Valid Inequalities for the Synchronization Bus Timetabling Problem

Omar J. Ibarra-Rojas
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
omore@yalma.fime.uanl.mx

P. Fouilhoux
Laboratoire d’ Informatique de Paris 6
Université Pierre et Marie Curie
Pierre.Fouilhoux@lip6.fr

S. Kedad-Sidhoum
Laboratoire d’ Informatique de Paris 6
Université Pierre et Marie Curie
sidhoum@lip6.fr

Yasmin A. Rios-Solis
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
yasmin@yalma.fime.uanl.mx

October 2012

1Corresponding author
Abstract

Bus transit network planning is a complex process that is divided into several subproblems such as: line planning, timetable generation, vehicle scheduling, and crew scheduling. Timetabling generation is the stage where the departure times for all trips of each line are scheduled. In this work, we focus on the Synchronization Bus Timetabling problem (SBT) that allows to maximize passenger transfers, avoids bus bunching, and includes flexibility that is crucial for many transit networks in Latin America. A timetabling solution that is not close to the optimum has strong repercussions in the vehicle and crew scheduling problems since a sequential resolution approaches are often required for solving the entire planning process. Therefore, we develop five families of valid inequalities that drastically strengthen the integer linear programming model of SBT. Three of them manage to bound the number of synchronizations while the other two are lifting ones. Experimental results show that the enhanced mixed integer linear programming yields high quality solutions (less than 1% of relative deviation from the optimal solution) in minutes for large instances based on real transit networks (200 bus lines and 40 synchronization nodes).

Keywords: urban transit network; bus timetabling; synchronization; mixed integer linear programming; valid inequalities.
1 Introduction

Planning and operating a public transportation network in a relatively big city (3 millions people) is an enormous task if the aims are to reduce costs without reducing the quality service for the users. Bus transit network problem consists of four main subproblems (Ceder, 2007). First one is the line planning problem that defines the routes, stops, and frequency for each bus line in a specific territory. Then, the timetable generation determines the departure times (and arrival times) of all the trips of the lines to achieve a quality service level such as maximum synchronization to allow transfer events and avoid bus bunching events for different lines (this is our case of study). Third subproblem is the vehicle scheduling problem that assigns vehicles to sets of trips of each bus line with the aim of minimizing the number of vehicles or vehicle costs. Finally, the crew scheduling problem defines tasks assigned to drivers subject to work regulation constraints (e.g., work time limit and lunch time for drivers) with the objective of minimizing crew costs.

Bus transit network planning problem is commonly tackled by sequential approaches. Therefore, obtaining high quality solutions in short time for each one of the subproblems is an important issue since it is often necessarily to iterate several times to find a suitable solution for the entire planning process. Particularly, timetable generation is a delicate task since it has repercussions in the principal operational subproblems; the vehicle and crew scheduling.

We are interested in the Synchronization Bus Timetabling problem (SBT) since it represents the characteristics of our case of study: the bus network of Monterrey, Mexico, which is similar to many Latin American cities. These characteristics are the following. The bus network is owned by several private companies. In total, it has more than 300 bus lines. The timetable is only for the company’s administration. Passengers do not know at what time a bus arrives at each stop instead, they only have an estimate of their waiting time. Then, Almost evenly spaced departure times for the trips of each line are seek. An important proportion of the lines pass near or by downtown. Thus, transfers must be favored in this area. The case where different lines converge on a specific node is frequent. Avoiding bus bunching between these lines is needed to improve the quality service. Finally,

Considering the characteristics of our case of study, an interesting objective not only for companies but also for passengers is maximizing the synchronization of different lines to allow passenger transfers and avoid bus bunching along the transit network.

Related studies such as Ceder (2007), Ceder and Tal (2001), and Ceder (2011) address the bus timetabling problem of maximizing the number of synchronizations which are defined as simultaneous arrivals at common nodes subject to headway bounds (a headway is the separation time between consecutive trips for the same line). Constructive algorithms...
were developed for this case. Later, an extension of Ceder’s studies is presented by Eranki (2004) where a synchronization is redefined as arrivals within a time window. In Liu et al. (2007), the authors reformulate the timetabling problem presented in Ceder et al. (2001) as a special case of the uncapacitated knapsack problem and implement a Tabu search to solve their formulation. Ávila and López (2012) propose to integrate the frequencies computation together with the synchronization timetabling generation in a multiobjective approach. Finally, in Ibarra-Rojas and Rios-Solis (2012), a mixed integer linear programming (MILP) is presented for our case of study, i.e., the NP-hard Synchronization Bus Timetabling Problem. The authors show that real size instances cannot be efficiently solved with a branch-and-bound algorithm (B&B) so they propose a multistart Iterated Local Search algorithm.

In bus networks as our case of study, timetabling is not anymore part of the strategical planning process but it is part of the operational one. Indeed, it must be recomputed several times a day since the number of disruptions (bus failures, crash accidents, and so on) is close to 10% of the vehicles used per day. Additionally, the absenteeism of the drivers is high affecting the whole planning process. With this in mind, the only way to operate this kind of network is by modifying the timetables frequently in an efficient way and with the best possible quality of the solutions. Although most of related timetabling problems in literature are solved by heuristics, we present an exact approach to solve BTP using integer programming techniques.

Integer Programming has been used to solve transport networks problems. For example, Fügenschuh (2009) presents the scholar bus scheduling problem, where starting times of schools and starting times of scholar buses must be synchronized to minimize the number of vehicles to transport all students. They develop different families of valid inequalities leading to a branch-and-cut algorithm. Another example is Quadrifoglio et al. (2008) that present a problem to optimize a weighted objective function based on vehicle resources and quality service for a transport network where deviation in routes of buses is allowed to serve more demand. They define logic constraints to reduce the feasible space which allows to reduce 90% of cpu time adding these logic cuts at the begging of a B&B algorithm.

Most of the exact approaches for timetabling problems are for the periodic case of railway systems since they are based on the Periodic Event Scheduling Problem (PESP). The PESP structure allows to define valid inequalities to generate cutting plane approaches (Caimi et al., 2011; Giesemann, 2002; Liebchen, 2004; 2007; Liebchen and Möhring, 2002, 2007). To the best of our knowledge, most of the bus timetabling problems different than the periodic case are not solved using exact approaches (an excellent review can be found in Guihaire and Hao (2008)). One exception is the work of Schröder and Solchenbach (2006) that implements a B&B algorithm to solve the timetabling problem with the objective of minimize the waiting time cost subject to limited deviation from an initial timetabling.
The problem is represented by a quadratic semi-assignment formulation. The relative small size of the real life instances (twelve lines and four transfer nodes) allows to obtain optimal solutions using a B&B.

Our main contributions are to define five families of valid inequalities that are added to the SBT MILP to tighten its continuous relaxation. The first three families of valid inequalities bound the number of possible synchronizations that can occur at a node for a specific trip. The other two families of valid inequalities are of the lifting type and we build two algorithms based on a preprocessing to generate these inequalities. Additionally, the SBT MILP has big $M$ parameters that we manage to tighten. We solve the enhanced MILP with a B&B algorithm finding optimal solutions for most of the real size instances. Moreover, the B&B converges to solutions with less than 3% of gap in less than five minutes.

The rest of this paper is organized as follows. In Section 1.1 we briefly summarize the SBT MILP formulation that we will enhance with our proposed valid inequalities that are defined in Section 2. The SBT MILP has big $M$ parameters that are tighten in Section 4. Experimental results for instances based in a real transit network are presented in Section 5 where we show the impact of each combination of the valid inequalities families. Finally, conclusions and future research areas are addressed in Section 6.

1.1 Synchronization Bus Timetabling Problem

The synchronization bus timetabling problem aims to achieve passenger transfers and avoid bus bunching at specific nodes in the transit network subject to almost even headways. A headway is the separation time between consecutive trips for the same line. For example, if first and second trip of some line depart at minutes 3 and 23, respectively, there is a headway of 20 minutes between these two trips.

Synchronization events can be defined by a separation of arrival times of two trips at some node. Specifically, large separation times between arrivals of trips at bus bunching nodes or small separation times between arrivals of trips at transfers nodes. Figure 1 shows two synchronizations. Case (a) represents a bunching node where lines $i$ and $j$ converges and then, they share a segment of their routes. The numbers by the side of node $b$ represent the arrival times of a pair of trips of these lines. Arrival separation between trips of lines $i$ and $j$ must be at least of 5 minutes to avoid bus bunching. Case (b) represents a transfer node $b'$ where passengers would like to go from one trip of line $i$ to some trip of line $j$. A separation of 3 minutes between arrivals represents the desired waiting time of passengers that make this transfer.

Because of uncertainties in travel times, passengers prefer flexible transfers than punctual ones. Therefore, a synchronization event is defined if the separation time between
arrivals of two trips at some node is between a given time window. For example, instead of consider a separation time of 3 min between arrivals of two trips to define a synchronization (case (b) of Figure 1) we can consider an arrival separation between 2 and 6 minutes.

Demand behavior is also variable along the day so there are different planning periods to achieve more accurate deterministic data. Particularly, planning periods can be classified as rush hours (more demand) and valley hours (less demand) for different types of days such as working day, weekend, Christmas, holiday, rainy day, and so on.

The type of bus network we are interested in can be represented by a set of lines denoted as $I$ and a set of synchronization nodes denoted as $B$. For each line $i \in I$, there exists a set of lines $J(i)$ that share synchronization nodes with line $i$. Set $B_{ij}$ represents all the synchronization nodes for the pair of lines $(i, j \in J(i))$. On the other hand, the parameters for the integer linear program of SBT are the following.

- $T$: Planning period in minutes. For example, 2 hours in rush periods in the morning or 3 hours in valley periods in the afternoon.
- $f^i$: Frequency (number of trips) of line $i \in I$.
- $t^i_b$: Travel time of line $i$ from initial node (depot) to node $b$.
- $[w_b, W_b]$: Waiting time window to define a synchronization at node $b$.
- $h^i$ and $H^i$: Minimum and maximum headways for line $i$, respectively. These headway times are defined as $h^i = \frac{T}{f^i} - \delta^i$ and $H^i = \frac{T}{f^i} + \delta^i$, where $\delta^i$ is a given headway amplitude factor. Parameter $\delta^i$ gives flexibility to timetables but it also assures almost even headways for each line.
The main decisions in SBT are represented by $X^i_p$: the departure time of trip $p \in \{1, \ldots, f^i\}$ of line $i \in I$. Moreover, a synchronization happens if departure times of two trips leads to separation of arrival times within a time window at a synchronization node. This is represented with the following decision variables.

$$Y^i_{pq} = \begin{cases} 1 & \text{if the } p\text{th trip of line } i \text{ arrives first at node } b, \text{ and synchronizes with the } q\text{th trip of line } j, \\ 0 & \text{otherwise.} \end{cases}$$

Considering the previous parameters and decision variables, the mixed integer linear program for SBT is given by

$$\max F_{SBT}(Y) = \sum_{i \in I} \sum_{j \in J(i)} \sum_{b \in B^{ij}} f^i f^j \sum_{p=1}^{f^i} \sum_{q=1}^{f^j} Y^i_{pq}$$

s.t.

$$X^i_1 \leq H^i \quad \forall i \in I$$

$$T - H^i \leq X^i_{f^i} \leq T \quad \forall i \in I$$

$$h^i \leq X^i_{p+1} - X^i_p \leq H^i \quad \forall i \in I, \ p = 1, \ldots, f^i - 1$$

$$\left( X^i_q + t^i_b \right) - \left( X^i_p + t^i_b \right) \geq w_b + M \left( 1 - Y^i_{pq} \right) \quad \forall i \in I, \ j \in J(i), \ b \in B^{ij}, \ p = 1, \ldots, f^i, \ q = 1, \ldots, f^j$$

$$\left( X^i_q + t^i_b \right) - \left( X^i_p + t^i_b \right) \leq W_b + M \left( 1 - Y^i_{pq} \right) \quad \forall i \in I, \ j \in J(i), \ b \in B^{ij}, \ p = 1, \ldots, f^i, \ q = 1, \ldots, f^j$$

$$X^i_p \in \mathbb{R}, Y^i_{pq} \in \{0,1\} \quad \forall i \in I, \ j \in J(i), \ b \in B^{ij}, \ p = 1, \ldots, f^i, \ q = 1, \ldots, f^j$$

The objective function maximizes the number of synchronizations. Constraints (1) and (2) force to first trip and last trip of each line to depart near the beginning and near the end of the planning period $T$, respectively. Constraints (3) separate consecutive trips of line $i$ by an amount of time between the minimum ($h^i$) and the maximum ($H^i$) headways. Constraints (4) and (5) activate the synchronization variables $Y^i_{pq}$ if the difference between arrivals of the $q$th trip of line $j$ and the $p$th trip of line $i$ at node $b$ is within $[w_b, W_b]$ where $M$ is a very large constant (which is bounded in section 4).

Notice that variables $Y^i_{pq}$ define oriented synchronizations, i.e., $Y^i_{pq} = 1$ represents the passengers transfer from trip $p$ of line $i$ to trip $q$ of line $j$ but not vice versa. Moreover, constraints (2) together with (1) guarantee that the entire planning period is covered by the trips even if uneven headways are allowed. These are some of the differences between the models studied in Ceder and Tal (2001) and in Eranki (2004). As it can be seen in
Section 5, the SBT formulation as it is, is intractable when solved by a B&B. While most of the timetabling problems are solved by heuristic algorithms, our main contributions are five families of valid inequalities that drastically improve the MILP so real size instances can be solved within minutes with a B&B. We introduce these valid inequalities in the following two sections.

2 Valid Inequalities based on headways

An alternative to obtain tighter formulations for optimization problems is using valid inequalities to cut fractional solutions of the linear relaxations of integer programs or cut non-optimal feasible solutions ([Wolsey 1998](#Wolsey:1998) [Nemhauser and Wolsey 1999](#Nemhauser:1999)). Our exact solution methodology adds different families of valid inequalities to SBT MILP to obtain a tighter formulation (strengthening phase). Then, this improved formulation is solved by a B&B (solving phase) to obtain a high quality solution (optimal for most of our instances as it is shown in Section 5). In the following, we introduce three families of valid inequalities for the synchronization bus timetabling problem obtained by headway parameters and the propagation of headway constraints.

2.1 Knapsack Inequalities

Using a logical deduction process, we can take advantage of the headway parameters to define a family of valid inequalities for each trip to be synchronized. For instance, consider two lines \(i\) and \(j\) to be synchronized at some node \(b\) such that the minimum headway time \(h^j\) of line \(j\) is greater than the length of the waiting time window of node \(b\), i.e., \(h^j > W_b - w_b\). In this case, if some trip \(p\) of line \(i\) (denoted as \((p, i)\)) synchronizes with another trip \((q, j)\), the synchronization of \((p, i)\) with trips \(q - 1\) or \(q + 1\) of line \(j\) is impossible since there would not be enough time. Figure 2 shows this case, where \(a_{pb}^i\) and \(a_{qb}^j\) represent the arrival times of trips \((p, i)\) and \((q, j)\) at node \(b\), respectively, and \(S^i_{pb}\) represents the synchronization time window \([a_{pb}^i + w_b, a_{qb}^j + W_b]\) of trip \((p, i)\). Notice that it does not matter the arrival time of trip \((q, j)\) at node \(b\), the arrival time of trips \((q - 1, j)\) and \((q + 1, j)\) at node \(b\) are always out of the synchronization window \(S^i_{pb}\) of trip \(p\) of line \(i\).

In the basis of the above, synchronization variables \(Y_{ijpq}^{p(q-1)b}\) and \(Y_{ijpq}^{p(q+1)b}\) are zero if \(Y_{ijpq}^{pqb} = 1\). Moreover, synchronization variables \(Y_{ijpq}^{q'qb}\) are 0 if variable \(Y_{ijpq}^{pqb} = 1\) for all \(q' \neq q\).

Therefore, \(\sum_{q=1}^{f} Y_{ijpq}^{pqb} \leq 1\) is a valid inequality for each trip \((p, i), j \in J(i)\), and synchronization node \(b \in B_{ij}\) such that \(h^j > W_b - w_b\). Generalizing the previous idea, we can obtain different sets of valid inequalities related to the possible number of synchronizations between each trip \(p\) of some line \(i\) with another line \(j\) or vice versa, i.e., inequalities related to the possible
number of synchronizations between each trip \( q \) of some line \( j \) with another line \( i \). We define these “knapsack inequalities” as follows.

\[
\sum_{q=1}^{f_j} Y_{pqb}^{ij} \leq 1 + \frac{W_b - w_b}{h^j} \quad \forall i \in I, j \in J(i), b \in B^{ij}, p = 1, \ldots, f^i \quad (7)
\]

\[
\sum_{p=1}^{f^i} Y_{pqb}^{ij} \leq 1 + \frac{W_b - w_b}{h^i} \quad \forall i \in I, j \in J(i), b \in B^{ij}, q = 1 \ldots f^j \quad (8)
\]

Figure \( \ref{fig:sub-matrix} \) shows the sub-matrix of synchronization variables related with a pair of lines \((i, j) \in J(i)\) to be synchronized at some node \( b \in B^{ij} \). The marked area in the left and right panels show the decision variables of the left side of constraints \((7)\) and \((8)\) for trips \( p = 2 \) and \( q = 2 \), respectively. Notice that the decision variables of knapsack inequalities are rows and columns of the related sub-matrix of synchronization variables. Moreover, quantity \( 1 + \left\lfloor \frac{W_b - w_b}{h^j} \right\rfloor \) of \((7)\) represents the maximum number of trips of line \( j \) that could arrive within the synchronization window \( S_{pb}^i \), i.e, the maximum number of synchronization variables that could be one in each row. Similarly, quantity \( 1 + \left\lfloor \frac{W_b - w_b}{h^i} \right\rfloor \) represents the maximum number of synchronization variables that could be one in each column. Theorem 1 proofs the validity of the previous ideas.

**Theorem 1.** Knapsack inequalities \((7)\) and \((8)\) are valid for SBT MILP.

**Proof.** To prove that inequalities \((7)\) are valid, we suppose there exists a feasible solution of SBT such that

\[
\sum_{q=1}^{f_j} Y_{pqb}^{ij} = r > 1 + \left\lfloor \frac{W_b - w_b}{h^j} \right\rfloor
\]

for some lines \( i, j \in J(i), \) node \( b \in B^{ij}, \) and trip \( p \). This means, that there exists \( r \) trips \( q_1 < q_2 < \cdots < q_r \) of line \( j \) that may synchronize with trip \((p, i)\). Thus, arrival times of these trips are within the feasible synchronization time window \( S_{pb}^i = [a_{pb}^i + w_b, a_{pb}^i + W_b] \)
Figure 3: Variables present in knapsack inequalities for some pair of lines \((i, j \in J(i))\) and synchronization node \(b \in B^j\). Marked variables in the left panel are the decisions variables related to inequalities (7) for trip \(p = 2\). Right panel illustrates the variables related to inequalities (8) for trip \(q = 2\).

thus, \(a_{q_1b}^j - a_{q_1b}^j \leq W_b - w_b\). By headway constraints (3) of the SBT MILP, the minimum separation time between arrival time of trips \((q_1, j)\) and \((q_r, j)\) is \((r - 1)h^j\). Then,

\[
a_{q_rb}^j - a_{q_1b}^j \geq (r - 1)h^j \geq 1 + \left[\frac{W_b - w_b}{h^j}\right] h^j \geq \left(1 + \left(\frac{W_b - w_b}{h^j} - 1\right)\right) h^j = W_b - w_b
\]

which is a contradiction. \(\square\)

The proof of valid inequalities (8) is analogous to the proof of (7). Notice that the number of these inequalities is of order \(O(|Y|)\), where \(Y\) represent the set of synchronization variables. In Section 5 we show that these inequalities are quite efficient to tighten dual bounds for SBT MILP.

2.2 Half-Cross Inequalities

Another family of valid inequalities can be obtained by combining the knapsack inequalities of SBT. To achieve this, we remark that the sum of (7) and (8) for some lines \(i, j \in J(i)\), node \(b\), trips \(p\), and \(q\) leads to the following valid inequality.

\[
2Y_{pqb}^{ij} + \sum_{q' \neq q} Y_{pq'b}^{ij} + \sum_{p' \neq p} Y_{p'q'b}^{ij} \leq \text{row}_{\text{sync}}_{b}^{ij} + \text{col}_{\text{sync}}_{b}^{ij}
\]

Where \(\text{row}_{\text{sync}}_{b}^{ij} = 1 + \left[\frac{W_b - w_b}{h^j}\right]\) (maximum number of synchronizations in each row) and \(\text{col}_{\text{sync}}_{b}^{ij} = 1 + \left[\frac{W_b - w_b}{h^i}\right]\) (maximum number of synchronizations in each column). Notice that the decision variables of these inequality are represented by a cross in the related sub-matrix of the synchronization variables for pair of lines \((i, j \in J(i))\) and node \(b\) (see Figure 4).
Figure 4: Sub-matrix of synchronization variables related to lines $i, j \in J(i)$, and node $b \in B^{ij}$. The marked area show the decisions variables related to inequalities (9) for trips $p = 2$ and $q = 2$.

By (7) and (8) and making $Y_{ijpq}^{ij} = 1$ for some node $b$, and trips $p$ and $q$, inequality (9) is equivalent to

$$Y_{ijpq}^{ij} + \sum_{q' > q} Y_{ijpq'}^{ij} + \sum_{p' > p} Y_{ijp'q}^{ij} \leq \text{row}_{\text{sync}}^{ij}_b + \text{col}_{\text{sync}}^{ij}_qb - 1 \quad (10)$$

On the other hand, $Y_{ijpq}^{ij} = 0$ implies that inequality (10) is not valid. However, we identify a subset of decisions variables of (10) that satisfy this bound. These variables are used to define a new family of valid inequalities named as “half-cross” due their shape in the matrix of synchronization variables. The definition of half-cross inequalities is the following.

$$Y_{ijpq}^{ij} + \sum_{q' > q} Y_{ijpq'}^{ij} + \sum_{p' > p} Y_{ijp'q}^{ij} \leq \text{row}_{\text{sync}}^{ij}_b + \text{col}_{\text{sync}}^{ij}_qb - 1 \quad \forall \ i \in I, \ j \in J(i), \ b \in B^{ij} \quad (11)$$

$$Y_{ijpq}^{ij} + \sum_{q' < q} Y_{ijpq'}^{ij} + \sum_{p' < p} Y_{ijp'q}^{ij} \leq \text{row}_{\text{sync}}^{ij}_b + \text{col}_{\text{sync}}^{ij}_qb - 1 \quad \forall \ i \in I, \ j \in J(i), \ b \in B^{ij} \quad (12)$$

The marked area in the left and right panels of Figure 5 show the decision variables of inequalities (11) and (12) for the synchronization variable $Y_{22b}^{ij}$. Notice that decision variables of these half-cross inequalities can be seen as the variable $Y_{ijpq}^{ij}$ plus segments of row $p = 2$ and column $q = 2$.

Indeed, as we shown in the next theorem, these half cross inequalities can be bounded based on the bounds previously obtained for knapsack constraints.

**Theorem 2.** Half-cross inequalities (11) and (12) are valid for SBT.

**Proof.** Let us prove that inequalities (11) are valid. Suppose there is a feasible solution that
and (8) we have

\[ Y_{ij} \] violates \((11)\) for some trips \((p, i), (q, j), \) and node \(b \in B^{ij} \). Then, we have

\[ Y_{pqb}^{ij} + \sum_{q' > q} Y_{pq'b}^{ij} + \sum_{p' > p} Y_{p'qb}^{ij} > \text{row-sync}_{ij}^{qb} + \text{col-sync}_{ij}^{qb} - 1. \]

As we mentioned before, in the case that \( Y_{pqb}^{ij} = 1 \) the proof is trivial by knapsack inequalities. Therefore, we only consider the case of \( Y_{pqb}^{ij} = 0 \). By knapsack inequalities (7) and (8) we have \( \sum_{q' > q} Y_{pq'b}^{ij} = \text{row-sync}_{ij}^{qb} \) and \( \sum_{p' > p} Y_{p'qb}^{ij} = \text{col-sync}_{ij}^{qb} \). This means that there are \( \text{row-sync}_{ij}^{qb} \) trips of line \( j \) that synchronize with trip \((p, i)\) and \( \text{col-sync}_{ij}^{qb} \) trips of line \( i \) that synchronize with trip \((q, j)\).

Using the minimum headway times and waiting time windows, we can conclude that the latest arrival time \( a^{i}_{(q+1)b} \) of trip \((q + 1, j)\) at node \(b\) happens when the \( \text{row-sync}_{ij}^{qb} \) (\( r \) for simplicity) trips of line \( j \) synchronized with trip \((p, i)\) are \( q + 1, q + 2, \ldots, q + r \) and these trips are separated by their minimum headway time \( h^{j} \). Figure 6 shows this case. We can notice that a separation of more than \( h^{j} \) minutes for trips of line \( j \), and synchronization of trip \((p, i)\) with posterior trips than \((q + r, j)\) leads to an earlier arrival time \( a^{i}_{(q+1)b} \) at node \(b\). Thus, the latest arrival time of \( a^{i}_{(q+1)b} \) is given by \( a^{i}_{pb} + W_{b} - (r - 1)h^{j} \).

Analogously, the earliest arrival time \( a^{j}_{qb} \) of trip \((q, b)\) at node \(b\) happens when trip \((q, j)\) synchronizes with \( \text{col-sync}_{ij}^{qb} \) (\( c \) for simplicity) trips of line \( i \). These trips are \( p + 1, p + 2, \ldots, p + c \) since \( Y_{pqb}^{ij} = 0 \). Therefore, \( a^{j}_{qb} \in S_{(p+1)b}^{i} \), i.e., the earliest arrival time \( a^{j}_{qb} \) is given by \( a^{i}_{(p+1)b} + w_{b} \). Figure 7 shows this case. Then, \( a^{i}_{(p+1)b} + w_{b} + h^{j} \leq a^{i}_{pb} + W_{b} - (n - 1)h^{j} \) since \( a^{j}_{qb} + h^{j} \leq a^{i}_{(p+1)b} \).

Replacing \( r \) and simplifying the previous expression we have that

\[
a^{i}_{(p+1)b} \leq a^{i}_{pb} + W_{b} - w_{b} - h^{j} - \left[ \frac{W_{b} - w_{b}}{h^{j}} \right] h^{j} < a^{i}_{pb} + W_{b} - w_{b} - h^{j} - \left( \frac{W_{b} - w_{b}}{h^{j}} - 1 \right) h^{j} = a^{i}_{pb} \]

Figure 5: Left and right panels show the decisions variables related to half-cross inequalities \((11)\) and \((12)\), respectively for synchronization variable \( Y_{22b}^{ij} \).
Figure 6: Latest arrival time $a^j_{(q+1)b}$ of trip $(q+1,j)$ at node $b$ considering that there are $r$ trips of line $j$ that synchronize with trip $(p,i)$.

Figure 7: Earliest arrival time $a^j_{qb}$ of trip $(q,j)$ at node $b$ considering that this trip synchronizes with trip $(p+1,i)$.

which is a contradiction.

The proof for inequalities (12) is straight forward using the same idea as proof of (11). Notice that the right side of half-cross inequalities are not too different compared with the right side of (9) obtained by the sum of knapsack inequalities of SBTP. The importance of half-cross inequalities is that we can use tighter bounds for the variables of the half-cross inequalities (11) and (12) if the minimum headway parameters $h^i$ and $h^j$ are at least one minute. We define these “advanced half-cross inequalities” for all trips $(p,i)$, $(q,j)$ with $j \in J(i)$, and node $b \in B^{ij}$ as follows.

\[
Y^i_{pj} + \sum_{q' > q} Y^ij_{p'q'b} + \sum_{p' > p} Y^ij_{pq'b} \leq 1 + \max \left\{ \left[ \frac{W_b - w_b}{h^j} \right], \left[ \frac{W_b - w_b}{h^i} \right] \right\} \tag{13}
\]

\[
Y^i_{pjb} + \sum_{q' < q} Y^ij_{pq'b} + \sum_{p' < p} Y^ij_{p'qb} \leq 1 + \max \left\{ \left[ \frac{W_b - w_b}{h^j} \right], \left[ \frac{W_b - w_b}{h^i} \right] \right\} \tag{14}
\]

Indeed for practical cases, the minimum headway parameters are greater than two minutes as points out Ceder (2007). Next, we present a theorem to show that advanced half-cross inequalities are valid.

**Theorem 3.** Advanced half-cross inequalities (13) and (14) are valid if minimum headway
times are at least one minute.

Proof. Suppose there is a feasible solution that violates (13) for some trips \((p, i), (q, j)\), and node \(b \in B^{ij}\). In the case that \(Y_{pqb}^{ij} = 0\) we have

\[
Y_{pqb}^{ij} + \sum_{q' > q} Y_{pq'b}^{ij} + \sum_{p' > p} Y_{qpb}^{ij} = r + c > \max \left\{ \text{row}_{sb}^{ij}, \text{col}_{sb}^{ij} \right\}
\]

where \(\sum_{q' > q} Y_{pq'b}^{ij} = r\) and \(\sum_{p' > p} Y_{qpb}^{ij} = c\). Then, there are \(r\) trips of line \(j\) that synchronize with trip \((p, i)\) and there are \(c\) trips of line \(i\) that synchronize with trip \((q, j)\).

As we did for proof of theorem 2 (see Figure 6), we can conclude that the latest arrival time \(a_{(q+1)b}^{ij}\) of trip \((q+1, j)\) at node \(b\) could happen when the \(r\) trips of line \(j\) synchronized with trip \((p, i)\) are \(q + 1, q + 2, \ldots, q + r\) and these trips are separated by their minimum headway time \(h^j\). Thus, the latest arrival time of \(a_{(q+1)b}^{ij}\) is given by \(a_{pb}^{ij} + W_b - (r - 1)h^j\).

Similarly, the earliest arrival time \(a_{qb}^{ij}\) of trip \((q, j)\) at node \(b\) happens when the \(c\) trips of line \(i\) synchronized with trip \((q, j)\) are \(p + 1, p + 2, \ldots, p + c\) and these trips are separated by their minimum headway time \(h^i\). Figure 8 shows this case where we can notice that in any other case a separation of more than \(h^i\) minutes for trips of line \(i\), and synchronization of trip \((q, j)\) with posterior trips of \((p + c, i)\) leads to a later arrival time of trip \((q, j)\) at node \(b\). Thus, the earliest arrival time of \(a_{qb}^{ij}\) is given by \(a_{(p+1)b}^{ij} + W_b + (c - 1)h^i\).

![Figure 8: Earliest arrival time \(a_{qb}^{ij}\) of trip \((q, j)\) at node \(b\) considering that this trip synchronizes with \(c\) trips of line \(i\).](image)

By headway constraints (3) of the MIP formulation of SBT, arrival times of trips \((p, i)\), \((p + 1, i)\), \((q, j)\), and \((q + 1, j)\) satisfy \(a_{pb}^{ij} + h^i \leq a_{(p+1)b}^{ij}\) and \(a_{qb}^{ij} + h^j \leq a_{(q+1)b}^{ij}\). In the basis of the above, we have that

\[
(a_{pb}^{ij} + h^i) + w_b + (c - 1)h^i + h^j \leq a_{(p+1)b}^{ij} + w_b + (c - 1)h^i + h^j \leq a_{pb}^{ij} + W_b - (r - 1)h^j.
\]

Thus \(a_{pb}^{ij} + h^i + w_b + (c - 1)h^i + h^j \leq a_{pb}^{ij} + W_b - (r - 1)h^j\). Simplifying the previous expression we have that \(rh^j + ch^i \leq W_b - w_b\), i.e., \(r + ch^i \leq \frac{W_b - w_b}{h^j}\).

Considering that headway times \(h^i\) and \(h^j\) are at least one minute and (13) is violated,
\[ r + c \leq r + c h^i \leq \frac{W_b - w_b}{h} \leq \max \left\{ row\_sync^{ij}_b, col\_sync^{ij}_b \right\} - 1 < r + c - 1 \]

which is a contradiction. The case of \( Y_{pqb}^{ij} = 1 \) is straightforward.

The proof for inequalities (14) is analogous to the proof of (13). The families of valid inequalities (11), (12), (13), and (14) are related to the number of synchronizations for a specific pair of lines. As it happens with the knapsack constraints, the number of half-cross inequalities is of order \( O(|Y|) \) for each one of them.

There exists another type of valid inequalities that define a relation between synchronization variables for different related pair of lines \((i, j) \in J(i)\) and \((i', j') \in J(i')\). These inequalities are developed in the next section.

### 3 Lifting inequalities for SBT

Several methodologies such as lifting procedures (for more details see Nemhauser and Wolsey (1999)) are used to obtain different families of valid inequalities. The lifting inequalities are defined for a subset of the original feasible space and then, they are “lifted” to the original feasible set. For example, consider a set \( S \subset B^n \). Given \( \delta \in \{0, 1\} \), we define the following subset of the feasible space \( S^\delta = S \cap \{x \in B^n : x_1 = \delta\} \). Assume we have the following inequality

\[ \sum_{j=2}^{n} \pi_j x_j \leq \pi_0 \quad (15) \]

If (15) is valid for \( S^0 \), set \( S^1 \neq \emptyset \), and we consider a parameter \( \zeta \geq \max \{ \sum_{j=2}^{n} \pi_j x_j : x \in S^1 \} \), the following is a valid inequality for \( S \).

\[ \sum_{j=2}^{n} \pi_j x_j \leq \pi_0 (1 - x_1) + \zeta x_1 \quad (16) \]

This procedure can be used to generate different valid inequalities for SBT by fixing each synchronization variable, and considering a bound for the rest of the synchronization variables as it happens with inequality (16).

Ibarra-Rojas and Rios-Solis (2012) propagate constraints (1)-(3) of SBT MILP to define feasible departure, arrival, and synchronization time windows leading to a identification of synchronization variables that are impossible to be one. This idea is used by the authors to define a preprocessing stage (Theorem 4) that removes the impossible synchronizations...
and their related constraints (since they are redundant) from the SBT MILP. Although the preprocessing stage improves the B&B performance for small instances, there is not a remarkable difference using this preprocessing for large instances. However, these time windows can be also used to identify impossible synchronizations given a new feasible set. For example, the feasible set obtained by fixing a synchronization variable $Y_{ijpq}^k$ to one. Therefore, we could obtain bounds for some of the synchronization variables in terms of the value of a specific synchronization variable and define lifting inequalities for SBT. First of all, we recall the definition of the feasible departure, arrival, and synchronization time windows.

The feasible departure time windows for each trip $p$ of line $i$ denoted as $D_p^i$ is defined with the propagation of constraints (1)-(3) as follows.

$$D_p^i = \left[ \max \left\{ (p-1)h^i, T - \left( f^i - (p-1) \right) H^i \right\}, \min \left\{ pH^i, T - \left( f^i - p \right) h^i \right\} \right].$$  \hspace{1cm} (17)

Figure 9 shows an example of the departure time window construction for trip $p=8$ of line $i$. This example has a planning period of $T=30$ minutes, a frequency of $f^i=10$, and headway parameters $h^i=2$ and $H^i=4$. Figure 9 has four time lines. First line shows the earliest departure time $7h^i=14$ for the eighth trip, assuming that first trip departs at its earliest time $X_{i1}^i = 0$. Second time line shows the latest departure time $\min \{T, 8H^i\} = 30$ for the eighth trip, assuming that first trip departs at its latest time $X_{i1}^i = H^i = 4$. Third line shows the earliest departure time $\max \{0, 30-3H^i\} = 18$ for the eighth trip, assuming that the last trip departs at its earliest time $X_{jf_i}^i = T - H^i = 26$. Finally, fourth line shows the latest departure time $T - 2h^i = 26$ of the eighth trip, assuming that the last trip departs at its latest time $X_{jf_i}^i = T$. Therefore, the intersection of the earliest and latest departure times $[\max \{14, 18\}, \min \{30, 26\}]$ results in the feasible departure time window $D_8^i$ (see marked area of Figure 9).

Arrival time windows for each trip $(p, i)$ at node $b$ denoted as $A_p^i b$ are obtained by shifting $D_p^i$ by $t_b$ time units. Therefore,

$$A_p^i b = \left[ \left\lfloor \left\lceil left(D_p^i) + t_b \right\rceil \right\rfloor, \left\lfloor \left\lceil right(D_p^i) + t_b \right\rceil \right\rfloor \right],$$

where $left(D_p^i)$ and $right(D_p^i)$ represent lower and upper limit of time window $D_p^i$, respectively. Similarly, the synchronization time window $S_p^i b$ for each trip $(p, i)$ and node $b$ is defined as

$$S_p^i b = \left[ \left\lfloor \left\lceil left(A_p^i) + w_b \right\rceil \right\rfloor, \left\lfloor \left\lceil right(A_p^i) \right\rceil + W_b \right\rfloor \right].$$

The following theorem shows how we can identify impossible synchronizations.

**Theorem 4.** For any trips $p$ and $q$ of lines $i$ and $j \in J(i)$, respectively, and any synchro-
3.1 Fix-sync inequalities

Now, we define a procedure to obtain lifting inequalities for SBT which are called “fix-sync inequalities”. The general idea to obtain these inequalities is to consider a synchronization variable $Y_{ijk}^q$ (denoted now as $y$ for simplicity in notation) and see what implications arise if $y = 1$. Once the value of this variable is assumed to be one, we compute the new feasible departure time windows $D_i^k$ and $D_j^q$ for trips of lines $i$ and $j$, respectively. These updated departure time windows affect other synchronization variables of lines that synchronize with lines $i$ and $j$. By Theorem 4 there exists a set of synchronization variables $E_y$, such that every variable in $E_y$ must be zero due to feasibility, i.e., $E_y$ would be the set of impossible synchronization assuming $y = 1$. Therefore, the following is a valid inequality.

$$\sum_{y' \in E_y} y' \leq M(1 - y) \quad \forall y$$

where $M$ is an upper bound of $\sum_{y' \in E_y} y'$. It is not necessary to prove that inequalities (18) are valid since by their definition, a violation of these constraints implies a feasibility violation of Theorem 4.

To determine bounds for the big $M$ parameters is by itself an optimization problem since we need to maximize the number of synchronization in SBT subject to $y = 1$. Instead of solving another optimization problem, we can define inequalities for subsets of $E_y$ with known bounds. To achieve this, let $E_{y'}$ be the synchronization variables of the left side

Figure 9: Feasible departure time window $D_k^i$ for the eighth trip of line $i$ corresponding to an instance with a planning period $T = 30$ minutes, $f^i = 10$, and $\delta_i = 0.33\%$ (Ibarra-Rojas and Rios-Solis 2012).

synchronization node $b \in B^{ij}$ of SBT, $S_{pb}^i \cap A_{qb}^j = \emptyset$, if and only if, $Y_{ijk}^{pq}$ is forced to be zero due feasibility, and constraints (4) and (5) related to this indexes are redundant.
of the knapsack inequality \[ \sum_{q' = 1}^{t'} Y_{p'q'v'} \leq 1 + \left\lceil \frac{W_{v' - w_{p'}}}{h_{p'}} \right\rceil \] for some pair of lines \((i', j' \in J(i'))\) affected by making \(y = 1\) \((\{i', j'\} \cap \{i, j\} \neq \emptyset)\), node \(b' \in B_{i'j'}\), and trip \(p'\). Then, the following inequality is valid for \(E_{p'}^y = E^y \cap E_{p'}\) since \(E_{p'}^y \subset E^y\) and \(E_{p'}^y \subset E_{p'}\).

\[ \sum_{y' \in E_{p'}^y} y' \leq \left( 1 + \left\lceil \frac{W_{b'} - w_{p'}}{h_{p'}} \right\rceil \right) (1 - y) \] (19)

Analogously, we define the following fix-sync inequalities:

\[ \sum_{y' \in E_{p'}^y} y' \leq \left( 1 + \left\lceil \frac{W_{b'} - w_{p'}}{h_{p'}} \right\rceil \right) (1 - y) \] (20)

Algorithm 1 shows the steps to generate the fix-sync inequalities (19). The first step, is to compute the feasible departure time windows \(\overline{D}_p^i\) and \(\overline{D}_q^j\) for trips \((p, i)\) and \((q, j)\), respectively, assuming that they are synchronized at node \(b\), i.e., \(Y_{pqb} = 1\) (steps 1 and 4). Then, we consider the feasible departure times windows of trips \((p, i)\) and \((q, j)\) to propagate constraints (3) and update the departure time windows for the rest of the trips \(i\) and \(j\) (step 5). For example, if \(p' > p\) \((p' < p\), respectively), \(\overline{D}_p^i\) of trip \((p', i)\) is given by the intersection of the original departure time window \(D_p^i\) and

\[
\left(\overline{\text{left}}(\overline{D}_p^i) + (p' - p)h^i, \overline{\text{right}}(\overline{D}_p^i) + (p' - p)H^i\right)
\]

\[
\left(\overline{\text{left}}(\overline{D}_p^i) - (p - p')H^i, \overline{\text{right}}(\overline{D}_p^i) - (p - p')h^i\right).
\]

Then, we identify different pairs of lines \((i', j' \in J(i'))\) affected by the modification of lines \(i\) and \(j\) (steps 6 and 7). Next, we create a valid inequality for each trip \(p'\) of line \(i'\) (steps 8 and 9). To define this inequality, we identify the possible synchronizations of trips \(p'\) of line \(i'\) with line \(j'\) that become impossible after updating the feasible solution space of the departure times (steps 10, 11, and 12). We include these impossible synchronization variables to the left side of the inequality related to trip \(p'\) (step 13). Finally, we add fix-sync inequality of type (19) (step 17).

As we did for fix-sync inequalities, we can combine the variables of each set \(E^y\) of impossible synchronization by forcing \(y = 1\) with “advanced half-cross” inequalities (13) and (14) to obtain new ones (named “fix-sync1”). If we consider advanced half-cross inequalities (13), we define \(E_{p'q'v'}\) as the synchronization variables of the left side of inequality \(Y_{p'q'v'} + \sum_{q'' > q'} Y_{p''q'v'} + \sum_{p'' > p'} Y_{p''q'v'} \leq 1 + \max \left\{ \left\lceil \frac{W_{v' - w_{p'}}}{h_{p'}} \right\rceil, \left\lceil \frac{W_{v' - w_{p'}}}{h_{p'}} \right\rceil \right\} \) for some pair of lines \((i', j' \in J(i'))\) affected by making \(y = 1\), node \(b' \in B_{i'j'}\), trips \(p'\) and \(q'\). Then, the following
Algorithm 1 \textit{Lift}(SBT)

\textbf{Input:} SBT formulation

\textbf{Output:} stronger SBT formulation

1: \textbf{for} (each possible synchronization \( y = Y_{pqb}^{ij} \)) \textbf{do}
2: \quad \( y \leftarrow 1 \) and \( [\alpha, \beta] = S_{q_b}^j \cap A_{p_b}^i \)
3: \quad \( D_{q_b}^j = [\alpha - t_{i_b}^j, \beta - t_{i_b}^j] \)
4: \quad \( D_{p_b}^i = [\alpha - W_b - t_{i_b}^j, \beta - w_b - t_{i_b}^j] \cap D_{p_b}^i \)
5: \quad \text{Recalculate} \( D_{p_b}^{i'} \text{ and } D_{q_b}^{j'} \), for all trips \( p' \neq p \) and \( q' \neq q \)
6: \quad \textbf{for} \((i', j') \in J(i'), b' \in B(i', j') \) \textbf{do}
7: \quad \quad \textbf{if} \((\{i', j'\} \cap \{i, j\} \neq \emptyset \) and \((i', j', b') \neq (i, j, b)) \textbf{then}
8: \quad \quad \quad \textbf{for} \( (p' = 1 \text{ to } f_{i'}) \) \textbf{do}
9: \quad \quad \quad \quad \textbf{for} \( (q' = 1 \text{ to } f_{j'}) \) such that \( S_{p'}^{i'} \cap A_{q'}^{j'} \neq \emptyset \) \textbf{do}
10: \quad \quad \quad \quad \quad \text{Calculate new windows} \( S_{p'}^{i'} \text{ and } A_{q'}^{j'} \)
11: \quad \quad \quad \quad \quad \textbf{if} \((S_{p'}^{i'} \cap A_{q'}^{j'} = \emptyset) \textbf{ then}
12: \quad \quad \quad \quad \quad \quad E_{p'}^{y'} \leftarrow E_{p'}^{y'} \cup \{Y_{p'q'}^{i'j'}\}
13: \quad \quad \quad \quad \quad \textbf{end if}
14: \quad \quad \quad \quad \textbf{end for}
15: \quad \quad \quad \textbf{end for}
16: \quad \quad \quad \text{Add inequality} \sum_{y' \in E_{p'}^{y'}} y' \leq \left( 1 + \left[ \frac{W_{q'} - w_{q'}}{h - y'} \right] \right)(1 - y) \text{ to SBT MILP}
17: \quad \textbf{end for}
18: \quad \textbf{end if}
19: \textbf{end for}
20: \textbf{end for}
inequality is valid for $E^y_{p'q'} = E^y \cap E_{p'q'}^y$ since $E_{p'q'}^y \subseteq E^y$ and $E_{p'q'}^y \subseteq E_{p'q'}$. 

$$\sum_{y' \in E^y_{p'q'}} y' \leq \left( 1 + \max \left\{ \left\lceil \frac{W_{y'} - w_{y'}}{h_{y'}} \right\rceil, \left\lfloor \frac{W_{y'} - w_{y'}}{h_{y'}} \right\rfloor \right\} \right) (1 - y)$$

Algorithm 1 must be modified to generate the fix-sync inequalities (21). Instead of exploring the entire row $p$ for impossible synchronization (steps 8-17), we need to explore the half-cross defined by inequalities (13) or (14) for trips $(p, i)$ and $(q, j)$. In Algorithm 1 is presented the algorithm for generating the family of valid inequalities (21) using advanced half-cross inequalities (13).

Summarizing, we defined five families of valid inequalities. The number of all the inequalities is polynomial since all of them are related to the number of synchronization variables. By adding these inequalities to the SBT MILP formulation we obtain a tighter formulation. As it can be seen in Section 5, this strengthen formulation leads to improved dual bounds that soar the B&B performance.

4 Big M Parameters

An important aspect in integer programming is to compute tight big $M$ parameters to reduce the computational time of solving the linear relaxation of integer programs. In a similar way that we use feasible departure, arrival, and synchronization time windows to define lifting inequalities, we can use them to bound big $M$ parameters for constraints (4) and (5) of the SBT MILP. Particularly, consider the feasible arrival time window $A^i_{ipb}$ for each line $i$, trip $p$, and synchronization node $b$. Recall that the earliest arrival time of trip $(q, j)$ at node $b$ is $left \left( A^j_{jpb} \right)$ and the latest arrival time of trip $(p, i)$ at node $b$ is $right \left( A^i_{ipb} \right)$. Therefore, the minimum difference of arrival times between trips $(q, j)$ and $(p, i)$ at node $b$ is $right \left( A^i_{ipb} \right) - left \left( A^j_{jpb} \right)$, i.e.,

$$\left( X^j_{jpb} + t^j_{b} \right) - \left( X^i_{ipb} + t^i_{b} \right) \geq right \left( A^i_{ipb} \right) - left \left( A^j_{jpb} \right).$$

Similarly, we can remark that the maximum difference of arrival times between trips $(q, j)$ and $(p, i)$ at node $b$ satisfies the following.

$$\left( X^j_{jpb} + t^j_{b} \right) - \left( X^i_{ipb} + t^i_{b} \right) \leq right \left( A^j_{jpb} \right) - left \left( A^i_{ipb} \right)$$

In the basis of the above, we can replace $M$ by $m_{ipqb} = right \left( A^i_{ipb} \right) - left \left( A^j_{jpb} \right) - w_b$ and $M_{ipqb} = right \left( A^i_{ipb} \right) - left \left( A^j_{jpb} \right) - W_b$ for synchronization constraints (4) and (5), respectively. In this way, we define a synchronization event if $Y^i_{ipb} = 1$, i.e., arrival separation
is within \([w_b, W_b]\) or redundant constraint if \(Y_{pqb}^{ij} = 0\).

After adding the families of valid inequalities and compute the big \(M\) parameters for constraints \(4\) and \(5\) we can execute a B&B algorithm to solve our tighter SBT MILP.

5 Experimental Results

To perform the experimental analysis, we implement the B&B algorithm of CPLEX 12.3 with default options, except for the gap that is set to 0% and limited to one hour of execution time on a iMac OS X with an Intel Core 2 Duo 3.06 GHz processor and 4 GB RAM.

We use the instance benchmark of [Ibarra-Rojas and Rios-Solis (2012)] that is based on information provided by a real bus transport company. The instances size is determined by the number of lines \(|I|\), the number of synchronization nodes \(|B|\) along the network, and the headway amplitude factor \(\delta^i\) which is responsible of the size of the feasible solution space of the departure time variables. Thus, the larger the headway amplitude factor \(\delta^i\) is, the harder is the instance. The name of the instance types and their parameters are summarized in Table 1.

<table>
<thead>
<tr>
<th>Instance</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>I</td>
<td>)</td>
<td>15</td>
<td>15</td>
<td>40</td>
<td>40</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>(</td>
<td>B</td>
<td>)</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>([\delta_{min}^i, \delta_{max}^i])</td>
<td>[5,15]</td>
<td>[10,25]</td>
<td>[5,15]</td>
<td>[10,25]</td>
<td>[5,15]</td>
<td>[10,25]</td>
<td>[5,15]</td>
<td>[10,25]</td>
</tr>
</tbody>
</table>

Table 1: Instance benchmark and their parameter values ([Ibarra-Rojas and Rios-Solis 2012]).

All the instance types have the following common characteristics: a planning period of \(T = 240\) min; the frequency \(f^i\) for each line \(i\) is randomly generated between [13,18]; the travel time \(t_{ib}^i\) from depot to synchronization node \(b\) for each line \(i\) is between [20,60]; the minimum (maximum) waiting time for each synchronization node \(b\) is within [3,5] ([9,12]); finally, the number of different pairs of lines to synchronize at each node \(b\) is between 1 and 7. We randomly generate ten instances for each one of the eight instance types to analyze the algorithm performance.

By parameters definition, minimum headways \(h^i\) are greater than the length of waiting time windows \(W_b - w_b\) for all instance types. Therefore, the particular cases of the half-cross inequalities \(11\) and \(12\) are also particular cases of the advanced half-cross inequalities.
and [14]. Particularly, these inequalities are defined as follows.

\[
Y_{ij}^{pq} + \sum_{q' > q} Y_{ij}^{pq'}b + \sum_{p' > p} Y_{ij}^{pqb} \leq 1 \quad \forall \ i \in I, \ j \in J(i), \ b \in B^{ij}, \ p = 1, \ldots, f^i, \ q = 1, \ldots, f^j,
\]

\[
Y_{ij}^{pq} + \sum_{q' < q} Y_{ij}^{pq'}b + \sum_{p' < p} Y_{ij}^{pqb} \leq 1 \quad \forall \ i \in I, \ j \in J(i), \ b \in B^{ij}, \ p = 1, \ldots, f^i, \ q = 1, \ldots, f^j.
\]

As we will show in numerical results, there exists diversity in the quality of solutions for SBT using the different combination of valid inequalities. This diversity is due to the nature of the valid inequalities and their specific impact on the SBT formulation, i.e., the different combinations of valid inequalities remove fractional solutions of the linear relaxation of SBT in different ways.

Before presenting the experimental results let us show a small example to point out the impact of each one of the valid inequalities. In Figure 10 we show an example of feasible values for some synchronization variables for a small instance using the different families of valid inequalities. The matrix of numbers represent the decision variables \( Y_{ij}^{pq} \) corresponding to a specific pair of lines \((i, j)\) and synchronization node \( b \). Particularly, element on row \( p \) and column \( q \) represents the synchronization variable for trips \((p, i)\), \((q, j)\), and node \( b \).

Matrix \( W \) of Figure 10 (SBT MILP without valid inequalities) fractional values of synchronization variables define a band near the diagonal of the sub-matrix of the decision variables. We know that for all the instances generated in this study it is possible to synchronize each trip \((p, i)\) with at most one trip \((q, j)\) at some node \( b \), i.e., the sum of each row or column is at most one. Therefore, the dual bound obtained by solution shown in matrix \( W \) is not tight enough. Matrix \( K \) shows the addition of the knapsack inequalities to SBT. We can notice that the fractional values are distributed in a smaller band near the diagonal of the sub-matrix of variables and the sum of each row and column is at most one, thus obtaining a tighter dual bound.

Adding half-cross inequalities (matrix \( H \)) leads to even a small dispersion of fractional solution since these inequalities also restrict the sum of synchronization variables in row and column segments simultaneously. On the other hand, fix-sync inequalities (matrix \( F \)) are defined using knapsack inequalities. Therefore, the bound obtained is very close to the obtained using that knapsack inequalities. However, fix-sync inequalities are defined for each synchronization variable instead of each row and column as it happens with knapsack constraints. Then, the linear relaxation of SBT MILP has a large number of inequalities that may lead to a slow convergence of a B&B. The same happens with fix-sync1 inequalities.

Implementing all families of valid inequalities (matrix \( A \)) lead to the smaller band of fractional solutions but the number of valid inequalities in the BTP MILP is very large thus solving linear relaxation of BTP takes more computational resources compared to
the previous cases. This is the reason that implementing all the valid inequalities do not warranties the best results.

Summarizing Figure 10 solving the linear relaxation of SBTP MILP using our proposed valid inequalities leads to fractional solutions really close to integer values. The impact of the valid inequalities is related with the instance structure. For example, fix-sync inequalities are not very useful if there are few related pair of lines \((i,j)\) and \((i',j')\) since there would be many redundant constraints and most of these constraints would be included in knapsack and half-cross inequalities. On the other hand, if there are several related pair of lines, both fix-sync inequalities could complement knapsack and half-cross inequalities.

Next, we present numerical results of our exact approach for the eight instances types. The computational effort to generate the valid inequalities for SBT MILP is negligible (less than one second) for all type of instances. Therefore, we measure the execution time of the B&B for solving the SBT MILP using different combinations of the valid inequalities proposed in this work.

In the first block of rows (rows one to four) of Table 2 we can see that the original formulation of SBT presented is intractable using the B&B algorithm of CPLEX. A remarkable difference arises when one family of valid inequalities is added to SBTP. However, in our case of study there is not one specific type of valid inequality that leads to the best results considering both indicators gap and time.

Individually, half-cross inequalities \((/ /H/)\) give the best results. On the other hand,
fix-sync1 inequalities ( / / /F1) seems to be weaker. The best results of both fix-sync inequalities are obtained for small instances such as A1–A4. Nevertheless, the number of constraints explodes for larger instances such as A5–A8, the quality is lost, and slow convergence is noticeable.

The combinations of valid inequalities leads to high quality solutions for SBT using small execution times. Moreover, using all families of valid inequalities does not guaranty the best quality for solutions of SBT. Particularly, the combination of knapsack and half-cross inequalities ( /K/H/) leads to the best results considering gap and time for seven and six of the instance types, respectively. This observation can be illustrated in the sixth block of rows in Table 2. Another observation is that combinations ( /K/ /F) and ( /K/ /F1) could not find a small dual bound for the first instance of type A8. Therefore, mean gap is calculated based on the other nine instances of type A8.
gap 47.868% 154.749% 68.4% 174.013% 60.915% 181.143% 153.294% 288.761%  
gap dev 23.666 32.139 14.611 5.624 20.997 14.812 27.181 50.004  
time 3600 3600 3600 3600 3600 3600 3600 3600  
time dev 0.815 0.717 0.869 0.72 0.73 1.836 2.165 0.771  

Table 2: Results for benchmark instances using the B&B algorithm of CPLEX 12.3 and our proposed families of valid inequalities. Mark (*) indicates the existence of an instance which could not be solved by B&B algorithm.

As we mentioned before, fix-sync inequalities seems to be the weakest. Results in Table 3 shows the slow convergence using any type of fix-sync inequalities for large instances such as A7 and A8. On the other hand, it can be noticed that instances types A1 and A4 were easier to solve compared with the rest of instances types since optimal solution were found for most of the combinations of valid inequalities.
Table 3: Results for benchmark instances using the B&B algorithm of CPLEX 12.3 and our proposed valid inequalities.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>0%</td>
<td>0.174%</td>
<td>0.242%</td>
<td>0%</td>
<td>0.822%</td>
<td>0.009%</td>
<td>0.355%</td>
<td>0.095%</td>
</tr>
<tr>
<td>gap_dev</td>
<td>0</td>
<td>0.55</td>
<td>0.767</td>
<td>0</td>
<td>0.812</td>
<td>0.028</td>
<td>0.328</td>
<td>0.153</td>
</tr>
<tr>
<td>time</td>
<td>49.435</td>
<td>391.337</td>
<td>665.897</td>
<td>158.891</td>
<td>3337.19</td>
<td>840.164</td>
<td>3600</td>
<td>2600.32</td>
</tr>
<tr>
<td>time_dev</td>
<td>133.765</td>
<td>1131.03</td>
<td>1068.99</td>
<td>249.403</td>
<td>860.397</td>
<td>1025.97</td>
<td>3.878</td>
<td>1031.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| gap       | 0%    | 0.179%| 0.263%| 0%    | 0.74% | 0.022%| 0.355%| 3.562%
| gap_dev   | 0     | 0.568 | 0.832 | 0     | 0.639 | 0.069 | 0.386 | 10.429|
| time      | 37.583| 397.6 | 847.968| 258.23| 3431.74| 1703.15| 3600  | 2879.52|
| time_dev  | 64.693| 1127.12| 1353.06| 354.236| 553.778| 1134.96| 4.358 | 1095.55|
|           |       |       |       |       |       |       |       |       |
| gap       | 0%    | 0.229%| 0.263%| 0%    | 0.612%| 0.061%| 0.621%| 5.517%
| gap_dev   | 0     | 0.726 | 0.832 | 0     | 0.564 | 0.128 | 0.68  | 11.216|
| time      | 77.684| 484.151| 782.896| 342.829| 3370.43| 1718.03| 3600  | 3313|
| time_dev  | 152.963| 1121.08| 1353.06| 354.236| 553.778| 1134.96| 3.878 | 683.835|
|           |       |       |       |       |       |       |       |       |
| gap       | 0%    | 0.161%| 0.242%| 0%    | 0.858%| 0.009%| 0.342%| 0.168%
| gap_dev   | 0     | 0.784 | 0.832 | 0     | 0.454 | 0.028 | 0.466 | 0.244 |
| time      | 53.59| 376.089| 637.611| 109.037| 938.277| 1195.74| 3600  | 3313|
| time_dev  | 143.566| 1134.06| 1082.07| 126.455| 949.593| 1195.74| 3.16  | 600.101|
|           |       |       |       |       |       |       |       |       |
| gap       | 0%    | 0.248%| 0.263%| 0%    | 0.744%| 0.052%| 1.821%| 4.342%
| gap_dev   | 0     | 0.784 | 0.832 | 0     | 0.454 | 0.129 | 4.508 | 10.285|
| time      | 77.304| 472.531| 902.229| 322.126| 3338.28| 1671.02| 3600  | 3211.98|
| time_dev  | 169.5| 1132.54| 1167.79| 464.957| 856.438| 1235.27| 3.16  | 600.101|
|           |       |       |       |       |       |       |       |       |
| gap       | 0%    | 0.227%| 0.263%| 0%    | 0.679%| 0.038%| 0.398%| 1.202%
| gap_dev   | 0     | 0.718 | 0.832 | 0     | 0.472 | 0.079 | 0.628 | 3.422 |
| time      | 71.659| 427.517| 775.813| 198.64| 3312.26| 1679.95| 3600  | 2937.67|
| time_dev  | 148.555| 1128.69| 1057.8| 202.074| 933.965| 1408.01| 3.538 | 878.077|
|           |       |       |       |       |       |       |       |       |
| gap       | 0%    | 0.25% | 0.326%| 0%    | 0.681%| 0.04% | 0.478%| 1.693%
| gap_dev   | 0     | 0.79  | 1.03  | 0     | 0.482 | 0.125 | 0.452 | 4.781 |
| time      | 48.655| 440.849| 857.378| 345.204| 3414.3| 1286.7 | 3600  | 2928.96|
| time_dev  | 122.377| 1124.03| 1030.98| 468.441| 614.672| 1005.54| 2.465 | 898.299|
|           |       |       |       |       |       |       |       |       |
| gap       | 0%    | 0.188%| 0.284%| 0%    | 0.9%  | 0.076%| 0.367%| 11.24%
| gap_dev   | 0     | 0.593 | 0.898 | 0     | 0.72  | 0.166 | 0.317 | 15.907|
| time      | 61.164| 408.798| 783.564| 242.785| 3336.58| 1433.07| 3600  | 3166.56|
| time_dev  | 133.381| 1126.71| 1035.49| 212.266| 854.367| 1272.65| 3.093 | 776.8 |

Considering all experiments, a large number of the instances were solved optimally using at least one family of valid inequalities. Moreover, non-optimal mean gaps in Table 2 of each specific instance type represent the existence of some particularly complex instance
where the execution of B&B reaches the time limit. Details for all the 800 experiments (ten instances for each one of the eight instance types) can be found in [http://yalma.fime.uanl.mx/~yasmin/Yasmin_Rios-Solis/Instances.html](http://yalma.fime.uanl.mx/~yasmin/Yasmin_Rios-Solis/Instances.html).

Another important result is the fast convergence of B&B to small gaps using our proposed valid inequalities. Particularly, most of the instances converge to gaps less than 3% in less than one minute. Therefore, we can use a small execution time limit (or gap limit) for the B&B to obtain high quality solutions in seconds. To show the fast convergence of the B&B using the proposed valid inequalities, we implement the combination “K/H/ /” using a stop criterion of 3% of relative gap. Table 4 compares this combination with the implementation of a multi-start iterated local search (MILS) presented in [Ibarra-Rojas and Rios-Solis (2012)](http://yalma.fime.uanl.mx/~yasmin/Yasmin_Rios-Solis/Instances.html) using the same experimental conditions. To compute the gap for the MILS implementation, we took the deviation of the feasible solution founded by MILS and the best bound found by the B&B algorithm of CPLEX.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>1.938%</td>
<td>1.275%</td>
<td>2.557%</td>
<td>0.849%</td>
<td>2.108%</td>
<td>1.238%</td>
<td>1.738%</td>
<td>1.729%</td>
</tr>
<tr>
<td>gap dev</td>
<td>1.142</td>
<td>0.989</td>
<td>0.440</td>
<td>0.571</td>
<td>0.553</td>
<td>0.869</td>
<td>0.428</td>
<td>0.750</td>
</tr>
<tr>
<td>time</td>
<td>1.733</td>
<td>2.981</td>
<td>334.509</td>
<td>4.47</td>
<td>41.917</td>
<td>18.623</td>
<td>91.245</td>
<td>85.485</td>
</tr>
<tr>
<td>time dev</td>
<td>0.952</td>
<td>3.270</td>
<td>1040.41</td>
<td>3.958</td>
<td>31.510</td>
<td>9.941</td>
<td>75.568</td>
<td>64.206</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>15.74%</th>
<th>15.09%</th>
<th>34.08%</th>
<th>21.94%</th>
<th>39.65%</th>
<th>24.14%</th>
<th>34.20%</th>
<th>26.20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>0.0511</td>
<td>0.115</td>
<td>0.068</td>
<td>0.047</td>
<td>0.048</td>
<td>0.027</td>
<td>0.027</td>
<td>0.029</td>
</tr>
<tr>
<td>gap dev</td>
<td>0.049</td>
<td>0.474</td>
<td>1.401</td>
<td>1.799</td>
<td>6.930</td>
<td>8.644</td>
<td>26.674</td>
<td>29.352</td>
</tr>
<tr>
<td>time</td>
<td>0.409</td>
<td>0.474</td>
<td>1.401</td>
<td>1.799</td>
<td>6.930</td>
<td>8.644</td>
<td>26.674</td>
<td>29.352</td>
</tr>
<tr>
<td>time dev</td>
<td>0.143</td>
<td>0.351</td>
<td>0.378</td>
<td>0.626</td>
<td>1.660</td>
<td>2.125</td>
<td>5.209</td>
<td>3.903</td>
</tr>
</tbody>
</table>

Table 4: Results of combination “K/H/ /” with a stop criterion of 3% of relative gap using the B&B of CPLEX 12.3 and the metaheuristic MILS.

Notice than our exact approach leads to the best results for all instance types in terms of relative gaps. Although, MILS is more efficient considering the execution time, there is not a remarkable difference of execution times since our exact approach reaches the stop criterion in less than 50 seconds for most of the instances.

In summary, we obtain optimal solution for most of the instances in a reasonable time (less than one hour) implementing our proposed valid inequalities. Moreover, we can use a specific stop criterion such as time limit, or deviation limit to obtain high quality solutions in short execution times. These characteristics are very important since recalculation of timetables is usually needed to obtain a solution of the entire transit network planning problem. Therefore, this study presents a tool for planners to improve the quality and efficiency of the whole transportation process.
6 Conclusions

We define an exact solution approach for the NP-hard Synchronization Bus Timetabling problem. This problem determines the departure time for all trips of each line to allow passenger transfers and avoid bus bunching between different lines, subject to almost even headways. STB MILP has characteristics such as symmetries in the feasible solution space, bad quality dual bounds (obtained by linear relaxation). However, the constraints that induce the flexibility in the STB MILP allow us to define different families of valid inequalities to bound the number of synchronizations related to a specific trip and other ones of the lifting type.

Our exact approach is based on solving a tighten formulation of the synchronization bus timetabling problem using our proposed valid inequalities and the B&B algorithm of CPLEX. By analyzing the linear relaxation of our tighter formulation, we conclude that the proposed valid inequalities are very useful to obtain tighter dual bounds and eliminate symmetries of solutions in the relaxed problem thus, improving B&B behavior. Moreover, the fractional solutions are very close to integer values leading to excellent numerical results such as find high quality solutions (optimal for most cases) for SBT in a short time, and a fast convergence to solutions with less than 3% of relative deviation from the optimal solution in less than two minutes. These results are very interesting if we want an integrated approach with the vehicle scheduling since we have a stronger formulation that can be used in an sequential approach or in a integration with other subproblems of the entire transit network planning problem.

Although, we obtain high quality solutions in a short time, there are theoretical interesting research opened by our proposed solution methodology for the synchronization bus timetabling problem such as the following. Find tighter upper bounds for big $\mathcal{M}$ parameters of fix-sync inequalities. Determine the dimension of the valid inequalities proposed in this study. Integrate the synchronization bus timetabling with other subproblems of transit network planning such as vehicle and crew scheduling. Finally, define the timetabling problem considering the entire day as a planning period.

References


Giesemann, C., 2002. Periodic timetable generation. Published in a Seminar of the University of Constance.


A Lifting Algorithm for Generating Equations (21)

Algorithm 2 Lifting (SBT)

Input: SBT formulation
Output: stronger SBT formulation

1: for (each possible synchronization \( y = Y_{pqb}^{ij} \)) do
2: \( y \leftarrow 1 \) and \( [\alpha, \beta] = S_q^j \cap A_p^i \)
3: \( D_q^j = [\alpha - t_b^j, \beta - t_b^j] \)
4: \( D_{p_b}^i = [\alpha - W_b - t_b^i, \beta - w_b - t_b^i] \cap D_{p_b}^i \)
5: Recalculate \( D_{p_{b'}}^i \) and \( D_{q_{b''}}^j \), for all trips \( p' \neq p \) and \( q' \neq q \)
6: for \((i', j') \in J(i'), b' \in B(i', j')\) do
7: if \((\{i', j'\} \cap \{i, j\} \neq \emptyset \) and \((i', j', b') \neq (i, j, b))\) then
8: for \((p' = 1 \text{ to } f_{i'} \) and \( q' = 1 \text{ to } f_{j'}\) do
9: \( E_{y_{p'q'}}^y \leftarrow \emptyset \)
10: for \((q'' \geq q' \text{ to } f_{j'}\) such that \( S_{p'_{b'}q''_b'} \cap A_{q'_{b''}} \neq \emptyset\) do
11: Calculate new windows \( S_{p'_{b'}q''_b'} \) and \( A_{q'_{b''}} \)
12: if \((S_{p'_{b'}q''_b'} \cap A_{q'_{b''}} = \emptyset\) then
13: \( E_{y_{p'q'}}^y \leftarrow E_{y_{p'q'}}^y \cup \{Y_{p'_{b'}q''_b'}\} \)
14: end if
15: end for
16: end for
17: for \((p'' > p' \text{ to } f_{i'}\) such that \( S_{p'_{b'}q''_b'} \cap A_{q'_{b''}} \neq \emptyset\) do
18: Calculate new windows \( S_{p''_{b'}q''_b'} \) and \( A_{q''_{b'}} \)
19: if \((S_{p''_{b'}q''_b'} \cap A_{q''_{b'}} = \emptyset\) then
20: \( E_{y_{p'q'}}^y \leftarrow E_{y_{p'q'}}^y \cup \{Y_{p''_{b'}q''_b'}\} \)
21: end if
22: end for
23: Add inequality \( \sum_{y' \in E_{y_{p'q'}}} y' \leq \left( 1 + \max \left\{ \left\lfloor \frac{W_{b'} - w_{b'}}{h'} \right\rfloor, \left\lfloor \frac{W_{b'} - w_{b'}}{h'} \right\rfloor \right\} \right) (1 - y) \) to SBT MILP
24: end if
25: end for
26: end for
Acknowledgments

This research was partially supported by grant 101857 from the Mexican National Council for Science and Technology (CONACYT). O.J. Ibarra-Rojas wishes to acknowledge graduate scholarship from CONACYT.