
Volume 185, Issue 2, 1 March 2008
ISSN 0377-2217
ELSEVIER
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
Available online at
ScienceDirect
www.sciencedirect.com
www.elsevier.com/locate/ejor

Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 185 (2008) 481–494
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

Mobility allowance shuttle transit (MAST) services: MIP formulation and strengthening with logic constraints

Luca Quadrifoglio [a,*], Maged M. Dessouky [b,1], Fernando Ordóñez [b,1]

[a] *Zachry Department of Civil Engineering, Texas A&M University, College Station, TX 77843, USA*
[b] *Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA 90089-0193, USA*

Received 6 April 2006; accepted 13 December 2006
Available online 14 January 2007

Abstract

We study a hybrid transportation system referred to as mobility allowance shuttle transit (MAST) where vehicles may deviate from a fixed path consisting of a few mandatory checkpoints to serve demand distributed within a proper service area. In this paper we propose a mixed integer programming (MIP) formulation for the static scheduling problem of a MAST type system. Since the problem is NP-Hard, we develop sets of logic cuts, by using reasonable assumptions on passengers' behavior. The purpose of these constraints is to speed up the search for optimality by removing inefficient solutions from the original feasible region. Experiments show the effectiveness of the developed inequalities, achieving a reduction up to 90% of the CPU solving time for some of the instances.

Keywords: Transportation; Integer programming; Constraints satisfaction

1. Introduction

We study a hybrid transportation system referred to as the mobility allowance shuttle transit (MAST) where vehicles may deviate from a fixed path consisting of a few mandatory checkpoints to serve demand distributed within a proper service area. A MAST system is described by a set of vehicles driving along a base fixed-route and serving a specific geographic area. The base route can be laid out around a loop or between two terminals. Vehicles must stop at a set of checkpoints along the main path. The checkpoints are conveniently located at major transfer points or high density demand zones, are relatively far from each other and have fixed departure times. Given a proper amount of slack time, vehicles are allowed to deviate from the fixed path to serve (pick-up and/or drop-off) customers at their desired locations, as long as they are within a service area.

The idea behind a MAST system is to combine the flexibility of demand responsive transit (DRT) systems with the low cost operability of fixed-route systems and tries to fulfill the recent goals of transit agencies, which are seeking ways to increase their service flexibility in a cost efficient way. A small scale version of such a system has been tested in

* Corresponding author. Tel.: +1 979 458 4171.
E-mail addresses: lquadrifoglio@civil.tamu.edu (L. Quadrifoglio), maged@usc.edu (M.M. Dessouky).
[1] Tel.: +1 213 740 4891.

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2006.12.030

Los Angeles County for one nighttime bus line servicing mostly night-shift employees of local firms. The vehicle moves back and forth several times between two terminals stopping at one additional checkpoint in the middle of the route and it is allowed to deviate within half a mile from either side of the main route.

MAST systems have only recently been approached by researchers. Quadrifoglio et al. (2007) developed a customized insertion heuristic scheduling algorithm to handle a large amount of demand dynamically. Continuing work is presented in Quadrifoglio and Dessouky (2007), where the authors evaluated the sensitivity to the shape of the service area of the effectiveness of the above mentioned heuristic. Zhao and Dessouky (2007) studied the optimal service capacity of a MAST system through a stochastic approach. Quadrifoglio et al. (2006) employed continuous approximations to evaluate the performance of MAST systems and help in their design phase.

Some work approached hybrid systems in which different vehicles perform the fixed and variable portions. Aldaihani et al. (2004) developed a continuous approximation model for designing such a service. Scheduling heuristics based on a hybrid system include the decision support system of Liaw et al. (1996), the insertion heuristic of Hickman and Blume (2000) and the tabu heuristic of Aldaihani and Dessouky (2003). Another work studying a combination of fixed and flexible service can be found in Cortés and Jayakrishnan (2002).

Other types of hybrid transportation systems have been studied by a few researchers. The work of Daganzo (1984) describes a checkpoint DRT system that combines the characteristics of both fixed route and door-to-door service. 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. However, the MAST system conceptually differs from it, since it allows also for door-to-door requests. Malucelli et al. (1999) provide a general overview of flexible transportation systems. Crainic et al. (2001) incorporate the hybrid fixed and flexible concept in a more general network setting, providing also a mathematical formulation.

MAST systems can be considered as a special case of the pickup and delivery problem (PDP) and can be formulated as mixed integer programs (MIP). There has been a significant amount of research on the PDP. Savelsbergh and Sol (1995), Desaulniers et al. (2000) and Cordeau and Laporte (2003) provide comprehensive reviews on PDP systems, examining mathematical formulations and solutions approaches presented by different authors. More recently, a branch-and-cut algorithm to solve the single vehicle PDP without capacity constraints is described in Lu and Dessouky (2004). Other optimization algorithms for different variants of the PDP include the work of Psaraftis (1980), Psaraftis (1983), Kalantari et al. (1985), Desrosiers et al. (1986), Fischetti and Toth (1989), Dumas et al. (1991), and Ruland and Rodin (1997). While PDP systems focus strictly on point-to-point transport services, the hybrid characteristics of the MAST service add 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.

In this paper, we propose a MIP formulation of the single-vehicle MAST scheduling problem and we develop sets of "logic cuts" based on realistic assumptions on passenger behavior. We test and demonstrate their effectiveness for a variety of demand scenarios by solving to optimality some sets of problems using CPLEX 9.0.

The reminder of this paper is structured as follows. In Section 2, we develop the basic formulation of a MAST system. In Section 3, we present the logic constraints. Section 4 describes the experimental results. Finally, we provide the conclusions in Section 5.

2. Formulation

The MAST system considered consists of a single vehicle, initially associated with a predefined schedule along a fixed-route consisting of C checkpoints identified by $c = 1, 2, \dots, C$; 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 moves back and forth between 1 and C. A trip 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 trips. Since the end-terminal of a trip r corresponds to the start-terminal of the following trip $r + 1$, the total number of stops at the checkpoints is $\mathrm{TC} = (C - 1) \times R + 1$. Hence, the initial schedule's array is represented by an ordered sequence of stops $s = 1, \dots, \mathrm{TC}$ and their scheduled departure times are assumed to be constraints on the system which cannot be violated.

The service area is represented by a rectangular region defined by $L \times W$, 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 Fig. 1).

Each checkpoint c is scheduled to be visited by the vehicle R times. Note that for terminal checkpoints $c = 1$ and $c = C$ the ending checkpoint of a trip r coincides with the starting checkpoint of the following trip $r + 1$.

The demand is defined by a set of requests. Each 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 the service area. 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

All customers but the PD requests need a booking process to use the service. While checkpoints are identified by $i = 1, \ldots, \mathrm{TC}$, non-checkpoint requests (NP or ND) are identified by $i = \mathrm{TC} + 1, \ldots, \mathrm{TS}$, where TS represents the total number of stops.

In this paper, we consider a static scenario in which all the demand is known in advance. We also assume one customer per request, no vehicle capacity constraint and a deterministic environment.

We define the following notation for the system:

- $R =$ number of trips
- $\mathrm{RD} = \{1, \ldots, R\} =$ set of trips
- $C =$ number of checkpoints
- $\mathrm{TC} = (C - 1) \times R + 1 =$ total number of stops at the checkpoints in the schedule
- $N_0 = \{1, \ldots, \mathrm{TC}\} =$ set of stops at the checkpoints
- $\theta_i =$ scheduled departure time of checkpoint stop $i\ \forall i \in N_0[\theta_1 = 0]$
- $K_{\mathrm{PD}} =$ set of PD requests
- $K_{\mathrm{PND}} =$ set of PND requests
- $K_{\mathrm{NPD}} =$ set of NPD requests
- $K_{\mathrm{NPND}} =$ set of NPND requests
- $K_{\mathrm{HYB}} = K_{\mathrm{PND}} \cup K_{\mathrm{NPD}} =$ set of hybrid requests (PND and NPD types)
- $K = K_{\mathrm{PD}} \cup K_{\mathrm{HYB}} \cup K_{\mathrm{NPND}} =$ set of all requests
- $\tau_k =$ ready time of request $k\ \forall k \in K$
- $\mathrm{TS} = \mathrm{TC} + |K_{\mathrm{PND}}| + |K_{\mathrm{NPD}}| + 2 \times |K_{\mathrm{NPND}}| =$ total number of stops
- $N_n = \{\mathrm{TC} + 1, \ldots, \mathrm{TS}\} =$ set of non-checkpoint stops
- $N = N_0 \cup N_n =$ set of all stops
- $\delta_{i,j} =$ rectilinear travel time between i and j $\forall i,j \in N$
- $b_i =$ service time for boardings and disembarkments at stop $i\ \forall i \in N/\{1\}$
- $A =$ set of all arcs in the network

PD requests are guaranteed to be served at their chosen service checkpoints identified by their index $i \in N_0$, since we assume no capacity constraint on the vehicle. NPND requests have their own stops identified by their index $i \in N_n$, which will be placed somewhere in the schedule. We therefore identify the following vectors that map pick-up

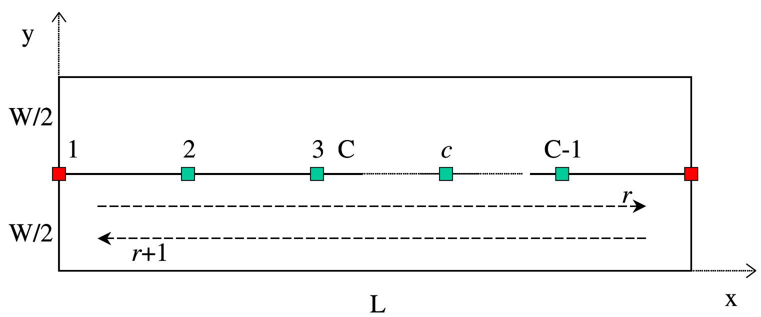

Fig. 1. MAST system.

and drop-off stops for each request (except the checkpoints of the hybrid ones):

- $\mathrm{ps}(k) \in N$ = pick-up stop of each request k $\forall k \in K/K_{\mathrm{PND}}$.
- $\mathrm{ds}(k) \in N$ = drop-off stop of each request k $\forall k \in K/K_{\mathrm{NPD}}$.

Hybrid requests (PND and NPD) instead do not have a priori a uniquely identified node in N corresponding to their checkpoint service point. In other words, each pick-up and drop-off stop of all requests uniquely corresponds to a node in N, with the exception of the pick-up stop of PND requests and the drop-off stop of the NPD requests. In fact, these can be associated to a number of occurrences of their chosen checkpoint (either a P or a D), 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 (i^*) as soon as possible and dropped off at the checkpoint $c = 4$ in the first trip $r = 1$. It could occur that, because of lack of slack time due to other requests, the NP stop i^* cannot be placed in the schedule before $c = 4$ during the first trip. As a result, the customer will have to be dropped off at a successive occurrence of $c = 4$ in the schedule. A similar example could be developed for PND requests. Thus we have:

- $\mathrm{pc}(k,r) \in N_0$ = collection of all the occurrences in the schedule (one for each $r \in \mathrm{RD}$) of the pick-up checkpoint of each request k $\forall k \in K_{\mathrm{PND}}$.
- $\mathrm{dc}(k,r) \in N_0$ = collection of all the occurrences in the schedule (one for each $r \in \mathrm{RD}$) of the drop-off checkpoint of each request k $\forall k \in K_{\mathrm{NPD}}$.

The *variables* of the system are the following:

- $x_{i,j} = \{0,1\}$ $\forall (i,j) \in A$ = binary variables indicating if an arc (i,j) is used ($x_{i,j} = 1$) or not ($x_{i,j} = 0$).
- t_i = departure time from stop i $\forall i \in N$.
- $\bar{t}_i$ = arrival time at stop i $\forall i \in N/\{1\}$.
- p_k = pick-up time of request k $\forall k \in K$.
- d_k = drop-off time of request k $\forall k \in K$.
- $z_{k,r} = \{0,1\}$ = binary variable indicating whether the checkpoint stop of the hybrid request k (a pick-up if $k \in K_{\mathrm{PND}}$ or a drop-off if $k \in K_{\mathrm{NPD}}$) is scheduled in trip r $\forall r \in \mathrm{RD}$.

The problem can now be formulated as a mixed integer linear program, where ω_1, ω_2 and ω_3 are proper weights:

$$\min \ \omega_1\left(\sum_{(i,j)\in A}\delta_{i,j}x_{i,j}\right)+\omega_2\left(\sum_{k\in K}(d_k-P_k)\right)+\omega_3\left(\sum_{k\in K}(p_k-\tau_k)\right) \tag{1}$$

subject to

$$\sum_i x_{i,j}=1 \quad \forall j\in N/\{1\}, \tag{2}$$

$$\sum_j x_{i,j}=1 \quad \forall i\in N/\{\mathrm{TC}\}, \tag{3}$$

$$t_i=\theta_i \quad \forall i\in N_0, \tag{4}$$

$$p_k=t_{\mathrm{ps}(k)} \quad \forall k\in K/K_{\mathrm{PND}}, \tag{5}$$

$$d_k=\bar{t}_{\mathrm{ds}(k)} \quad \forall k\in K/K_{\mathrm{NPD}}, \tag{6}$$

$$\sum_{r\in \mathrm{HYBR}(k)} z_{k,r}=1 \quad \forall k\in K_{\mathrm{HYB}}, \tag{7}$$

$$p_k\geqslant t_{\mathrm{pc}(k,r)}-M(1-z_{k,r}) \quad \forall k\in K_{\mathrm{PND}}, \forall r\in \mathrm{RD}, \tag{8}$$

$$p_k\leqslant t_{\mathrm{pc}(k,r)}+M(1-z_{k,r}) \quad \forall k\in K_{\mathrm{PND}}, \forall r\in \mathrm{RD}, \tag{9}$$

$$d_k\geqslant \bar{t}_{\mathrm{dc}(k,r)}-M(1-z_{k,r}) \quad \forall k\in K_{\mathrm{NPD}}, \forall r\in \mathrm{RD}, \tag{10}$$

$$d_k\leqslant \bar{t}_{\mathrm{dc}(k,r)}+M(1-z_{k,r}) \quad \forall k\in K_{\mathrm{NPD}}, \forall r\in \mathrm{RD}, \tag{11}$$

$$p_k\geqslant \tau_k \quad \forall k\in K, \tag{12}$$

$$d_k> p_k \quad \forall k\in K, \tag{13}$$

$$\bar{t}_j\geqslant t_i+x_{i,j}\delta_{i,j}-M(1-x_{i,j}) \quad \forall (i,j)\in A, \tag{14}$$

$$t_i\geqslant \bar{t}_i+b_i \quad \forall i\in N/\{1\}. \tag{15}$$

The objective function (1) 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 the 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 (2) and (3) allow each stop (except nodes 1 and TC) to have exactly one incoming arc and one outgoing arc equal to 1, so that all the stops will be visited once.

Constraints (4) force the departure times from each checkpoint to be fixed, since they are prescheduled like in a fixed-route line.

Constraints (5) establish for each request (except the PND) the equality between the pick-up time and the departure time of its corresponding node.

Similarly, constraints (6) establish for each request (except the NPD) the equality between the drop-off time and the arrival time of its corresponding node.

Constraints (7) allow exactly one z variable to be equal to 1 for each hybrid request, assuring that a unique ride will be selected for their pick-up or drop-off checkpoint.

Constraints (8) and (9) fix the value of the p_k variables for each request $k \in K_{\mathrm{PND}}$, depending on the variable z chosen. Constraints (10) and (11) do the same for the d_k variables for each request $k \in K_{\mathrm{NPD}}$. We let M represent a number large enough to cause the constraints to become irrelevant when $z_{k,r} = 0$. An $M = \theta_{\mathrm{TC}} - \theta_1$ is big enough to serve this purpose.

Constraints (12) prevent the departure times of each customer from being earlier than its ready time. Constraints (13) are the precedence constraints for each request: pick-up must be scheduled before the corresponding drop-off.

Constraints (14) are the key constraint in the formulation. They define 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 $M = \theta_{\mathrm{TC}} - \theta_1$ is large enough to be effective) assures that for any $x_{i,j} = 0$ the constraints become irrelevant. By using time stamps, these constraints guarantee that every feasible solution does not contain inner loops, but a single path from node 1 to node TC. Thus, they serve as subtour elimination constraints and they are similar to the Miller-Tucker-Zemlin (MTZ) constraints. Constraints (15) make sure that at each node the departure time is always bigger than the arrival time plus the service time.

The problem is a special case of the pick-up and delivery problem (PDP) that is known to be NP-Hard. The above formulation is sufficient to find the optimal solution (if it exists) of a given instance of the MAST problem. However, the CPU time to reach optimality can be greatly reduced by removing unnecessary binary variables and especially by adding logic constraints. The elimination of evidently infeasible arcs to reduce the size of the problem has been performed, but it is not shown here for brevity. In Section 3 instead we define and describe the logic constraints.

3. Logic constraints

The above formulation is sufficient to find the optimal solution of the problem, but it is ineffective in the sense that it includes many feasible inefficient solutions and thus has a weak LP relaxation. The purpose of this section is to identify inequalities linking together some of the variables to reduce the feasible region identified by constraints (2)–(15) and possibly speed up the search for optimality. The challenge is to make sure that these new constraints are legitimate and will only remove feasible but not optimal solutions from the problem.

A way to speed up the search for optimality and be able to solve larger instances in a reasonable time is to "tighten" the model by adding constraints ("cuts") to the formulation. Legitimate cuts should never cause the optimal solution to change; their purpose is to help solvers to reach optimality faster.

As noted by Schrijver (1986), a constraint (either equality or inequality) is classified as *valid* if it reduces the dimensions of the relaxed feasible region, but all integer feasible solutions of the original model still satisfy it. The purpose of these constraints is to reduce the size of the relaxed feasible region, ideally making it the convex hull of the integer feasible solutions which would allow an LP algorithm to solve the problem. Wolsey (1989, 2003) provide comprehensive surveys about the research on the development of effective valid constraints for MIP formulations.

Another category of constraints are the so called "logic cuts". These constraints are not *valid* because their purpose is to reduce the feasible region by eliminating some integer feasible solutions that are provably not optimal by some logic considerations. These "logic cuts" can be indeed very effective. They may significantly shrink the feasible region, even by some orders of magnitude, and they allow improving the quality of the LP relaxation bound, considerably speeding up the reduction of the optimality gap throughout the iterations of the solver. As a result, they can be extremely beneficial in reducing the CPU time in the search for optimality. However, adding too many of them can also cause the formulation to become cumbersome, forcing solvers to spend too much time while solving LP relaxation sub-problems, increasing the total CPU time. Therefore, their identification and addition to the formulation must be careful and wise, since it may not always be effective. Developments of logic constraints can be found, for example, in Andalaft et al. (2003) for forest harvesting related optimization problems. Related research has been performed earlier by Kirby et al. (1986) and Guignard et al. (1994, 1998).

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 corresponding ND or NP stop, once these are placed in the schedule. In order to prove this we need to assume $\omega_2 > \omega_3$ in the objective function (1), which implies that customers would prefer to wait for pick-up rather than to ride the vehicle. Note that the waiting time is defined as the difference between the pick-up time and the ready time ($p_k - \tau_k\ \forall k \in K$). This would generally not be 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 their time onboard instead of waiting at their pick-up stop, especially when facing bad weather conditions and/or unsafe areas. However, in a MAST system, once the schedule is done, customers know in advance the expected time for pick-up and drop-off. Thus, given that the drop-off time is fixed, they would reasonably prefer to have their scheduled pick-up times as late as possible to make their ride shorter and consequently their 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. Also, PND and PD customers would spend their waiting time at the checkpoints, most likely large comfortable and equipped stations rather than outdoor possibly unsafe bus stops.

More formally, we can state the following Proposition 1 for NPD requests, which will disembark the vehicle as early as possible after being picked up to minimize their ride time.

Proposition 1. *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. Consider a request $k \in K_{\mathrm{NPD}}$ and assume that the optimal solution, call it (I), drops off request k during trip r°, i.e. $z_{k,r^\circ} = 1$, and has ps(k) scheduled between dc$(k, r^* - 1)$ and dc(k, r^*), with $r^\circ > r^*$. The objective function can be written as $Z = \Delta + \omega_2(d_k - p_k)$, where Δ includes all the terms in Z except the ride time term of k; therefore its value would be $Z_{\mathrm{I}} = \Delta + \omega_2(\bar{t}_{\mathrm{dc}(k,r^\circ)} - p_k)$, since $d_k = \bar{t}_{\mathrm{dc}(k,r^\circ)}$ (depending on the values of the $z_{k,r}$, indicating at which occurrence of the drop-off checkpoint the customer disembarks the vehicle, d_k could be equal to $\bar{t}_{\mathrm{dc}(k,r^*)}$, $\bar{t}_{\mathrm{dc}(k,r^*+1)}$, ..., $\bar{t}_{\mathrm{dc}(k,\mathrm{R})}$, with $\bar{t}_{\mathrm{dc}(k,r^*)} < \bar{t}_{\mathrm{dc}(k,r^*+1)} < \cdots < \bar{t}_{\mathrm{dc}(k,\mathrm{R})}$). Another feasible solution (II) of the problem can be identified by setting $z_{k,r^\circ} = 0$ and $z_{k,r^*} = 1$, thus $d_k = \bar{t}_{\mathrm{dc}(k,r^*)}$, and leaving everything else unchanged (the customer would basically disembark the vehicle at an earlier occurrence of its drop-off checkpoint). Its $Z_{\mathrm{II}} = \Delta + \omega_2(\bar{t}_{\mathrm{dc}(k,r^*)} - p_k)$. Since $\bar{t}_{\mathrm{dc}(k,r^*)} < \bar{t}_{\mathrm{dc}}(k, r^\circ)$, we have $Z_{\mathrm{II}} < Z_{\mathrm{I}}$. This is a contradiction. □

In parallel, we can develop and prove the following Proposition 2 for PND requests, which will board the vehicle as late as possible, minimizing their ride time and therefore maximizing their waiting time.

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. Consider a request $k \in K_{\mathrm{PND}}$ with $\tau_k \leqslant t_{\mathrm{pc}(k,r^\circ)}$ and assume that the optimal solution, call it (I), picks up request k during trip r°, i.e. $z_{k,r^\circ} = 1$, and has ds(k) scheduled between pc(k, r^*) and pc$(k, r*+1)$, with $r^\circ < r*$. The objective function can be written as $Z = \Delta + \omega_2(d_k - p_k) + \omega_3(p_k - \tau_k)$, where Δ includes all the terms in Z except the ride time and the waiting time terms of k, and can be rearranged as $Z = \Delta + \omega_2 d_k - \omega_3\tau_k + p_k(\omega_3 - \omega_2)$; therefore its value would be $Z_{\mathrm{I}} = \Delta + \omega_2 d_k - \omega_3\tau_k + t_{\mathrm{pc}(k,r^\circ)}(\omega_3 - \omega_2)$, since $p_k = t_{\mathrm{pc}(k,r^\circ)}$ (depending on the values of the $z_{k,r}$, indicating at which occurrence of the pick-up checkpoint the customer boards the vehicle, p_k could be equal to $t_{\mathrm{pc}(k,r^\circ)}, t_{\mathrm{pc}(k,r^\circ+1)}, \ldots, t_{\mathrm{pc}(k,r^*)}$, with $t_{\mathrm{pc}(k,r^\circ)} < t_{\mathrm{pc}(k,r^\circ+1)} < \cdots < t_{\mathrm{pc}(k,r^*)}$). Another feasible solution (II) of the problem can be identified by setting $z_{k,r^\circ} = 0$ and $z_{k,r^*} = 1$, thus $p_k = t_{\mathrm{pc}(k,r^*)}$, and leaving everything else unchanged (the customer would basically board the vehicle at a later occurrence of its pick-up checkpoint). Its $Z_{\mathrm{II}} = \Delta + \omega_2 d_k - \omega_3\tau_k + t_{\mathrm{pc}(k,r^*)}(\omega_3 - \omega_2)$. Since $\omega_3 - \omega_2 < 0$ by assumption and $t_{\mathrm{pc}(k,r^\circ)} < t_{\mathrm{pc}(k,r^*)}$, we have $Z_{\mathrm{II}} < Z_{\mathrm{I}}$. This is a contradiction. □

Note that the opposite assumption on the weights ($\omega_2 < \omega_3$) would just reverse the above result, having customers getting onboard as soon as possible and we would still be able to produce logic cuts similar to the ones developed shortly.

Although the logic behind the above Propositions may seem obvious to a human mind, it is not explicitly stated in the formulation and the solver would still consider several feasible but inefficient solutions (violating the above propositions) as possible candidates while searching for optimality. Therefore, based on the above Propositions, we develop three different groups of valid inequalities to add to the formulation.

To formally develop the constraints we define the following notation:

- A_n = arcs in N_n, including all arcs (i,j) $\forall i,j \in N_n$, with $i \neq j$.
- $A_{0,n}$ = arcs from N_0 to N_n, including all arcs (i,j) $\forall i \in N_0/\{\mathrm{TC}\}$ $\forall j \in N_n$.
- $A_{n,0}$ = arcs from N_n to N_0, including all arcs (i,j) $\forall i \in N_n$, $\forall j \in N_0/\{1\}$.
- $q(i) \in K$ = corresponding request of each non-checkpoint stop i $\forall i \in N_n$.

3.1. Group #1

The first group of inequalities is developed by directly applying Propositions 1 and 2. They include constraints linking the z variables to the t variables (departure times) of non-checkpoint stops of hybrid requests and constraints linking the z variables to some of the x variables.

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

$$t_{\mathrm{ds}(k)} < z_{k,r}\theta_j + M(1 - z_{k,r}), \qquad (16)$$

with $j = \mathrm{pc}(k, r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$.

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

Symmetrically for NPD requests a legitimate set of inequalities is represented by

$$t_{\mathrm{ps}(k)} > z_{k,r}\theta_i - M(1 - z_{k,r}), \qquad (17)$$

with $i = \mathrm{dc}(k, r-1)$ $\forall k \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}/\{1\}$.

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

We can also include the following inequalities for PND requests:

$$x_{\mathrm{ds}(k),j} \leqslant z_{k,r}, \qquad (18)$$

with $\mathrm{pc}(k,r) < j \leqslant \mathrm{pc}(k, r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall (\mathrm{ds}(k), j) \in A_{n,0}$.

By Proposition 1, if $z_{k,r} = 1$, $\mathrm{ds}(k)$ must be scheduled between $\mathrm{pc}(k,r)$ and $\mathrm{pc}(k, r+1)$ and all arcs originating from $\mathrm{ds}(k)$ and ending at a checkpoint j cannot exist whenever j is not included in that interval. These arcs would in fact infeasibly require the vehicle to go from $\mathrm{ds}(k)$ to a checkpoint scheduled before its pick-up $\mathrm{pc}(k,r)$ or to skip $\mathrm{pc}(k, r+1)$ going directly from $\mathrm{ds}(k)$ to a checkpoint scheduled after $\mathrm{pc}(k, r+1)$.

Similarly we have:

$$x_{i,\mathrm{ds}(k)} \leqslant z_{k,r}, \qquad (19)$$

with $\mathrm{pc}(k,r) \leqslant i < \mathrm{pc}(k, r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall (i, \mathrm{ds}(k)) \in A_{0,n}$.

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

Symmetrically for NPD requests we have that

$$x_{i,\mathrm{ps}(k)} \leqslant z_{k,r}, \qquad (20)$$

with $\mathrm{dc}(k, r-1) \leqslant i < \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}/\{1\}$, $\forall (i, \mathrm{ps}(k)) \in A_{0,n}$.

$$x_{\mathrm{ps}(k),j} \leqslant z_{k,r}, \qquad (21)$$

with $\mathrm{dc}(k, r-1) < j \leqslant \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}/\{1\}$, $\forall (\mathrm{ps}(k), j) \in A_{n,0}$.

3.2. Group #2

A second group of inequalities includes constraints linking z and x variables by making use of Propositions 1 and 2 along with the ready times τ of the requests.

For PND requests we have that

$$\tau_{q(i)} + \delta_{i,j} + b_j \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{\mathrm{ds}(k),i}), \qquad (22)$$

with $i = \mathrm{ps}(q(i))$, $j = \mathrm{pc}(k, r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall (\mathrm{ds}(k), i) \in A_n$.

By Proposition 1, if $z_{k,r} = 1$, ds(k) must be scheduled between pc(k,r) and pc($k, r+1$) and these constraints impose that any arc originating from the ds(k) of a PND request to any non-checkpoint pick-up i is not allowed if the vehicle would not be able to reach checkpoint pc($k, r+1$) on time by passing through i, because of too high $\tau_{q(i)}$, even using the quickest way possible. The M causes these constraints to become irrelevant if either $z_{k,r}$ or $x_{\mathrm{ds}(k),i}$ are not equal to 1.

Similarly,

$$\tau_{q(i)} + (\delta_{\mathrm{ps}(q(i)),\mathrm{ds}(k)} + \delta_{\mathrm{ds}(k),i} + \delta_{i,j}) + (b_{\mathrm{ds}(k)} + b_i + b_j) \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{\mathrm{ds}(k),i}), \tag{23}$$

with $i = \mathrm{ds}(q(i))$, $j = \mathrm{pc}(k, r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall (\mathrm{ds}(k), i) \in A_n$.

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

Analogous constraints can be developed for arcs (i, ds(k)) as follows:

$$\tau_{q(i)} + (\delta_{i,\mathrm{ds}(k)} + \delta_{\mathrm{ds}(k),j}) + (b_{\mathrm{ds}(k)} + b_j) \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{i,\mathrm{ds}(k)}), \tag{24}$$

with $i = \mathrm{ps}(q(i))$, $j = \mathrm{pc}(k, r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall (i, \mathrm{ds}(k)) \in A_n$.

Table 1
System parameters, common to all cases

L	10 miles
W	1 mile
C	3
$\delta_{s,s+1}$ ($s = 1, \ldots, \mathrm{TC} - 1$)	12 minutes
b_s ($s = 1, \ldots, \mathrm{TS}$)	18 seconds
$\omega_1/\omega_2/\omega_3$	0.4/0.4/0.2

$$\tau_{q(i)} + (\delta_{\mathrm{ps}(q(i)),i} + \delta_{i,\mathrm{ds}(k)} + \delta_{\mathrm{ds}(k),j}) + (b_i + b_{\mathrm{ds}(k)} + b_j) \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{i,\mathrm{ds}(k)}), \tag{25}$$

with $i = \mathrm{ds}(q(i))$, $j = \mathrm{pc}(k, r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall (i, \mathrm{ds}(k)) \in A_n$.

For NPD requests the four constraints above can be developed likewise:

$$\tau_{q(i)} + \delta_{i,j} + b_j \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{\mathrm{ps}(k),i}), \tag{26}$$

with $i = \mathrm{ps}(q(i))$, $j = \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}$, $\forall (\mathrm{ps}(k), i) \in A_n$.

$$\tau_{q(i)} + (\delta_{\mathrm{ps}(q(i)),\mathrm{ps}(k)} + \delta_{\mathrm{ps}(k),i} + \delta_{i,j}) + (b_{\mathrm{ps}(k)} + b_i + b_j) \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{\mathrm{ps}(k),i}), \tag{27}$$

with $i = \mathrm{ds}(q(i))$, $j = \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}$, $\forall (\mathrm{ps}(k), i) \in A_n$.

$$\tau_{q(i)} + (\delta_{i,\mathrm{ps}(k)} + \delta_{\mathrm{ps}(k),j}) + (b_{\mathrm{ps}(k)} + b_j) \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{i,\mathrm{ps}(k)}), \tag{28}$$

with $i = \mathrm{ps}(q(i))$, $j = \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}$, $\forall (i, \mathrm{ps}(k)) \in A_n$.

$$\tau_{q(i)} + (\delta_{\mathrm{ps}(q(i)),i} + \delta_{i,\mathrm{ps}(k)} + \delta_{\mathrm{ps}(k),j}) + (b_i + b_{\mathrm{ps}(k)} + b_j) \leqslant z_{k,r}\theta_j + M(2 - z_{k,r} - x_{i,\mathrm{ps}(k)}), \tag{29}$$

with $i = \mathrm{ds}(q(i))$, $j = \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}$, $\forall (i, \mathrm{ps}(k)) \in A_n$.

3.3. Group #3

A third group of inequalities links z and x variables by applying the results from the Propositions to pairs of hybrid requests. We indeed know by Proposition 1 (2) 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. For any given pair of hybrid requests, the direct path connecting together their non-checkpoint stops identified by the appropriate x variable is not allowed if

Table 2
System parameters specific to each case

Parameters	Cases							
	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
$\lvert K_{\mathrm{PD}}\rvert$	1	1	1	2	1	1	1	1
$\lvert K_{\mathrm{PND}}\rvert$	2	2	5	6	1	3	5	8
$\lvert K_{\mathrm{NPD}}\rvert$	1	2	4	6	1	2	5	7
$\lvert K_{\mathrm{NPND}}\rvert$	1	1	1	2	0	1	1	1
TS	10	15	20	25	15	20	25	30

the intervals where the non-checkpoint stops are supposed to be included in, identified by the corresponding z variables, do not overlap.

Therefore, the following relationships can be written:

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ds}(k),\mathrm{ds}(h)}), \tag{30}$$

with $i = \mathrm{pc}(h,s)$, $j = \mathrm{pc}(k,r+1)$ $\forall k,h \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall s \in \mathrm{RD}$.

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ds}(h),\mathrm{ds}(k)}), \tag{31}$$

with $i = \mathrm{pc}(h,s)$, $j = \mathrm{pc}(k,r+1)$ $\forall k,h \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall s \in \mathrm{RD}$.

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ps}(k),\mathrm{ps}(h)}), \tag{32}$$

with $i = \mathrm{dc}(h,s-1)$, $j = \mathrm{dc}(k,r)$ $\forall k,h \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}$, $\forall s \in \mathrm{RD}/\{1\}$.

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ps}(h),\mathrm{ps}(k)}), \tag{33}$$

with $i = \mathrm{dc}(h,s-1)$, $j = \mathrm{dc}(k,r)$ $\forall k,h \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}$, $\forall s \in \mathrm{RD}/\{1\}$.

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ps}(k),\mathrm{ds}(h)}), \tag{34}$$

with $i = \mathrm{pc}(h,s), j = \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{NPD}}, \forall h \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}$, $\forall s \in \mathrm{RD}$.

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ds}(h),\mathrm{ps}(k)}), \tag{35}$$

with $i = \mathrm{pc}(h,s)$, $j = \mathrm{dc}(k,r)$ $\forall k \in K_{\mathrm{NPD}}$, $\forall h \in K_{\mathrm{PND}}$, $\forall r \in \mathrm{RD}$, $\forall s \in \mathrm{RD}$.

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ds}(k),\mathrm{ps}(h)}), \tag{36}$$

with $i = \mathrm{dc}(h,s-1)$, $j = \mathrm{pc}(k,r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall h \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall s \in \mathrm{RD}/\{1\}$.

$$z_{h,s}\theta_i - z_{k,r}\theta_j < M(3 - z_{h,s} - z_{k,r} - x_{\mathrm{ps}(h),\mathrm{ds}(k)}), \tag{37}$$

with $i = \mathrm{dc}(h,s-1)$, $j = \mathrm{pc}(k,r+1)$ $\forall k \in K_{\mathrm{PND}}$, $\forall h \in K_{\mathrm{NPD}}$, $\forall r \in \mathrm{RD}/\{R\}$, $\forall s \in \mathrm{RD}/\{1\}$.

Table 3
Customer type distribution of MTA line 646

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

Table 4
CPLEX runs, subset A1

Cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
Case: *A1a* TS = 10: $R = 2$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 2$; $\lvert K_{\mathrm{NPD}}\rvert = 1$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
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%
Cuts	var	bin	lin	con	sec	n	10^3 i	rel	opt	ub	lb	gap
Case: *A1b* TS = 15: $R = 4$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 2$; $\lvert K_{\mathrm{NPD}}\rvert = 2$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
None	114	79	35	146	0.16	182	1.34	101.0	141.22	–	–	0.0%
#1	109	75	34	146	0.08	23	0.31	101.1	141.22	–	–	0.0%
#2	114	79	35	174	0.16	140	0.92	101.0	141.22	–	–	0.0%
#3	114	79	35	228	0.20	189	1.56	101.0	141.22	–	–	0.0%
All	109	75	34	245	0.09	11	0.30	101.1	141.22	–	–	0.0%
Cuts	var	bin	lin	con	sec	10^3 n	10^3 i	rel	opt	ub	lb	gap
Case: *A1c* TS = 20: $R = 4$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 5$; $\lvert K_{\mathrm{NPD}}\rvert = 4$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
None	226	176	50	273	44.35	59.59	449.6	129.9	191.3	–	–	0.0%
#1	219	171	48	309	6.59	6.93	71.9	129.9	191.3	–	–	0.0%
#2	226	176	50	332	37.95	40.87	408.4	129.9	191.3	–	–	0.0%
#3	226	176	50	451	40.5	38.34	385.8	129.9	191.3	–	–	0.0%
All	219	171	48	493	5.35	4.25	54.8	129.9	191.3	–	–	0.0%
Cuts	var	bin	lin	con	sec	10^3 n	10^6 i	rel	opt	ub	lb	gap
Case: *A1d* TS = 25: $R = 4$; $\lvert K_{\mathrm{PD}}\rvert = 2$; $\lvert K_{\mathrm{PND}}\rvert = 6$; $\lvert K_{\mathrm{NPD}}\rvert = 6$; $\lvert K_{\mathrm{NPND}}\rvert = 2$												
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%

For example in constraints (30) if $z_{h,s} = 1$ and $z_{k,r} = 1$ we know that $\mathrm{ds}(h)$ must be scheduled between $\mathrm{pc}(h,s)$ and $\mathrm{pc}(h,s+1)$; similarly $\mathrm{ds}(k)$ must be scheduled between $\mathrm{pc}(k,r)$ and $\mathrm{pc}(k,r+1)$. Therefore, the direct path from $\mathrm{ds}(k)$ to $\mathrm{ds}(h)$, identified by $x_{\mathrm{ds}(k),\mathrm{ds}(h)}$, cannot be allowed if checkpoint $\mathrm{pc}(h,s)$ is not scheduled earlier than $\mathrm{pc}(k,r+1)$ 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 $x_{\mathrm{ds}(k),\mathrm{ds}(h)}$ are equal to 0.

3.4. Other constraints

We note that it would be possible to develop several other valid inequalities similar to the ones already described. Equations from (16) to (37) shrink the feasible region by rendering infeasible some 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 non-checkpoint stops and ending at j. However, the number of constraints added to the formulation would be exponentially high, most likely slowing down the solution search instead of being effective.

4. Experimental results

In this section, we evaluate the effectiveness of the groups of 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 using a 3.2 GHz CPU with 2 GB RAM. We refer to Fig. 1 for the geometry of the MAST system considered and Table 1 summarizes the assumed parameters, common to all cases and consistent with the real data of the MTA Line 646 in Los Angeles County.

We run two sets of experiments: in set A we assume a difference between the scheduled departure times of two consecutive checkpoints ($\theta_{s+1} - \theta_s$, $s = 1, \ldots, \mathrm{TC} - 1$) of 17.5 minutes; in set B we assume

Table 5
CPLEX runs, subset A2

Cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
Case: *A2a* TS = 15: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 1$; $\lvert K_{\mathrm{NPD}}\rvert = 1$; $\lvert K_{\mathrm{NPND}}\rvert = 0$												
None	68	35	33	83	0.02	9	106	80	101.1	–	–	0.0%
#1	68	35	33	106	0.01	0	59	80	101.1	–	–	0.0%
#2	68	35	33	87	0.01	9	119	80	101.1	–	–	0.0%
#3	68	35	33	101	0.02	7	103	80	101.1	–	–	0.0%
All	68	35	33	128	0.01	0	61	80	101.1	–	–	0.0%
Cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
Case: *A2b* TS = 20: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 3$; $\lvert K_{\mathrm{NPD}}\rvert = 2$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
None	129	84	45	156	0.12	191	978	126.1	164.5	–	–	0.0%
#1	129	84	45	194	0.10	17	366	126.1	164.5	–	–	0.0%
#2	129	84	45	184	0.11	142	853	126.1	164.5	–	–	0.0%
#3	129	84	45	316	0.17	188	1164	126.1	164.5	–	–	0.0%
All	129	84	45	382	0.09	10	304	126.1	164.5	–	–	0.0%
Cuts	var	bin	lin	con	sec	n	10^3 i	rel	opt	ub	lb	gap
Case: *A2c* TS = 25: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 5$; $\lvert K_{\mathrm{NPD}}\rvert = 5$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
None	287	226	61	353	41.20	27,267	392.8	162	212	–	–	0.0%
#1	284	223	61	437	2.03	893	12.0	162	212	–	–	0.0%
#2	287	226	61	435	38.72	20,315	374.5	162	212	–	–	0.0%
#3	287	226	61	739	73.96	29,313	556.7	162	212	–	–	0.0%
All	284	223	61	819	1.83	524	8.5	162	212	–	–	0.0%
Cuts	var	bin	lin	con	sec	10^6 n	10^6 i	rel	opt	ub	lb	gap
Case: *A2d* TS = 30: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 8$; $\lvert K_{\mathrm{NPD}}\rvert = 7$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
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	1377	36,000	5.1	138	186.7	?	295.3	257.8	12.7%
All	409	334	75	1428	12,273	3.7	65	186.7	293.9	–	–	0.0%

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 trips (R) compared to subset A1 (and B1). In each subset we consider 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 Table 2. The NP and ND locations are sampled from a spatial uniform distribution over the whole service area ($W \times L$); 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.

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). As mentioned in Section 1, the MAST scheduling problem can be considered as a special case of the PDP. The traditional single-vehicle PDP has been solved optimally for sizes up to about 30 nodes (Kalantari et al., 1985; Fischetti and Toth, 1989; Ruland and Rodin, 1997), which is about the same size of the MAST problems solved in this paper.

We tried to maintain the ratio between the different types of requests as close as possible to the real demand data of MTA Line 646, which has a distribution described in Table 3.

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

Table 6
CPLEX runs, subset B1

Cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
Case: *B1a* TS = 10: $R = 2$; $\lvert K_{PD}\rvert = 1$; $\lvert K_{PND}\rvert = 2$; $\lvert K_{NPD}\rvert = 1$; $\lvert K_{NPND}\rvert = 1$												
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%
Cuts	var	bin	lin	con	sec	n	10^3 i	rel	opt	ub	lb	gap
Case: *B1b* TS = 15: $R = 4$; $\lvert K_{PD}\rvert = 1$; $\lvert K_{PND}\rvert = 2$; $\lvert K_{NPD}\rvert = 2$; $\lvert K_{NPND}\rvert = 1$												
None	124	89	35	156	0.56	695	7.91	105.8	164.9	–	–	0.0%
#1	123	88	35	199	0.19	126	1.39	105.8	164.9	–	–	0.0%
#2	124	89	35	188	0.50	643	5.46	105.8	164.9	–	–	0.0%
#3	124	89	35	256	0.62	815	7.25	105.8	164.9	–	–	0.0%
All	123	88	35	309	0.25	89	1.55	105.8	164.9	–	–	0.0%
Cuts	var	bin	lin	con	sec	10^3 n	10^6 i	rel	opt	ub	lb	gap
Case: *B1c* TS = 20: $R = 4$; $\lvert K_{PD}\rvert = 1$; $\lvert K_{PND}\rvert = 5$; $\lvert K_{NPD}\rvert = 4$; $\lvert K_{NPND}\rvert = 1$												
None	247	197	50	299	619.0	723.3	5.58	132.8	217.8	–	–	0.0%
#1	244	195	49	351	49.0	60.7	0.47	132.8	217.8	–	–	0.0%
#2	247	197	50	400	355.7	319.9	3.33	132.8	217.8	–	–	0.0%
#3	247	197	50	639	508.1	460.2	4.03	132.8	217.8	–	–	0.0%
All	244	195	49	742	32.0	27.2	0.31	132.8	217.8	–	–	0.0%
Cuts	var	bin	lin	con	sec	10^6 n	10^6 i	rel	opt	ub	lb	gap
Case: *B1d* TS = 25: $R = 4$; $\lvert K_{PD}\rvert = 2$; $\lvert K_{PND}\rvert = 6$; $\lvert K_{NPD}\rvert = 6$; $\lvert K_{NPND}\rvert = 2$												
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%

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 complete results of one instance of each case for subset A1, A2, B1 and B2 are shown in Tables 4–7, respectively.

The results in the tables show that cuts "#1" are the most effective, followed by "#2" and then by "#3", which are efficient in roughly half of the cases (compared to the "none" runs). The synergistic effect of grouping them all the cuts together ("all" runs) is beneficial in most cases; however, in some cases, adding cuts "#1" alone to the formulation is still the best choice. The improvement due to the logic cuts can be observed in any instance, reaching a reduction of CPU time up to 90% or more in some of them compared to the "none" runs (see cases "c"). The larger problem size cases "d" do not always reach optimality but the effect of cuts can be noted by looking at the smaller "gap" values, which are tightened because of better lower bounds "lb". In case A2d, the optimality gap is still 6.6% after 10 CPU hours with the original formulation ("none"); cuts "#2" and "#3" are not effective; yet optimality is reached after about 3 CPU hours with cuts "#1" or "all". We also note that the relaxed optimal values ("rel") are about the same in each run for each case; this means that the cuts do not improve the initial value of the lower optimality bound, but they are effective in speeding up the rise of it throughout the iterations.

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 solution run time is consistently larger in all instances. For example in case A1d CPLEX is able to reach the optimal solution in each run relatively fast, while in case B1d CPLEX could 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.

The significant results show how effective the methodology can be. The original MIP formulation

Table 7
CPLEX runs, subset B2

Cuts	var	bin	lin	con	sec	n	i	rel	opt	ub	lb	gap
Case: *B2a* TS = 15: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 1$; $\lvert K_{\mathrm{NPD}}\rvert = 1$; $\lvert K_{\mathrm{NPND}}\rvert = 0$												
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%
Cuts	var	bin	lin	con	sec	10^3 n	10^3 i	rel	opt	ub	lb	gap
Case: *B2b* TS = 20: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 3$; $\lvert K_{\mathrm{NPD}}\rvert = 2$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
None	172	127	45	205	2.76	5.93	35.04	139.1	190.9	–	–	0.0%
#1	172	127	45	261	0.47	0.54	4.46	139.1	190.9	–	–	0.0%
#2	172	127	45	243	1.99	3.47	24.98	139.1	190.9	–	–	0.0%
#3	172	127	45	365	2.16	2.93	22.86	139.1	190.9	–	–	0.0%
All	172	127	45	459	0.96	1.15	9.99	139.1	190.9	–	–	0.0%
Cuts	var	bin	lin	con	sec	10^3 n	10^6 i	rel	opt	ub	lb	gap
Case: *B2c* TS = 25: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 5$; $\lvert K_{\mathrm{NPD}}\rvert = 5$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
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	1004	1007	501	6.57	143.5	222.1	–	–	0.0%
All	327	266	61	1329	51	27	0.46	143.5	222.1	–	–	0.0%
Cuts	var	bin	lin	con	sec	10^6 n	10^6 i	rel	opt	ub	lb	gap
Case: *B2d* TS = 30: $R = 6$; $\lvert K_{\mathrm{PD}}\rvert = 1$; $\lvert K_{\mathrm{PND}}\rvert = 8$; $\lvert K_{\mathrm{NPD}}\rvert = 7$; $\lvert K_{\mathrm{NPND}}\rvert = 1$												
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	1826	36,000	4.5	136	196.6	?	333.2	270.8	18.7%
All	566	490	76	2157	36,000	7.1	150	196.6	?	332.8	305.6	8.2%

is enough to fully represent the MAST scheduling problem and find an optimum for any given instance. However, "complicating" the model by adding logic constraints can be extremely effective to guide solvers in finding optimality faster, which could be crucial for NP-Hard problems. This would suggest applying the methodology for more complicated MAST systems (multiple-vehicle and/or MAST networks).

5. Conclusions

In this paper, we propose a mixed integer programming (MIP) formulation of the static scheduling problem of a mobility allowance shuttle transit (MAST) system, a hybrid transit solution combining fixed and flexible types of services. Since it is a NP-Hard problem, we develop sets of "logic cuts" based on reasonable assumptions on passengers' behavior and whose purpose is to remove inefficient and therefore uninteresting solutions from the feasible region to speed up the search for optimality.

Experimental results on several instances show the effectiveness of the cuts, which are able to reduce the CPU solution time by up to more than 90% for some cases. Specifically, cuts "#1" provide the best overall results that always effective, followed in general by cuts "#2" and cuts "#3", which are not always effective. The synergistic effect of including all the cuts together further reduces the CPU solution time in many cases.

Future research may consider developing a solution algorithm which would efficiently "add and lift" the logic constraints in the formulation throughout the iterations and possibly reduce the CPU solution time even more. In addition, the same methodology could be used to further strengthen the MAST scheduling optimization problem formulation by looking for different logic constraints.

Acknowledgment

The research reported in this paper was partially supported by the National Science Foundation under Grant NSF/USDOT-0231665.

References

Aldaihani, M.M., Dessouky, M., 2003. Hybrid scheduling methods for paratransit operations. Computers & Industrial Engineering 45, 75–96.

Aldaihani, M.M., Quadrifoglio, L., Dessouky, M.M., Hall, R.W., 2004. Network design for a grid hybrid transit service. Transportation Research A 38, 511–530.

Andalaft, N., Andalaft, P., Guignard, M., Magendzo, A., Wainer, A., Weintraub, A., 2003. A problem of forest harvesting and road building solved through model strengthening and Lagrangean relaxation. Operations Research 51, 613–628.

Cordeau, J.F., Laporte, G., 2003. The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. 4OR 1 (2), 89–101.

Cortés, C.E., Jayakrishnan, R., 2002. Design and operational concepts of a high coverage point-to-point transit system. Transportation Research Record 1783, 178–187.

Crainic, T.G., Malucelli, F., 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., 1984. Checkpoint dial-a-ride systems. Transportation Research B 18, 315–327.

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., 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 Science 6, 301–325.

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

Fischetti, M., Toth, P., 1989. An additive bounding procedure for combinatorial optimization problems. Operations Research 37, 319–328.

Guignard, M., Ryu, C., Spielberg, K., 1994. Model tightening for integrated timber harvest and transportation planning. In: Proceedings of the International Symposium Systems Analysis Management Decisions Forestry, Valdivia, Chile, pp. 364–369.

Guignard, M., Ryu, C., Spielberg, K., 1998. Model tightening for integrated timber harvest and transportation planning. European Journal of Operational Research 111, 448–460.

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

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.

Kirby, M.W., Hager, W., Wong, P., 1986. Simultaneous planning of wildland transportation alternatives. TIMS Studies in the Management Sciences 21, 371–387.

Liaw, C.F., White, C.C., 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.

Lu, Q., Dessouky, M.M., 2004. An exact algorithm for the multiple vehicle pickup and delivery problem. Transportation Science 38, 503–514.

Malucelli, F., Nonato, M., Pallottino, S., 1999. Demand adaptive systems: Some proposals on flexible transit. In: Ciriania, T., Johnson, E., Tadei, R. (Eds.), Operations Research in Industry, pp. 157–182.

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

Quadrifoglio, L., Dessouky, M.M., 2007. Sensitivity analyses over the service area for mobility allowance shuttle transit (MAST) services. Lecture Notes in Economics and Mathematical Systems (in press).

Quadrifoglio, L., Hall, R.W., Dessouky, M.M., 2006. Performance and design of mobility allowance shuttle transit (MAST) services: Bounds on the maximum longitudinal velocity. Transportation Science 40 (3), 351–363.

Quadrifoglio, L., Dessouky, M.M., Palmer, K., 2007. An insertion heuristic for scheduling mobility allowance shuttle transit (MAST) services. Journal of Scheduling 10, 25–40.

Ruland, K.S., Rodin, E.Y., 1997. The pickup and delivery problem: Faces and branch-and-cut algorithm. Computers & Mathematics with Applications 33 (12), 1–13.

Savelsbergh, M.W.P., Sol, M., 1995. The general pickup and delivery problem. Transportation Science 29, 17–29.

Schrijver, A., 1986. Theory of linear and integer programmingWiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, New York.

Wolsey, L.A., 1989. Strong formulations for mixed integer programming: A survey. Mathematical Programming 45, 173–191.

Wolsey, L.A., 2003. Strong formulations for mixed integer programs: Valid inequalities and extended formulations. Mathematical Programming 97, 423–447.

Zhao, J., Dessouky, M.M., 2007. Optimal service capacity for a single bus mobility allowance shuttle transit (MAST) system, submitted for publication.