfer centers and serve simultaneously as pick-up and drop-off points, like regular fixed-route stops. Delays at the checkpoints would result in undesirable deviations from a predetermined fixed schedule and passengers missing their connections in case of late arrivals.

Although MAST systems can be considered as a special case of the Pickup and Delivery Problem (PDP) with time windows and there has been a significant amount of research on DRT systems like the PDP, we are unaware of any work performed on specifically studying systems such as the MAST service. The purpose of this research is to address the gap in the research community by studying this hybrid fixed and flexible type of service from a design, theoretical and operational points of view, providing insights of its challenges and foreseeing its performance for an utilization in large scale as an alternative to conventional public services.

The contribution of this research is as follows:

- From a design point of view, by utilizing a continuous approximations model, we provide insights about the relationships among the longitudinal velocity of the vehicles and the main parameters of the service to help in the design process (Chapter 4).
- From an operational perspective we develop scheduling tools. We first look at the problem in a static scenario (Chapter 5). MAST systems are mathematically formulated as mixed integer linear

programs. Since they are NP-Hard problems, we develop a set of effective valid inequalities.

- For a dynamic operating scenario (Chapter 6) we propose a customized insertion heuristic scheduling algorithm that makes use of proper control parameters.
- Finally, we present conclusions and future research (Chapter 7).

2 Literature review

Hybrid types of transportation systems have been only recently approached by researchers. Zhao and Dessouky (2004) studied the optimal service capacity through a stochastic approach. Malucelli et al. (1999) also approached the problem including it in a general overview of flexible transportation systems. Crainic et al. (2001) described the MAST concept and incorporated it in a more general network setting providing also a mathematical formulation.

The hybrid type of service that we are studying consists of the same vehicle performing the fixed and variable portions of the trip. There has been some work in studying hybrid systems in which different vehicles perform the fixed and variable portions. In the latter case, local service is provided by on-demand vehicles and line-haul service is provided by a fixed route line. Passengers switch vehicles at a transfer station. Aldaihani et al. (2004) develop a continuous approximation model for designing such a service. There has been some work in developing operational scheduling and routing policies for this latter type of hybrid system. Liaw et al. (1996) develop a scheduling heuristic based on a system in Ann Arbor, Michigan. Hickman and Blume (2000) develop an insertion heuristic and test it on a data set from Houston, Texas. Aldaihani and Dessouky (2003) develop a tabu search heuristic and test it on a data set from Antelope Valley in California. They show that shifting some of the demand to a hybrid service route (18.6% of the requests) reduces the on-demand vehicle distance by 16.6% without significantly increasing the trip times.

In the first part of this thesis we make use of continuous approximations to analyze the problem. As noted by Daganzo (1991), the main objective of this approach is to obtain reasonable solutions with as little information as possible. Hall (1986) also pointed out that continuous approximation are useful to develop models that are easy for humans to comprehend; on the other hand, he observed that these models should not replace but supplement the more detailed mathematical programming models. There is a significant body of work in the literature on continuous approximation models for transportation systems. Most of the work has been developed to provide decision support tools for strategic planning in the design process. Langevin et al. (1996) provide a detailed overview of the research performed in the field. They concentrate primarily on freight distribution systems, while we focus on public transport; but most of the issues of interest are common to both fields.

The second and third part of this thesis focus on finding a solution of the MAST scheduling problem utilizing an exact and an heuristic approach. As mentioned, MAST systems are related to DRT systems because they can be considered as a special case of the PDP with time windows and there is a significant body of work in the literature on routing and scheduling DRT systems. Savelsbergh and Sol (1995) and Desaulniers et al. (2000) provide detailed reviews of the PDP, examining mathematical formulations and solutions approaches presented by different authors. Due to the combinatorial nature of the problem (the PDP is NP-Hard) exact optimization methods are theoretically interesting but practically unsolvable. Therefore, most of the research efforts focus on heuristic approaches.

Here below we present a literature review of the most relevant works done in the field, categorized by their research focus.

General Continuous approximations:

Pioneering research on continuous approximation model dates back to the fifties. Beardwood et al. (1959) provided the first approximation formula to estimate the length of a Traveling Salesman Problem (TSP) tour in a compact zone with uniform demand density. Stein (1978a) and Jaillet (1988) integrated their work by estimating the value of the TSP tour length in case of Euclidean and rectilinear metrics. In general, geometrical probability has been extensively studied to provide estimates on the average distances among points for different shapes. We mention in this area the work of Ghosh (1951), Fairthorne (1965), Schweizer (1968), Christofides and Elion (1969), Bouwkamp (1977), Ruben (1978), Daganzo (1980, 1984b), Vaughan (1984), Koshizuka and Kurita (1991) and Stone (1991). Similar works on estimating TSP length have been developed from a more theoretical and multi-dimensional point of view by Verblunsky (1951), Rhee (1993) and Stadje (1995).

Continuous Models:

Szplett (1984) provides a review of the research performed on continuous models specifically for public transport. In this area we cite the work of Holroyd (1965), Newell (1979), Mandl (1980), Ceder and Wilson (1986), LeBlanc (1988), Chang and Schonfeld (1991a, 1991b), Chien and Schonfeld (1997) and Aldaihani et al. (2004) that studied the optimality of bus network systems. Lesley (1976a, 1976b), Vaughan and Cousins (1977), Wirasinghe and Ghoneim (1981) and Kuah and Perl (1988) analyzed the optimality of spacing between bus stops. The work of Daganzo (1984a) is especially related to our research because it introduces the concept of "strip strategy", providing an approximate estimate of the optimal width of a corridor in order to minimize the distance between points and therefore the length of the TSP tour while employing a simple no-backtracking routing policy along the strip.

Continuous models have also been utilized to examine DRT systems. Daganzo (1978) used an approximate analytical model to study many-to-many DRT systems. Jacobson (1980) and Bélisle (1989) also made use of an analytical model for DRT systems. Diana et al. (2005) provide an analytical model to determine the fleet size of a DRT system.

Exact algorithms:

Exact approaches to solve DRT systems provide optimal solutions, but the combinatorial nature of the problem limits the applicability of these methods to very small instances; therefore, they provide a good theoretical insight, but practically can not be used to solve real situations.

Psaraftis (1980) describes an exact backwards dynamic programming solution approach for the single vehicle Dial-a-Ride problem for static and dynamic environments without time windows. A forward dynamic programming approach is then presented in Psaraftis (1983a), to handle cases with time windows. The complexity of those algorithms is exponential and they can solve only small problems up to 10 customers.

Another dynamic programming approach for the single vehicle PDP with time windows is described in Desrosiers et al. (1986). The adopted techniques are very efficient and the running time of the algorithm increases slower with the problem size if the time windows are tighter. The increased efficiency of the procedure allows handling instances with up to 20 customers.

Sexton and Bodin (1985a, 1985b) and Sexton and Choi (1986) describe a Benders' decomposition approach to solve the single vehicle PDP with time windows and capacity constraints. The latter paper introduced the concept of soft instead of hard time windows, meaning that the time windows can be violated, but the solution will be arbitrarily and proportionally penalized for this gap in the objective function; an infinity penalty function would transform the problem to the hard time windows case. Note that the routing sub-problems are solved by a heuristic. Therefore, their optimization approach is not entirely an exact algorithm;

should the routing problem be solved optimally, the entire procedure would become an exact algorithm.

Dumas et al. (1991) present a Dantzig-Wolfe approach for optimally solving the multiple vehicles PDP with time windows and capacity constraints. The master problem is iteratively solved by a column generation algorithm and a branch-and-bound exploration tree, while the constrained shortest path sub-problems are solved by a forward dynamic programming algorithm.

Savelsbergh and Sol (1998) propose a branch-and-price based algorithm to solve the dynamic multi vehicle PDP. Their approach uses a sophisticated column management technique and incorporates heuristics in the pricing procedure to allow the algorithm to solve large instances quickly. Instances up to 30 customers were successfully solved.

Kalantari et al. (1985) apply a branch and bound algorithm to the PDP. All the arcs that violate the active precedence constraints are precluded in each branch. Fischetti and Toth (1989) develop an additive bounding procedure suitable for a branch-and-bound algorithm for the single vehicle PDP. Rulan and Rodin (1997) introduce a polyhedral approach and a branch-and-cut algorithm to solve the single vehicle PDP without capacity constraints.

An exact algorithm approach is described in Lu and Dessouky (2004). The optimization procedure utilizes a branch-and-cut technique to solve the multi vehicle PDP. An effective application of valid equality and inequality constraints helps the algorithm to reach solutions faster. Instances with up to 5 vehicles and 17 customers have been successfully optimized in a reasonable time.

Clustering algorithms:

Clustering approaches use the intuitive idea of merging together in a single point requests that are physically close to each other. The problem instances are reduced in size and therefore exact approaches can then be applied efficiently.

Ioachim et al. (1995) develop a clustering algorithm to solve the multi vehicle PDP with time windows. The requests are grouped together in mini-clusters and the problem is then solved by a column generation approach and compared to an existing parallel insertion heuristic. In addition, in order to allow the algorithm to handle even larger problems, the original network is reduced to a sub-network by eliminating some arcs.

Min (1989) also proposes a clustering heuristic approach to solve the vehicle routing problem. In addition, he allows single nodes to serve simultaneously as pickup and delivery points as often happens in practice; this concept is relevant for this research because, in MAST systems, checkpoints are exactly of this type.

The work of Daganzo (1984c) describes a checkpoint DRT system that combines the characteristics of both fixed route and door-to-door service. In a checkpoint system, a service request is still made but the pick-up and drop-off points are not at the door but at centralized locations called checkpoints. The author shows that a checkpoint system, intermediate between fixed route and DRT systems, can be useful only for a narrow range of demand density in a given service area. However, the MAST system conceptually differs from the checkpoint only system described, since it allows also for door-to-door requests.

Stein (1978a, 1978b) develops a probabilistic analysis of the PDP. His analysis shows how optimal path lengths can be bounded with high probability by a constant function of the service area and the amount of random requests from a uniform distribution. Based on these findings, he proposes heuristics to solve single and multiple vehicle problems for static and dynamic environments, basically partitioning the service area into sub-regions (clustering).

Local search techniques:

Local search techniques are those heuristics that start from an initial feasible solution and "move" locally in the neighborhood of the solution space. The main drawback is that the solution found might be a local optimum, potentially very far from the global optimal.

Psaraftis (1983b, 1983c) presents two heuristic approaches for the single vehicle Dial-a-Ride problem with no time windows. In the first one is described a *k*-interchange local search heuristic approach: the algorithm performs a local improvement of a current solution by substituting *k* arcs with new ones while maintaining feasibility; the algorithm complexity, with N requests, is $O(N^k)$. The second one is based on a Minimum Spanning Tree procedure: the solution is constructed from an initial MST following a simple but effective procedure; the complexity of this algorithm is O(N^2).

Local search procedures are reported in Van Der Bruggen et al. (1993). The single vehicle PDP with time windows is solved by an arc interchange procedure with variable depth. In addition, a simulated annealing procedure is introduced to prevent the algorithm from getting stuck into a local optimal.

Healy and Moll (1995) present another local search technique for the Dial-a-Ride problem. They introduce a new procedure called *sacrificing*: basically the algorithm is allowed to proceed not only towards lower cost feasible solutions, but also towards higher cost solutions and broader feasibility neighborhood.

Tabu search techniques have been applied by Rochat and Taillard (1995), Badeau et al. (1997) and Landrieu et al. (2001) for the vehicle routing problem with hard time windows; by Taillard et al. (1997) for the soft windows case; by Nanry and Barnes (2000) and Cordeau and Laporte (2003) to solve the multi vehicle PDP.

Insertion Heuristics:

Insertion heuristics are probably the most popular techniques. Campbell and Savelsbergh (2004) justify their extensive use in practice, because they are very fast and capable to handle large problems, provide good solutions compared to optimality, can easily handle complicating constraints and can be simply implemented in dynamic environments.

Jaw et al. (1986) illustrate a heuristic algorithm for the static multi vehicle PDP with time windows. The algorithm takes in consideration one customer at a time, evaluates all feasible insertions for pickup and delivery points and selects the one with the minimum cost; a new vehicle is assigned if no existing routes can accommodate a new customer. The insertion approach used in this work demonstrates its effective applicability for large amounts of customers because of its computational speed.

Madsen et al. (1995) implement an insertion heuristic approach for a partly dynamic multi vehicle PDP. Requests known in advance are considered as static, while real-time requests are handled in a sequential fashion.

Potvin and Roussean (1993) and Liu and Shen (1999) develop parallel regret insertion heuristic algorithms for the multi vehicle routing problem and with time windows. These algorithms create routes in parallel and use a generalized regret measure over all un-routed customers in order to select the next candidate for insertion. Diana and Dessouky (2004) apply the same concept for the PDP, with an appropriate metric that helps to overcome the myopic behavior that is often the drawback of such a method.

A parallel insertion heuristic is proposed by Toth and Vigo (1997) to solve the static multi vehicle PDP with soft time windows. The insertion algorithm is very fast and applied to the transportation of handicapped persons. A tabu search is also proposed to improve the solution generated by the insertion heuristic. Their algorithm is tested with real life instances in the city of Bologna, Italy and their results outperform the hand made schedules.

Teodorovich and Radivojevic (2000) use a fuzzy logic insertion approach to solve dynamically the Dial-a-Ride problem. This approximate reasoning algorithm allows the insertion procedure to be executed in real time.

Dessouky, Rahimi, and Weidner (2003) develop an insertion procedure with objective function that includes both cost and environmental impact objectives. Experimental analysis on data sets representing dial-a-ride operations in Los Angeles County show that the best fleet composition is not, necessarily, a fleet comprised exclusively of vehicles selected to optimize one objective or the other.

Lu and Dessouky (2005) present a new insertion based construction heuristic to solve the multi-vehicle pickup and delivery problem with hard time windows. The new heuristic does not only consider the classical incremental distance measure in the insertion evaluation criteria but also the cost of reducing the time window slack due to the insertion. They also present a new non-standard measure, *Crossing Length Percentage,* to evaluate the 'visual attractiveness' of the route. The effect of using the proposed measure to guide the construction heuristic in obtaining a higher quality solution has also been investigated. They compared the heuristic to a standard insertion heuristic on different benchmarking problems, and the computational results show that the proposed heuristic performs better with respect to both the standard and non-standard measures.

Theoretical and technologically based works:

A few authors provide theoretical insights to the problem, while recent developments in the field based on technologically advanced systems, have attracted the attention of the researchers in the latest years.

Feuerstein and Stougie (2001) investigate the best possible competitive ratio for an on-line single server dial-a-ride problem. They show that no heuristic algorithm can have a *competitive ratio* better than 2, where the *competitive ratio* is the worst case ratio between the objective value produced by the algorithm and the optimal value.

Dial (1995) proposes a fully automated routing and scheduling system, where the customer is the only human interacting with it in the entire process of booking a ride. The system is embedded in a decentralized control strategy.

Horn (2002b) develops an algorithm for the scheduling and routing of a fleet of vehicles that is embedded in a modeling framework for the assessment of the performance of a general public transport system with the latter being presented in Horn (2002a).

Fu (2002) presents a simulation model to test if and how the introduction of technologically advanced paratransit services can be beneficial. The results seem promising, but in some cases a decline on the overall performance of the system is observed.

3 Description of MAST systems

A Mobility Allowance Shuttle Transit (MAST) system is represented by a fleet of vehicles serving a set of customers' requests. Vehicles follow a fixed-route line (back and forth between two terminal checkpoints or around a loop, see Figure 1) composed by an ordered set of stops (*checkpoints*) associated with prescheduled departure times.

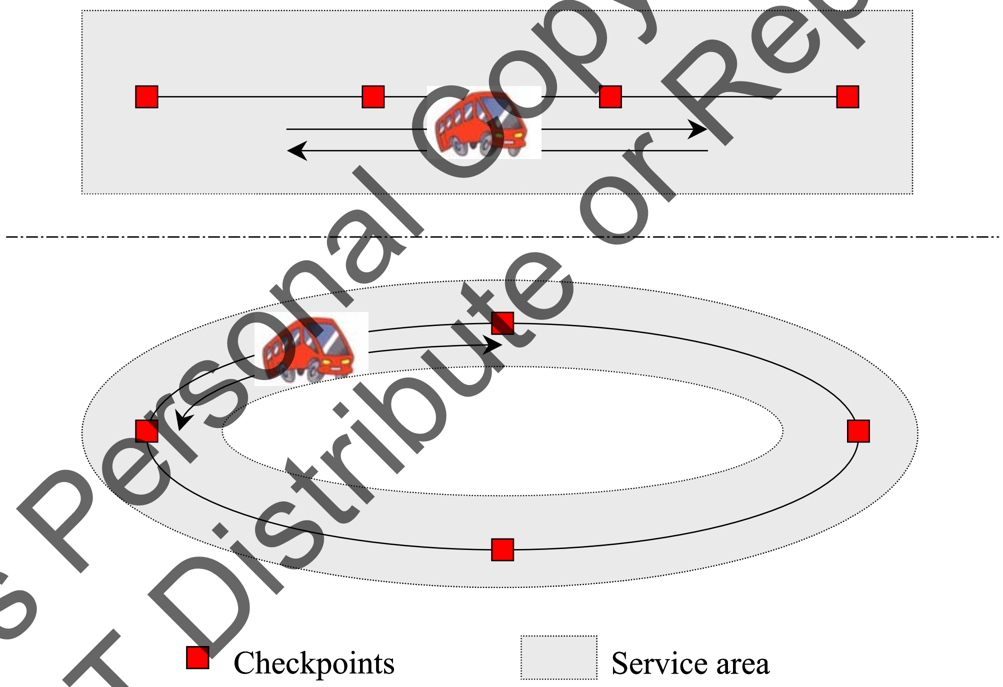

Figure 1 – Possible configurations of MAST systems

Each customer's request is defined by pick-up/drop-off service stops and a ready time for pick-up. The MAST service can respond to four different types of requests: pick-up (P) and drop-off (D) at the checkpoints; non-checkpoint pick-up

(NP) and drop-off (ND), representing customers picked up/dropped off at any location within a *service area* designed around the base fixed-route. A certain amount of *slack time* between any consecutive pair of checkpoints is needed in order to allow deviations to serve NP or ND requests.

There are consequently four different possible types of customers' requests:

- PD ("regular"): pick-up and drop-off at the checkpoints
- PND ("hybrid"): pick-up at the checkpoint, drop-off not at the checkpoint
- NPD ("hybrid"): pick-up not at the checkpoint, drop-off at the checkpoint
- NPND ("random"): pick-up and drop-off not at the checkpoints

PD requests rely only on already scheduled checkpoints and they use the service like a regular fixed-route line; therefore, they just show up at their pick-up checkpoint, not needing any booking or scheduling procedure. The other types of requests need to make reservations instead (by phone, internet or at terminals located at the checkpoints) to schedule one or both their non-checkpoint service stops.

The service can work dynamically, so that customers may book their rides (or show up at the checkpoints) at any moment before or during the service. An effective MAST system needs to rely on recent developments in communication and computation technologies that allow real-time information about pick-up/drop-off requests to be used to re-route the vehicle by means of a scheduling algorithm. Its necessity becomes more evident in case a MAST system serves a heavy demand and relies on a fleet of vehicles and/or MAST networks. Ideally after each request the vehicles' routes are updated in real time and customers are immediately notified whether their request has been accepted, postponed or rejected and are provided with an approximate time (or time windows) for their pick-up and/or drop-off at their non-checkpoint locations.

4 Design perspective

One of the factors that has to be taken into consideration in the design process of a MAST system is its *viability*. In fact, in case of public transport, the main purpose of the vehicle is to move customers along a primary direction (see Figure 2). The more customers that are served, the slower the vehicle would move along this direction because of the deviations needed for pick-ups and drop-offs.

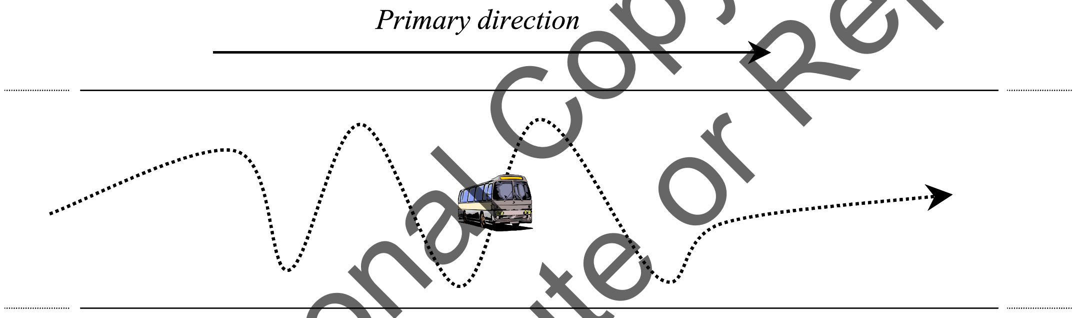

Figure 2 – Primary direction of MAST services

The purpose of this chapter is to provide insights about the *viability* of MAST systems to help in the design process of the main system parameters. The service is defined *viable* if the velocity of the vehicle along this direction is kept high enough to maintain the service attractive to customers while serving a sufficient demand. A minimum threshold value of the velocity can be used to set the maximum slack time allowable between checkpoints and to determine the maximum demand level that can be served by one vehicle and the number of vehicles to be employed per line.

4.1 System definition

The MAST system model considered for our analysis is described by a linear corridor of width W and length L >> W, oriented in a horizontal direction. The demand is assumed to be known in advance and is represented by a set of passenger trips that occur with density ρ per unit area. We do not need to know the distribution of the trip length for the purpose of this Chapter. Pick-up and drop-off points are uniformly distributed across the width and the length of the corridor. Vehicles follow rectilinear paths within the corridor and travel with constant speed v, except at pick-up/drop-off points where there is a constant stop time of b.

Vehicles serve passenger trips by following a forward progression through the corridor in either a left-right or right-left direction. This means that a left-right (right-left) vehicle rides from the checkpoint at the left (right) end to the checkpoint at the right (left) end of the corridor and only serves customers whose drop-off is to the right (left) of the pick-up (see Figure 3). This is only a reasonable operating policy, but not necessarily optimal. We also assume no mandatory checkpoints in between.

The general system is represented by several vehicles riding along the corridor, but we assume that ρ represents one cycle of the demand served by two vehicles with infinite capacity: a left-right one and a right-left one. The problem is symmetrical and we analyze only the left-right case, with a demand density per unit area of ρ/2 trips and ρ stops, since each trip has two service points (we are simplifying the analysis by assuming that ρ includes NPND requests only).

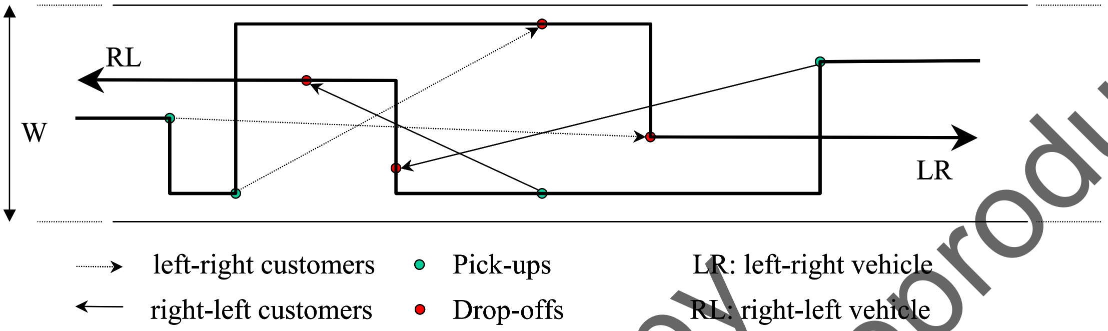

Figure 3 – Right-left and left-right vehicles

The longitudinal velocity V of the vehicle is defined by the rate at which the vehicle moves in the horizontal direction which has the average given by

$$V = \frac{L}{\frac{p}{v} + \rho WLb} \tag{4.1}$$

where p is the length of a rectilinear Hamiltonian path among all the service points, respecting the customer precedence constraints (for each customer the drop-off must be scheduled after the pick-up) and ρWLb represents the total stop time (counting pick-up and drop-off stops). The problem (P) is to minimize p (with optimal value p*), which corresponds to maximize V (with optimal value V*).

Because P is NP-Hard, its combinatorial nature causes it to be unsolvable in reasonable time for a large number of stops; in addition, we do not know the exact locations of the demand points but only their distribution. Therefore, we proceed by generating lower and upper bounds on V*, assuming continuous approximations of the system parameters. We also provide an estimate of V* based on the results from Beardwood (1959) and we perform a simulation to compute the velocity when utilizing an insertion heuristic algorithm.

A summary of the system parameters and the notation used in this chapter are as follows:

W	Width of corridor (miles)
L	Length of the corridor (miles); L >> W
v	Average vehicle speed (miles/hour)
b	Stop time while serving pick-up/drop-off (hours)
ρ	Demand density (customers/miles^2)
	= stop density for the left-right case (stops/miles^2)
V	Longitudinal velocity of the vehicle (miles/hour)
V*	Optimal (maximum) value of V
$i \in N$	Set of all "left-right" stops (pick-ups and drop-offs)
x_i	longitudinal coordinate of *i*, increasing from left to right (miles)
y_i	lateral coordinate of *i*, increasing from bottom to top (miles)

4.2 Lower bound of V*

Let's consider a no-backtracking policy, allowing the vehicle to move only in the forward direction (left to right) and serve all the demand, as illustrated in Figure 4. The numbers represent the customers, where "+" is a pick-up and "-" a drop-off.

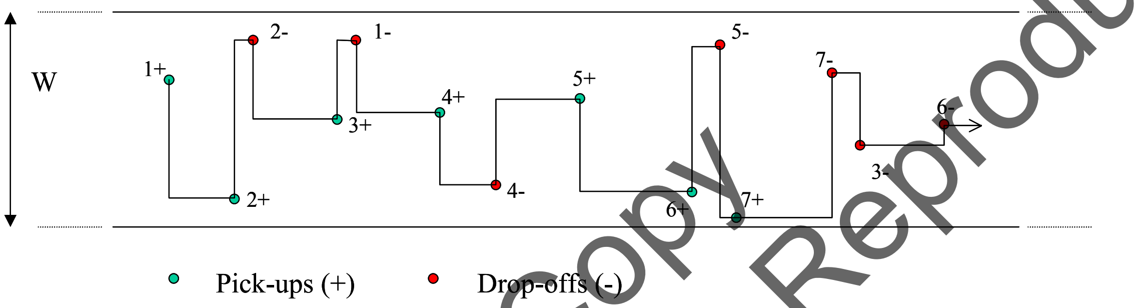

Figure 4 – No-backtracking policy

Since by assumption the customers served by a left-right vehicle have their drop-off always on the right of their pick-up, this policy guarantees feasibility because all origins are served before their destination points, satisfying all the customer precedence constraints. However, this policy is not necessarily optimal. In fact, the solution could be improved by simply removing the arbitrary "no-backtracking" constraint and developing a better routing strategy. Thus, this simple no-backtracking policy provides a feasible lower bound on V*, but it does not provide optimality. For our purpose it is useful because we can compute its closed form solution.

Because of the uniformity of the demand we know that the expected value of the lateral (along the vertical direction) distance l_y traveled by the vehicle between any pair of stops is given by

$$E(l_y) = \frac{W}{3} \tag{4.2}$$

and the expected value of the time t_y spent by the vehicle while moving laterally when traveling between two consecutive points given by

$$E(t_y) = \frac{E(l_y)}{v} = \frac{W}{3v} \tag{4.3}$$

In a corridor L×W with given density ρ there are ρWL stops and the expected value of the total time t spent by the vehicle while driving along the corridor is given by

$$E(t) = \rho WL\left[E(t_y) + b\right] + \frac{L}{v} = L\left[\rho W\left(\frac{W}{3v} + b\right) + \frac{1}{v}\right] \tag{4.4}$$

where L/v is the time spent by the vehicle while moving longitudinally along the corridor and b is the service time per stop.

Finally, the lower bound on V* is formally computed by L×E(1/t) which is very well approximated and lower bounded by L/E(t). In fact we know by the Jensen inequality that E(1/t) ≥ 1/E(t). Therefore, we compute the lower bound V^l by

$$L \times E\left(\frac{1}{t}\right) \geq V^l = \frac{L}{E(t)} = \frac{v}{1 + \rho W\left(bv + \frac{W}{3}\right)} \tag{4.5}$$

We can also verify by the following Table 1 the good estimates provided by Equation (4.5) on the true lower bound L×E(1/t) computed by simulation for different values of ρ. The simulation values are obtained by averaging 30 replications for each ρ considered. In each replication we considered 5,000 stops uniformly distributed in a corridor of width W = 0.5 miles, a v = 30 miles/hour and b = 30 seconds.

Table 1 – V^l values: analytical vs. simulation

	V^l (miles/hour)	
ρ	Equation (4.5)	Simulation
1	24.83	24.83
5	14.69	14.69
10	9.73	9.73
50	2.63	2.63
100	1.37	1.37

Note that V^l is inversely proportional to W, b and ρ. With $\rho = 0$ (no demand) the vehicle would have its $V^l = v$ as expected.

4.3 Optimality of no-backtracking policy

Before estimating the first upper bound we want to focus on the “strip strategy” introduced by Daganzo (1984a). He showed that good solutions of the TSP tour, for any shape of the service area, are obtained by cutting a swath covering the whole area and having the vehicle drive along the resulting long strip while serving the demand uniformly distributed in the area. He claimed that selecting a proper width of the strip a simple no-backtracking policy produces good results in terms of the total distance traveled.

We want to determine if there exists any sufficient condition on the locations of the demand points that would guarantee optimality of a no-backtracking routing policy. This would allow us to select a subset of points that satisfy this condition so that we can utilize the no-backtracking routing policy to serve them optimally. The longitudinal velocity to serve this subset will be an upper bound on V*.

To find out whether this sufficient condition exists, let’s consider a left-right vehicle following a Hamiltonian path (α) among a set of demand points. Referring to Figure 5, consider points *j*, *h* and *k*. We assume that $x_h \le x_j$ and $x_h \le x_k$ and that the backtracking sub-sequence …-*j*-*h*-*k*-… is part of path α. We want to determine if there exists a condition on x_h with respect to x_j and x_k to guarantee that a reinsertion of *h* earlier in the schedule in a no-backtracking fashion will always lead to a shorter total distance traveled.

It is always possible to identify two consecutive points *a* and *b* earlier in the schedule such that $x_a \leq x_h \leq x_b$ (at the limit, we can have *a* be the checkpoint in the far left and/or $b \equiv j$). Therefore, we have path α following the sequence …-*a*-*b*-…-*j*-*h*-*k*-….

Consider another path (β) that follows the sequence …-*a*-*h*-*b*-…-*j*-*k*-… with point *h* reinserted between *a* and *b* in a no-backtracking fashion.

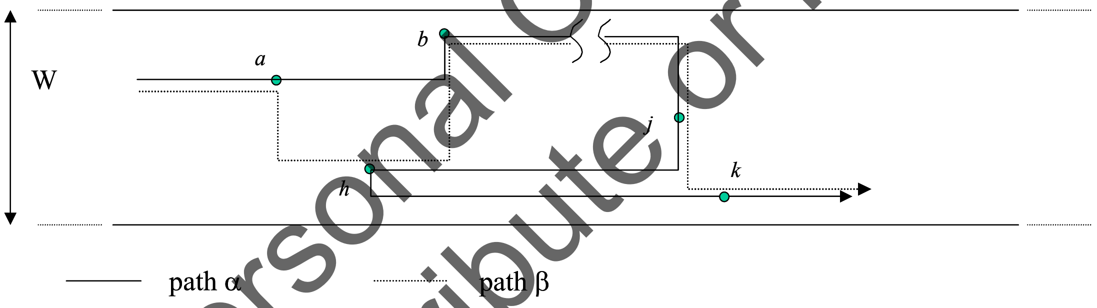

Figure 5 – Two different paths to serve points *a*, *b*, *j*, *h*, *k*

Let's compute the rectilinear distance driven in the two cases, considering only the relevant portions of the sequences that differ between α and β. Path α yields to the following distance l_α:

$$l_\alpha = x_b - x_a + |y_b - y_a| + x_j - x_h + |y_j - y_h| + x_k - x_h + |y_k - y_h| \tag{4.6}$$

For the path β the distance l_β is given by

$$l_\beta = x_h - x_a + |y_h - y_a| + x_b - x_h + |y_b - y_h| + |x_k - x_j| + |y_k - y_j| \tag{4.7}$$

We want to determine the minimum longitudinal distance between *h* and *j* and/or *h* and *k* needed to guarantee that path β will always be better than path α in terms of minimizing the total distance traveled. Therefore, we impose the condition $l_\beta \le l_\alpha$ and after a few passages we obtain the following inequality:

$$\begin{aligned} & x_j + x_k - |x_k - x_j| - 2x_h \ge \\ & |y_h - y_a| + |y_b - y_h| - |y_b - y_a| + |y_k - y_j| - |y_j - y_h| - |y_k - y_h| \end{aligned} \tag{4.8}$$

Depending on the random lateral position of the points along the corridor, the maximum possible value for $|y_h - y_a| + |y_b - y_h| - |y_b - y_a|$ is 2W when *h* is located on the opposite edge of the corridor with respect to *a* and *b*; while $|y_k - y_j| - |y_j - y_h| - |y_k - y_h|$ can be at most equal to 0 when *h* is located laterally in between *j* and *k*. Otherwise, it is less than 0. Therefore, the right-hand side of Equation (4.8) is less than or at most equal to 2W and the inequality becomes

$$\min(x_j, x_k) - x_h \ge W \tag{4.9}$$

that is the sufficient condition on the longitudinal position of *h*, with respect to the closest (longitudinally) point between *j* and *k*, that would guarantee that the reinsertion of *h* somewhere earlier in the schedule in a no-backtracking fashion between some points *a* and *b* would always lead to a better solution in terms of shorter distance traveled.

Given the result obtained by Equation (4.9) we can state the following.

Proposition 1. *Given a set of points randomly distributed along a corridor of width W and length L, the shortest Hamiltonian rectilinear path from the first point on the far left to the last point on the far right is the sequence of points ordered by increasing longitudinal coordinate (no-backtracking), as long as the minimum longitudinal distance between any pair of points is at least W.*

Proof. Consider a set of points identified by $i = 1, 2, 3,\ldots$ and ordered by increasing longitudinal coordinate (no-backtracking) and let the minimum longitudinal distance between any pair of points be at least W. Assume that there exists an optimal sequence Λ ordered not following a no-backtracking policy; the position of each *i* in Λ is identified by $\lambda(i)$. Let's consider the smallest point $i_0 \neq \lambda(i_0) \in \Lambda$. We can show by Equation (4.9) that by reinserting i_0 in Λ such that $i_0 = \lambda(i_0)$ and readjusting all other $\lambda(i)$ accordingly leads to a better solution. But this is a contradiction, because we supposed Λ to be optimal, therefore, the no-backtracking policy is optimal. □

4.4 First upper bound on V*

To create an upper on V* we first identify a subset $G[g(1), g(2), g(3),\ldots] \subseteq N$ of points such that the longitudinal distance between any pair of them is as small as possible but at least W. By Proposition 1 we know that the optimal routing policy to serve the subset G is given by a no-backtracking sequence. We then assume that all the points $i \in N$, but $i \notin G$, will be served as well, but that no additional lateral deviations are required to reach them. This is a subproblem P' (with optimal value p'*) of the original problem P. We know by construction that $p'^* \leq p^*$, because in computing the total distance traveled in P' we are ignoring some of the vertical deviations and possible backtracking portions of the path needed to attain p*. Therefore, this policy guarantees optimality of the subproblem P', without assuring feasibility of P, and represents a lower bound on the total minimum distance traveled (thus, an upper bound V*).

To construct the subset G from the set N, we can use the following algorithm:

Algorithm 1

1. g(1) is the first point on the far left of the corridor
2. g(i+1) is the longitudinally closest point to the right of g(i) after a “jump” of W units of length to the right of g(i); with i = 1, 2, 3,…
3. Repeat step 2 until there are no more points

As an example, referring to Figure 6, we first include 1+ in the subset G; then, from its horizontal coordinate x_{1+} we move W units of length to its right and we include in G the longitudinally closest point to the right of the location x_{1+}+W (point 4+); and we proceed in this fashion including in G points 6+, 6-, etc…, until the end.

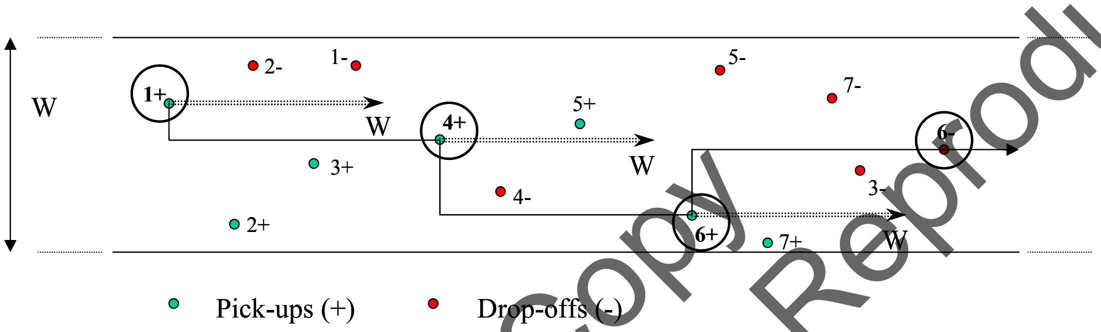

Figure 6 – subset G: longitudinal distance of at least *w* among points.

We know from Equation (4.2) that the expected lateral distance driven by the vehicle while moving between any pair of points is given by $E(l_y) = W/3$. Equation (4.3) provides the expected time $E(t_y) = W/3v$ spent by the vehicle while moving along l_y.

The expected longitudinal distance $E(l_x)$ between two consecutive points in the subset G is given by

$$E(l_x) = W + \frac{1}{\rho W} \tag{4.10}$$

where W is the minimum step and 1/ρW is the expected longitudinal distance to be traveled in order to find the next closest point on the right. This occurs because with uniformly, independently and randomly scattered points on the corridor, the positions along the side of the strip at which points lie form (locally) a Poisson process with rate ρW. Thus, the expected number of points E(n_G) in G is given by

$$E(n_G) = \frac{L}{W + \frac{1}{\rho W}} = \frac{\rho W L}{1 + \rho W^2} \qquad (4.11)$$

We note that higher values of n_G would lead to smaller gaps between p'* and p*, because of the smaller number of stops (and lateral deviations) ignored. Thus, the bound is tighter for narrower corridors (smaller W) and sparser demand density (lower ρ).

The expected value of the total time t' spent by the vehicle while moving along the corridor, including the stop time b to serve each point in N and the time spent moving longitudinally, is given by

$$E(t') = E(n_G)E(t_y) + \frac{L}{v} + \rho W L b = \frac{L}{v}\left[1 + \rho W b v + \frac{\rho W^2}{3\left(1 + \rho W^2\right)}\right] \qquad (4.12)$$

Finally, the upper bound V^u on V^* would be given by $L \times E(1/t')$ which does not have a closed form solution, but it is very well approximated by $L/E(t')$ and therefore given by

$$V^u \cong \frac{L}{E(t')} = \frac{v}{1 + \rho W b v + \dfrac{\rho W^2}{3\left(1 + \rho W^2\right)}} \tag{4.13}$$

Again, we can verify the good estimate of V^u provided by Equation (4.13) by looking at the following Table 2. The simulation values are obtained as for Table 1 (30 replications; 5,000 stops uniformly distributed; $W = 0.5$ miles; $v = 30$ miles/hour and $b = 30$ seconds).

Table 2 – V^u values: analytical vs. simulation

	V^u (miles/hour)	
ρ	Equation (4.13)	Simulation
1	25.17	25.18
5	16.57	16.57
10	12.06	12.06
50	3.97	3.97
100	2.17	2.17

As for V^l, V^u is inversely proportional to W, b and ρ; with $\rho = 0$, $V^u = v$.

4.5 Second upper bound on V*

To produce the second upper bound we again remove constraints from problem P. The Hamiltonian path among all the points requires exactly one incoming arc and one outgoing arc at each node of the network so that all the points are connected to complete the tour. We remove the first assumption allowing unlimited incoming arcs at any node, but we still require exactly one outgoing arc from each node. In addition, we remove the customer precedence constraints. This is another subproblem P″ (with optimal value p″*) of the original problem P. p″* is given by the summation over all the stops of the arcs connecting any stop to its closest neighbor. In other words, we are stating that from each stop the vehicle has to travel at least to its closest neighbor; the sum over all stops produces p″*, which is a lower bound on p* and which therefore yields to an upper bound on V*.

We know that uniformly and randomly scattered points follow a spatial Poisson distribution. Specifically, the number of points Γ(A) within the area A is a Poisson random variable and its distribution is given by

$$\Pr\left[\Gamma(\mathrm{A}) = q\right] = \frac{(\rho\mathrm{A})^q}{q!} e^{-\rho\mathrm{A}} \qquad q = 0, 1, 2, 3, \ldots \qquad (4.14)$$

with expected value equal to ρA.

Let D be the random variable indicating the distance of the closest neighbor from any stop $i \in \mathrm{N}$. We can say that

$$\overline{F}(d) = \Pr(\mathrm{D} > d) = \Pr[\Gamma(\mathrm{A}(d)) = 0] = e^{-\rho \mathrm{A}(d)} \tag{4.15}$$

where A(d) is the area around i within rectilinear distance d falling in the corridor.

We want to calculate the expected value E(D). For the purpose of this analysis we assume the limiting case where L/W→∞, ignoring the effect on the calculation given by the left and right ends of the corridor.

Let y be the distance of a random stop $i \in \mathrm{N}$ from the nearest edge of the corridor, which we suppose to be the bottom one without loss of generality. Depending on d we can have three different scenarios to compute A(d) as shown in Figure 7.

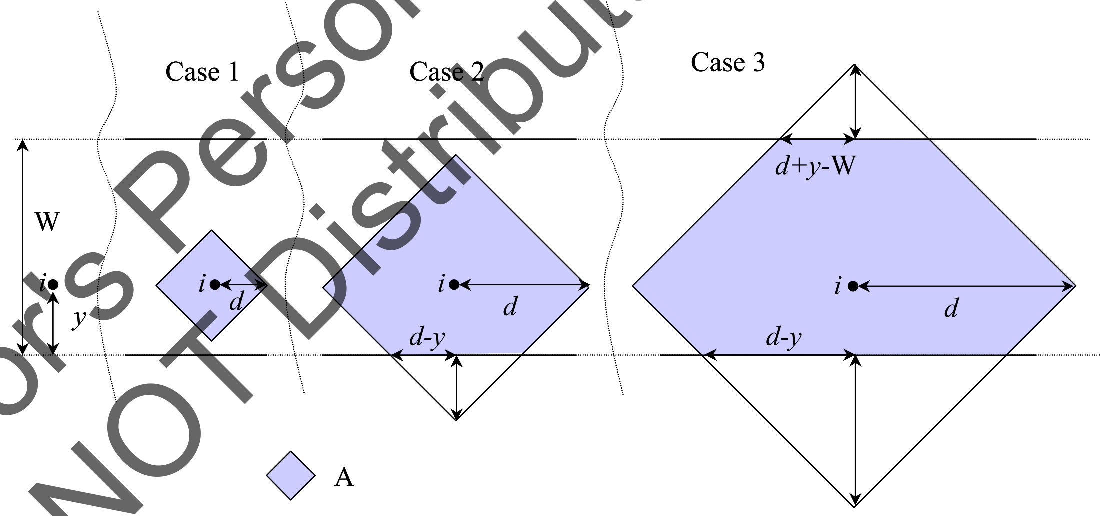

Figure 7 – A depending on d

Case 1. $$\mathrm{A}(d) = 2d^2 \qquad 0 \le d \le y \tag{4.16}$$

Case 2. $$\mathrm{A}(d) = 2d^2 - (d-y)^2 = d^2 - y^2 + 2dy \qquad y \le d \le \mathrm{W}\text{-}y \tag{4.17}$$

Case 3. $$\begin{aligned} \mathrm{A}(d) &= 2d^2 - (d-y)^2 - (d+y-\mathrm{W})^2 = \\ &= \mathrm{W}(2d+2y-\mathrm{W}) - 2y^2 \end{aligned} \qquad d \ge \mathrm{W}\text{-}y \tag{4.18}$$

The expected value of D depending on *y* is given by

$$\mathrm{E}(\mathrm{D}(y)) = \int_0^\infty \overline{F}(d)\,\mathrm{d}\,d = \int_0^\infty e^{-\rho \mathrm{A}(d)}\,\mathrm{d}\,d \tag{4.19}$$

Averaging over all values of *y* between 0 and W/2 we finally obtain

$$\mathrm{E}(\mathrm{D}) = \frac{2}{\mathrm{W}} \int_0^{\mathrm{W}/2} \mathrm{E}(\mathrm{D}(y))\,\mathrm{d}\,y \tag{4.20}$$

Note that the analysis for W/2 ≤ *y* ≤ W is symmetrically the same.

Equation (4.20) does not have a closed form solution, but we can examine two limiting scenarios, depending on the value of the parameter $\mathrm{W}\sqrt{\rho}$, that is an indication of the effect of the edges of the strip on the calculation of E(D).

If $\mathrm{W}\sqrt{\rho} \to \infty$ we can approximate E(D) by considering only Case 1 and compute A(*d*) by Equation (4.16). For the majority of the points, the probability of finding the closest point in an area defined by Case 2 or Case 3 is negligible, either because the edges are too far (large W) or because the density is very high. Therefore,

$$\mathrm{E}(\mathrm{D}) \approx \int_0^\infty e^{-2\rho d^2}\,\mathrm{d}\,d = \frac{1}{2}\sqrt{\frac{\pi}{2\rho}} \approx \frac{0.63}{\sqrt{\rho}} \tag{4.21}$$

If $\mathrm{W}\sqrt{\rho} \to 0$ we can approximate E(D) by considering only Case 3 and compute A(*d*) by Equation (4.18). For the majority of the points, the probability of finding the closest point in an area defined by Case 1 or Case 2 is negligible, either because W is very small or the density is very low. Therefore, we obtain

$$\mathrm{E}(\mathrm{D}) \approx \int_0^\infty e^{-\rho\left[\mathrm{W}(2d+2y-\mathrm{W})-2y^2\right]}\,\mathrm{d}\,d = \frac{1}{2\rho\mathrm{W}} \tag{4.22}$$

that also corresponds to the expected distance of the closest point (in either direction) in a one-dimensional case with all the points uniformly distributed along a line with linear density ρW.

We performed a numerical integration on Equations (4.19) and (4.20) with $W = 1$ for different values of ρ. The results are shown in the following Figure 8 along with the figures computed by simulations performed with $L >> W$, in dimensionless form.

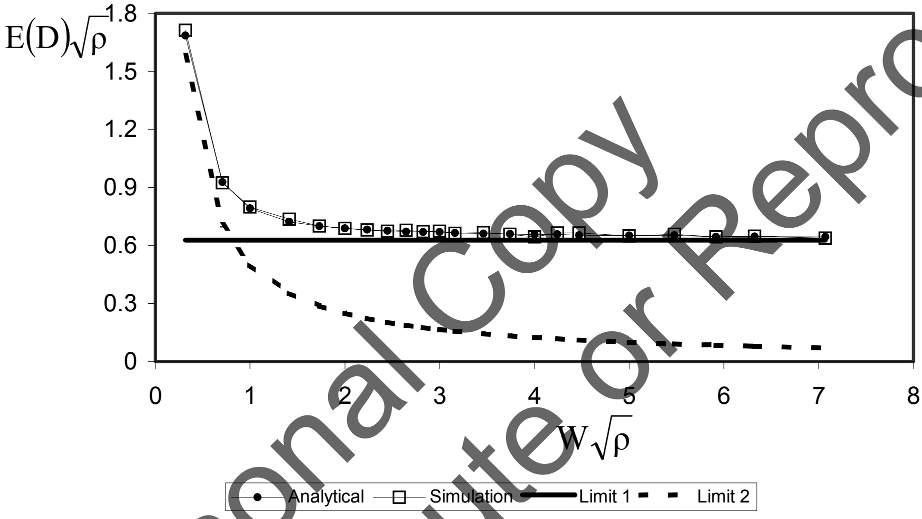

Figure 8 – E(D) vs. simulation and asymptotic limits

“Analytical” refers to the values computed by numerical integration; “Limit 1” and “Limit 2” refer to Equation (4.21) and (4.22) respectively. The chart shows that the “Analytical” curve is asymptotically bounded by the two limits for $W\sqrt{\rho} \to 0$ and $W\sqrt{\rho} \to \infty$ as expected. In addition, the “Simulation” curve closely matches the “Analytical” curve, confirming that assuming $L/W\to\infty$ instead of $L >> W$ did not affect the results even for lower values of ρ.

The expected value of the time t″ spent by the vehicle is given by

$$E(t'') = \rho W L\left[b + \frac{E(D)}{v}\right] \tag{4.23}$$

Finally, we have that the upper bound on V* is given by L×E(1/t″), very well approximated by L/E(t″). Therefore the second upper bound V^{u2} on V* is given by

$$V^{u2} \cong \frac{L}{E(t'')} = \frac{v}{\rho W[bv + E(D)]} \tag{4.24}$$

As before, we can verify the good estimate of V^{u2} provided by Equation (4.24) by looking at the following Table 3. The simulation values are obtained as for Table 1 (30 replications; 5,000 stops uniformly distributed; W = 0.5 miles; v = 30 miles/hour and b = 30 seconds).

Table 3 – V^{u2} values: analytical vs. simulation

	V^{u2} (miles/hour)	
ρ	Equation (4.24)	Simulation
1	42.53	42.49
5	20.29	20.30
10	12.65	12.64
50	3.49	3.49
100	1.90	1.90

As for V^l and V^u, V^{u2} is inversely proportional to W, b and ρ. With $\rho \to 0$ or $\rho \to \infty$, by applying Equation (4.21) and (4.22), the asymptotic values of V^{u2} are given by

$$\lim_{\rho\to\infty} V^{u2} = \frac{v}{\rho W b v + 0.63 W \sqrt{\rho}} \tag{4.25}$$

$$\lim_{\rho\to 0} V^{u2} = 2v \tag{4.26}$$

4.6 Approximate value for V*

We know by Beardwood et al. (1959) and Jaillet (1988) that the length T of the optimal TSP tour for rectilinear metric visiting M points distributed randomly in a region of area A is approximated by the following formula:

$$T = 0.97\sqrt{AM} \tag{4.27}$$

This formula provides better approximations with large values of M.

To make use of this result for our case, we assume that the MAST vehicle is driving along a long corridor that is shaped as a loop, having the starting and ending checkpoints of the vehicle coincide. Since we assumed L >> W, we can approximate the ring-shaped service area A by

$$A = LW \tag{4.28}$$

and estimate the optimal length of the tour by Equation (4.27) for different values of M given by

$$M = \rho A = \rho WL \tag{4.29}$$

Since the total time t_a spent to complete a loop is given by

$$t_a = \frac{T}{v} + Mb = LW\sqrt{\rho}\left(\frac{0.97}{v} + \sqrt{\rho}b\right) \tag{4.30}$$

The resulting approximation of the optimal longitudinal velocity V^a is given by

$$V^a = \frac{L}{t_a} = \frac{v}{\rho Wbv + 0.97W\sqrt{\rho}} \tag{4.31}$$

which has the same form as the asymptotic value of V^{u2} for ρ going to infinity, given by Equation (4.25), with 0.97 replacing 0.63. As for V^l, V^u and V^{u2}, V^a is inversely proportional to W, b and ρ. However, V^a goes to infinity when ρ goes to zero, confirming that Equation (4.31) does not provide good estimates for low ρ.

We need to emphasize that V^a is neither an upper nor a lower bound of V^* and it does not consider the customer precedence constraints.

4.7 Insertion heuristic simulation value for V*

In addition to the bounds and the approximation formula, we computed V^i, representing the longitudinal velocity obtained by simulation for different values of ρ, while implementing a simple insertion heuristic algorithm to schedule the uniformly distributed demand. Insertion algorithms generally provide good feasible solutions, but they do not guarantee optimality. Therefore, the resulting curve represents a lower bound for V^* as well, but since it does not have a closed form solution it can not be quickly computed like V^l for different scenarios.

4.8 Viability

We are now able to plot the lower bound V^l, the upper bounds V^u and V^{u2}, the approximate value V^a, respectively using Equations (4.5), (4.13), (4.24) and (4.31), and V^i from the insertion heuristic to analyze the results. The curves are drawn utilizing a vehicle speed $v = 30$ miles/hour and a service time $b = 30$ sec for each stop. We analyze two different cases, with $W = 1$ (see Figure 9), consistent with the existing MAST system (line 646 in the Los Angeles County), and $W = 0.5$ miles (Figure 10).

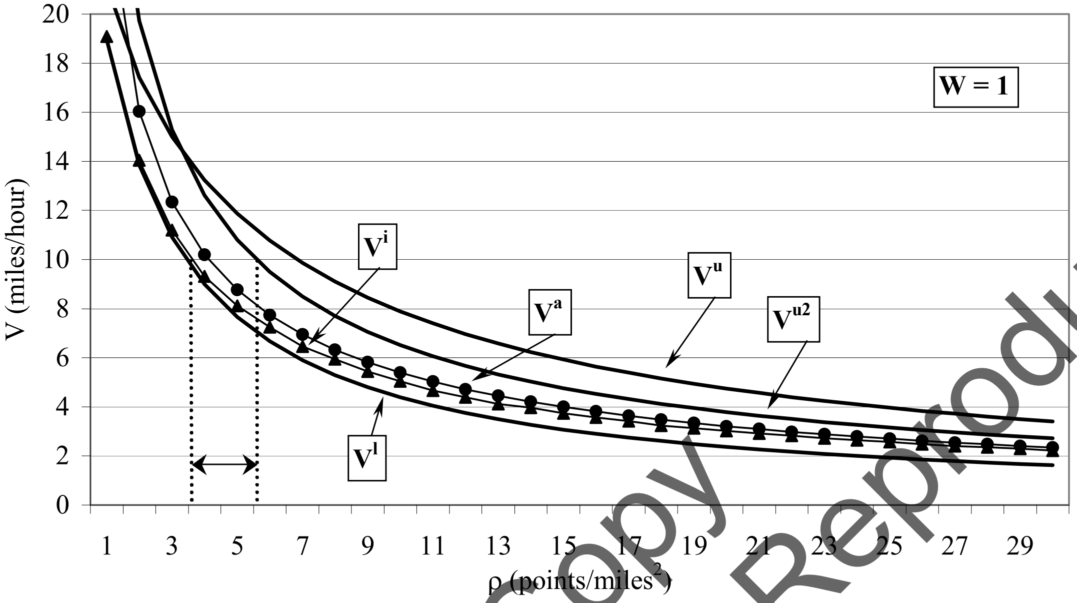

Figure 9 – Longitudinal velocity (V) vs. demand density (ρ); W = 1

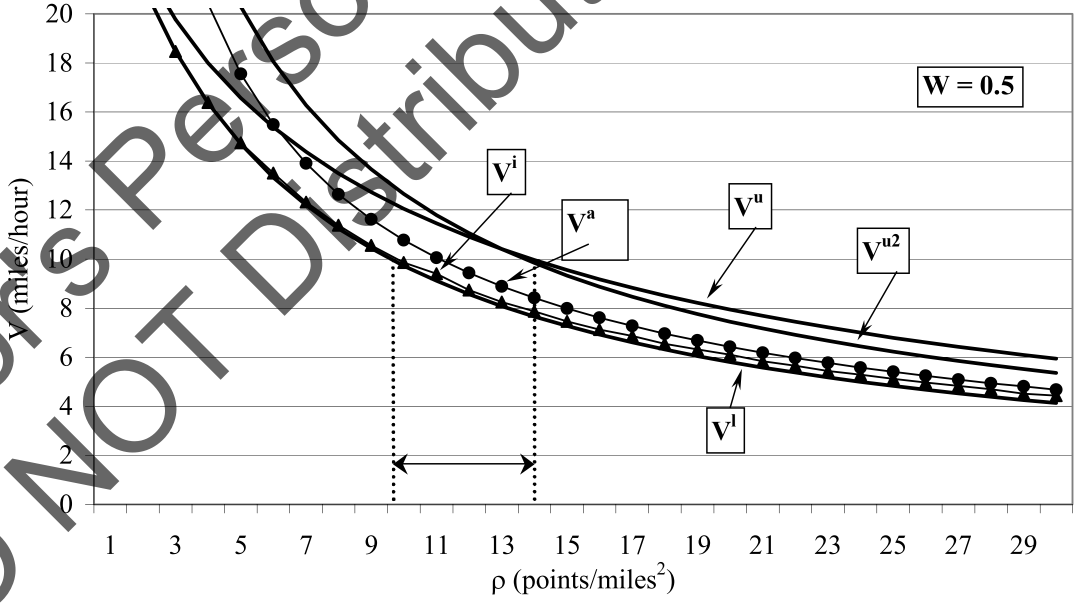

Figure 10 – Longitudinal velocity (V) vs. demand density (ρ); W = 0.5

We note that in both charts V^l and V^u converge for lower values of ρ (as expected since they are both equal to v for $\rho = 0$), while V^{u2} goes to 2v. V^{u2} is a tighter bound than V^u for higher ρ and this is more evident for the case with W = 1. The gap between V^l and V^u/V^{u2} does not diverge significantly with increasing ρ maintaining a reasonably narrow range. The approximate value V^a falls in the middle of this range, except for smaller ρ, because Equation (4.31) is no longer a good estimate for low demand density. The insertion heuristic curve V^i lies a little above V^l; the gap between them slightly increases with ρ, showing that the improvement provided by the insertion heuristic algorithm over the no-backtracking policy is more evident for denser demand. This gap is smaller for W = 0.5, because the narrower corridor guarantees better solutions from the no-backtracking policy (in accordance with Proposition 1).

Even though MAST services are designed to provide a comfortable door-to-door service, customers would probably perceive the service as being too slow if the velocity along the primary direction would fall below a threshold value. According to a random check of the timetables of various fixed-route bus lines in the Los Angeles County, regular fixed-route buses generally achieve an overall average velocity along their routes of about 15 miles per hour, depending on the number of stops placed in the route and the number of customers to be served (they can go as quick as 20 miles/hour for interurban fast lines and they can go as slow as 10 miles/hour for downtown services). We assume that MAST customers would be willing to sacrifice some of this velocity for the convenience of being picked up and dropped off at their desired locations. This might be true for NPND, NPD and PND types of customers (as defined in Chapter 3), but MAST systems are designed serve also the "regular" PD customers that rely only on already scheduled checkpoints for both their service points and that clearly would not welcome any reduction on the velocity of the service, since they do not require any deviations from the route.

As an example, we assume the minimum acceptable V of a MAST system to be 10 miles/hour, 33% less than that of average fixed-route buses; we suppose that below this level the demand would radically drop, because it is too inconvenient. From the charts we note that the demand density that can be served corresponding to this value is in the range of $\rho = 4\text{-}6$ customers/mile^2 (when $W = 1$ mile) and $\rho = 10\text{-}14$ customers/mile^2 (when $W = 0.5$ miles), according to the values provided by the bounds. Recall that ρ represents the density of the stops served only by the left-right vehicle, which corresponds also to the density of all customers served by both vehicles of the systems (the right-left vehicle and the left-right one). Therefore, the system would be able to serve at least 4 customers every mile of the corridor (when $W = 1$ mile) or at least $10\times0.5 = 5$ customers every mile of the corridor (when $W = 0.5$ miles), not considering "regular" PD customers being picked up and dropped off at the checkpoints. The existing MAST line 646 currently serves a very low nighttime demand of about $\rho = 1$ customers/miles^2; the width of the service area is $W = 1$ mile that allows the system to properly serve all the customers, maintaining a relatively high longitudinal velocity of almost 20 miles/hour. Heavier demands would require either to lower the longitudinal velocity maintaining the same size of the service area or to narrow the width of the strip keeping the same longitudinal velocity or to add more vehicles thereby reducing the cycle length.

While designing a MAST system a planner can make use of the information provided in Figure 9 and Figure 10 to schedule the time difference between checkpoints in order to set the velocity of the service and therefore establish the maximum slack time allowed for deviations. In areas where most customers are of the PD type and would use the service as a regular fixed-route line between the checkpoints and only a small portion of them would take advantage of the door-to-door characteristic, the demand would probably significantly increase for higher values of V and it might be more convenient (also for capacity constraints) to design the MAST system assigning a larger amount of "faster" vehicles to the line, instead of fewer slower buses.

5 Static operating scenario: schedule optimization

In this chapter we look at the problem of scheduling MAST services in static scenarios, where all the demand is supposed to be known in advance. We provide a mathematical formulation of the problem as a mixed integer linear program and we aim to optimize it by finding the best schedule. Since the problem is NP-Hard, we develop a set of valid inequalities to be added to the formulation in order to increase the lower optimality bound and speed up the search for optimality.

5.1 Formulation

The system considered consists of a single vehicle, initially associated with a predefined schedule along a fixed-route consisting of C checkpoints identified by $c = 1, 2, \ldots, C$; two of them are terminals located at the extremities of the route ($c = 1$ and $c = C$) and the remaining C-2 intermediate checkpoints are distributed along the route. The vehicle is moving back and forth between 1 and C. A ride *r* is defined as a portion of the schedule beginning at one of the terminals and ending at the other one after visiting all the intermediate checkpoints; the vehicle's schedule consists of R rides. Since the end-terminal of a ride *r* corresponds to the start-terminal of the following ride *r*+1, the total number of stops at the checkpoints is $TC = (C-1) \times R+1$. Hence, the initial schedule's array is represented by an ordered sequence of stops $s = 1, \ldots, TC$ and their scheduled departure times are assumed to be constraints on the system which can not be violated.

The service area is represented by a rectangular region defined by L×W, where L (on the x axis) is the distance between terminals 1 and C and W/2 (on the y axis) is the maximum allowable deviation from the main route in either side (see Figure 11).

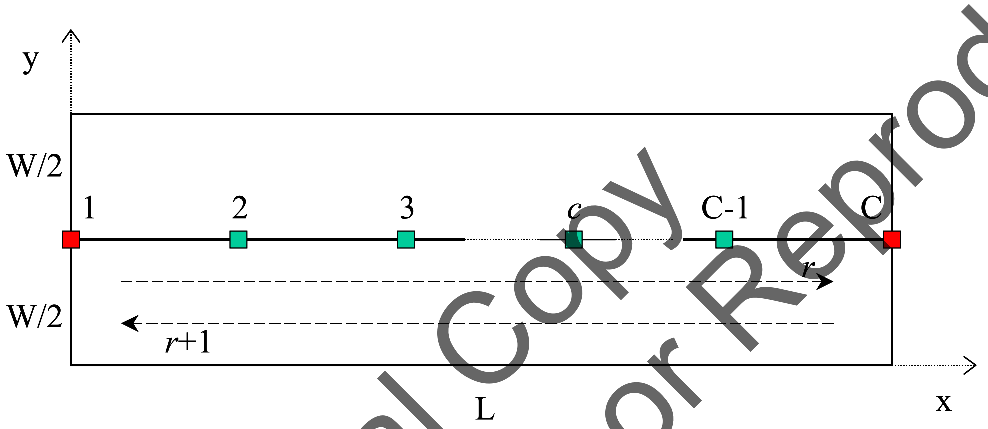

Figure 11 – MAST system

Each checkpoint *c* is scheduled to be visited by the vehicle R times. The stop indexes $s_k(c)$ identifying them in the schedule (stop index *s* of the k^{th} occurrence of checkpoint *c* in the schedule) are computed by the following sequence:

$$s_r(c)=\left\{1+(\mathrm{C}-1)(r-1)+\frac{(\mathrm{C}-1)+(-1)^r[(\mathrm{C}-1)-2(c-1)]}{2}\right\} \quad r=1,\ldots,\mathrm{R} \qquad (5.1)$$

Note that for terminal checkpoints *c* = 1 and *c* = C the ending checkpoint of a ride *r* coincides with the starting checkpoint of the following ride *r*+1. For example:

with C = 5, R = 4 and r = 1, 2, 3, 4, checkpoint c = 2 will have $s_r(2)$ = 2, 8, 10, 16; while terminal checkpoint c = 5 will have $s_r(5)$ = 5, 5, 13, 13.

While checkpoints are identified by s = 1,…, TC, non-checkpoint customer requests (NP or ND) are identified by s = TC+1,…, TS where TS represents the current total number of stops. The problem is to minimize a properly defined objective function by finding the optimal sequence of stops identified by the integer index $\alpha(s)$, s = 1,…, TS, representing the position of any stop s in the vehicle's schedule, with $\alpha(1) = 1$ and $\alpha(\text{TC}) = \text{TS}$ being respectively the first and last checkpoints of the service.

To simplify the problem we assume no capacity constraint and a deterministic environment, with one customer per request.

We define the following notation for the system:

- v = the vehicle speed
- R = number of rides
- RD = {1,…,R} = set of rides
- C = number of checkpoints
- TC = (C-1)×R+1 = total number of stops at the checkpoints in the schedule
- N_0 = {1,…, TC} = set of stops at the checkpoints
- tdc_i, $\forall i \in N_0$ = departure times from checkpoints [$tdc_1 = 0$]
- pd = number of PD requests

- pnd = number of PND requests
- npd = number of NPD requests
- npnd = number of NPND requests
- PD = {1,…, pd} = set of PD requests
- PND = {pd+1,…, pd+pnd} = set of PND requests
- NPD = {pd+pnd+1,…, pd+pnd+npd} = set of NPD requests
- NPND = {pd+pnd+npd+1,…, pd+pnd+npd+npnd} = set of NPND requests
- HYB = PND $\cup$ NPD = set of hybrid requests (PND and NPD types)
- K = PD $\cup$ HYB $\cup$ NPND = set of all requests
- τ_k, $\forall k \in$K = ready times of requests
- TS = TC+pnd+npd+2×npnd = total number of stops
- N_n = {TC+1,…, TS} = set of non-checkpoint stops
- cs(i)$\in$K, $\forall i \in N_n$ = corresponding request of each non-checkpoint stop
- N = $N_0 \cup N_n$ = set of all stops
- $d_{i,j}$, $\forall i,j \in$N = rectilinear distance between i and j
- b_i, $\forall i \in$N/{1} = service time for boardings and disembarkments at stop i

PD customers are guaranteed to be served at their chosen service checkpoints identified by their index $s \in N_0$, since we assume no capacity constraint on the vehicle. NPND customers have their own stops placed somewhere in the schedule and identified by their index $s \in \mathrm{N}_n$. Hybrid customers (PND and NPD) instead do not have a priori a uniquely identified stop for their checkpoint service point. In fact, they will be served at one of the occurrences of their chosen checkpoint (either a P or a D) identified by Equation (5.1), depending on where their non-checkpoint stop (either a ND or a NP) is positioned in the schedule. For example, consider a MAST system with C = 5 and R = 4 and assume that a NPD request would like to be picked up at its NP stop (s^*) as soon as possible and dropped off at the checkpoint $c = 4$ in the first ride $r = 1$ identified by $s_1(4) = 4$, from Equation (5.1). It could occur that, because of lack of slack time due to other requests, the NP stop s^* can not be placed in the schedule before $s_1(4) = 4$, so that $\alpha(s^*) > \alpha(4)$. As a result the customer will have to be dropped off at one of the successive occurrence of $c = 4$ in the schedule ($s_r(4) = 6, 12, 14$, for $r = 2, 3, 4$). A similar example could be developed for PND customers.

Therefore, we can further identify the following:

- pu(k)$\in$N, $\forall k \in \mathrm{K} \cap \mathrm{PND}$ = pick-up of all requests except PND
- do(k)$\in$N, $\forall k \in \mathrm{K} \cap \mathrm{NPD}$ = drop-off of all requests except NPD
- PU(k,r)$\in \mathrm{N}_0$, $\forall k \in \mathrm{PND}$, $\forall r \in \mathrm{RD}$ = set of possible pick-up checkpoints for each PND request, obtainable from the sequence $s_k(c)$ in Equation (5.1), where $c \leq \mathrm{C}$ represents their pick-up checkpoint.

- DO(k,r)$\in$N$_0$, $\forall k\in$NPD, $\forall r\in$RD = set of possible drop-off checkpoints for each NPD customer, obtainable from the sequence $s_k(c)$ in Equation (5.1), where $c \leq$ C represents their drop-off checkpoint.

We also note that not all the occurrences of a checkpoint c are feasible candidates to be selected as P or D checkpoint of a hybrid request, depending on their ready times τ_k. Therefore, we define the following sets:

- HYBR(k) $\subset$ RD, $\forall k\in$HYB = set of feasible rides (depending on τ_k) for each hybrid request to be picked-up or dropped-off at their checkpoint. Formally:

  $\forall k\in$PND, HYBR(k) includes all $r\in$RD s.t.: $\tau_k \leq$ tdc$_i$, $\forall i\in$PU(k,r)

  $\forall k\in$NPD, HYBR(k) includes all $r\in$RD s.t.:

  $$\max(\tau_k,\mathrm{tdc}_i) + \mathrm{d}_{i,j}/\mathrm{v} + \mathrm{b}_j \leq \mathrm{tdc}_j,\ \forall j\in \mathrm{DO}(k,r), \text{ with } i = \mathrm{pu}(k)$$

The problem is represented by a network and the sets of arcs are defined as follows:

- A_0 = arcs in N_0, including only arcs $(i,i+1)$, with $i = 1,\ldots,$ TC-1, because the checkpoints are already ordered sequentially in the schedule.
- A_n = arcs in N_n, including all arcs (i,j), $\forall i,j \in \mathrm{N}_\mathrm{n}$, with $i \neq j$.
- $\mathrm{A}_{0,\mathrm{n}}$ = arcs from N_0 to N_n, including all arcs (i,j), $\forall i \in \mathrm{N}_0/\{\mathrm{TC}\}$, $\forall j \in \mathrm{N}_\mathrm{n}$.
- $\mathrm{A}_{\mathrm{n},0}$ = arcs from N_n to N_0, including all arcs (i,j), $\forall i \in \mathrm{N}_\mathrm{n}$, $\forall j \in \mathrm{N}_0/\{1\}$.
- $\mathrm{A} = \mathrm{A}_0 \cup \mathrm{A}_\mathrm{n} \cup \mathrm{A}_{0,\mathrm{n}} \cup \mathrm{A}_{\mathrm{n},0}$ = set of all arcs

The variables of the system are the following:

- $x_{i,j} = \{0,1\}$, $\forall (i,j) \in \mathrm{A}$ = binary variables indicating if an arc (i,j) is used ($x_{i,j} = 1$) or not ($x_{i,j} = 0$).
- ta_i, $\forall i \in \mathrm{N}/\{1\}$ = arrival time at stop i.
- td_i, $\forall i \in \mathrm{N}$ = departure time from stop i.
- $z_{k,r} = \{0,1\}$, $\forall k \in \mathrm{HYB}$, $\forall r \in \mathrm{HYBR}(k)$ = binary variable indicating whether customer k is picked-up ($k \in \mathrm{PND}$) or dropped-off ($k \in \mathrm{NPD}$) in ride r (in this case $z_{k,r} = 1$, otherwise $z_{k,r} = 0$).
- TD_k, $\forall k \in \mathrm{PND}$ = departure time of customer k.
- TA_k, $\forall k \in \mathrm{NPD}$ = arrival time of customer k.
- it_i, $\forall i \in \mathrm{N}/\{1\}$ = idle time spent at node i.

We note that once a feasible solution of the problem is found the indexes $\alpha(i)$, $\forall i \in \mathrm{N}$ are trivially determined either by the x, the td or the ta variables.

Finally, we propose the following mixed integer linear programming formulation for the MAST system, where ω_1, ω_2 and ω_3 are weights:

$$\begin{aligned}
&\min \omega_1 \left(\sum_{(i,j)\in \mathrm{A}} \mathrm{d}_{i,j} x_{i,j} \right) \Big/ \mathrm{v} + \\
&\quad \omega_2 \left(\sum_{k\in \mathrm{K}\cap \mathrm{HYB}} \left(ta_{\mathrm{do}(k)} - td_{\mathrm{pu}(k)} \right) + \sum_{k\in \mathrm{PND}} \left(ta_{\mathrm{do}(k)} - TD_k \right) + \sum_{k\in \mathrm{NPD}} \left(TA_k - td_{\mathrm{pu}(k)} \right) \right) + \\
&\quad \omega_3 \left(\sum_{k\in \mathrm{K}\cap \mathrm{PND}} \left(td_{\mathrm{pu}(k)} - \tau_k \right) + \sum_{k\in \mathrm{PND}} \left(TD_k - \tau_k \right) \right)
\end{aligned} \tag{5.2}$$

subject to:

$$\sum_i x_{i,j} = 1 \qquad \forall j \in \mathrm{N}/\{1\} \tag{5.3}$$

$$\sum_j x_{i,j} = 1 \qquad \forall i \in \mathrm{N}/\{\mathrm{TC}\} \tag{5.4}$$

$$td_i = \mathrm{tdc}_i \qquad \forall i \in \mathrm{N}_0 \tag{5.5}$$

$$\sum_{r\in \mathrm{HYBR}(k)} z_{k,r} = 1 \qquad \forall k \in \mathrm{HYB} \tag{5.6}$$

$$TD_k \geq td_{\mathrm{PU}(k,r)} - \mathrm{M}(1\text{-}z_{k,r}) \qquad \forall k \in \mathrm{PND},\ \forall r \in \mathrm{HYBR}(k) \tag{5.7}$$

$$TD_k \leq td_{\mathrm{PU}(k,r)} + \mathrm{M}(1\text{-}z_{k,r}) \qquad \forall k \in \mathrm{PND},\ \forall r \in \mathrm{HYBR}(k) \tag{5.8}$$

$$TA_k \geq ta_{\mathrm{DO}(k,r)} - \mathrm{M}(1\text{-}z_{k,r}) \qquad \forall k \in \mathrm{NPD},\ \forall r \in \mathrm{HYBR}(k) \tag{5.9}$$

$$TA_k \leq ta_{\mathrm{DO}(k,r)} + \mathrm{M}(1\text{-}z_{k,r}) \qquad \forall k \in \mathrm{NPD},\ \forall r \in \mathrm{HYBR}(k) \tag{5.10}$$

$$td_{\mathrm{pu}(k)} \geq \tau_k \qquad \forall k \in \mathrm{K} \cap \mathrm{PND} \tag{5.11}$$

$$TD_k \geq \tau_k \qquad \forall k \in \mathrm{PND} \qquad (5.12)$$

$$ta_{\mathrm{do}(k)} > td_{\mathrm{pu}(k)} \qquad \forall k \in \mathrm{NPND} \qquad (5.13)$$

$$ta_{\mathrm{do}(k)} > TD_k \qquad \forall k \in \mathrm{PND} \qquad (5.14)$$

$$TA_k \geq td_{\mathrm{pu}(k)} \qquad \forall k \in \mathrm{NPD} \qquad (5.15)$$

$$ta_j \geq td_i + x_{i,j}\mathrm{d}_{i,j}/\mathrm{v} - \mathrm{M}(1 - x_{i,j}) \qquad \forall (i,j) \in \mathrm{A} \qquad (5.16)$$

$$td_i = ta_i + \mathrm{b}_i + it_i \qquad \forall i \in \mathrm{N}/\{1\} \qquad (5.17)$$

$$it_i \geq 0 \qquad \forall i \in \mathrm{N}/\{1\} \qquad (5.18)$$

$$\sum_{(i,j)\in \mathrm{A}} x_{i,j}\left(\mathrm{d}_{i,j}/\mathrm{v} + \mathrm{b}_i\right) + \sum_{i \in \mathrm{N}/\{1\}} it_i = \mathrm{tdc}_{\mathrm{TC}} - \mathrm{tdc}_1 \qquad (5.19)$$

The objective function (5.2) minimizes the weighted sum of three different factors, namely the total miles driven by the vehicle, the total ride time of all customers and the total waiting time of all customers, defined as the time interval between the ready time and pick-up time. This definition allows optimizing in terms of both the vehicle variable cost (first term) and the service level (the last two terms); modifying the weights accordingly we can emphasize one factor over the others as needed.

Network constraints (5.3) and (5.4) allow each stop (except node 1 and TC) to have exactly one incoming arc and one outgoing arc equal to 1, so that all stops will be visited once.

Constraint (5.5) forces the departure times from the checkpoint to be fixed, since they are prescheduled like in a fixed-route line.

Constraints (5.6) allow exactly one *z* variable to be equal to 1 for each hybrid customer, meaning that a unique ride will be chosen for their pick-up or drop-off checkpoint.

Constraints (5.7) and (5.8) fix the value of the *TD* variables for each customer $k \in$PND depending on the chosen *z* variable. Constraints (5.9) and (5.10) do the same for NPD requests. M represents a number large enough to cause the constraints to become irrelevant when $z_{k,r} = 0$. An M = tdc_{TC}-tdc_1 is big enough to serve this purpose.

Constraints (5.11) and (5.12) prevent the departure times of each customer from being earlier than its ready time.

Equations (5.13), (5.14) and (5.15) are the precedence constraints for each request. Pick-up must be scheduled before the corresponding drop-off.

Constraint (5.16) is the key constraint in the formulation. It defines that for each $x_{i,j} = 1$ the arrival time at *j* should be no less than the departure time from *i* plus the time needed to travel between *i* and *j*. The last term with the M (also in this case an M = tdc_{TC}-tdc_1 is large enough to be effective) assures that for any $x_{i,j} = 0$ the constraints become irrelevant. This constraint also guarantees that every feasible solution does not contain inner loops, but a single path from node 1 to node TC.

Constraint (5.17) links together arrival time, departure time and idle time for each stop *i* in the network. Constraint (5.18) ensures no negative idle times.

Constraint (5.19) is a balance equation and prevents the system from finding unrealistic solutions having idle time in between stops and not only at stops.

The following constraint can also be added to the formulation, in case we would like to allow idling only at the checkpoints:

$$it_i = 0 \qquad \forall i \in \mathrm{N_n} \qquad (5.20)$$

The problem is a special case of the Pick-up and Delivery Problem (PDP) that is known to be NP-Hard and unsolvable in reasonable time for large enough instances.

5.2 Elimination of infeasible arcs

In order to reduce the size of the problem we can exclude from the network several infeasible arcs and therefore many of the x variables from the formulation. Specifically we can redefine the sets of arcs $\mathrm{A_n}$, $\mathrm{A_{0,n}}$, $\mathrm{A_{n,0}}$ as follows:

- $\mathrm{A_n}$ = arcs in $\mathrm{N_n}$, including all arcs (i,j), $\forall i,j \in \mathrm{N_n}$, with $i \neq j$ and excluding the following infeasible arcs:
  a. arcs (i,j) s.t. $i = \mathrm{do}(k)$ and $j = \mathrm{pu}(k)$, $\forall k \in \mathrm{NPND}$
  b. arcs (i,j) s.t.:
     $(\mathrm{d}_{k,i}+\mathrm{d}_{i,j}+\mathrm{d}_{j,k+1})/\mathrm{v} + (\mathrm{b}_i+\mathrm{b}_j+\mathrm{b}_{k+1}) > \mathrm{tdc}_{k+1}-\mathrm{tdc}_k$, $\forall k \in \mathrm{N_0}/\{\mathrm{TC}\}$.
     These arcs are infeasible because the vehicle does not have sufficient time to go from checkpoint k to i to j to checkpoint

k+1, for any pair of consecutive checkpoints *k* and *k*+1. This is not acceptable, since the vehicle must pass by all checkpoints $k \in N_0$ and has to meet the departure deadline tdc_k.

c. arcs (i,j) with $k \in NPD = cs(j)$ or $cs(i)$ and $h = DO(k,R)$ be the last possible drop-off checkpoint in the schedule for *k*. Arcs (i,j) are infeasible if the vehicle would not be able to arrive at *h* in time to meet the departure time tdc_h, because of too high ready times τ_k. Formally arc (i,j) is infeasible if one of the following five conditions is verified:

$cs(i) = k \in NPD$, $h = DO(k,R)$ and

$$\tau_k + (d_{i,j}+d_{j,h})/v + (b_j+b_h) > tdc_h$$

$cs(i) = k \in NPD$, $h = DO(k,R)$, $j = pu(cs(j))$ and

$$\tau_{cs(j)} + d_{j,h}/v + b_h > tdc_h$$

$cs(i) = k \in NPD$, $h = DO(k,R)$, $j = do(cs(j))$ and

$$\tau_{cs(j)} + (d_{pu(cs(j)),i}+d_{i,j}+d_{j,h})/v + (b_i+b_j+b_h) > tdc_h$$

$cs(j) = k \in NPD$, $h = DO(k,R)$, $i = pu(cs(i))$ and

$$\tau_{cs(i)} + (d_{i,j}+d_{j,h})/v + (b_j+b_h) > tdc_h$$

$cs(j) = k \in NPD$, $h = DO(k,R)$, $i = do(cs(i))$ and

$$\tau_{cs(i)} + (d_{pu(cs(i)),i}+d_{i,j}+d_{j,h})/v + (b_i+b_j+b_h) > tdc_h$$

- $A_{0,n}$ = arcs from N_0 to N_n, including all arcs (i,j), $\forall i \in N_0/\{TC\}$, $\forall j \in N_n$ and excluding the following infeasible arcs:

  a. arcs (i,j) s.t. $(d_{i,j}+d_{j,i+1})/v + (b_j+b_{i+1}) > tdc_{i+1}-tdc_i$, $\forall i \in N_0/\{TC\}$.

     These arcs are infeasible because the vehicle does not have sufficient time to go from checkpoint i to j to checkpoint i+1, for any pair of consecutive checkpoints i and i+1. This is not acceptable, since the vehicle must pass by all checkpoints $i \in N_0$ and has to meet the departure deadline tdc_i.

  b. arcs (i,j) s.t. $cs(j) = k \notin PND$, $i < TC$, $\tau_k + d_{j,i+1}/v + b_{i+1} > tdc_{i+1}$.

     These arcs are infeasible because they would not allow the vehicle to reach checkpoint i+1 on time for its departure time tdc_{i+1}.

  c. arcs (i,j) s.t. $cs(j) = k \in PND$, $\tau_k > tdc_i$ or $PU(k,1) > i$.

     These arcs are infeasible because the earliest possible pick-up checkpoint for k is later in the schedule compared to i. The vehicle leaving i must pass by the checkpoint pick-up of k before going to j.

  d. arcs (i,j) s.t. $cs(j) = k \in NPD$, $i \geq DO(k,R)$.

     These arcs are infeasible because i must be earlier in the schedule compared to $DO(k,R)$, the last possible drop-off checkpoint for k, in order to allow the vehicle to go from i to j.

e. arcs (i,j) s.t. $i = 1$, $j = \text{do}(k)$ and $k \in \text{NPND}$.

   These arcs are infeasible because the vehicle needs to pass by pu(k) first.

f. arcs (i,j) s.t. $j = \text{do}(k)$, $k \in \text{NPND}$, $\tau_k + d_{\text{pu}(k),i}/\text{v} + b_i > \text{tdc}_i$.

   These arcs are infeasible because τ_k does not allow the vehicle to go from pu(k) to i on time for its departure time tdc_i (pu(k) must be scheduled before $j = \text{do}(k)$).

- $A_{n,0}$ = arcs from N_n to N_0, including all arcs (i,j), $\forall i \in N_n$, $\forall j \in N_0/\{1\}$ and excluding the following infeasible arcs:

  a. arcs (i,j) s.t. $(d_{j\text{-}1,i}+d_{i,j})/\text{v} + (b_i+b_j) > \text{tdc}_j – \text{tdc}_{j\text{-}1}$, $\forall j \in N_0/\{1\}$.

     These arcs are infeasible because the vehicle would not be able time wise to go from checkpoint j-1 to i to checkpoint j, for any pair of consecutive checkpoints j-1 and j. This is not acceptable, since the vehicle must pass by all checkpoints $j \in N_0$ and has to meet the departure deadline tdc_j.

  b. arcs (i,j) s.t. $\text{cs}(i) = k \notin \text{PND}$, $\tau_k + d_{i,j}/\text{v} + b_j > \text{tdc}_j$.

     These arcs are infeasible because they would not allow the vehicle to reach checkpoint j on time for its departure time tdc_j.

c. arcs (i,j) s.t. $i = \mathrm{do}(k)$, $k \in \mathrm{NPND}$, $\tau_k + (\mathrm{d}_{\mathrm{pu}(k),i}+\mathrm{d}_{i,j})/\mathrm{v} + (\mathrm{b}_i+\mathrm{b}_j) > \mathrm{tdc}_j$.

   These arcs are infeasible because τ_k does not allow the vehicle to go from pu(k) to i to j on time for its departure time (pu(k) must be before i = do(k)).

d. arcs (i,j) s.t. $\mathrm{cs}(i) = k \in \mathrm{PND}$, $\tau_k > \mathrm{tdc}_{j\text{-}1}$ or $\mathrm{PU}(k,1) \geq j$.

   These arcs are infeasible because the earliest possible pick-up checkpoint for k is later (or equal) in the schedule compared to j.

e. arcs (i,j) s.t. $\mathrm{cs}(i) = k \in \mathrm{NPD}$, $j > \mathrm{DO}(k,\mathrm{R})$.

   These arcs are infeasible because j is later in the schedule compared to the last possible drop-off checkpoint for k.

f. arcs (i,j) s.t. $j = \mathrm{TC}$, $i = \mathrm{pu}(k)$, $k \in \mathrm{NPND}$.

   These arcs are infeasible because the vehicle needs to pass by do(k) first.

- $\mathrm{A} = \mathrm{A}_0 \cup \mathrm{A}_\mathrm{n} \cup \mathrm{A}_{0,\mathrm{n}} \cup \mathrm{A}_{\mathrm{n},0}$ = set of all arcs

5.3 Valid inequalities

The purpose of this section is to identify valid inequalities linking together some of the variables of the MAST system formulation in order to reduce the feasible region identified by constraints from (5.3) to (5.20) and possibly speed up the search for the optimal solution of the problem. The challenge is to make sure that these new constraints will only remove feasible but not optimal solutions from the problem.

In order to develop some of the inequalities we assume $\omega_2 > \omega_3$, giving a higher weight to the travel time term vs. the waiting time term in the objective function (5.2). This means that we assume that customers would rather wait for their pick-up instead of spending time onboard the vehicle. This is generally not true if customers do not know the schedule and face random arrivals of buses at their pick-up locations; in fact, they would probably rather spend the time onboard instead of waiting at their pick-up stop, especially when facing bad weather conditions and/or unsafe areas. However, a MAST system provides a door-to-door transportation service built on reservations with prescheduled departure times from checkpoints. Therefore, customers know in advance the expected departure times from their pick-up locations (either a P or a NP) and in this case this assumption is rather reasonable. In fact most customers, given that the arrival time at destination is fixed, would reasonably prefer to have their scheduled pick-up times as late as possible to make the ride shorter and consequently the wait longer. This is particularly true for NPD and NPND customers that would spend their waiting time at their NP stop (home or office or other convenient locations) and not at an outdoor bus stop.

The underlying concept behind all the inequalities developed in this section is that hybrid customers will be choosing their P or D checkpoints as close as possible to their ND or NP stop, once these are placed in the schedule. In fact NPD customers will disembark the vehicle as early as possible after being picked up to minimize their ride time. A PND request will instead board the vehicle as late as possible to minimize their ride time and consequently maximize their waiting time, since we assume $\omega_2 > \omega_3$. More formally we can develop and prove the following propositions.

Proposition 2. *If $\omega_2 > \omega_3$, a necessary condition for optimality is that PND customers must board the vehicle at the last occurrence of their P checkpoint prior to their scheduled ND drop-off stop.*

Proof. Suppose that in the optimal solution we have $\alpha(\mathrm{PU}(k,r^*)) < \alpha(\mathrm{do}(k)) < \alpha(\mathrm{PU}(k,r^*+1))$ and that $\tau_\mathrm{k} \le td_{\mathrm{PU}(k,r^\circ)}$, with $r^\circ \le r^*$, for a request $k \in$PND. The objective function value can be written as $Z = \Delta + \omega_2(ta_{do(k)}\text{-}TD_k) + \omega_3(TD_k\text{-}\tau_k)$, where Δ includes all the terms in Z except the ride time and the waiting time terms of k. TD_k could be equal to $td_{\mathrm{PU}(k,r^\circ)}$, $td_{\mathrm{PU}(k,r^\circ+1)}$,…, $td_{\mathrm{PU}(k,r^*)}$, depending on $z_{k,r}$ indicating at which occurrence of the pick-up checkpoint the customer boards the vehicle. Rearranging the terms we have $Z = \Delta + \omega_2 ta_{do(k)} - \omega_3\tau_k + TD_k(\omega_3\text{-}\omega_2)$. Since $td_{\mathrm{PU}(k,r^\circ)} \le td_{\mathrm{PU}(k,r^\circ+1)} \le \ldots \le td_{\mathrm{PU}(k,r^*)}$, $\omega_3\text{-}\omega_2 < 0$ by assumption and Z is optimal, TD_k must be the largest possible, thus equal to $td_{\mathrm{PU}(k,r^*)}$. This means that PND customers get onboard at the last occurrence of their pick-up checkpoint preceding their do(k), minimizing their ride time. □

Proposition 3. *A necessary condition for optimality is that NPD customers must disembark the vehicle at the first occurrence of their D checkpoint following their scheduled NP pick-up stop.*

Proof. Suppose that in the optimal solution we have $\alpha(\mathrm{DO}(k,r^*\text{-}1)) < \alpha(\mathrm{pu}(k)) < \alpha(\mathrm{DO}(k,r^*))$ for a request $k \in \mathrm{NPD}$. The objective function value can be written as $Z = \Delta + \omega_2(TA_k\text{-}td_{\mathrm{pu}(k)})$, where Δ includes all the terms in Z except the ride time term of k. TA_k could be equal to $ta_{\mathrm{DO}(k,r^*)}$, $ta_{\mathrm{DO}(k,r^*+1)}$,…, $ta_{\mathrm{DO}(k,\mathrm{R})}$, depending on $z_{k,r}$ indicating at which occurrence of the drop-off checkpoint the customer disembarks the vehicle. Since $ta_{\mathrm{DO}(k,r^*)} \leq ta_{\mathrm{DO}(k,r^*+1)} \leq \ldots \leq ta_{\mathrm{DO}(k,\mathrm{R})}$ and Z is optimal, TA_k must be the smallest possible, thus equal to $ta_{\mathrm{DO}(k,r^*)}$. This means that NPD customers disembark at the first occurrence of their drop-off checkpoint following their pu(k). □

We are now able to develop three different groups of valid inequalities described in the following sections.

5.3.1 Group #1

The first group of valid inequalities includes constraints linking the *z* variables to the *td* variables (departure times) of non-checkpoint stops of hybrid customers and constraints linking the *z* variables to some of the *x* variables.

For a PND request a valid set of inequalities is represented by

$$td_{\mathrm{do}(k)} < z_{k,r}\mathrm{tdc}_j + \mathrm{M}(1-z_{k,r}), \qquad (5.21)$$

with $j = \mathrm{PU}(k,r+1)$, $\forall k \in \mathrm{PND}$, $\forall r \in \mathrm{HYBR}(k)/\{\mathrm{R}\}$

Because of Proposition 2 these inequalities force the ND stop of each PND request to be scheduled before the next occurrence in the schedule of the checkpoint chosen as pick-up. If $z_{k,r} = 1$ the PND customer is picked up at his/her checkpoint PU(*k,r*) in ride *r* and the constraint imposes that the do(*k*) has to be scheduled before PU(*k,r*+1) by setting an upper bound on the departure time $td_{\mathrm{do}(k)}$. If $z_{k,r} = 0$ the constraint becomes irrelevant because of the M.

Symmetrically for NPD requests a valid inequality is represented by

$$td_{\mathrm{pu}(k)} > z_{k,r}\mathrm{tdc}_i - \mathrm{M}(1-z_{k,r}), \qquad (5.22)$$

with $i = \mathrm{DO}(k,r-1)$, $\forall k \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)/\{1\}$

Because of Proposition 3, these inequalities force the NP stop of each NPD request to be scheduled after the previous occurrence in the schedule of the checkpoint chosen as drop-off. If $z_{k,r} = 1$ the NPD customer is dropped off at his/her checkpoint DO(k,r) in ride r and the constraint imposes that the pu(k) has to be scheduled after DO(k,r-1) by setting a lower bound on the departure time $td_{\mathrm{pu}(k)}$. If $z_{k,r} = 0$ the constraint becomes irrelevant because of the M.

We can also include the following inequalities for PND requests:

$$x_{\mathrm{do}(k),j} \leq z_{k,r}, \qquad (5.23)$$

$\forall k \in \mathrm{PND}, \forall r \in \mathrm{HYBR}(k)/\{\mathrm{R}\}, \forall(\mathrm{do}(k),j) \in \mathrm{A}_{\mathrm{n},0}$ s.t. $\mathrm{PU}(k,r) < j \leq \mathrm{PU}(k,r+1)$

By Proposition 2, do(k) must be scheduled in between PU(k,r) and PU(k,r+1) and all arcs originating from do(k) and ending at a checkpoint j can not exist whenever j is not included in that interval. These arcs would in fact unfeasibly require the vehicle to go from do(k) to a checkpoint scheduled before its pick-up PU(k,r) or to skip PU(k,r+1) going directly from do(k) to a checkpoint scheduled after PU(k,r+1).

Similarly we have:

$$x_{i,\mathrm{do}(k)} \leq z_{k,r}, \qquad (5.24)$$

$\forall k \in \mathrm{PND}, \forall r \in \mathrm{HYBR}(k)/\{\mathrm{R}\}, \forall(i,\mathrm{do}(k)) \in \mathrm{A}_{0,\mathrm{n}}$ s.t. $\mathrm{PU}(k,r) \leq i < \mathrm{PU}(k,r+1)$

All arcs originating from a checkpoint i and ending at do(k) are eliminated whenever i is not included in the interval [PU(k,r), PU(k,r+1)) identified by $z_{k,r} = 1$.

Symmetrically for NPD requests we have that

$$x_{i,\mathrm{pu}(k)} \leq z_{k,r}, \tag{5.25}$$

$\forall k \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)/\{1\}$, $\forall (i,\mathrm{pu}(k)) \in \mathrm{A}_{0,\mathrm{n}}$ s.t. $\mathrm{DO}(k,r\text{-}1) \leq i < \mathrm{DO}(k,r)$

$$x_{\mathrm{pu}(k),j} \leq z_{k,r}, \tag{5.26}$$

$\forall k \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)/\{1\}$, $\forall (\mathrm{pu}(k),j) \in \mathrm{A}_{\mathrm{n},0}$ s.t. $\mathrm{DO}(k,r\text{-}1) < j \leq \mathrm{DO}(k,r)$

5.3.2 Group #2

A second group of valid inequalities links z and x variables making use of the ready times τ of the customers involved. For PND customers we have that

$$\tau_{\mathrm{cs}(i)} + \mathrm{d}_{i,j}/\mathrm{v} + \mathrm{b}_j \leq z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2\text{-}z_{k,r}\text{-}x_{\mathrm{do}(k),i}), \tag{5.27}$$

$j = \mathrm{PU}(k,r+1)$, $\forall k \in \mathrm{PND}$, $\forall r \in \mathrm{HYBR}(k)/\{\mathrm{R}\}$, $\forall (\mathrm{do}(k),i) \in \mathrm{A}_\mathrm{n}$, s.t. $i = \mathrm{pu}(\mathrm{cs}(i))$

Since by Proposition 2 do(k) must be scheduled in between PU(k,r) and PU(k,r+1), these constraints impose that any arc originating from the do(k) of a PND customer to any non-checkpoint pick-up i is not allowed if the vehicle would not be able to reach checkpoint PU(k,r+1) on time by passing through i, because of too high

$\tau_{\mathrm{cs}(i)}$, even using the quickest way possible. The M causes these constraints to become irrelevant if either $z_{k,r}$ or $x_{\mathrm{do}(k),i}$ are not equal to 1.

Similarly,

$$\tau_{\mathrm{cs}(i)} + (\mathrm{d}_{\mathrm{pu}(\mathrm{cs}(i)),\mathrm{do}(k)}+\mathrm{d}_{\mathrm{do}(k),i}+\mathrm{d}_{i,j})/\mathrm{v} + (\mathrm{b}_{\mathrm{do}(k)}+\mathrm{b}_i+\mathrm{b}_j) \le$$
$$z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2-z_{k,r}-x_{\mathrm{do}(k),i}), \qquad (5.28)$$

$j = \mathrm{PU}(k,r+1)$, $\forall k\in\mathrm{PND}$, $\forall r\in\mathrm{HYBR}(k)/\{\mathrm{R}\}$, $\forall(\mathrm{do}(k),i)\in\mathrm{A_n}$, s.t. $i = \mathrm{do}(\mathrm{cs}(i))$

Any arc originating from the do(k) of a PND customer k to any non-checkpoint drop-off i is not allowed if the vehicle is not able to go from the pick-up point pu(cs(i)) to do(k) to i to checkpoint PU(k,r+1) on time, because of too high $\tau_{\mathrm{cs}(i)}$, even using the quickest way possible. The M causes these constraints to become irrelevant if either $z_{k,r}$ or $x_{\mathrm{do}(k),i}$ are not equal to 1.

Similar constraints can be developed for arcs (i,do(k)) as follows:

$$\tau_{\mathrm{cs}(i)} + (\mathrm{d}_{i,\mathrm{do}(k)}+\mathrm{d}_{\mathrm{do}(k),j})/\mathrm{v} + (\mathrm{b}_{\mathrm{do}(k)}+\mathrm{b}_j)\le z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2-z_{k,r}-x_{i,\mathrm{do}(k)}), \qquad (5.29)$$

$j = \mathrm{PU}(k,r+1)$, $\forall k\in\mathrm{PND}$, $\forall r\in\mathrm{HYBR}(k)/\{\mathrm{R}\}$, $\forall(i,\mathrm{do}(k))\in\mathrm{A_n}$, s.t. $i = \mathrm{pu}(\mathrm{cs}(i))$

$$\tau_{\mathrm{cs}(i)} + (\mathrm{d}_{\mathrm{pu}(\mathrm{cs}(i)),i}+\mathrm{d}_{i,\mathrm{do}(k)}+\mathrm{d}_{\mathrm{do}(k),j})/\mathrm{v} +(\mathrm{b}_i+\mathrm{b}_{\mathrm{do}(k)}+\mathrm{b}_j) \le$$
$$z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2-z_{k,r}-x_{i,\mathrm{do}(k)}), \qquad (5.30)$$

$j = \mathrm{PU}(k,r+1)$, $\forall k\in\mathrm{PND}$, $\forall r\in\mathrm{HYBR}(k)/\{\mathrm{R}\}$, $\forall(i,\mathrm{do}(k))\in\mathrm{A_n}$, s.t. $i = \mathrm{do}(\mathrm{cs}(i))$

For NPD customers the four constraints above can be similarly developed:

$$\tau_{\mathrm{cs}(i)} + \mathrm{d}_{i,j}/\mathrm{v} + \mathrm{b}_j \le z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2-z_{k,r}-x_{\mathrm{pu}(k),i}), \qquad (5.31)$$

$j = \mathrm{DO}(k,r)$, $\forall k \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall(\mathrm{pu}(k),i) \in \mathrm{A_n}$, s.t. $i = \mathrm{pu}(\mathrm{cs}(i))$

$$\tau_{\mathrm{cs}(i)} + (\mathrm{d}_{\mathrm{pu}(\mathrm{cs}(i)),\mathrm{pu}(k)} + \mathrm{d}_{\mathrm{pu}(k),i} + \mathrm{d}_{i,j})/\mathrm{v} + (\mathrm{b}_{\mathrm{pu}(k)} + \mathrm{b}_i + \mathrm{b}_j) \le$$
$$z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2-z_{k,r}-x_{\mathrm{pu}(k),i}), \qquad (5.32)$$

$j = \mathrm{DO}(k,r)$, $\forall k \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall(\mathrm{pu}(k),i) \in \mathrm{A_n}$, s.t. $i = \mathrm{do}(\mathrm{cs}(i))$

$$\tau_{\mathrm{cs}(i)} + (\mathrm{d}_{i,\mathrm{pu}(k)} + \mathrm{d}_{\mathrm{pu}(k),j})/\mathrm{v} + (\mathrm{b}_{\mathrm{pu}(k)} + \mathrm{b}_j) \le z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2-z_{k,r}-x_{i,\mathrm{pu}(k)}), \qquad (5.33)$$

$j = \mathrm{DO}(k,r)$, $\forall k \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall(i,\mathrm{pu}(k)) \in \mathrm{A_n}$, s.t. $i = \mathrm{pu}(\mathrm{cs}(i))$

$$\tau_{\mathrm{cs}(i)} + (\mathrm{d}_{\mathrm{pu}(\mathrm{cs}(i)),i} + \mathrm{d}_{i,\mathrm{pu}(k)} + \mathrm{d}_{\mathrm{pu}(k),j})/\mathrm{v} + (\mathrm{b}_i + \mathrm{b}_{\mathrm{pu}(k)} + \mathrm{b}_j) \le$$
$$z_{k,r}\mathrm{tdc}_j + \mathrm{M}(2-z_{k,r}-x_{i,\mathrm{pu}(k)}), \qquad (5.34)$$

$j = \mathrm{DO}(k,r)$, $\forall k \in \mathrm{PND}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall(i,\mathrm{pu}(k)) \in \mathrm{A_n}$, s.t. $i = \mathrm{do}(\mathrm{cs}(i))$

5.3.3 Group #3

A third group of valid inequalities is represented by linking *z* and *x* variables of pairs of hybrid customers. The following relationships can be written:

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3\text{-}z_{h,s}\text{-}z_{k,r}\text{-}x_{\mathrm{do}(k),\mathrm{do}(h)}), \tag{5.35}$$

$i = \mathrm{PU}(h,s)$, $j = \mathrm{PU}(k,r+1)$, $\forall k,h \in \mathrm{PND}$, $\forall r \in \mathrm{HYBR}(k)/\{\mathrm{R}\}$, $\forall s \in \mathrm{HYBR}(h)$

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3\text{-}z_{h,s}\text{-}z_{k,r}\text{-}x_{\mathrm{do}(h),\mathrm{do}(k)}), \tag{5.36}$$

$i = \mathrm{PU}(h,s)$, $j = \mathrm{PU}(k,r+1)$, $\forall k,h \in \mathrm{PND}$, $\forall r \in \mathrm{HYBR}(k)/\{\mathrm{R}\}$, $\forall s \in \mathrm{HYBR}(h)$

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3\text{-}z_{h,s}\text{-}z_{k,r}\text{-}x_{\mathrm{pu}(k),\mathrm{pu}(h)}), \tag{5.37}$$

$i = \mathrm{DO}(h,s\text{-}1)$, $j = \mathrm{DO}(k,r)$, $\forall k,h \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall s \in \mathrm{HYBR}(h)/\{1\}$

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3\text{-}z_{h,s}\text{-}z_{k,r}\text{-}x_{\mathrm{pu}(h),\mathrm{pu}(k)}), \tag{5.38}$$

$i = \mathrm{DO}(h,s\text{-}1)$, $j = \mathrm{DO}(k,r)$, $\forall k,h \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall s \in \mathrm{HYBR}(h)/\{1\}$

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3\text{-}z_{h,s}\text{-}z_{k,r}\text{-}x_{\mathrm{pu}(k),\mathrm{do}(h)}), \tag{5.39}$$

$i = \mathrm{PU}(h,s)$, $j = \mathrm{DO}(k,r)$, $\forall k \in \mathrm{NPD}$, $\forall h \in \mathrm{PND}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall s \in \mathrm{HYBR}(h)$

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3\text{-}z_{h,s}\text{-}z_{k,r}\text{-}x_{\mathrm{do}(h),\mathrm{pu}(k)}), \tag{5.40}$$

$i = \mathrm{PU}(h,s)$, $j = \mathrm{DO}(k,r)$, $\forall k \in \mathrm{NPD}$, $\forall h \in \mathrm{PND}$, $\forall r \in \mathrm{HYBR}(k)$, $\forall s \in \mathrm{HYBR}(h)$

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3\text{-}z_{h,s}\text{-}z_{k,r}\text{-}x_{\mathrm{do}(k),\mathrm{pu}(h)}), \tag{5.41}$$

$i = \mathrm{DO}(h,s\text{-}1)$, $j = \mathrm{PU}(k,r+1)$,

$\forall k \in \mathrm{PND}$, $\forall h \in \mathrm{NPD}$, $\forall r \in \mathrm{HYBR}(k)/\{\mathrm{R}\}$, $\forall s \in \mathrm{HYBR}(h)/\{1\}$

$$z_{h,s}\mathrm{tdc}_i - z_{k,r}\mathrm{tdc}_j < \mathrm{M}(3-z_{h,s}-z_{k,r}-x_{\mathrm{pu}(h),\mathrm{do}(k)}), \qquad (5.42)$$

$i = \mathrm{DO}(h,s\text{-}1),\ j = \mathrm{PU}(k,r+1),$

$\forall k\in\mathrm{PND},\ \forall h\in\mathrm{NPD},\ \forall r\in\mathrm{HYBR}(k)/\{\mathrm{R}\},\ \forall s\in\mathrm{HYBR}(h)/\{1\}$

We know by Proposition 2 (3) that the non-checkpoint stop of a PND (NPD) request must be included in the interval between the chosen pick-up (drop-off) checkpoint and its next (previous) occurrence in the schedule. The above constraints say that given any pair of hybrid requests, the direct path connecting together their non-checkpoint stops identified by the appropriate *x* variable is not allowed if the intervals where the non-checkpoint stops are supposed to be included in, identified by the corresponding *z* variables, do not overlap. For example in constraints (5.35) if $z_{h,s} = 1$ and $z_{k,r} = 1$ we know that do(*h*) must be scheduled between PU(*h,s*) and PU(*h,s*+1); similarly do(*k*) must be scheduled between PU(*k,r*) and PU(*k,r*+1). Therefore, the direct path from do(*k*) to do(*h*), identified by $x_{\mathrm{do}(k),\mathrm{do}(h)}$, can not be allowed if checkpoint PU(*h,s*) is not scheduled earlier than PU(*k,r*+1) and the intervals do not overlap because the vehicle would have to pass by those checkpoints first, not allowing a direct path that would skip them. The M causes these constraints to become irrelevant if either $z_{h,s}$, $z_{k,r}$ or $x_{\mathrm{do}(k),\mathrm{do}(h)}$ are equal to 0.

5.3.4 Other valid inequalities

We note that it would be possible to develop several other valid inequalities similar to the ones already described. Equations from (5.23) to (5.42) reduce the feasible region by eliminating direct arcs from some stop *i* to some stop *j*, identified by $x_{i,j}$. Utilizing the same logic, we could forbid any *path* beginning at *i*, passing through one or more other non-checkpoint stops and ending at *j*. However, the number of constraints added to the formulation would be too high, slowing down the solution search instead of being effective.

5.4 Experimental results

In this section we evaluate the effectiveness of the inequalities defined above by solving different instances of the problem, including none, one or all of them in the formulation. All the runs are performed utilizing CPLEX 9.0 with default settings in a 3.2 GHz CPU with 2GB RAM. We refer to Figure 11 for the geometry of the MAST system considered and the following Table 4 summarizes the assumed parameters, common to all cases.

Table 4 – System parameters, common to all cases

L	10 miles
W	1 mile
C	3
$d_{s,s+1}$ (s = 1,…, TC-1)	5 miles
v	25 miles/hr
b_s (s = 1,…, TS)	18 sec
ω_1 / ω_2 / ω_3	0.4 / 0.4 / 0.2

The above data are consistent with the real data of the MTA line 646 in Los Angeles.

We run two sets of experiments: in set A we assume a difference between the scheduled departure times of two consecutive checkpoints (tdc_{s+1}-tdc_s, s = 1,…, TC-1) of 17.5 minutes; in set B we assume 25 minutes instead. As a result the slack time is approximately 25% in set A and 50% in set B, since the direct time among two consecutive checkpoint is about 12.5 minutes.

In each set we consider two different subsets of runs. In subset A2 (and B2) we assume larger number of rides (R) compared to subset A1 (and B1). In each subset we assume four cases (i.e., for subset A1: A1a, A1b, A1c and A1d) so that moving from the smallest (A1a) to the largest (A1d) case we have a 5 unit increase in the total number of stops in the network (TS). We assume a different number of requests of each type, as shown in the following Table 5. The NP and ND locations are sampled from a spatial uniform distribution over the whole service area (W×L); while the ready times are sampled from a uniform distribution starting from half an hour before the beginning of the service to the end of it.

Table 5 – System parameters specific to each case

Cases Parameters	A1a B1a	A1b B1b	A1c B1c	A1d B1d	A2a B2a	A2b B2b	A2c B2c	A2d B2d
R	2	4	4	4	6	6	6	6
TC	5	9	9	9	13	13	13	13
PD	1	1	1	2	1	1	1	1
PND	2	2	5	6	1	3	5	8
NPD	1	2	4	6	1	2	5	7
NPND	1	1	1	2	0	1	1	1
TS	10	15	20	25	15	20	25	30

As a result we have TS going from 10 to 25 for subsets A1 (and B1) and from 15 to 30 for subsets A2 (and B2).

We tried to maintain the ratio between the different types of requests as close as possible to the real data of MTA line 646 in Los Angeles, which have a distribution described in the following Table 6.

Table 6 – Customer type distribution of MTA line 646

Type	PD	PND	NPD	NPND
%	10%	40%	40%	10%

The results are shown in the following tables. Each table includes four cases. In each case we solve the problem with five different formulations: without adding any groups of inequalities (none), adding only one group at a time (#1, #2 or #3) or adding all the groups together (all). For each run we show the size of the problem solved (after the presolve routine in CPLEX): total variables (var), divided into binary (bin) and linear (lin) and total number of constraints (con). The following columns show the time to reach optimality in seconds (sec), the number of nodes visited in the branch and bound tree (n), the number of simplex iterations performed (i), the relaxed optimal value (rel) and the real optimum (opt). We stopped CPLEX after a maximum solving time of 10 hours (36,000 seconds), recording the upper (ub) and lower (lb) bounds and the gap reached at that time. The results for subset A1, A2, B1 and B2 are shown in Table 7, Table 8, Table 9 and Table 10 respectively.

Table 7 – CPLEX runs, subset A1

Case: **A1a** **TS=10: R=2; PD=1; PND=2; NPD=1; NPND=1**

cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
none	52	29	23	64	0.03	35	156	60.8	84.9	/	/	0.0%
#1	52	29	23	67	0.02	21	98	60.9	84.9	/	/	0.0%
#2	52	29	23	66	0.01	24	118	60.8	84.9	/	/	0.0%
#3	52	29	23	66	0.02	24	118	60.8	84.9	/	/	0.0%
all	52	29	23	71	0.03	26	185	60.9	84.9	/	/	0.0%

Case: **A1b** **TS=15: R=4; PD=1; PND=2; NPD=2; NPND=1**

cuts	var	bin	lin	con	sec	n	10^3 i	rel	opt	ub	lb	gap
none	93	58	35	119	0.10	126	1.14	90.7	117.2	/	/	0.0%
#1	92	57	35	139	0.08	110	0.94	90.7	117.2	/	/	0.0%
#2	93	58	35	137	0.09	128	1.22	90.7	117.2	/	/	0.0%
#3	93	58	35	131	0.13	194	1.55	90.7	117.2	/	/	0.0%
all	92	57	35	169	0.09	94	1.06	90.7	117.2	/	/	0.0%

Case: **A1c** **TS=20: R=4; PD=1; PND=5; NPD=4; NPND=1**

cuts	var	bin	lin	con	sec	10^3 n	10^3 i	rel	opt	ub	lb	gap
none	190	140	50	241	8.10	16.1	97	142.9	220.7	/	/	0.0%
#1	184	135	49	278	2.41	3.0	30	142.9	220.7	/	/	0.0%
#2	190	140	50	298	7.13	12.2	90	142.9	220.7	/	/	0.0%
#3	190	140	50	429	10.16	14.5	117	142.9	220.7	/	/	0.0%
all	184	135	49	441	1.71	2.2	18	142.9	220.7	/	/	0.0%

Case: **A1d** **TS=25: R=4; PD=2; PND=6; NPD=6; NPND=2**

cuts	var	bin	lin	con	sec	10^3 n	10^6 i	rel	opt	ub	lb	gap
none	279	216	63	343	419	327	3.80	154.1	242.4	/	/	0.0%
#1	273	211	62	390	81	64	0.77	154.1	242.4	/	/	0.0%
#2	279	216	63	416	186	131	1.69	154.1	242.4	/	/	0.0%
#3	279	216	63	503	269	192	2.20	154.1	242.4	/	/	0.0%
all	273	211	62	563	80	53	0.73	154.1	242.4	/	/	0.0%

Table 8 – CPLEX runs, subset A2

Case: **A2a** **TS=15: R=6; PD=1; PND=1; NPD=1; NPND=0**

cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
none	65	32	33	77	0.01	1	117	92.5	97.6	/	/	0.0%
#1	65	32	33	100	0.01	0	102	92.6	97.6	/	/	0.0%
#2	65	32	33	83	0.01	0	89	92.5	97.6	/	/	0.0%
#3	65	32	33	99	0.01	0	94	92.5	97.6	/	/	0.0%
all	65	32	33	128	0.01	0	69	92.6	97.6	/	/	0.0%

Case: **A2b** **TS=20: R=6; PD=1; PND=3; NPD=2; NPND=1**

cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
none	147	101	46	189	0.15	20	503	116.0	138.8	/	/	0.0%
#1	144	99	45	229	0.15	7	401	116.0	138.8	/	/	0.0%
#2	147	101	46	201	0.20	53	752	116.0	138.8	/	/	0.0%
#3	147	101	46	265	0.21	27	565	116.0	138.8	/	/	0.0%
all	144	99	45	311	0.16	7	420	116.0	138.8	/	/	0.0%

Case: **A2c** **TS=25: R=6; PD=1; PND=5; NPD=5; NPND=1**

cuts	var	bin	lin	con	sec	n	10^3 i	rel	opt	ub	lb	gap
none	257	196	61	326	1.32	849	9.49	138.7	174.7	/	/	0.0%
#1	253	193	60	407	0.68	160	2.83	138.7	174.7	/	/	0.0%
#2	257	196	61	478	1.25	626	9.21	138.7	174.7	/	/	0.0%
#3	257	196	61	748	1.70	698	9.50	138.7	174.7	/	/	0.0%
all	253	193	60	927	0.92	196	3.36	138.7	174.7	/	/	0.0%

Case: **A2d** **TS=30: R=6; PD=1; PND=8; NPD=7; NPND=1**

cuts	var	bin	lin	con	sec	10^6 n	10^6 i	rel	opt	ub	lb	gap
none	418	342	76	503	36,000	14.3	242	186.7	?	294.1	274.7	6.6%
#1	409	334	75	604	10,316	3.8	60	186.7	293.9	/	/	0.0%
#2	418	342	76	671	36,000	12.1	227	186.7	?	295.2	267.4	9.4%
#3	418	342	76	1,377	36,000	5.1	138	186.7	?	295.3	257.8	12.7%
all	409	334	75	1,428	12,273	3.7	65	186.7	293.9	/	/	0.0%

Table 9 – CPLEX runs, subset B1

Case: **B1a**	**TS=10: R=2; PD=1; PND=2; NPD=1; NPND=1**											
cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
none	67	43	24	85	0.04	64	403	81.2	114.7	/	/	0.0%
#1	67	43	24	91	0.03	27	221	81.8	114.7	/	/	0.0%
#2	67	43	24	87	0.04	50	324	81.2	114.7	/	/	0.0%
#3	67	43	24	85	0.04	64	403	81.2	114.7	/	/	0.0%
all	67	43	24	93	0.03	25	217	81.8	114.7	/	/	0.0%
Case: **B1b**	**TS=15: R=4; PD=1; PND=2; NPD=2; NPND=1**											
cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
none	90	55	35	114	0.06	31	351	93.8	132.3	/	/	0.0%
#1	90	55	35	135	0.05	37	312	93.8	132.3	/	/	0.0%
#2	90	55	35	132	0.05	27	267	93.8	132.3	/	/	0.0%
#3	90	55	35	126	0.06	75	457	93.8	132.3	/	/	0.0%
all	90	55	35	165	0.05	32	302	93.8	132.3	/	/	0.0%
Case: **B1c**	**TS=20: R=4; PD=1; PND=5; NPD=4; NPND=1**											
cuts	var	bin	lin	con	sec	10^6 n	10^6 i	rel	opt	ub	lb	gap
none	242	192	50	294	2,190	2.27	20.1	174.2	278.4	/	/	0.0%
#1	239	190	49	357	155	0.14	1.5	174.2	278.4	/	/	0.0%
#2	242	192	50	369	2,174	2.21	19.0	174.2	278.4	/	/	0.0%
#3	242	192	50	578	1,459	1.12	11.7	174.2	278.4	/	/	0.0%
all	239	190	49	669	116	0.09	1.1	174.2	278.4	/	/	0.0%
Case: **B1d**	**TS=25: R=4; PD=2; PND=6; NPD=6; NPND=2**											
cuts	var	bin	lin	con	sec	10^6 n	10^6 i	rel	opt	ub	lb	gap
none	398	336	62	452	36,000	20.2	249	193.0	?	312.8	293.0	6.3%
#1	398	336	62	506	36,000	17.5	235	193.0	?	312.8	304.4	2.7%
#2	398	336	62	552	36,000	17.0	246	193.0	?	312.8	293.4	6.2%
#3	397	335	62	590	36,000	14.4	215	193.0	?	312.8	295.6	5.5%
all	397	335	62	744	36,000	15.3	219	193.0	?	312.8	299.8	4.1%

Table 10 – CPLEX runs, subset B2

Case: **B2a** **TS=15: R=6; PD=1; PND=1; NPD=1; NPND=0**

cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
none	86	53	33	107	0.03	4	144	92.6	103.3	/	/	0.0%
#1	86	53	33	146	0.02	0	129	92.7	103.3	/	/	0.0%
#2	86	53	33	113	0.03	4	115	92.6	103.3	/	/	0.0%
#3	86	53	33	129	0.02	0	156	92.6	103.3	/	/	0.0%
all	86	53	33	174	0.01	0	82	92.7	103.3	/	/	0.0%

Case: **B2b** **TS=20: R=6; PD=1; PND=3; NPD=2; NPND=1**

cuts	var	bin	lin	con	sec	10^3 n	10^3 i	rel	opt	ub	lb	gap
none	190	144	46	229	1.27	1.44	13.3	123.3	168.1	/	/	0.0%
#1	187	142	45	281	1.23	0.82	12.3	123.3	168.1	/	/	0.0%
#2	190	144	46	269	1.56	1.67	18.0	123.3	168.1	/	/	0.0%
#3	190	144	46	363	1.53	1.45	15.0	123.3	168.1	/	/	0.0%
all	187	142	45	449	1.19	0.80	11.1	123.3	168.1	/	/	0.0%

Case: **B2c** **TS=25: R=6; PD=1; PND=5; NPD=5; NPND=1**

cuts	var	bin	lin	con	sec	10^3 n	10^6 i	rel	opt	ub	lb	gap
none	327	266	61	393	589	388	5.49	143.5	222.1	/	/	0.0%
#1	327	266	61	518	64	41	0.62	143.5	222.1	/	/	0.0%
#2	327	266	61	593	489	314	4.36	143.5	222.1	/	/	0.0%
#3	327	266	61	1,004	1,007	501	6.57	143.5	222.1	/	/	0.0%
all	327	266	61	1,329	51	27	0.46	143.5	222.1	/	/	0.0%

Case: **B2d** **TS=30: R=6; PD=1; PND=8; NPD=7; NPND=1**

cuts	var	bin	lin	con	sec	10^6 n	10^6 i	rel	opt	ub	lb	gap
none	567	491	76	654	36,000	12.0	198	196.6	?	332.8	278.7	16.3%
#1	566	490	76	839	36,000	7.5	168	196.6	?	332.8	298.3	10.4%
#2	567	491	76	908	36,000	7.4	161	196.6	?	334.9	283.3	15.4%
#3	567	491	76	1,826	36,000	4.5	136	196.6	?	333.2	270.8	18.7%
all	566	490	76	2,157	36,000	7.1	150	196.6	?	332.8	305.6	8.2%

In Table 7 (subset A1) cuts #1 and cuts #2 are always effective, with cut #1 consistently improving its efficacy when increasing the problem size. Cut #3 is effective for the smallest (A1a) and largest (A1d) instances, but not for the intermediate cases. Cuts #1 perform better than the others for most cases, but the synergetic effect of including all inequalities at ones (all) is even better for the larger cases (A1c and A1d), where the solving time is reduced approximately by a factor of 5.

Table 8 (subset A2) shows that cuts #1 are always effective and improve their efficacy with increasing size of the problem. In A2d CPLEX is able to reach the optimum in less than 3 hours when including cuts #1 in the formulation, while in the "none" case the gap is still 6.6% after 10 hours. Cuts #2 and cuts #3 are instead not useful for this subset: they worsen the CPLEX performance in each case and they do not help synergistically in the "all" runs, as they did for subset A1.

In Table 9 (subset B1) the results are similar to the ones for subset A1. All cuts are effective in all cases, with cuts #1 being the best and with a good synergistic effect in the "all" run in case B1c, where the solving time is reduced by a factor of almost 20. However, the "all" run does not perform better than the "#1" run in B1d, showing that cuts #2 and #3 are not useful in this instance when added to cuts #1.

In Table 10 (subset B2) the results show that cuts #1 are consistently the best, cuts #2 slightly improve the performance in all cases, while cuts #3 worsen it in each instance except the smallest one. In the heavier case B2d CPLEX is not able to reach optimality in any of the cases after 10 CPU hours, but the effect of cuts #1 is clear from the tightened gap due to higher lower bounds. The "all" runs show a good synergistic effect of the cuts and the solving time is reduced by a factor of 11 in the B2c case and the gap is the smallest in the B2d case.

We note that increasing the slack time from 25% (Set A) to 50% (Set B) expands the feasible region, because more stops could be placed between any pair of consecutive checkpoints in the schedule. As a result, the solving time is consistently larger in all instances. For example in case A1d CPLEX is able to reach the solution in each run relatively fast, while in case B1d CPLEX can not find the optimal solution in any run after the 10 hours maximum solving time allowed. Similarly A2d can be solved faster than B2d and so forth.

In conclusion, the developed valid inequalities are effective in most cases considered either allowing the solver to reach the optimal solution faster or tightening the optimality gap by producing higher lower bounds. Specifically, cuts #1 are consistently effective in almost all cases and their performance improves when applied to larger instances. Cuts #2 show good results in several cases. Cuts #3 are effective only for a few instances. The synergistic effect of including all cuts in the formulation ("all" runs) is positive for most of the cases in subsets A1, B1 and B2.

6 Dynamic operating scenario: scheduling algorithm

In this chapter we look at the problem from a dynamic operational point of view and we develop an insertion heuristic scheduling algorithm for the MAST system. The challenge mainly resides in defining the logic to best operate the vehicle under a dynamic and multi-criteria environment. In particular we need to set the insertion feasibility rules for any given customer at any point in time because inserting a new request in the vehicle's schedule even if feasible at that time, might not be best overall. The algorithm should decide in real time whether accepting a request and provide customers with time windows for their pick-up and/or drop-off service points. An insertion heuristic approach is used because it is computationally fast and it can easily handle complicating constraints in a dynamic environment (Campbell and Savelsbergh, 2003) such as the MAST system.

System

The MAST system considered is the same as described at the beginning of Chapter 5 and represented by Figure 11, consisting of a single vehicle moving along the route back and forth between 1 and C for a total of R rides. The total number of stops at the checkpoints is $TC = (C\text{-}1)\times R+1$ and the initial schedule's array is represented by an ordered sequence of stops $s = 1,\ldots,$ TC at the checkpoints; their scheduled departure time $td_s = \mathrm{tdc}_s$ are assumed to be constraints of the system which can not be violated.

At any moment before or during the ride a customer (PD, PND, NPD or NPND) may call in (or show up at the checkpoints), specifying the locations of both pick-up and drop-off points. We assume that customers will be ready to be picked up at the moment of their request. However, the system could easily handle reservations for future pick-ups by limiting the search for insertion in the portion of the schedule following the ready time specified by the customer.

As mentioned we identify checkpoints by $s = 1,\dots,$ TC and non-checkpoint stops by $s =$ TC+1,…, TS. The index $\alpha(s)$, $s = 1,\dots,$ TS represents the current position of any stop s in the schedule. The problem is then to determine the indexes $\alpha(s)$ $\forall s$ and the departure times td_s for non-checkpoint stops, while not violating td_s for checkpoint stops.

Slack time

In order to allow deviations from the main route to serve NP and ND requests between two consecutive checkpoints, identified by s and s+1, there should be a certain amount of slack time in the schedule. Let $st^{(0)}_{s,s+1}$ be the slack given by the schedule and is computed as follows:

$$st^{(0)}_{s,s+1} = td_{s+1} - td_s - \mathrm{d}_{s,s+1}/\mathrm{v} - \mathrm{b}_{s+1} \qquad s = 1,\dots,\mathrm{TC}\text{-}1 \qquad (6.1)$$

As more pick-ups and drop-offs occur off the base route, the slack is reduced. Let $st_{s,s+1}$ be the available slack that can be used to route the vehicle off the base route. Initially (no requests made yet),

$$st_{s,s+1} = st^{(0)}_{s,s+1} \qquad s = 1,\ldots, \mathrm{TC}\text{-}1 \qquad (6.2)$$

Idle policy

We assume that the vehicle, driving from checkpoint s to s+1, follows a no-idle policy until all the requests in between them have been satisfied. The unused slack time $st_{s,s+1}$ possibly still available when arriving to checkpoint s+1 is spent as idle time (note that while the vehicle is idle at s+1, new upcoming customer requests can still be inserted in the schedule before s+1 using $st_{s,s+1}$ if feasible and best at the moment, meaning that the vehicle leaves s+1 to serve the new requests and comes back to s+1 before td_{s+1}).

Arrival times

While td_s represents the scheduled departure time at stop s, we define ta_s as the arrival time at stop s. Because of the idle policy, we have for non-checkpoint stops (s > TC) $td_s = ta_s + \mathrm{b}_s$ and for checkpoint stops ($s \leq$ TC), $td_s \geq ta_s + \mathrm{b}_s$ and their initial values are:

$$ta_{s+1} = td_{s+1} - st_{s,s+1} - \mathrm{b}_s \qquad s = 1,\ldots, \mathrm{TC}\text{-}1 \qquad (6.3)$$

Bus motion

We assume that the vehicle follows a *rectilinear motion*, allowing the vehicle to move only along the horizontal or vertical direction; this is a good approximation of the real world, since vehicles ride along streets, which often form a grid.

Furthermore, whenever a horizontal or vertical direction can be equally chosen to reach the next scheduled stop, the vehicle prefers the one that keeps it closer to the central x axis of the service area. This behavior guarantees a better service to the future expected demand under the assumption of uniform distribution of non-checkpoint requests.

6.1 Control parameters

In order to improve the insertion algorithm effectiveness we define and make use of control parameters that are a function of the future expected demand (*usable slack time*) and the relative position of the new request with respect to the current position of the vehicle (*backtracking distance*). In order to define the former we first need to introduce the concept of *bucket*.

6.1.1 Buckets

The MAST insertion algorithm does not explicitly add a constraint to limit the maximum allowable ride time of each customer as the Dial-a-Ride algorithms generally do. Instead, it obtains a similar result working with “buckets”. The underlying concept is that for PND and NPD type of customers one of the service points (either a P or a D checkpoint) is already part of the schedule; therefore, the algorithm attempts to insert the corresponding ND and NP stops in the “vicinity” of the first occurrence of those checkpoints in the schedule’s array. If not feasible, the algorithm checks for insertion in the “vicinity” of the following occurrences of the checkpoint of interest, one by one, till feasibility is found. Clearly, this postponement causes a delay for the entire trip, but the ride time will be upper bounded. This logic is consistent with Proposition 1 and 2, stating that customers would minimize their ride time by boarding or disembarking the vehicle at the checkpoint as close as possible time wise to their inserted non-checkpoint stop.

In order to define buckets, let’s consider the schedule’s array as shown in Table 11, illustrating the checkpoints only with their corresponding stop index *s*. We know by Equation (5.1) that each checkpoint *c* is scheduled to be visited by the vehicle a number of times, with different stop indices $s_k(c)$ (stop index of the k^{th} occurrence of checkpoint *c* in the schedule), depending on how many rides (R) are planned.

Definition: For every checkpoint *c*, we define a **bucket of *c***, in general, as a portion of the schedule delimited by two successive occurrences of *c*, namely all the stops *s* in the schedule’s array such that $\alpha[s_k(c)] \leq \alpha(s) < \alpha[s_{k+1}(c)]$ for any allowable *k*.

Table 11 – Schedule's array and buckets

ride	s	Checkpoints c
1	1	1
	2	2
	3	3
	…	…
	c	c
	…	…
	C-1	C-1
	C	C
2	C+1	C-1
	…	…
	2(C-1)+1-(c-1)	c
	…	…
	2C-2	2
	2C-1	1
3	2C	2
	…	…
	2(C-1)+1+(c-1)	c
	…	…
	3C-2	C
…	…	…
r	r(C-1)+1-(c-1)	c
	…	…
	r(C-1)+1	1
$r+1$	…	…
	r(C-1)+1+(c-1)	c
	…	…
	(r+1)(C-1)+1	C
…	…	…
R	TC=R(C-1)+1	1 or C

1st bucket of c=1

bucket of c=2

another bucket of c=2

For PND customers the buckets of their checkpoints of interests are defined by all the stops s such that $\alpha(\text{PU}(k,r)) \leq \alpha(s) < \alpha(\text{PU}(k,r+1))$, $\forall r \in \text{RD}/\{\text{R}\}$.

For NPD customers the buckets of their checkpoints of interests are defined by all the stops s such that $\alpha(\text{DO}(k,r\text{-}1)) \leq \alpha(s) < \alpha(\text{DO}(k,r))$, $\forall r \in \text{RD}/\{1\}$.

The buckets' definition for NPND type customers needs to be revised since they do not rely on checkpoints for pick-ups and drop-offs; so we identify the buckets with the rides. More formally, let's characterize the sequence representing the occurrences of any terminal checkpoint (c = 1 or C):

$$s_k(1 \text{ or C}) = 1 + (\mathrm{C}-1)(k-1) \qquad k = 1,\dots, \mathrm{R}+1 \qquad (6.4)$$

We have that, for NPND type customers, a bucket represents all the stops s such that $\alpha[s_k(1 \text{ or C})] \leq \alpha(s) < \alpha[s_{k+1}(1 \text{ or C})]$ for any allowable k as described in Equation (6.4).

6.1.2 Usable slack time

The slack time is a crucial resource needed to serve customers. When this resource is scarce, the system is not able to properly satisfy new requests and it is forced to postpone or reject them. Therefore, a MAST service needs to be particularly careful about accepting customer requests that require a lot of slack time consumption preventing future requests from being fully satisfied. In fact, an insertion that appears to be good at the time of its placement in the schedule may not be so, if we consider future expected customer requests. We therefore need to define a parameter that properly controls the consumption of slack time.

$st_{s,s+1}$ represents the current available unused slack time between two consecutive checkpoints s and s+1; while $st^{(0)}_{s,s+1}$ is the slack time initially available before any insertion has been performed. We define the *usable slack time* $st^{u}_{s,s+1}$ as the maximum amount of slack time that any customer request is allowed to consume for its insertion between s and s+1. It represents an upper bound on the usable amount of slack time and it prevents a single insertion from consuming too much of it. $st^{u}_{s,s+1}$ is defined as a function of the future expected demand between s and s+1 and is not related to the actual unused slack time $st_{s,s+1}$ and therefore $st^{u}_{s,s+1}$ can be greater or lower than $st_{s,s+1}$ depending on the circumstances. As we will see in the insertion feasibility Section 6.2.1, a request will be allowed to consume the minimum value among $st^{u}_{s,s+1}$ and $st_{s,s+1}$ for its insertion.

We assume that the demand rate λ (# of requests per unit time in the service area L×W) of non-checkpoint's requests (NP and ND) is uniformly distributed in the service area and constant over time. The time interval between two checkpoints s and s+1 is defined by td_{s+1}-td_s, while the ratio between the area covered by the segment of the route from s and s+1 and the total service area is given by $\frac{|\mathrm{x}_s - \mathrm{x}_{s+1}|}{\mathrm{L}}$ (where x_s and x_{s+1} are the x coordinate values of s and s+1 with respect to the service area). Consequently, the expected demand between s and s+1 (total # of insertion requests) arising during td_{s+1}-td_s, $\Lambda_{s,s+1}$, is estimated as follows (see Figure 12):

$$\Lambda_{s,s+1} = \lambda \frac{|x_s - x_{s+1}|}{L} (td_{s+1}\text{-}td_s) \qquad (6.5)$$

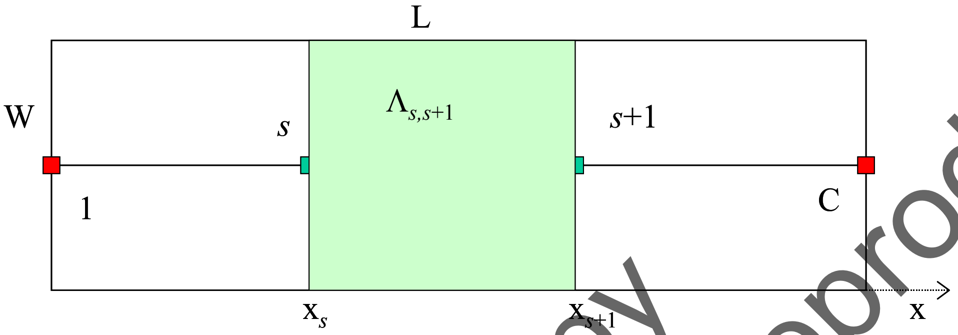

Figure 12 – Portion of service area covered by the segment between *s* and *s*+1

As soon as the vehicle departs from *s* at td_s, the expected residual demand drops linearly until reaching the zero value at td_{s+1}. Hence, the expected residual demand as a function of the current clock time t_{now}, $\Lambda_{s,s+1}^{(t_{now})}$, may be expressed as (see Figure 13):

$$\Lambda_{s,s+1}^{(t_{now})} = \begin{cases} \Lambda_{s,s+1} & t_{now} < td_s \\ \Lambda_{s,s+1}\left(1 - \dfrac{t_{now} - td_s}{td_{s+1} - td_s}\right) & td_s \le t_{now} \le td_{s+1} \\ 0 & t_{now} > td_{s+1} \end{cases} \qquad (6.6)$$

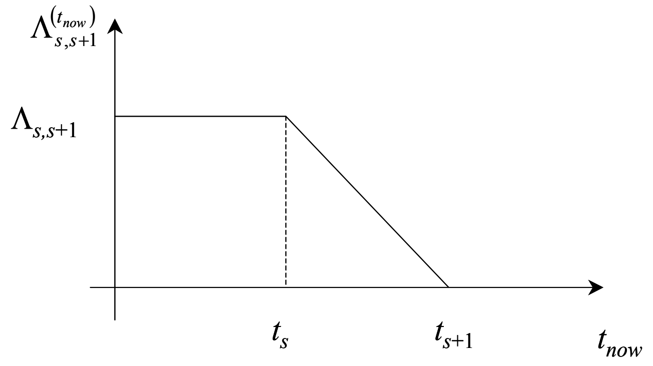

Figure 13 – Expected residual demand between *s* and *s*+1 as a function of t_{now}

We define the parameter $\pi_{s,s+1}$ as a function of the expected demand as follows:

$$\pi_{s,s+1} = 1 + \left(\frac{\pi^{(0)}_{s,s+1} - 1}{\Lambda_{s,s+1}}\right)\Lambda^{(t_{now})}_{s,s+1} \qquad \text{with } 0 \le \pi^{(0)}_{s,s+1} \le 1 \qquad (6.7)$$

Since $0 \le \Lambda^{(t_{now})}_{s,s+1} \le \Lambda_{s,s+1}$, we have that $\pi^{(0)}_{s,s+1} \le \pi_{s,s+1} \le 1$ and $\pi^{(0)}_{s,s+1}$ can be set accordingly. We finally define the usable slack time, $st^{u}_{s,s+1}$, as follows:

$$st^{u}_{s,s+1} = \pi_{s,s+1} st^{(0)}_{s,s+1} \qquad (6.8)$$

If the residual expected demand $\Lambda^{(t_{now})}_{s,s+1} \rightarrow 0$, then $\pi_{s,s+1} \rightarrow 1$ and $st^{u}_{s,s+1} \rightarrow st^{(0)}_{s,s+1}$. Whereas, when $\Lambda^{(t_{now})}_{s,s+1}$ attains its maximum ($\Lambda_{s,s+1}$), $\pi_{s,s+1}$ reaches its minimum value, $\pi^{(0)}_{s,s+1}$, and so does $st^{u}_{s,s+1} = \pi^{(0)}_{s,s+1}\, st^{(0)}_{s,s+1}$.

Combining Equations (6.6), (6.7) and (6.8) we finally derive the expression for the *usable slack time*, $st^{u}_{s,s+1}$, as a function of t_{now} (see Figure 14):

$$st^{u}_{s,s+1} = \begin{cases} \pi^{(0)}_{s,s+1} st^{(0)}_{s,s+1} & t_{now} < td_{s} \\ \left[1 + \left(\pi^{(0)}_{s,s+1} - 1\right)\left(1 - \dfrac{t_{now} - td_{s}}{td_{s+1} - td_{s}}\right)\right] st^{(0)}_{s,s+1} & td_{s} \le t_{now} \le td_{s+1} \\ st^{(0)}_{s,s+1} & t_{now} > td_{s+1} \end{cases} \qquad (6.9)$$

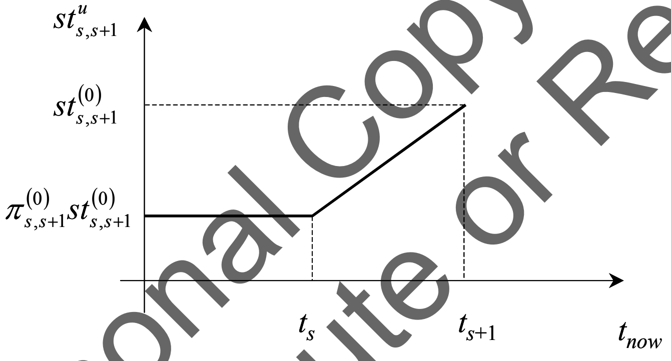

Figure 14 – *Usable slack time*

Let's now consider a non-checkpoint request q located at the edge of the service area, such that $y_q = 0$ or $y_q = W$ and $x_s \le x_q \le x_{s+1}$ and let's assume that the schedule between s and s+1 is empty (no previously inserted stops). In order to be inserted, the q request would require an amount of slack time st_q given by the time needed by the vehicle to deviate from the x axis, serve the q request and come back to the x axis ($st_q = W/v+b_q$). Since the minimum amount of usable slack time from

Equation (6.9) is given by $st^{u}_{s,s+1} = \pi^{(0)}_{s,s+1}\, st^{(0)}_{s,s+1}$, we need to have $st^{u}_{s,s+1} \geq st_q$ to prevent the *q* request from being rejected. Hence we define:

$$\pi^{(0)\min}_{s,s+1} = (\mathrm{W/v+b}_q)/\, st^{(0)}_{s,s+1} \tag{6.10}$$

as the minimum value of $\pi^{(0)}_{s,s+1}$ that guarantees every non-checkpoint request *q* to be considered for insertion between *s* and *s*+1 with empty schedule, regardless of the location of *q* as long as $\mathrm{x}_s \leq x_q \leq x_{s+1}$.

Setting $\pi^{(0)}_{s,s+1} < \pi^{(0)\min}_{s,s+1}$ would prevent the algorithm from working properly, because some customers would be rejected not because of system saturation or end of service, but because of improper parameter setting. Clearly, setting $\pi^{(0)}_{s,s+1} = 0$ would result in having $st^{u}_{s,s+1} = 0$ for $t_{now} < td_s$, preventing any requests before td_s from being considered for insertion. On the contrary, $\pi^{(0)}_{s,s+1} = 1$ causes $st^{u}_{s,s+1} = st^{(0)}_{s,s+1}$ at any time and customers requests would have no limit on the amount of slack time allowed to be consumed for their insertion.

A proper value of $\pi^{(0)}_{s,s+1}$ in between $\pi^{(0)\min}_{s,s+1}$ and 1 allows the system to control the consumption of slack time. Any request occurring before td_s can use at most the minimum value of $st^{u}_{s,s+1} = \pi^{(0)}_{s,s+1}\, st^{(0)}_{s,s+1}$ because there is an expected demand of future customers that should be properly served with the remaining slack time. Whereas, if a customer request occurs towards the end of the ride from *s* to *s*+1, it is allowed to consume a bigger portion of the slack time until a maximum of $st_{s,s+1}^{(0)}$ because the chance of having additional requests before the vehicle reaches the next checkpoint *s*+1 is very low.

6.1.3 Backtracking distance

The insertion procedure can cause the vehicle to drive back and forth with respect to the direction of a ride *r*, not only consuming the extra slack time, but also having a negative impact on the customers already onboard, which may perceive this behavior as an additional delay. Therefore, we limit the amount of backtracking in the schedule. The backtracking distance indicates how much the vehicle drives backwards on the x axis while moving between two consecutive stops to either pick up or drop off a passenger at a non-checkpoint stop with respect to the direction of the current ride. More formally, as shown in Figure 15, given any two consecutive stops identified by *a* and *b* [such that $\alpha(a)+1 = \alpha(b)$] and the vector $\hat{d}_{a,b}$ representing the distance from *a* to *b*, the backtracking distance $bd_{a,b}$ is defined as the negative component of the projection of $\hat{d}_{a,b}$ along the unit vector $\hat{d}_r$, representing the direction of the current ride *r* (1→C or vice versa, parallel to the x axis) as follows:

$$bd_{a,b} = -\min(0, \hat{d}_r \cdot \hat{d}_{a,b}) \tag{6.11}$$

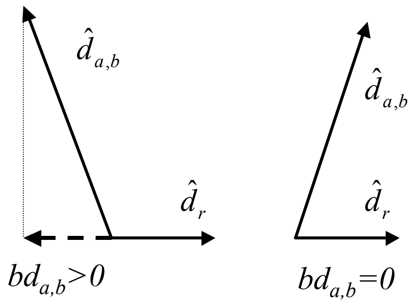

Figure 15 – Backtracking distance

The backtracking parameter (BACK > 0) is defined as the maximum allowable backtracking distance that the vehicle can ride between any two consecutive stops. BACK is a parameter and can be set accordingly; clearly with BACK ≥ L any backtracking is allowed.

6.2 Algorithm description

6.2.1 Feasibility

While evaluating a customer request, the algorithm needs to determine the feasibility of the insertion of a new stop (let's identify it by $s = q$) between any two consecutive stops a and b already scheduled. The extra time needed for the insertion is computed as follows:

$$\Delta t_{a,q,b} = (d_{a,q}+d_{q,b}-d_{a,b})/v - b_q \tag{6.12}$$

Let *m* and *m*+1 be the checkpoints prior and after stops *a* and *b* in the schedule. The algorithm computes also the backtracking distances $bd_{a,q}$ and $bd_{q,b}$ by Equation (6.11). Finally, it is feasible to insert *q* between *a* and *b* if (see Figure 16):

$$\begin{cases} \Delta \mathrm{t}_{a,q,b} \leq \min(st_{m,m+1},\ st^{u}_{m,m+1}) & (6.13) \\ bd_{a,q} \leq \mathrm{BACK} & (6.14) \\ bd_{q,b} \leq \mathrm{BACK} & (6.15) \end{cases}$$

The algorithm does not need to check feasibility with respect to the vehicle capacity because we assume it to be infinite.

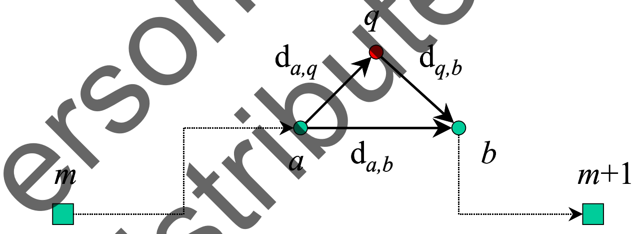

Figure 16 – Insertion feasibility of *q*

6.2.2 Cost function

When searching for the best insertion among the feasible ones, the algorithm computes a COST for each of them and selects the one with the minimum value. Let's assume that the insertion of a stop *q* between *a* and *b* is feasible and we need to compute its COST. The system's entities affected by an insertion are:

- The vehicle, in terms of how many extra miles it has to drive.
- The customer requesting the insertion, in terms of how long the ride time is.
- The passengers already onboard and waiting to be dropped off, in terms of how much longer they have to stay onboard.
- The previously inserted customers in the schedule waiting to be picked up at the NP stops, in terms of how much longer their pick-up time is delayed and also in terms of how much their expected ride time changes.

Thus, the algorithm computes the following quantities:

- ΔPT: the sum over all passengers of the extra ride time, including the ride time of the customer requesting the insertion.
- ΔPW: the sum over all passengers of the extra waiting time at the already inserted NP stops.

Finally, the cost function is defined as:

$$\mathrm{COST} = w_1 \times \Delta t_{a,q,b} + w_2 \times \Delta PT + w_3 \times \Delta PW \qquad (6.16)$$

where w_1, w_2 and w_3 are the weights, which can be modified as needed to emphasize one factor over the others. $\Delta t_{a,q,b}$ corresponds to the consumption of the slack time (the resource needed by the system to serve more customers). During heavy demand periods, we should assign a higher value to this scarce resource by increasing w_1 with respect to w_2 and w_3. In contrast, during low demand periods, the opposite is true and the COST function should emphasize more the service quality for the customers rising w_2 and w_3 over w_1.

We note that w_1, w_2 and w_3 are comparable but not equal to ω_1, ω_2 and ω_3 defined in Section 5.1. In fact the cost function (6.16) evaluates the incremental cost brought by a new insertion, while the objective function (5.2) measures the whole cost of the system. In addition, the third term in (6.16) includes only the total extra time that already inserted customers would have to wait at their NP stops, being their pick-up delayed because of the new insertion. Whereas the third term in 5.2 (weighted by ω_3) includes the total waiting time, which is the sum over all customers of the time interval from the ready time to the departure time. In the global optimization of the system (Chapter 5) we assume a static environment and therefore it is not possible to identify how much of this total time is extra waiting time, which is anyhow small compared to the total.

6.2.3 Insertion procedure

PD type

PD type requests do not need any insertion procedure since both pick-up and drop-off points are checkpoints and they are already part of the schedule. However, once the PD type customers are onboard, they are important in evaluating the COST of any other insertion.

PND type

PND type customers need to have their ND stop *q* inserted in the schedule. The algorithm checks for insertion's feasibility in the buckets of the P checkpoint. Since the ND stop can not be scheduled before P, the first bucket to be examined is the one starting with the first occurrence of P following the current position of the vehicle (bucket delimited by $s_{k'}$(P) and $s_{k'+1}$(P) with $k' = \min_k s_k(\mathrm{P}), \text{ s.t. } td_{s_k(\mathrm{P})} \geq t_{now}$). Among the feasible insertions between all pairs of consecutive stops *a*, *b* in the first bucket, the algorithm selects the one with the minimum COST and then stops. The customer is therefore scheduled to be picked up at $s_{k'}$(P) and dropped off at the ND inserted stop *q*. If no feasible insertions are found in the first bucket, the algorithm repeats the procedure in the second bucket (assuming that the customer will be picked up at the beginning of the second bucket corresponding to the following occurrence of P, $s_{k'+1}$(P)). The process is repeated bucket by bucket until at least one feasible insertion is found.

NPD type

NPD type customers need to have their NP stop q inserted in the schedule. Similarly, the algorithm checks for insertion's feasibility in the buckets of the D checkpoint. The first bucket to be examined is the one delimited by the current position of the vehicle (x$_b$, y$_b$) and the first occurrence of D following the current position of the vehicle ($s_{k'}$(D) with $k' = \min_k s_{k'}(\mathrm{D}), \text{s.t. } td_{s_k(\mathrm{D})} \geq t_{now}$). In general, (x$_b$, y$_b$) does not correspond to a stop. Therefore, the first pair of points between which the algorithm checks for feasibility is represented by (x$_b$, y$_b$) and the first stop to be visited afterwards, as shown in Figure 17.

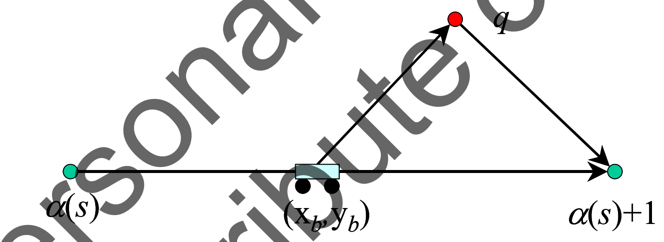

Figure 17 – Insertion from current vehicle position

Among the feasible insertions in the first bucket, the algorithm selects the one with the minimum COST and then stops. The customer is therefore scheduled to be picked up at the inserted NP stop q and dropped off at $s_{k'}$(D). If no feasible insertions are found in the first bucket, the algorithm repeats the procedure in the second bucket (forcing the customer to be dropped off at the end of the second bucket, corresponding to the following occurrence of D, $s_{k'+1}$(D)). This process is repeated bucket by bucket until at least one feasible insertion is found.

NPND type

A NPND type customer requires the insertion of two new stops *q* and *q'*; therefore, the insertion procedure will be performed by a O(TS^2) procedure, meaning that for each feasible insertion of the NP stop *q*, the algorithm checks feasibility for the ND stop *q'*. A NPND feasibility is granted when both NP and ND insertions are simultaneously feasible. The search for NPND feasibility is performed with the additional constraint of having *q* scheduled before *q'*.

Recall that buckets correspond to the rides for a NPND type customer. The search for NPND feasibility is performed in at most two consecutive buckets meaning that when checking for NP insertion feasibility in bucket *i* and *i*+1, the algorithm looks for ND insertion feasibility only in bucket *i* and *i*+1.

The algorithm starts checking the NPND feasibility in the first bucket delimited by the current position of the vehicle (x$_b$, y$_b$) and the end of the current ride *r*. This is the first occurrence in the schedule of one of the terminal checkpoints $s = 1$ or $s = \mathrm{C}$, namely $s_{k'}(1 \text{ or C}) = \min\limits_{k} s_k(1 \text{ or C}), \text{s.t. } td_{s_k(1 \text{ or C})} \geq t_{now}$. Among all feasible NPND insertions in the first bucket, the algorithm selects the one with the minimum COST. If no NPND feasibility is found, the algorithm will then check pairs of two consecutive buckets at a time, increasing the "checking-range" by one bucket at each step (buckets 1/2, then buckets 2/3, …, *i*/*i*+1, etc.). While checking buckets *i*/*i*+1, we already know that NPND insertion is infeasible in bucket *i* (because it has been already established before in the procedure while checking buckets *i*-1/*i*). Therefore, while NP insertion feasibility needs to be considered in both buckets (since NPND insertion infeasibility in bucket *i* does not prevent NP insertion to be feasible in *i*), ND insertion needs to be checked only in bucket *i*+1. The procedure will continue till at least one NPND feasible insertion is found.

Rejection policy

The general assumption while performing the insertion procedure is a no-rejection policy from both the MAST service and the customers. Thus, the algorithm attempts to insert the customer requests checking if necessary the whole existing schedule bucket by bucket and rejection may occur only if there is no feasibility at all. It may occur, for example with a very high demand rate or when a customer request arrives towards the end of the service. On the other hand, the customers are assumed to never reject the insertion proposed by the algorithm and there is no negotiation between the MAST system and the customers.

6.2.4 Update procedure

Once a minimum COST feasible insertion is selected, a new stop *q* (either a NP or a ND request) has been successfully scheduled between two points *a* and *b* in a portion of the schedule delimited by checkpoints m and m+1, and the variables of the system need to be updated.

The slack time will be updated as follows:

$$st_{m,m+1} = st_{m,m+1} - \Delta \mathrm{t}_{a,q,b} \tag{6.17}$$

The departure and arrival times will also be updated (delayed) as follows:

$$td_s = td_s + \Delta \mathrm{t}_{a,q,b} \qquad \forall s \text{ s.t. } \alpha(s) \in [\alpha(b), \alpha(m+1)) \tag{6.18}$$

$$ta_s = ta_s + \Delta \mathrm{t}_{a,q,b} \qquad \forall s \text{ s.t. } \alpha(s) \in [\alpha(b), \alpha(m+1)] \tag{6.19}$$

Since the departure times td_s of checkpoints ($s \le$ TC) are constraints of the system and act as "time-barriers", all the stops that are not in the portion of the schedule where the insertion takes place (between m and m+1) are not affected. We can therefore identify six different cases:

- Customers having both pick-up and drop-off stops scheduled before q are not affected by the insertion.
- Customers having their pick-up stop before q and their drop-off stop in between q and m+1 will have their ride time increased because their drop-off stop will be delayed as given by Equation (6.19).

- Customers having their pick-up stop before *q* and their drop-off stop after *m*+1 will not be affected by the insertion because the departure time t_{m+1} will remain unchanged.
- Customers having both their pick-up and drop-off stops in between *q* and *m*+1 will have both of them delayed by the same amount as given by Equations (6.18) and (6.19). Therefore, their waiting time at the pick-up stop will be increased but their ride time will remain unchanged.
- Customers having their pick-up stop in between *q* and *m*+1 and their drop-off stop after *m*+1 will have their waiting time at the pick-up stop increased as given by Equation (6.18) and their ride time decreased by the same amount because their drop-off stop will not be affected.
- Customers having both their pick-up and drop-off stops after *m*+1 will not be affected.

Time windows

The algorithm provides customers at the time of the request with time windows for their pick-up and drop-off locations. To do so, it computes the earliest departure time from *q*, etd_q, as follows:

$$etd_q = td_a + \mathrm{d}_{a,q}/\mathrm{v} + \mathrm{b}_q \qquad (6.20)$$

where td_a represents the current departure time from stop a. Also the departure time of q is initialized likewise:

$$td_q = td_a + d_{a,q}/\mathrm{v} + \mathrm{b}_q = etd_q \tag{6.21}$$

It can easily be shown that etd_q is a lower bound for any further updates of td_q.

The algorithm then computes the latest departure time from q, ltd_q, as follows:

$$ltd_q = etd_q + st_{m,m+1} \tag{6.22}$$

We prove that ltd_q is an upper bound for td_q by the following contradiction argument. Let's use the superscript β (with $\beta = 0,\ldots, f$) to indicate the β^{th} update of a variable and suppose that $td_q^{(f)} > ltd_q$, we have $td_q^{(f)}\text{-}td_q^{(0)} > ltd_q\text{-}td_q^{(0)}$. We also know by Equation (6.18) that:

$$\begin{aligned} td_q^{(f)}\text{-}td_q^{(0)} &= (td_q^{(f)}\text{-}td_q^{(f\text{-}1)})+\ldots+(td_q^{(\beta)}\text{-}td_q^{(\beta\text{-}1)})+\ldots+(td_q^{(1)}\text{-}td_q^{(0)})= \\ &= \Delta \mathrm{t}_\mathrm{f}+\ldots+\Delta \mathrm{t}_\beta+\ldots+\Delta \mathrm{t}_1 = \sum_{k=1}^{f} \Delta t_k \end{aligned} \tag{6.23}$$

and from Equations (6.21) and (6.22), $ltd_q - td_q^{(0)} = ltd_q - etd_q = st_{m,m+1}$, but this would imply $\sum_{k=1}^{f} \Delta t_k > st_{m,m+1}$, meaning that the sum of the extra time needed for insertions after the insertion of q had exceeded the total slack time available after the insertion of q and this is a contradiction since the feasibility check would have prevented this from happening. Therefore, Equation (6.22) says that future possible insertions between m and q will delay td_q to a maximum total amount of time bounded by the currently available slack time.

In a similar fashion, the earliest and latest arrival times, eta_q and lta_q, are computed. As a result, the customer, once accepted, is provided with etd_q, ltd_q, eta_q and lta_q knowing that their actual times td_q and ta_q will be bounded by these values:

$$etd_q \leq td_q \leq ltd_q \tag{6.24}$$

$$eta_q \leq ta_q \leq lta_q \tag{6.25}$$

While a P request has $etd_P = td_P = ltd_P$ because the departure time from a checkpoint is a constant in a MAST system, a D request will have $eta_D \leq ta_D \leq lta_D$. Clearly NP and ND requests will also have $etd_{NP} \leq td_{NP} \leq ltd_{NP}$ and $eta_{ND} \leq ta_{ND} \leq lta_{ND}$.

6.3 Experimental results

In this section we discuss the results obtained by simulation analysis. The target is to show that the insertion heuristic developed in this paper can be a used as an efficient scheduling tool for real MAST systems. We test its performance on a simulation model of the actual MAST service represented by MTA Line 646 in Los Angeles. In order to perform this task, we first need to define the MAST system's performance measures.

6.3.1 Performance measures

We define the following performance parameters for a MAST system:

- MI: total miles driven by the vehicle
- PT: average ride time per passenger
- PW: average extra waiting time (*td_{NP} - etd_{NP}*) over NP requests only

These three indicators are directly related to the corresponding terms of the COST function in Equation (6.16): $\Delta t_{a,q,b}$, ΔPT, ΔPW. Thus, we can similarly define the overall performance Z of a MAST system as:

$$Z = w_1 \times MI/v + w_2 \times PT \times NC_T + w_3 \times PW \times NC_{NP} \quad (6.26)$$

where NC_T and NC_{NP} stand respectively for the total number of customers and the total number of NP customer requests (NPD and NPND types) served by the system. Z is in time units.

In addition, let's define:

- PI: average time interval between request/show up and earliest pick-up time (etd_P or etd_{NP}) per passenger
- PST: percentage of the total initial slack time (= $\sum_{s=1}^{\mathrm{TC}-1} st_{s,s+1}^{(0)}$) consumed

Given a total demand rate θ (customers/hour), we define the *saturation level* as the maximum demand that a system configuration can satisfy without becoming unstable. This level can be estimated by looking at the PI values. Given that the demand is uniform over time, for systems well below their saturation level, the PI values should be around half the headway of the system. A slightly larger value of PI, but constant over the simulation time, shows that the system is near the saturation level, but still below it. Even if a few customers have to wait longer to be picked up due to temporary congestions created by the randomness of the demand, the system on average is stable. If instead the PI value increases over the simulation time, then the system is unstable and the demand rate is above the saturation level. An indication of how much the demand rate is below the saturation level is given by the PST; values around 90% indicate that the demand rate is more or less at saturation level. In addition, since the slack time consumption is directly proportional to the miles driven, the PST and MI values are related to each other. Therefore, bigger values of MI also indicate a higher level of saturation.

6.3.2 Algorithm performance

As earlier noted, a MAST service already exists in San Pedro in Los Angeles County, Line 646. San Pedro is one of Los Angeles County's busiest commercial hubs, consisting of several warehouses, factories and offices. Bus lines offer regular fixed-route service in the area during the daytime. However, for safety reasons, employees of local firms working on night shifts have been finding it extremely inconvenient to walk to and wait at a bus stop. Therefore, MTA Line 646 offers a MAST service during nighttime, transporting passengers between one of the business areas in San Pedro to a nearby bus terminal.

The MAST system represented by Line 646 consists of a single vehicle covering a service area with L = 10 miles and W = 1 mile, with two terminal checkpoints and one intermediate checkpoint located in the middle. The duration of each ride is 30 minutes and the headway is 1 hour. The service operates for 4.5 hours (9 rides) each night. Given that v = 25 miles/hour, the system has very little slack time ($st_{s,s+1}^{(0)}$ = 2.5 minutes, for s = 1,…, TC-1; therefore, about 6 minutes per ride), allowing very few insertions of non-checkpoint requests, but this is justified by the very low actual demand (4-5 customers/hour, most of them being of type PND and NPD). These "light" conditions allow the bus operator to easily make all the decisions concerning accepting/rejecting customer requests and routing the vehicle since the system needs to deal with only 2-3 insertion requests per ride.

MTA is interested in testing the MAST concept for higher demand levels. However, at the current slack level, the system will not be able to accommodate more demand. Therefore, in order to evaluate the performance of the insertion algorithm for the higher demand cases we perform the simulation experiments assuming a larger slack time. A summary of the parameters values that are used in the experiments are shown in Table 12.

Table 12 – System parameters

L	10 miles
W	1 mile
C	3
$d_{s,s+1}$ (s = 1,…, TC-1)	5 miles
td_{s+1} - td_s (s = 1,…, TC-1)	25 min (t_1 = 0)
v	25 miles/hour
b_s (s = 1,…, TS)	18 sec
w_1 / w_2 / w_3	0.25 / 0.25 / 0.5

From Equation (6.1) we compute the values of the initial slack times $st_{s,s+1}^{(0)}$ = 12.7 minutes (s = 1,…, TC-1) that are about 50% of the time intervals between two consecutive checkpoints' departure times (td_{s+1} - td_s = 25 minutes).

In setting the COST function's weights, we assume that customers perceive the extra waiting time at stops (w_3) with more discomfort than the ride time on the vehicle (w_2) and that slack time consumption (w_1) and passengers' ride time (w_2) are equally weighted. This is the inverse of what we assumed in Section 5.1 for ω_3, compared to ω_1 and ω_2 (see Table 4), but as we noted earlier the waiting time weighted by w_3 and ω_3 are different in nature.

Given a total demand rate θ (customers/hour) constant over time, we also assume that the customer types are distributed as shown in Table 13, like assumed earlier (Table 6):

Table 13 – Customer type distribution

Type	PD	PND	NPD	NPND
%	10%	40%	40%	10%

We further assume that the checkpoint requests (P and D) are uniformly distributed among the C checkpoints and that non-checkpoint requests (NP and ND) are uniformly distributed in the service area. The simulation is run for 50 hours. We verified that this length of simulation time was sufficiently long to have all the performance parameters converge to their steady-state values for stable systems. According to the parameter values shown in Table 12, the total number of rides R = 60.

We first perform a set of runs setting the control parameters BACK = L and $\pi_{s,s+1}^{(0)} = 1$ (for all $s = 1,\ldots,$ TC-1) allowing any backtracking and any slack time consumption if available, thus giving the most freedom to the algorithm when checking for insertion feasibility. At these parameter settings (configurations A) we seek the saturation level of the system, by examining the PI and PST values for different values of the demand θ. The results are shown in Table 14.

Table 14 – Saturation level for configurations A

Configuration	A1	**A2**	A3
θ (customers/hour)	15	**20**	25
BACK (miles)	L	**L**	L
$\pi_{s,s+1}^{(0)}$	1	**1**	1
PI (min)	56.52	**61.67**	236.74
PST (%)	81.3%	**91.3%**	98.9%
saturation level?	below	**yes**	above
PW (min)	1.07	**1.23**	1.75
PT (min)	23.86	**25.86**	30.39
MI (miles)	1012.7	**1051.4**	1083.8

The findings show that the saturation level is around θ = 20 customers/hour (configuration A2). While A1 is a stable system relatively far from saturation (PST = 81.3%), A2 is right at the boundary because the PI value is higher than half the headway (50 minutes), but it does not increase over time. Hence, the system is stable, but since the slack time consumption is very high (PST = 91.3%), it is near the demand limit. Anything above θ = 20 would lead to system instability as shown by the results from A3, where the PI value is very high and keeps increasing along with the simulation run time and the PST is close to 100%.

Therefore, by allowing more slack time in the schedule ($st_{s,s+1}^{(0)}$ = 12.7 minutes instead of 2.5, for $s = 1,\ldots,$ TC-1) and setting BACK = L and $\pi_{s,s+1}^{(0)}$ = 1 (configurations A), MTA Line 646 would be able to serve a demand θ with up to 20 customers/hour assuming the customer type distribution of Table 13.

Now, keeping the demand at the saturation level (configuration A2), we want to observe the effect of modifying the usable slack time $st_{s,s+1}^{u}$. For this purpose, maintaining BACK = L, we vary the values of $\pi_{s,s+1}^{(0)}$ (for all $s = 1,\ldots,$ TC-1) in the range from 1 to $\pi_{s,s+1}^{(0)\min}$ (configurations B) to observe the effect of this control parameter. We compare the performances of each case by means of the object function Z, as defined in Equation (4.29). The simulation run time is again 50 hours. Each configuration is tested with exactly the same demand using CNR (Common Random Numbers). The results are summarized in Table 15. From Equation (4.13), $\pi_{s,s+1}^{(0)\min}$ is approximately equal to 0.22.

Table 15 – Effect of $\pi_{s,s+1}^{(0)}$ - configurations B

Configuration	B1 = A2	B2	B3	B4	**B5**	B6
θ (customers/hour)	20	20	20	20	**20**	20
BACK (miles)	L	L	L	L	**L**	L
$\pi_{s,s+1}^{(0)}$	1	0.75	0.5	0.4	**0.3**	$\pi_{s,s+1}^{(0)\min}$=0.22
PI (min)	61.67	55.87	54.59	51.56	**52.26**	51.60
PST (%)	91.3%	87.4%	82.3%	79.2%	**76.6%**	72.0%
saturation level?	yes	below	below	below	**below**	below
PW (min)	1.23	1.15	1.25	1.32	**1.41**	1.37
PT (min)	25.86	24.68	24.13	23.09	**22.60**	22.76
MI (miles)	1051.4	1021.7	989.0	968.2	**951.5**	921.7
Z	7149	6987	6853	6624	**6533**	6551

The figures reveal the positive effect of decreasing $\pi_{s,s+1}^{(0)}$ from 1 to almost $\pi_{s,s+1}^{(0)\min}$. All the performance parameters significantly improve their values, with the exception of PW, showing initially a progress, but then a progressive worsening. Also the Z values gradually drop and reach their minimum value with configuration B5 at $\pi_{s,s+1}^{(0)} \cong 0.3$, slightly greater than $\pi_{s,s+1}^{(0)\min}$. Due to the increased efficiency of the algorithm, all the configurations drop well below their saturation levels. Note that configuration B6 has lower PST and MI values, indicating a better performance in terms of the slack time consumption, but the overall performance Z shows a worsening of the service quality with respect to B5. These results show the benefit of controlling the consumption of slack time and saving some of it for future insertions.

Now, starting from configuration B5, we would like to observe the effect of limiting the backtracking distance. We perform another set of simulations (configurations C), keeping θ = 20 and $\pi^{(0)}_{s,s+1}$ = 0.3 and varying the BACK parameter from L to 0. The results are shown in Table 16.

Table 16 – Effect of BACK - configurations C

Configuration	C1 = B5	C2	C3	C4	C5	**C6**	C7	C8
θ (customers/hour)	20	20	20	20	20	**20**	20	20
BACK (miles)	L	1.5	0.8	0.5	0.3	**0.2**	0.1	0
$\pi^{(0)}_{s,s+1}$	0.3	0.3	0.3	0.3	0.3	**0.3**	0.3	0.3
PI (min)	52.26	52.26	52.35	51.70	52.19	**52.23**	51.28	51.84
PST (%)	76.6%	76.6%	75.8%	74.2%	72.8%	**72.4%**	71.2%	70.9%
saturation level?	below	below	below	below	below	**below**	below	below
PW (min)	1.41	1.41	1.39	1.38	1.37	**1.37**	1.42	1.43
PT (min)	22.60	22.60	22.62	22.46	22.34	**22.28**	22.36	22.94
MI (miles)	951.5	951.5	946.4	936.1	927.2	**924.2**	916.8	914.5
Z	6533	6533	6528	6478	6435	**6419**	6451	6596

There are no changes in the performance by lowering the value of the BACK parameter from L (configuration C1) down to about 1.5 miles (C2). This means that in the simulation there are no cases of an insertion with a backtracking distance bigger than 1.5 miles. Therefore, setting BACK to a value larger than 1.5 has no effect on the schedule. On the contrary, improvements in all the performance measures can be progressively seen in cases C3, C4, C5 and C6 (BACK = 0.8, 0.5, 0.3 and 0.2) while C7 and C8 (BACK = 0.1 and 0) show better values for PST and MI, but the overall performance Z slightly worsens due to the increasing values of PW and PT. All the cases are well below their saturation level and the best configuration according to Z is found by setting BACK = 0.2 miles, corresponding to case C6. These experiments illustrate the positive effect of limiting to a certain degree the amount of backtracking that the vehicle is allowed to do.

Case C6 represents a better configuration than A2 with respect to the overall performance Z and almost all the other parameters (with the exception of PW, slightly increased). In particular, the improved efficiency of the algorithm causes the MI and PST values to drop and the system is now well below saturation. We therefore look for the new saturation level for these more efficient parameter settings by performing another set of runs (configurations D, see Table 17) starting from configuration C6 and progressively increasing θ.

Table 17 – New saturation level - configurations D

Configuration	D1 = C6	**D2**	D3
θ (customers/hour)	20	**25**	30
BACK (miles)	0.2	**0.2**	0.2
$\pi_{s,s+1}^{(0)}$	0.3	**0.3**	0.3
PI (min)	52.23	**55.98**	77.58
PST (%)	72.4%	**86.8%**	95.9%
saturation level?	below	**yes**	above
PW (min)	1.37	**1.72**	1.92
PT (min)	22.28	**23.93**	29.00
MI (miles)	924.2	**983.4**	1020.6

As done for configurations A, we can estimate the saturation level for configurations D by looking at the stability of the PI value over the simulation time. The figures show that θ = 25 customers/hour (D2) approximately represent the limit for the system. Anything above this value would cause instability. Therefore, the adjustments made on the control parameters allow the insertion heuristics to handle a demand rate 25% larger than the initial configuration A2.

6.3.3 Comparison vs. optimality

We now provide an evaluation of the insertion heuristic algorithm by comparing its performance against optimality found by CPLEX, as described in Chapter 5.

In order to perform this task we need to slightly revise the COST function defined in Equation (6.16); in particular we need to modify the waiting time term so that it matches the corresponding term in the objective function given in the Equation (5.2), in order to have w_3 and ω_3 weighing the same thing. Thus, we have that

$$COST = w_1 \times \Delta t_{a,q,b} + w_2 \times \Delta PT + w_3 \times \Delta PW_T \tag{6.27}$$

where ΔPW_T now represents the sum over all passengers of the total waiting time, defined as the time interval between the ready time and the pick-up time. We also have that Equation (6.26) is modified accordingly as follows:

$$Z = w_1 \times MI/v + w_2 \times PT \times NC_T + w_3 \times PW_T \times NC_T \tag{6.28}$$

where PW_T now represents the average total waiting time of all customers.

For each subset (A1, A2, B1 and B3) solved by CPLEX in Section 5.4 we consider only the cases with heavier demand (A1d, A2d, B1d and B2d), since in all the other cases the heuristic reaches optimality. In Table 18 for each case we provide the Z value obtained by the insertion heuristic and by CPLEX. For the heuristic results we show the Z obtained with no control and with the best setting of the control parameters found for each case (if any). The CPLEX results presented are the best ones for each case depending on whether and which valid inequalities are added to the formulation.

Table 18 – Heuristic vs. optimality

	Heuristic				CPLEX				
	no control ($\pi_{s,s+1}^{(0)} = 1$; BACK = L)	best control							
case	Z	Z	$\pi_{s,s+1}^{(0)}$	BACK (miles)	cuts	opt	ub	lb	gap
A1d	**242.4**	already optimal			any	**242.4**	/	/	0.0%
A2d	**294.1**	no improvement			#1/all	**293.9**	/	/	0.0%
B1d	**323.1**	**314.1**	0.3	0.2	#1	?	**312.8**	**304.4**	2.7%
B2d	**344.1**	**332.8**	0.9	5	all	?	**332.8**	**305.6**	8.2%

The figures show that in the A1d case the heuristic reaches the optimal value 242.4 even with the default values of the control parameters ($\pi_{s,s+1}^{(0)} = 1$ and BACK = L). In the A2d case, the heuristic reaches a Z value of 294.1 very close to optimality (293.9) with the default values of the control parameters and we could not improve the result by modifying them. In cases B1d and B2d the heuristic with no control reaches the Z values of 323.2 and 344.1 respectively that are higher than the upper optimality bound found by CPLEX (312.8 and 332.8 correspondingly); a proper setting of the control parameters allows to improve the solutions substantially respectively down to 314.1 and 332.8.

In conclusion, the heuristic obtains results that are very close to optimality for the instances considered especially by properly modifying the values of the control parameters.

6.4 Sensitivity over service area

We now perform a simulation analysis to observe the behavior of the system when modifying the shape of the service area, maintaining constant the total square mileage. In particular we want to observe the effect of the control parameters in each configuration over their saturation level.

The assumed parameters of the systems are shown in the following Table 19 and are the same as the ones in Table 12, excluding L and W that are objects of our analysis. We note that the initial slack time available between any pair of consecutive checkpoints will vary depending on the assumed proportion between W and L: the smaller L, the larger the amount of slack time, because the checkpoints are closer.

Table 19 – System parameters

C	3
$d_{s,s+1}$ (s = 1,…, TC-1)	5 miles
td_{s+1} - td_s (s = 1,…, TC-1)	25 min (t_1 = 0)
v	25 miles/hour
b_s (s = 1,…, TS)	18 sec
w_1 / w_2 / w_3	0.25 / 0.25 / 0.5

Configuration A: W=1; L=12

The first analysis is done within a *slim* service area with L = 12 and W = 1, both in miles. The distance between checkpoints is 6 miles and the slack time available between any consecutive pair of them is therefore about 10.5 minutes. We first look for the saturation level of this system configuration setting the control parameters BACK = L and $\pi_{s,s+1}^{(0)} = 1$, allowing any backtracking and any slack time consumption if available, thus giving the most freedom to the algorithm when checking for insertion feasibility. The results are shown in Table 20.

Table 20 – Saturation level for Configuration A, BACK = L, $\pi_{s,s+1}^{(0)} = 1$

θ (customers/hour)	18
BACK (miles)	L
$\pi_{s,s+1}^{(0)}$	1
PW (min)	0.99
PT (min)	25.33
MI (miles)	1049.8

The system becomes unstable with a demand θ > 18 customers/hour, that is approximately the saturation level of this configuration.

We now look for the new capacity of the system with a proper setting of the control parameters, namely: BACK = 0.2 and $\pi_{s,s+1}^{(0)}$ = 0.3. The results summarized in Table 21 show that the saturation level is increased up to 21 customers/hour.

Table 21 – Saturation level for Configuration A, BACK = 0.2, $\pi_{s,s+1}^{(0)}$ = 0.3

θ (customers/hour)	21
BACK (miles)	0.2
$\pi_{s,s+1}^{(0)}$	0.3
PW (min)	1.43
PT (min)	25.42
MI (miles)	1018.2

The improvement on the capacity of the system is only 3 customers/hour (about 15% increase), but it shows the positive effect of the control parameters also on the total mileage MI, that has decreased by approximately 30 miles despite the increased demand, demonstrating an improved efficiency of the algorithm. The ride time (PT) remains about the same, while the extra waiting time at NP stops (PW) slightly increases, due to the heavier demand that leads to an increased number of insertions and postponement of pick-ups.

Configuration B: W=2; L=6

A similar analysis is performed over a service area with W = 2 and L = 6. The total square mileage is still equal to 12 and all the other parameters of the system are kept the same. However, given the different shape of the area, checkpoints are closer to each other and therefore the initial slack time available between any two pair of consecutive checkpoints is larger, about 18 minutes.

The next two tables show the figures for the saturation levels of this configuration. Table 22 shows the results with the maximum freedom given to the insertion procedure (BACK = L and $\pi^{(0)}_{s,s+1}$ = 1). Table 23 illustrates the findings with a proper setting of the control parameters, namely BACK = 0.3 and $\pi^{(0)}_{s,s+1}$ = 0.3.

Table 22 – Saturation level for Configuration B, BACK = L, $\pi^{(0)}_{s,s+1}$ = 1

θ (customers/hour)	12
BACK (miles)	L
$\pi^{(0)}_{s,s+1}$	1
PW (min)	1.36
PT (min)	20.59
MI (miles)	1054.5

Table 23 – Saturation level for Configuration B, BACK = 0.3, $\pi^{(0)}_{s,s+1}$ = 0.3

θ (customers/hour)	20
BACK (miles)	0.3
$\pi^{(0)}_{s,s+1}$	0.3
PW (min)	1.94
PT (min)	22.81
MI (miles)	933.5

In this case the improvement due to control parameter adjustment is more significant: the saturation level jumps from 12 to 20 customers/hour (66% increase) and the mileage (MI) is reduced by about 120 miles, even with the increased demand. The values of PT and PW increase slightly.

Configuration C: W=3; L=4

We now consider a service area with W = 3 and L =4. The total square mileage is again still equal to 12 and all the other parameters of the system are kept the same, but checkpoints are even closer to each other and the initial slack time available between any two pair of consecutive checkpoints is now about 20 minutes.

Table 24 shows the saturation level for this configuration with the maximum freedom given to the insertion procedure (BACK = L and $\pi_{s,s+1}^{(0)}$ = 1) and Table 25 shows the saturation level with a proper setting of the control parameters for this system (BACK = 0.5 and $\pi_{s,s+1}^{(0)}$ = 0.5).

Table 24 – Saturation level for Configuration C, BACK = L, $\pi_{s,s+1}^{(0)}$ = 1

θ (customers/hour)	12
BACK (miles)	L
$\pi_{s,s+1}^{(0)}$	1
PW (min)	1.73
PT (min)	17.37
MI (miles)	1047.3

Table 25 – Saturation level for Configuration C, BACK = 0.5, $\pi_{s,s+1}^{(0)}$ = 0.5

θ (customers/hour)	18
BACK (miles)	0.5
$\pi_{s,s+1}^{(0)}$	0.5
PW (min)	1.68
PT (min)	22.17
MI (miles)	964.0

The increase in the saturation level due to control parameter adjustments is significant, from 12 to 18 customers/hour (50% increase) and the mileage (MI) also is reduced by about 80 miles. A more significant increase of the PT value is observed for this Configuration.

The analysis shows that a proper setting of the control parameters could significantly improve the performance of the system for every configuration. The results also show that the *slim* Configuration A performs better with or without the involvement of the control parameters, even though with different emphasis in the two cases.

Without activating the control parameters (BACK = L and $\pi_{s,s+1}^{(0)} = 1$) Configuration A outperforms Configurations B and C in terms of system capacity (18 vs. 12 customers/hour), meaning that the insertion procedure is able to perform better in case of a slimmer service area and consequently a lesser amount of slack time. This is due to the fact that a "wild" consumption of the slack time is less likely to happen when there is a smaller amount of it available to begin with and the system is able to control itself better.

When properly setting the control parameters, every configuration benefits from it, but the improvements shown in Configuration B and C are much more evident than those in Configuration A and, while the *slim* case still performs better, the three "optimized" systems are comparable in terms of capacity and performance.

In addition we note that the longitudinal velocity V (along the x axis in Figure 11) of the vehicle decreases with the widening of the service area (Configurations B and C), because of the increased amount of time needed by the vehicle to serve points along the larger width. PD customers traveling only to/from checkpoints could perceive this slowness unfavorably, because on average they would experience ride times increasingly larger than the direct time needed to travel between their pick-up and drop-off. Therefore, only slimmer service areas, such as Configuration A would be suitable for public transportation purposes, where the longitudinal velocity V of the vehicle is not much slower than a fixed-route lines traveling between checkpoints. However, configurations with wider service area could very well be appropriate for the transportation of goods instead of people.

6.5 MAST/Fixed-route comparison

We now perform a comparison between the MAST service and a fixed-route bus service. For this purpose we assume the same service area (L×W = 10×1 miles) is served by a single vehicle fixed-route line consisting of 19 stops evenly distributed along the x axis (one stop every 0.5 miles). See Figure 18. We keep the same vehicle speed (v = 25 miles/hour) and the same b = 18 sec for all stops and we assume no slack time for the fixed-route since it does not have to drive off route. We note that in most transit systems there is also additional slack time added to the schedule due to random travel times. Since in this study we consider only deterministic travel times, we assume the slack time for accommodating random travel times is zero. Since the headway for the fixed route bus is 60 minutes, the scheduled/actual travel time between two consecutive stops is 1.5 minutes.

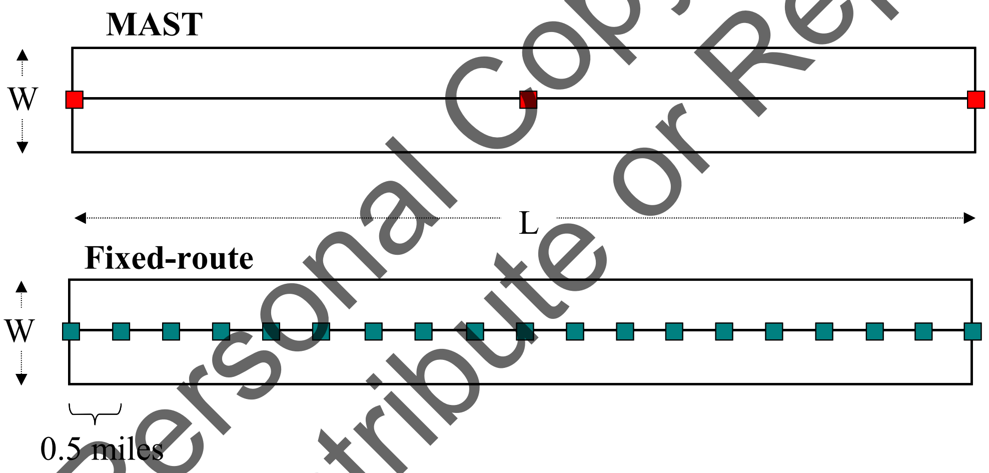

Figure 18 – MAST/Fixed-route systems comparison

In order to perform the comparison, we need to define an additional performance measure given by the average walking time per passenger (PWK) assuming a walking speed of 3 miles/hour. While the MAST system serves its customers point to point and no walking occurs, a fixed-route system forces NP and ND requests to walk to/from the nearest fixed stop in order to use the service. Note that the P and D requests could have a certain amount of walking time associated with it, but considering the same demand it would be equivalent for both systems. Consequently, we assume it to be zero. Therefore, the overall performance Z defined in Equation (6.26) slightly changes as follows:

$$Z = w_1 \times MI/v + w_2 \times PT \times NC_T + w_3 \times PW \times NC_{NP} + w_4 \times PWK \times NC_T \qquad (6.29)$$

where the new last term represents the contribution to Z of the amount of walking time and w_4 is its weight factor that we conservatively assume to be equal to 0.5 like w_3 (even though customers would probably perceive walking time with more discomfort that waiting time at a bus stop especially during nighttime for safety reasons).

We ran the simulation for the fixed-route service again for 50 hours so that R = 100 in this case, and we compare the results with the MAST configuration D2 using the same demand. The results are shown in Table 26.

Table 26 – MAST/fixed-route comparison

θ (customers/hour)	25	
System	MAST-D2	Fixed
PI (min)	55.98	30.21
PW (min)	1.72	0
PT (min)	23.93	14.00
PWK (min)	0	7.5
MI (miles)	983.4	1000
Z	8674.2	9862.1

We observe that the PI values (directly proportional to the headway of the system and not included in Z) clearly are in favor of the fixed-route system. However, it has been shown that for headways larger than 12-13 minutes the majority of the customers are aware of the schedule (Okrent, 1974) and this is true for all P requests showing up at bus stops (for both systems). Furthermore, as we already noticed, for NP requests PI represents the waiting time incurred from the customer's call (ready time τ) to the etd_{NP} that people most likely spend at an office, at home or in another comfortable location, not at a bus stop. Therefore, we do not consider PI as a valid parameter for this comparison.

The other figures show that the MAST system compared to the corresponding fixed-route results has a smaller PW (< 2 minutes) and a PT bigger by approximately 10 minutes, but MI is lower and there's no walking for the customers as opposed to the fixed-route system where on average customers walk 7.5 minutes. The overall performance Z is clearly in favor of the MAST system, confirming the validity of this innovative service compared to a conventional transportation system for this service region.

7 Conclusions and future research

In this research we analyzed the Mobility Allowance Shuttle Transit (MAST) service, an innovative fixed and flexible type of transportation system that merges the flexibility of demand responsive transit (DRT) systems and the low cost operability of fixed-route systems.

From a design point of view, we investigated the viability of MAST systems. Results show that the system is able to serve properly a reasonable demand while maintaining a relatively high longitudinal velocity, in order to make the service attractive to customers. The relationship between velocity and demand density can be beneficially used in the design process to set the parameters of the MAST system, such as slack time, size of the service area and number of vehicles to be employed per line.

From an operational point of view in static scenarios, the problem is mathematically formulated as a NP-Hard integer linear program and it is a special case of the Pickup and Delivery Problem (PDP). We developed and added to the formulation a set of proper and efficient valid inequalities that sped up the search for the optimal solution by raising the lower optimality bound.

The MAST scheduling problem is then examined from a dynamic operational perspective. We developed a customized insertion heuristic algorithm to schedule the service dynamically. Due to the dynamic nature of the environment, the algorithm makes effective use of a set of control parameters to reduce the consumption of slack time and enhance the algorithm performance. The results show the efficacy of the algorithm and its control parameters and demonstrate that the algorithm can be used as an effective method to automate scheduling of this line and other similar services. A comparison performed by simulation shows that the innovative hybrid characteristics of MAST services are competitive with conventional fixed-route ones and perform better under certain demand distributions.

Future research on MAST systems should focus on studying the multi-vehicle case and designing efficient networks of this type of service, in order to cover wider service areas and different demand distributions. The combinatorial nature of the problem would also need to develop and analyze efficient algorithms to schedule the vehicles interconnected within these networks.

References

Aldaihani M.M. and Dessouky, M. (2003), "Hybrid scheduling methods for paratransit operations", *Computers & Industrial Engineering*, 45, 75-96.

Aldaihani, M.M., Quadrifoglio, L., Dessouky, M. and Hall, R.W. (2004) "Network design for a grid hybrid transit service", *Transportation Research*, 38A, 511-530.

Badeau, P., Buertin, F., Gendreau, M., Potvin, J.-Y., Taillard, E. (1997) "A parallel tabu search heuristic for the vehicle routing problem with time windows", *Transportation Research*, 5C, 109-122.

Beardwood, J., Halton, J.H. and Hammersley, J.M. (1959) "The shortest path through many points", *Proceedings of the Cambridge Philosophical Society*, 55, 299-327.

Bélisle, J.P. (1989) "La modélisation analytique de transport porte-à-porte et ses applications au transport adapté", Cahier G-89-02, Ecole des Hautes Etudes Commerciales, Montreal.

Bouwkamp C.J. (1977) "On the average distance between points in two coplanar non-overlapping circular disks", *J. Appl. Sci. Engng.* A, 2, 183-186.

Campbell, A. M., and Savelsbergh, M. (2004) "Efficient insertion heuristics for vehicle routing and scheduling problems", *Transportation Science*, 38, 369-378.

Ceder, A. and Wilson, N. H. M. (1986) "Bus network design", *Transportation Research*, 20B, 331-344.

Chang, S.K. and Schonfeld, P.M. (1991a) "Multiple period optimization of bus transit systems", *Transportation Research*, 25B, 453-478.

Chang, S.K. and Schonfeld, P.M. (1991b) "Optimization models for comparing conventional and subscription bus feeder services", *Transportation Science*, 25, 281-298.

Chien, S. and Schonfeld, P.M. (1997) "Optimizing of grid transit system in heterogeneous urban environment", *Journal of Transportation Engineering*, 123, 28-35.

Christofides N. and Eilon S. (1969) "Expected distances in distribution problems", *Operations Research Quarterly*, 20, 437-443.

Cordeau, J.F. and Laporte, G. (2003) “A tabu search heuristic for the static multi-vehicle dial-a-ride problem”, *Transportation Research*, 37B, 579-594.

Crainic, T.G., Malucelli, F. and Nonato, M. (2001) “Flexible many-to-few + few-to-many = an almost personalized transit system”, TRISTAN IV, São Miguel Azores Islands, 435-440.

Daganzo, C.F. (1978) “An approximate analytic model of many-to-many demand responsive transportation systems”, *Transportation Research*, 12, 325-333.

Daganzo, C.F. (1980) “Network representation, continuum approximations and a solution to the spatial aggregation problem of traffic assignment”, *Transportation Research*, 14B, 229-239.

Daganzo, C.F. (1984a) “The length of tours in zones of different shapes”, *Transportation Research*, 18B, 135-145.

Daganzo, C.F. (1984b) “The distance traveled to visit N points with a maximum of C stops per vehicle: an analytic model and an application”, *Transportation Science*, 18, 331-350.

Daganzo, C.F. (1984c) “Checkpoint dial-a-ride systems”, *Transportation Research*, 18B, 315-327.

Daganzo, C.F. (1991) “logistic systems analysis”, *Lecture notes in Economics and Mathematical Systems*, 361, Springer-Verlag, Heidelberg, Germany.

Desaulniers, G. et al. (2000) “The VRP with pickup and delivery”, *Cahiers du GERARD*, G-2000-25, Ecole des Hautes Etudes Commerciales, Montréal.

Desrosiers, J., Dumas, Y. and Soumis, F. (1986) “A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows”, *American Journal of Mathematical and Management Sciences*, 6, 301-325.

Dessouky, M., Rahimi, M., and Weidner, M. (2003) “Jointly optimizing cost, service, and environmental performance in demand-responsive transit scheduling”, *Transportation Research*, 8D, 433-465.

Dial, R.B. (1995) “Autonomous dial-a-ride transit: introductory overview”, *Transportation Research*, 3C, 261-275.

Diana, M. and Dessouky, M. (2004) “A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows”, *Transportation Research*, 38B, 539-557.

Diana, M., Dessouky, M. and Xia, N. (2005) "A model for the fleet sizing of demand responsive transportation services with time windows", *Transportation Research B*, accepted with revisions in progress.

Dumas, Y., Desrosiers, J. and Soumis, F. (1991) "The pickup and delivery problem with time windows", *European Journal of Operational Research*, 54, 7-22.

Faithorne D. (1965) "Distances between pairs of points in towns of simple geometrical shapes", Proceedings of the 2nd International Symposium on the Theory of Traffic Flow, 391-406, OECD, Paris.

Feuerstein, E. and Stougie, L. (2001) "On-line single-server dial-a-ride problems", *Theoretical Computer Science*, 268, 91-105.

Fischetti, M., Toth, P. (1989) "An additive bounding procedure for combinatorial optimization problems", *Operations Research*, 37, 319-328.

Fu, L. (2002) "A simulation model for evaluating advanced dial-a-ride paratransit systems", *Transportation Research*, 36A, 291-307.

Gosh B. (1951) "Random distances within a rectangle and between two rectangles", *Bulletin of the Calcutta Mathematical Society*, 43, 17-24.

Hall R.W. (1986) "Discrete models/continuous models", *Omega International Journal of Management Science*, 14, 213-220.

Healy, P. and Moll, R. (1995) "A new extension of local search applied to the dial-a-ride problem", *European Journal of Operational Research*, 83, 83-104.

Hickman, M. and Blume, K. (2000) "A method for scheduling integrated transit service", 8th International Conference on Computer-Aided Scheduling of Public Transport (CASPT), Berlin, Germany.

Holroyd, E.M. (1965) "The optimum bus service: a theoretical model for a large uniform urban area", Proceedings 3rd International Symposium Transportation Traffic Theory, 303-328.

Horn, M.E.T. (2002a) "Multi-modal and demand-responsive passenger transport systems: a modeling framework with embedded control systems", *Transportation Research*, 36A, 167-188.

Horn, M.E.T. (2002b) "Fleet scheduling and dispatching for demand-responsive passenger services", *Transportation Research*, 10C, 35-63.

Ioachim, I. et al. (1995) "A request clustering algorithm for door-to-door handicapped transportation", *Transportation Science*, 29, 63-78.

Jacobson, J. (1980) "Analytical models for comparison of alternative service options for the transportation handicapped", *Transportation Research*, 14A, 113-118.

Jaillet P. (1988) "A priori solution of a traveling salesman problem in which a random subset of the customers are visited", *Operations Research*, 36, 929-936.

Jaw, J.J. et al. (1986) "A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows", *Transportation Research*, 20B, 243-257.

Kalantari, B., Hill, A.V., Arora, S.R. (1985) "An algorithm for the traveling salesman problem with pickup and delivery customers", *European Journal of Operational Research*, 22, 377-386.

Koshizuka, T. and Kurita, O. (1991) "Approximate formulas of average distances associated with regions and their applications to location problems", *Mathematical Programming*, 52, 99-123.

Kuah, G.K. and Perl, J. (1988) "Optimization of feeder bus routes and bus-stop spacing", *Journal of Transportation Engineering*, 114, 331-354.

Landrieu, A., Mati, Y. and Binder, Z. (2001) "A tabu search heuristic for the single vehicle pickup and delivery problem with time windows", *Journal of Intelligent Manufacturing*, 12, 497-508.

Langevin, A., Mbaraga, P. and Campbell, J.F. (1996) "Continuous approximation models in freight distribution: an overview", *Transportation Research*, 30B, 163-188.

LeBlanc, L.J. (1988) "Transit system network design", *Transportation Research*, 22B, 383-390.

Lesley L.J.S. (1976a) "Optimum bus-stop spacing: Part 1", *Traffic Engineering Control*, 17, 399-401.

Lesley L.J.S. (1976b) "Optimum bus-stop spacing: Part 2", *Traffic Engineering Control*, 17, 472-473.

Liaw, C.F., White, C.C. and Bander, J.L., (1996) "A decision support system for the bimodal dial-a-ride problem", *IEEE Transactions on Systems, Man, and Cybernetics*, 26(5), 552-565.

Liu, F.-H., Shen, S.-Y. (1999) "The fleet size and mix vehicle routing problem with time windows", *Journal of the Operational Research Society*, 50, 721-732.

Lu, Q. and Dessouky, M. (2004) "An exact algorithm for the multiple vehicle pickup and delivery problem", *Transportation Science*, 38, 503-514.

Lu, Q. and Dessouky, M. (2005) "A new insertion-based construction heuristic for solving the pickup and delivery problem with hard time windows," *European Journal of Operations Research*, to appear.

Madsen, O.B.G., Raven, H.F. and Rygaard, J.M. (1995) "A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives", *Annals of Operations Research*, 60, 193-208.

Mandl, C.E. (1980) "Evaluation and optimization of urban public transport networks", *European Journal of Operational Research*, 6, 31-56.

Malucelli, F., Nonato, M. and Pallottino, S. (1999) "Demand adaptive systems: some proposals on flexible transit", *Operations Research in Industry*, T. Ciriani, E. Johnson, R. Tadei (eds.), 157-182, McMillian.

Min, H. (1989) "The multiple vehicle routing problem with simultaneous delivery and pick-up points", *Transportation Research*, 23A, 377-386.

Nanry, W.P. and Barnes, J.W. (2000) "Solving the pickup and delivery problem with time windows using reactive tabu search", *Transportation Research*, 34B, 107-121.

Newell, G.F. (1979) "Some issues relating to the optimal design of bus routes", *Transportation Science*, 13, 20-35.

Okrent, M.M. (1974) "Effect of transit service characteristics on passenger waiting time", MS Thesis, Department of Civil Engineering, Northwestern University, Evanston, IL.

Potvin, J.-Y., Rousseau, J.-M. (1993) "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows", *European Journal of Operational Research*, 66, 331-340.

Psaraftis, H.N. (1980) "A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem", *Transportation Science*, 14, 130-154.

Psaraftis, H.N. (1983a) "An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows", *Transportation Science*, 17, 351-357.

Psaraftis, H.N. (1983b) “k-interchange procedures for local search in a precedence-constrained routing problem”, *European Journal of Operational Research*, 13, 391-402.

Psaraftis, H.N. (1983c) “Analysis of an $O(N^2)$ heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem”, *Transportation Research*, 17B, 133-145.

Rhee, W.T. (1993) “On the stochastic traveling salesperson problem for distributions with unbounded support”, *Mathematics of Operations Research*, 18, 252-259.

Rochat, Y., Taillard, E. (1995) “Probabilistic diversification and intensification in local search for vehicle routing”, *Journal of Heuristics*, 1, 147-167.

Ruben H. (1978) “On the distance between points in polygons”, *Lecture Notes in Biomathematics*, No 23 (Miles R. and Serra, J. Eds.), Springer-Verlag, Berlin.

Ruland, K.S. and Rodin, E.Y. (1997) “The pickup and delivery problem: faces and branch-and-cut algorithm”, *Computers and Mathematics with Applications*, 33, 1-13.

Savelsbergh, M.W.P. and Sol, M. (1995) “The general pickup and delivery problem”, *Transportation Science*, 29, 17-29.

Savelsbergh, M.W.P. and Sol, M. (1998) “Drive: dynamic routing of independent vehicles”, *Operations Research*, 46, 474-490.

Schweitzer P.A. (1968) “Moments of distances of uniformly distributed points”, *American Mathematical Monthly*, 75, 802-804.

Sexton, T.R. and Bodin, L.D. (1985a) “Optimizing single vehicle many-to-many operations with desired delivery times: 1. Scheduling”, *Transportation Science*, 19, 378-410.

Sexton, T.R. and Bodin, L.D. (1985b) “Optimizing single vehicle many-to-many operations with desired delivery times: 2. Routing”, *Transportation Science*, 19, 411-435.

Sexton, T.R. and Choi, Y. (1986) “Pickup and delivery of partial loads with soft time windows”, *American Journal of Mathematical and Management Sciences*, 6, 369-398.

Stadje W. (1995) “Two asymptotic inequalities for the stochastic traveling salesman problem”, *Sankhya: The Indian Journal of Statistics*, 57A1, 33-40.

Stein, D.M. (1978a) “An asymptotic probabilistic analysis of a routing problem”, *Mathematics of Operations Research*, 3, 89-101.

Stein, D.M. (1978b) “Scheduling dial-a-ride transportation problems”, *Transportation Science*, 12, 232-249.

Stone R.E. (1991) “Some average distance results”, *Transportation Science*, 25, 83-91.

Szplett, D.B. (1984) “Approximate procedures for planning public transit systems: a review and some examples”, *Journal of Advanced Transportation*, 18, 245-257.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., Potvin, J. (1997) “A tabu search heuristic for the vehicle routing problem with soft time windows”, *Transportation Science*, 31, 170-186.

Teodorovich, D. and Radivojevic, G. (2000) “A fuzzy logic approach to dynamic dial-a-ride problem”, *Fuzzy Sets and Systems*, 116, 23-33.

Toth, P., and Vigo, D. (1997) “Heuristic algorithm for the handicapped persons transportation problem”, *Transportation Science*, 31, 60-71.

Van Der Bruggen, L.J J., Lenstra, J.K. and Schuur, P.C. (1993) “Variable-depth search for the single-vehicle pickup and delivery problem with time windows”, *Transportation Science*, 27, 298-311.

Vaughan, R.J. (1984) “Approximate formula for average distances associated with zones”, *Transportation Science*, 18, 231-244.

Vaughan, R.J. and Cousins, E.A. (1977) “Optimum location of stops on a bus route”, Proceedings 7th International Symposium Transportation Traffic Theory, 697-719.

Verblunsky, S. (1951) “On the shortest path through a number of points” Proc. Amer. Math. Soc., 6, 904-913.

Wirasinghe, S.C. and Ghoneim, N.S.A. (1981) “Spacing of bus-stops for many-to-many travel demand”, *Transportation Science*, 15, 210-221.

Zhao, J. and Dessouky, M. (2004) “Optimal service capacity for a single-bus mobility allowance shuttle transit (MAST) system”, submitted for publication.