Valid inequalities for the synchronization bus timetabling problem

P. Fouilhoux, O.J. Ibarra-Rojas, S. Kedad-Sidhoum, Y.A. Rios-Solis

Abstract

Bus transit network planning is a complex process that is divided into several phases such as: line planning, timetable generation, vehicle scheduling, and crew scheduling. In this work, we address the timetable generation which consists in scheduling the departure times for all trips of each bus line. We focus on the Synchronization Bus Timetabling Problem (SBTP) that favors passenger transfers and avoids congestion of buses at common stops. These characteristics and the embedded flexibility of the SBTP are crucial for many transit networks in Latin America. A Mixed Integer linear Program (MIP) was proposed in the literature for the SBTP but it fails to solve real bus network instances. We develop in this paper four classes of valid inequalities for this MIP using combinatorial properties of the SBTP on the number of synchronizations. Experimental results show that the enhanced MIP yields high quality solutions using small computational times. In particular, instances based on real transit networks are solved within few minutes with a relative deviation from the optimal solution that is usually less than 2%.

Keywords: Integer Programming, Transportation, Timetabling, Valid


Inequalities, Urban Bus System.

1. Introduction

Planning and operating a public transportation network in a relatively big city (3 millions people) is an enormous task if the aim is to reduce the costs without degrading the quality service for the users. In [Ceder (2007)], the bus transit network problem is presented as a sequence of four main phases. The first one is the line planning phase that defines the routes, stops, and frequency for each bus line in a specific territory. Then, the timetable generation phase determines the departure times of all the trips of the lines to achieve a level of quality service. The third subproblem is the vehicle scheduling phase that assigns vehicles to trips associated to bus lines. Finally, the crew scheduling phase defines tasks assigned to drivers. The bus transit network problem is commonly tackled by sequential approaches. Therefore, obtaining high quality solutions in short time for each phase is an important issue since it is often necessary 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 operational subproblems which are vehicle and crew scheduling.

In this article, we focus on bus networks having the following characteristics. The passengers of a line have an estimate of their waiting time at each stop rather than knowing the whole timetable of the line. Therefore, regularly spaced departure times for the trips of each line are required. Moreover, since different lines frequently converge at specific stops of the network, it is necessary to reduce congestion to improve the quality service. In the sequel, a synchronization occurs when the timetable permits the arrival of two buses at the same stop so that the two buses do not bunch or permits passenger transfers without long waiting times. Therefore, the Synchronization Bus Timetabling Problem (SBTP) consists in determining the departure time of every bus trip in order to maximize the synchronization of different bus lines of a network. Thus, the SBTP corresponds exactly to the timetable generation phase for networks that
appear in different cities of Latin America, as for instance, the network of the
city of Monterrey in Mexico which has more than 300 bus lines.

Ibarra-Rojas & Rios-Solis (2012) have proved that the SBTP is NP-hard.
They have proposed a multi-start iterated local search algorithm and a MIP.
We denote by SBTP MIP this formulation that will be described in Section 2.
However, the SBTP MIP solved using a standard linear solver (CPLEX 12.3)
has not succeeded in efficiently solving real size instances within reasonable CPU
times. Since the number of disruptions such as bus failures or crash accidents,
is close to 10% of the vehicles used per day (in Monterrey), bus timetables
must be recomputed several times a day. Consequently, heuristic methods are
usually used to quickly produce timetables. However, these solving methods do
not ensure any guarantee on the solution quality. In this article, we present an
exact approach to solve the SBTP using integer programming techniques and
show that high quality solutions can be produced for real size instances within
a few minutes.

Particular cases of the SBTP have been considered in the literature. Ceder
& Tal (2001) and Ceder (2011) have introduced a simpler version of the SBTP
where synchronizations are defined as simultaneous arrivals at common stops
subject to bound constraints on the the separation time between consecutive
trips of the same line. Constructive algorithms were developed for this case.
An extension of Ceder’s studies is presented by Eranki (2004) where a syn-
cchronization is redefined as arrivals within a time window. Liu et al. (2007)
have reformulated the timetabling problem presented by Ceder et al. (2001)
and implemented a Tabu search to solve the problem.

Some closely related problems on transport networks have been studied us-
ing integer programming approaches. We can quote the scholar bus scheduling
problem introduced by Fügenschuh (2009) where starting times of schools and
starting times of scholar buses must be synchronized to minimize the number of
vehicles to transport all students. The authors have developed different families
of valid inequalities leading to a branch-and-cut algorithm. Quadrifoglio et al.
(2008) optimized a weighted objective function based on vehicle resources and
quality service for a transport network. They define logic constraints to reduce the feasible space which allows to reduce up to 90% of the CPU time when these cuts are added at the beginning of a branch-and-bound algorithm. In railway systems, timetabling problems have been extensively studied using integer programming techniques. In particular, the periodic timetabling case has special structure (not present in the SBTP) that favors the design of valid inequalities commonly implemented in branch-and-cut algorithms (Caimi et al. (2011); Giesemann (2002); Liebchen (2004, 2007); Liebchen & Möhring (2002, 2007); Schröder & Solchenbach (2006)). Unfortunately, timetabling problems in urban transport networks do not share the structure of periodic timetabling. Comprehensive reviews have been provided by Guihaire & Hao (2008) and Cacchiani & Toth (2012).

An inequality is said to be valid for a MIP if every solution of the MIP satisfies this inequality. Consequently, a valid inequality can be added to a MIP in order to obtain a stronger formulation (see Wolsey (1998)) that will be easier to solve. Our main contributions are to define four families of valid inequalities that will be used for the SBTP MIP. The first two families of valid inequalities bound the number of possible synchronizations that can occur at a node for a specific trip. The two other families are obtained from the previous ones using a generic lifting procedure. Additionally, some inequalities of the SBTP MIP are tightened by lowering some parameters. The enhanced SBTP MIP is then solved with a standard linear solver to obtain optimal solutions for most of the real size instances. Moreover, for all the instances, solutions with less than 3% of deviation from the optimum are found in less than five minutes.

The rest of the paper is organized as follows. In Section 2 we briefly recall the SBTP MIP that we will enhance with our proposed valid inequalities that are defined in Section 3. A generic lifting procedure is introduced and is applied in order to produce another two families of valid inequalities. Section 4 presents a way to tighten some inequalities by adjusting some parameters. Experimental results for instances based on a real transit network are presented in Section 5 where we show the impact of some combination of the valid inequalities families.
Finally, conclusions and future research areas are addressed in Section 7.

2. Problem definition and mathematical model

In this section, we briefly define the SBTP and present the SBTP MIP proposed by Ibarra-Rojas & Rios-Solis (2012).

2.1. The synchronization bus timetabling problem

The SBTP schedules the departure times for all bus line trips in order to maximize the synchronizations between buses. A formal definition of the problem is given in the following.

The type of bus network we are interested in can be represented by a set of lines denoted as $I$. As described in Section 1, the routes and the stops used by each bus line as well as the frequency of each line are determined before the timetable generation phase. We will call a synchronization node, a specific stop in the network where two lines cross each other and where passenger transfers are needed or congestion of buses at common stops can happen. We denote by $B$ the set of all synchronization nodes. Let $T$ be the planning horizon defined in time units. For example, $T$ will be 2 hours in rush periods in the morning or 3 hours in valley periods in the afternoon. We will denote by $f_i$ the frequency of line $i \in I$, that is to say the number of trips that have to be scheduled within $T$ for line $i$. We also assume that buses of the same line will run at regular speed and we denote by $t_{ib}, i \in I$, the travel time for a bus from an initial node, called depot, to a node $b \in B$.

A headway time is the separation time between consecutive trips of the same line. For example, if the first (resp. second) trip of a line starts at time unit 3 (resp. 23), there is a headway time of 20 time units between these two trips. Since regularly spaced departure times are required, each trip of a line $i$ of $I$ will start every $\frac{T}{f_i}$ time units with a flexibility of $\delta_i$ time units. Consequently, $h^i = \frac{T}{f_i} - \delta_i$ (resp. $H^i = \frac{T}{f_i} + \delta_i$) is the minimum (resp. maximum) headway time for a line $i \in I$. Moreover, the first (resp. last) trip of line $i$ must start before $H^i$. 

5
These headway times guarantee that the entire planning horizon is covered by the trips which is one of the main differences between the addressed model and the ones studied by Ceder & Tal (2001) and Eranki (2004). Consequently, a timetable given by the departure times $X_{ip}$ of each trip $p \in \{1, ..., f^i\}$ of every line $i \in I$ will satisfy $X_{ip} \leq H^i$, $h^i \leq X_{ip+1} - X_{ip} \leq H^i$ for every $i \in \{1, ..., f^i - 1\}$, $T - H^i \leq X_{if^i} \leq T$.

Figure 1 illustrates regularly spaced trips for a given line $i$ of frequency $f^i = 4$ with and without flexibility.

We recall that a synchronization at a stop between two bus trips occurs in two different cases; when the difference between their arrival times is long enough to reduce congestion of buses belonging to different lines at common stops in one hand or allow passenger transfers without long waiting times in another hand.

A transfer node is a particular synchronization node in the network where passengers may transfer from one line $i$ to some line $j$. To allow well timed passenger transfers at node $b$, the difference between two bus arrivals at this node has to be lower (resp. greater) than a specific value denoted $W_b$ (resp. $w_b$). A congestion node is a particular synchronization node in the network where two buses (of different lines) arriving simultaneously at this node will...
share a segment of their respective routes. Consequently, to reduce congestion of buses at node \( b \), the length of the time interval between two bus arrivals at node \( b \) has to be greater than a specific value denoted \( w_b \). In this case, it is convenient to consider a maximum separation time \( W_b \) between the bus arrivals at node \( b \) such that \( W_b - w_b < h^j \). In the basis of the above, a synchronization at node \( b \) for two bus trips happens if the difference of their arrival times is in the so-called waiting time window \([w_b, W_b]\).

For each line \( i \in I \), let us denote by \( I(i) \) the set of lines that share a synchronization node with line \( i \). Notice that for \( i, j \in I \), \( j \in I(i) \) does not imply \( i \in I(j) \). Indeed, it could be desirable in some situations to enforce the passenger transfers only from line \( i \) to line \( j \).

Therefore, the SBTP consists in determining the departure times \( X^i_p \) of each trip \( p \in \{1, \ldots, f^i\} \) of every line \( i \in I \) such that the headway time bounds are satisfied, \( X^i_1 \leq H^i, h^i \leq X^i_{p+1} - X^i_p \leq H^i \) for every \( i \in I \), \( T - H^i \leq X^i_f \leq T \), with the objective of maximizing the number of pairs of synchronized trips, that is, to maximize the number of pairs of trips \( p \in \{1, \ldots, f^i\} \) and \( q \in \{1, \ldots, f^j\} \) such that \( w_b \leq (X^j_q + t^j_b) - (X^i_p + t^i_b) \leq W_b \) is satisfied, for \( i \in I \), \( j \in I(i) \).

Figure 2 illustrates the two types of synchronization nodes. Case (1) represents a congestion node \( b \) where lines \( i \) and \( j \) (with \( h^j = 8 \)) converge and then 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. The minimum and maximum separation times to reduce congestion at this node \( b \) are \( w_b = 6 \) and \( W_b = 12 \).

Case (2) represents a node \( b \) where passengers would like to go from one trip of line \( i \) to some trip of line \( j \). In this case, 5 time units is the maximum desired passenger waiting time and 2 time units is the estimated required time for the transfer. Then, the waiting time window is given by \([w_b, W_b] = [2, 7]\).

2.2. Mixed integer program

We consider in the following an instance of SBTP and we now recall with the 0-1 MIP of the problem already described by Ibarra-Rojas & Rios-Solis (2012). For a given a line \( i \in I \), we will denote by \((p, i), p \in \{1, \ldots, f^i\}\), the
Consider the previous parameters and decision variables, the SBTP MIP is given by

$$\max \sum_{i \in I} \sum_{j \in I(i)} \sum_{b \in B^{ij}} \sum_{p=1}^{f^i} \sum_{q=1}^{f^j} Y_{pqb}^{ij}$$

s.t.

$$X_i^p \leq H_i^i \quad \forall i \in I$$  \hspace{2cm} (1)

$$T - H_i^i \leq X_i^{f_i} \leq T \quad \forall i \in I$$ \hspace{2cm} (2)

$$h_i \leq X_{p+1}^i - X_p^i \leq H_i^i \quad \forall i \in I, p = 1, \ldots, f_i - 1$$ \hspace{2cm} (3)

$$\left(X_j^b + t_{i}^j\right) - \left(X_p^i + t_{i}^p\right) \geq w_b + M \left(1 - Y_{pqb}^{ij}\right) \quad \forall i \in I, j \in I(i), b \in B^{ij}, p = 1, \ldots, f_i, q = 1, \ldots, f_j$$ \hspace{2cm} (4)

$$\left(X_q^j + t_{j}^q\right) - \left(X_p^i + t_{i}^p\right) \leq W_b + M \left(1 - Y_{pqb}^{ij}\right) \quad \forall i \in I, j \in I(i), b \in B^{ij}, p = 1, \ldots, f_i, q = 1, \ldots, f_j$$ \hspace{2cm} (5)

$$X_i^p \in \mathbb{R}, Y_{pqb}^{ij} \in \{0, 1\} \quad \forall i \in I, j \in I(i), b \in B^{ij}, p = 1, \ldots, f_i, q = 1, \ldots, f_j$$ \hspace{2cm} (6)
The objective function maximizes the total number of synchronizations. Constraints (1) and (2) guarantee that the entire planning horizon is covered by the trips. Constraints (3) impose that consecutive trips of line $i$ must happen with a minimum (resp. maximum) headway time $h^i$ (resp. $H^i$). Remark that the arrival time of trip $(p, i)$ at node $b$ is $X_p^i + t_b^i$. Hence Constraints (4) and (5) activate the synchronization variables $Y_{pq}^{ij}$ if the difference between the arrival times of $(p, i)$ and $(q, j)$ at node $b$ is within $[w_b, W_b]$ with $(p, i)$ arriving first at node $b$. $M$ is a large constant whose value will be defined in Section (4).

As mentioned by Ibarra-Rojas & Rios-Solis (2012), the SBTP MIP cannot be used to solve real instances of the SBTP using a standard linear programming solver. The main contributions of this paper is to strengthen SBTP MIP by introducing valid inequalities that allow to efficiently solve the real size instances of the SBTP.

3. Valid inequalities

To obtain tighter formulations for the SBTP, we add valid inequalities to the SBTP MIP before its resolution by a linear programming solver. In fact, adding valid inequalities permits to cut fractional solutions of the linear relaxations of integer programs or to cut non-optimal feasible solutions (Wolsey (1998); Nemhauser & Wolsey (1999)). In the following, we introduce two families of valid inequalities for the synchronization bus timetabling problem obtained from the headway time parameters and the propagation of Constraints (1), (2), and (3).

3.1. Synchronization inequalities

We can take advantage of the headway time parameters to define a family of valid inequalities for each trip to be synchronized. Let us consider two lines $i$ and $j$ to be synchronized at a given 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., $h^j > W_b - w_b$. If a trip $(p, i)$ synchronizes with another trip $(q, j)$, the
synchronization of \((p, i)\) with trips \((q-1, j)\) or \((q+1, j)\) is impossible since there would not be enough time units to ensure solution feasibility. Generalizing the previous idea, we obtain the following result.

**Lemma 1.** Let \(i\) and \(j\) be two distinct lines with \(j \in I(i)\) and \(b \in B^{ij}\), the maximum number of synchronizations between one trip of line \(i\) and all the trips of line \(j\) is \(1 + \left\lfloor \frac{W_b - w_b}{h^j} \right\rfloor\).

**Proof.** Let us suppose that there exists a feasible solution \((\tilde{X}, \tilde{Y})\) of the SBTP and a trip \((p, i)\) such that there exist \(r > 1 + \left\lfloor \frac{W_b - w_b}{h^j} \right\rfloor\) trips \(q_1 < q_2 < \cdots < q_r\) of line \(j\) that may synchronize with trip \((p, i)\). We will show that it is impossible to schedule these synchronized trips because of the imposed minimal headway times between trips. The arrival times of these trips are within the feasible synchronization time window \([\tilde{X}^{ij}_p + t^i_b + w_b, \tilde{X}^{ij}_p + t^i_b + W_b]\). Therefore, the difference between the arrival times \(\tilde{X}^{ij}_{q_r} - \tilde{X}^{ij}_{q_l}\) of trips \((q_l, j)\) and \((q_r, j)\) must be less than \(W_b - w_b\), consequently we obtain \(\tilde{X}^{ij}_{q_r} - \tilde{X}^{ij}_{q_l} \leq W_b - w_b\).

However, since solution \((\tilde{X}, \tilde{Y})\) corresponds to a regularly spaced schedule, we have that \(\tilde{X}^{ij}_{q_{l+1}} - \tilde{X}^{ij}_{q_l} \geq h^j\). We then have

\[
\tilde{X}^{ij}_{q_r} - \tilde{X}^{ij}_{q_l} \geq (r-1)h^j \geq \left(1 + \left\lfloor \frac{W_b - w_b}{h^j} \right\rfloor\right) h^j \geq \left(1 + \left( \frac{W_b - w_b}{h^j} - 1 \right)\right) h^j = W_b - w_b
\]

which is a contradiction.

Using Lemma 1, we derive the following inequalities that will be called synchronization inequalities.

\[
\sum_{q=1}^{f^j} Y^{ij}_{pqb} \leq 1 + \left\lfloor \frac{W_b - w_b}{h^j} \right\rfloor \quad \forall i \in I, j \in I(i), b \in B^{ij}, \forall p \in \{1, \ldots, f^i\} \quad (7)
\]

\[
\sum_{p=1}^{f^i} Y^{ij}_{pqb} \leq 1 + \left\lfloor \frac{W_b - w_b}{h^i} \right\rfloor \quad \forall i \in I, j \in I(i), b \in B^{ij}, \forall q \in \{1, \ldots, f^j\}. \quad (8)
\]

Let us consider two lines \(i\) and \(j\) with \(j \in I(i)\), a node \(b \in B^{ij}\), and a trip \((p, i)\).

Given a solution \((\tilde{X}, \tilde{Y})\) of the SBTP, we can remark that \(\sum_{q=1}^{f^j} \tilde{Y}^{ij}_{pqb}\) is exactly the number of synchronization between trips of line \(j\) and trip \((p, i)\) at node \(b\).

Consequently, we have the following theorem.
Theorem 1. Synchronization inequalities \( (7) \) and \( (8) \) are valid for the SBTP MIP.

3.2. Headway inequalities

Using similar ideas, we can devise the following class of inequalities.

\[
Y_{ij}^{pq} + \sum_{q'=q+1}^{f^i} Y_{ij}^{pq'}b + \sum_{p'=p+1}^{f^i} Y_{ij}^{p'q}b \leq 1 + \left\lfloor \frac{W_b - w_b}{\min(h^i, h^j)} \right\rfloor \quad \forall \, i \in I, \, \forall \, j \in I(i), \, \forall \, b \in B^i, \, \forall \, p \in \{1, \ldots, f^i\}, \forall \, q \in \{1, \ldots, f^j\} \quad (9)
\]

\[
Y_{ij}^{pq} + \sum_{q'=1}^{q-1} Y_{ij}^{pq'}b + \sum_{p'=1}^{p-1} Y_{ij}^{p'q}b \leq 1 + \left\lfloor \frac{W_b - w_b}{\min(h^i, h^j)} \right\rfloor \quad \forall \, i \in I, \, \forall \, j \in I(i), \, \forall \, b \in B^i, \, \forall \, p \in \{1, \ldots, f^i\}, \forall \, q \in \{1, \ldots, f^j\}. \quad (10)
\]

We will denote these inequalities as headway inequalities as they depend on the value of the desired minimum headway time between trips. We then have the following result.

Theorem 2. Headway inequalities \( (9) \) and \( (10) \) are valid for the SBTP MIP.

Proof. We will consider an inequality \( (9) \) corresponding to the two trips \((p, i)\), \((q, j)\) and a node \(b \in B^i\) (the proof for inequalities \( (10) \) is analogous). Let \((\tilde{X}, \tilde{Y})\) be a feasible solution of the SBTP. We consider the two quantities \(c\) and \(r\) defined as follows

\[
r = \sum_{q'=q+1}^{f^i} \tilde{Y}_{ij}^{pq'b} \quad \text{and} \quad c = \sum_{p'=p+1}^{f^i} \tilde{Y}_{ij}^{p'q'b}.
\]

If \(r = 0\) then from inequality \( (8) \) we know that

\[
\tilde{Y}_{ij}^{pq} + c \leq \sum_{p'=1}^{f^i} \tilde{Y}_{ij}^{p'q'b} \leq 1 + \left\lfloor \frac{W_b - w_b}{\min(h^i, h^j)} \right\rfloor \leq 1 + \left\lfloor \frac{W_b - w_b}{\min(h^i, h^j)} \right\rfloor .
\]

Similarly, if \(c = 0\), from inequality \( (7) \), we obtain that

\[
\tilde{Y}_{ij}^{pq} + r \leq \sum_{q'=1}^{q-1} \tilde{Y}_{ij}^{pq'b} \leq 1 + \left\lfloor \frac{W_b - w_b}{h^j} \right\rfloor \leq 1 + \left\lfloor \frac{W_b - w_b}{\min(h^i, h^j)} \right\rfloor .
\]

Let us now suppose that both \(r \geq 1\) and \(c \geq 1\). Since \(r \geq 1\) trip \((p, i)\) synchronizes with \(r\) trips among trips \(q + 1, \ldots, f^j\) of line \(j\). Let \(q'\)
be the first of these trips that is synchronized with \((p, i)\). Consequently, arrival times of trips \(q', q' + 1, \ldots, q' + (r - 1)\) at node \(b\) are within the time window \([\tilde{X}_b^j + t_b^i + w_b, \tilde{X}_b^j + t_b^i + W_b]\). Since the minimal headway time between two trips of line \(j\) is \(h_j^j\), the arrival time of trip \((q', j)\) at node \(b\) satisfies \(\tilde{X}_q^j + t_b^i \leq \tilde{X}_p^j + t_b^i + W_b - (r - 1)h_j^j\). Since \(\tilde{X}_q^j \leq \tilde{X}_q^{j+1}\), we finally obtain

\[
\tilde{X}_q^{j+1} + t_b^i \leq \tilde{X}_p^j + t_b^i + W_b - (r - 1)h_j^j.
\]

Similarly, since \(c \geq 1\) there are \(c\) trips among \(p + 1, \ldots, f^i\) of line \(i\) that synchronizes with \((q, j)\). Let \(p'\) be the last of these trips that synchronized with \((q, j)\), thus we know that \(\tilde{X}_q^j + t_b^i \geq \tilde{X}_p^{j'} + t_b^i + w_b\). Moreover, since trips \(p' - (c - 1), \ldots, p'\) are synchronized with \((q, j)\) and because of the imposed minimum headway times of line \(i\), we know that \(\tilde{X}_q^j + t_b^i \geq \tilde{X}_p^{j'-(c-1)} + t_b^i + w_b + (c - 1)h_i^i\).

Since \(\tilde{X}_p^{j'-(c-1)} \geq \tilde{X}_{p+1}^j\), we obtain

\[
\tilde{X}_q^j + t_b^i \geq \tilde{X}_{p+1}^j + t_b^i + w_b + (c - 1)h_i^i.
\]

Because of the minimum headway times between trips of line \(j\), we then get

\[
h_j^j \leq (\tilde{X}_{q+1}^j + t_b^i) - (\tilde{X}_q^j + t_b^i) \leq \tilde{X}_p^j - \tilde{X}_{p+1}^j + W_b - w_b - (r - 1)h_j^j - (c - 1)h_i^i.
\]

And, because of the minimum headway time between trips \((p, i)\) and \((p + 1, i)\), we have \(\tilde{X}_{p+1}^j - \tilde{X}_p^j \geq h_i^i\). Then, we obtain

\[
h_j^j \leq -h_i^i + W_b - w_b - (r - 1)h_j^j - (c - 1)h_i^i,
\]

which then becomes \(rh_j^j + ch_i^i \leq W_b - w_b\). W.l.o.g. we can suppose that \(h_j^j \leq h_i^i\).

We then obtain

\[
r + c \leq r + c \frac{h_i^i}{h_j^j} \leq \frac{W_b - w_b}{h_j^j}.
\]

Thus we obtain

\[
\bar{Y}_{pqj} + r + c \leq 1 + \frac{W_b - w_b}{h_j^j}
\]

which proves the validity of inequality 9.

\[\square\]
3.3. Lifting procedure

In this section, we present a generic lifting method that permits to compute new valid inequalities from the previous ones.

Let us first introduce a useful notation. Let \( \xi \) be the set of all 5-tuples \((i, j, p, q, b)\) with \( i \in I, j \in I(i), p \in \{1, \ldots, f^i\}, q \in \{1, \ldots, f^j\} \) and \( b \in B^j \). Notice that set \( \xi \) is in a one-to-one correspondence with the set of \( Y \)'s variables, that is to say, each 5-tuple \((i, j, p, q, b) \in \xi \) corresponds to a potential synchronization between trips \((p, i)\) and \((q, j)\) at node \( b \) so that trip \((p, i)\) arrives first at node \( b \).

Let us consider the following inequality:

\[
\sum_{(i, j, p, q, b) \in E} Y_{ij}^{pq} \leq \gamma \tag{11}
\]

where \( E \subset \xi \) and \( \gamma \) is an upper bound over the number of synchronizations in \( E \). The inequalities of type \( \text{(11)} \) are clearly valid. We note that synchronization and headway inequalities belong to this class.

Given \((i_0, j_0, p_0, q_0, b_0) \in \xi \), we define \( E_{p_0q_0b_0}^{i_0j_0} \) as a set of 5-tuples \((i, j, p, q, b)\) of \( E \) so that trips \((p, i)\) and \((q, j)\) cannot be synchronized at node \( b \) if trips \((p_0, i_0)\) and \((q_0, j_0)\) are synchronized at node \( b_0 \) and trip \((p_0, i_0)\) arrives first at node \( b_0 \), that is to say that for a given solution \((\tilde{X}, \tilde{Y})\) of the SBTP, \( \tilde{Y}_{p_0q_0b_0}^{i_0j_0} = 1 \) implies \( \tilde{Y}_{pq}^{ij} = 0 \) for all \((i, j, p, q, b) \in E_{p_0q_0b_0}^{i_0j_0} \).

Then inequality \( \text{(11)} \) can be lifted to create the following inequality

\[
\sum_{(i, j, p, q, b) \in E \cap E_{p_0q_0b_0}^{i_0j_0}} Y_{ij}^{pq} \leq \gamma \left(1 - \tilde{Y}_{p_0q_0b_0}^{i_0j_0}\right) \tag{12}
\]

which is clearly valid. In the following, we present a generic lifting procedure to obtain inequalities of type \( \text{(12)} \). This procedure will be applied to both synchronization and headway inequalities in order to obtain lifted synchronization inequalities and lifted headway inequalities. Remark that the set of variables \( Y \) of such a lifted inequality is related to different lines while the set of variables of a headway or a synchronization inequality is related to only two lines.

To compute the lifted inequalities, given a 5-tuple \((i, j, p, q, b)\), we will need to find a list of 5-tuples that can not be synchronized if \((i, j, p, q, b)\) is synchronized.
To achieve this, we will compare the feasible departure time window of the potentially synchronized trips, that is to say the time window during which a trip has to start. Some of the results introduced by Ibarra-Rojas & Rios-Solis (2012) for a preprocessing stage, can be used to devise these time windows.

Given trip \((p, i)\) with \(i \in I\) and \(p \in \{1, \ldots, f^i\}\), we denote by \(D_i^p = [d_i^p, D_i^p]\) a feasible departure time window of trip \((p, i)\). Ibarra-Rojas & Rios-Solis (2012) have been proved that

\[
D_i^p = [d_i^p, D_i^p] \subset [d_i^p, D_i^p] \subset [d_i^p, D_i^p] \subset [d_i^p, D_i^p].
\]

The time window \([d_i^p, D_i^p]\) constitutes a generic feasible departure time window. As it will turn out, when two trips synchronize, it will be possible to tighten this window. This can be done using logical inferences that are given by the following theorem which gathers several results given by Ibarra-Rojas & Rios-Solis (2012).

**Theorem 3.** Let \((i, j, p, q, b) \in \xi\) and \(D_i^p = [D_i^p, D_i^p]\) (resp. \(D_q^q = [D_q^q, D_q^q]\)) be a feasible departure time window of trip \((p, i)\) (resp. \((q, j)\)). By setting

\[
\alpha, \beta = [D_i^p + t_i^b, D_i^p + t_i^b + w_i] \cap [D_q^q + t_q^b, D_q^q + t_q^b],
\]

we obtain that

i) \((q, j)\) is synchronized with trip \((p, i)\) arriving first at \(b\) if and only if \([\alpha, \beta] \neq \emptyset\),

ii) and, in that case,

\[
[D_i^p \cap [\alpha - t_i^b, \beta - w_i - t_i^b]
\]

is a tighter feasible departure time window for trip \((p, i)\) and

\[
[D_q^q \cap [\alpha - t_q^b, \beta - w_q - t_q^b]
\]

is a tighter feasible departure time window for trip \((q, j)\).

An example of how to use the previous theorem is the following. Consider feasible departure times \([D_i^p, D_i^p] = [15, 20]\) and \([D_q^q, D_q^q] = [18, 25]\) for trips \((p, i)\) and \((q, j)\), respectively. Assume we have the parameters values \(t_i^b = 10\), \(t_q^b = 22\), and \([w_i, W_i] = [3, 8]\). Then, the arrival of trip \((q, j)\) at node \(b\) is within the interval \([18 + 22, 25 + 22] = [40, 47]\) and the arrival of trip \((p, i)\) at node \(b\) is within \([15 + 10, 20 + 10] = [25, 30]\). If it is possible to synchronize trip \((p, i)\) with trip \((q, j)\), arrival time of trip \((q, j)\) at node \(b\) must be within
\[ [25 + 3, 30 + 8] = [28, 38] \text{ which is impossible since } [28, 38] \cap [40, 47] = \emptyset. \]

Knowing a tighter feasible departure time window \( D_p^i = \left[ D_p^i, \overline{D}_p^i \right] \) for a given trip \((p, i)\), a tighter departure time window \( D_{p'}^i \), for all trips \( p' \neq p \), can be inferred using the headway times between trips of line \( i \). This can be done using the procedure \( \text{Propagate}(D_p^i) \).

\textbf{Algorithm 1 \textit{Propagate}(D_p^i)}
\begin{enumerate}
    \item \textbf{for} \( p' = 1 \) to \( p - 1 \) \textbf{do}
    \item \( D_{p'}^i := D_{p'}^i \cap \left[ D_p^i + (p' - p)h^i, \overline{D}_p^i + (p' - p)H^i \right] \)
    \item \textbf{end for}
    \item \textbf{for} \( p' = p + 1 \) to \( f^i \) \textbf{do}
    \item \( D_{p'}^i := D_{p'}^i \cap \left[ D_p^i - (p - p')H^i, \overline{D}_p^i - (p - p')h^i \right] \)
    \item \textbf{end for}
\end{enumerate}

Algorithm 2, called \textit{Generic Lifting}, shows the steps to generate lifted inequalities. The general idea to obtain these inequalities is to consider a 5-tuple \((i_0, j_0, p_0, q_0, b_0) \in \xi \) and see what implications arise if the corresponding synchronization is set, that is to say when \( Y_{i_0,j_0,p_0,q_0,b_0} = 1 \). Step 2 consists in computing the feasible departure time windows of every trip from Formula (13). Then, assuming that \((i_0, j_0, p_0, q_0, b_0)\) is synchronized at node \( b_0 \), we compute tighter feasible departure time windows \( D_{p_0}^{i_0} \) and \( D_{q_0}^{j_0} \) (Steps 3–5) using Theorem 3ii). Using procedure \textit{Propagate}, we then update the departure time windows for the rest of the trips of lines \( i_0 \) and \( j_0 \) (Step 6). We then apply a specific procedure for every inequalities \( \sum_{(i,j,p,q,b) \in E} Y_{pqb}^{ij} \leq \gamma \) of type (11): we determine a set \( E' \) of 5-tuples \((i, j, p, q, b)\) of \( E \) that cannot be synchronized and we then can create the corresponding lifted inequalities.

Unfortunately, Steps 7–9 must be dedicated to each of the two considered type of inequalities in order to consider sets \( E' \) with known upper bounds. In fact, lines 7–10 have to be replaced by Algorithm 4 for synchronization inequalities and Algorithm 5 for headway inequalities.

In order to find which trips cannot be synchronized, we introduce another procedure (Algorithm 3), called \textit{Test synchronization}. This procedure deter-
Algorithm 2 Generic Lifting

1: for each \((i_0, j_0, p_0, q_0, b_0)\) ∈ \(\xi\) do
2: \(D_i^p := \left[\overline{d}_i^p, \overline{t}_i^p\right]\) for every trip \((p, i)\)
3: \([\alpha, \beta] := \left[\overline{D}_i^{p_0} + t_i^{p_0} + w_{b_0}, \overline{D}_i^{p_0} + t_i^{p_0} + W_{b_0}\right] \cap \left[\overline{D}_j^{q_0} + t_j^{q_0} + \overline{D}_j^{q_0} + t_j^{q_0}\right]\)
4: \(D_j^{q_0} := [\alpha - t_j^{q_0}, \beta - t_j^{q_0}]\)
5: \(D_i^{p_0} := D_i^{p_0} \cap [\alpha - W_{b_0} - t_i^{p_0}, \beta - w_{b_0} - t_i^{p_0}]\)
6: Propagate\((D_i^{p_0})\) and Propagate\((D_j^{q_0})\)
7: for each inequality of type \((11)\) \(\sum_{(i,j,p,q,b) \in E} Y_{pq}^{ij} \leq \gamma\) do
8: Find a set \(E'\) of 5-tuples \((i, j, p, q, b)\) of \(E\) that cannot be synchronized
9: Create inequality \(\sum_{(i,j,p,q,b) \in E'} Y_{pq}^{ij} \leq \gamma \left(1 - Y_{p_0q_0}^{i_0j_0}\right)\)
10: end for
11: end for

mines if trip \((p, i)\) cannot be synchronized with trip \((q, j)\) due to the fact that \((i_0, j_0, p_0, q_0, b_0)\) is synchronized. Using Theorem 3, lines 2-3 tests if \((i, j, p, q, b)\) is a potential synchronization and then, if this synchronization becomes impossible after the update of the departure time windows.

Algorithm 3 Test synchronization\((i, j, p, q, b)\)

1: \([\alpha, \beta] := \left[\overline{D}_i^p + t_i^p + w_b, \overline{D}_i^p + t_i^p + W_b\right] \cap \left[\overline{D}_j^q + t_j^q + \overline{D}_j^q + t_j^q\right]\)
2: if \((\alpha, \beta) \neq \emptyset\) then
3: \(D_j^q := [\alpha - t_j^q, \beta - t_j^q]\)
4: \(D_i^p := D_i^p \cap [\alpha - W_b - t_i^p, \beta - w_b - t_i^p]\)
5: if \(\left[\overline{D}_i^p + t_i^p + w_b, \overline{D}_i^p + t_i^p + W_b\right] \cap \left[\overline{D}_j^q + t_j^q + \overline{D}_j^q + t_j^q\right] = \emptyset\) then
6: Return False
7: end if
8: end if
9: Return True

For creating lifted synchronization inequalities, Algorithm 4 enumerates every synchronization inequality of type \((7)\) which can be affected by the modification of lines \(i_0\) and \(j_0\) (Steps 11). Each potential trip \(q\) is then tested by
Algorithm 3 to know if \((i,j,p,q,b)\) becomes impossible and we then create a lifted synchronization inequality from the set \(E'\) which represents the set of impossible synchronizations. The procedure that generates the lifted synchronization inequalities from the synchronization inequalities (8) is analogous to Algorithm 4.

Algorithm 4 \textit{Create lifted synchronization inequalities} \((i_0,j_0,p_0,q_0,b_0)\)

1: if \((\{i,j\} \cap \{i_0,j_0\} \neq \emptyset \text{ and } (i,j,b) \neq (i_0,j_0,b_0))\) then
2: for \((p = 1 \text{ to } f^i)\) do
3: \(E' := \emptyset\)
4: for \((q = 1 \text{ to } f^j)\) do
5: if \(\text{Test\_synchronization}(i,j,p,q,b) = \text{False}\) then
6: \(E' := E' \cup \{(i,j,p,q,b)\}\)
7: end if
8: end for
9: Create \(\sum_{(i,j,p,q,b) \in E'} Y_{ij}^{pq} b \leq \left(1 + \left\lfloor \frac{W_{b}-w_{b}}{h_{b}} \right\rfloor \right) \left(1 - Y_{i_0,j_0}^{p_0,q_0,b_0} \right)\)
10: end for
11: end if

Using similar ideas, Algorithm 5 computes lifted headway inequalities from the headway inequalities of type (9). A similar algorithm can be devised for the headway inequalities of type (10).

4. Tightening Constraints (4) and (5)

An important aspect in integer programming is to compute tight 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 the lifting inequalities, we can use them to bound big \(M\) parameters for Constraints (4) and (5) of the SBTP MIP.

We can recall that the earliest arrival time of trip \((q,j)\) at node \(b\) is \(d_{j}^{q} + t_{b}^{j}\) and the latest arrival time of trip \((p,i)\) at node \(b\) is \(d_{i}^{p} + t_{b}^{i}\). Therefore, the
Algorithm 5 Create lifted headway inequalities \((i_0, j_0, p_0, q_0, b_0)\)

1: if \((\{i, j\} \cap \{i_0, j_0\} \neq \emptyset \text{ and } (i, j, b) \neq (i_0, j_0, b_0))\) then

2: for \((p = 1 \text{ to } f^i \text{ and } q = 1 \text{ to } f^j)\) do

3: \(E' := \emptyset\)

4: for \((q' \geq q \text{ to } f^j)\) do

5: if \(\text{Test}\_\text{synchronization}(i, j, p, q', b) = \text{False}\) then

6: \(E' := E' \cup \{(i, j, p, q', b)\}\)

7: end if

8: end for

9: for \((p' > p \text{ to } f^i)\) do

10: if \(\text{Test}\_\text{synchronization}(i, j, p', q, b) = \text{False}\) then

11: \(E' := E' \cup \{(i, j, p', q, b)\}\)

12: end if

13: end for

14: Create \(\sum_{(i, j, p, q, b) \in E'} Y_{pqb}^{ij} \leq \left(1 + \left[\frac{W_b - w_b}{\min(h^i, h^j)}\right]\right) \left(1 - Y_{p_0q_0b_0}^{i_0j_0}\right)\)

15: end for

16: end if
minimum difference of arrival times between trips \((q,j)\) and \((p,i)\) at node \(b\) is \(d^i_p + t^i_i - d^j_q - t^j_j\). Consequently, given a solution \((\tilde{X}, \tilde{Y})\) of the SBTP, we know that
\[
(\tilde{X}^j_{qb} + t^j_j) - (\tilde{X}^i_{pb} + t^i_i) \geq d^i_p + t^i_i - d^j_q - t^j_j.
\]

Similarly, we can remark that the maximum difference of arrival times between trips \((q,j)\) and \((p,i)\) at node \(b\) satisfies the following inequality
\[
(\tilde{X}^j_{qb} + t^j_j) - (\tilde{X}^i_{pb} + t^i_i) \leq d^j_q + t^j_j - d^i_p - t^i_i.
\]

In the basis of the above, for Constraints \((4)\) and \((5)\) corresponding to \((i,j,p,q,b) \in \xi\), we can replace \(M\) by \(m^{ij}_{pqb} = d^i_p + t^i_i - d^j_q - t^j_j\) and \(M^{ij}_{pqb} = d^j_q + t^j_j - d^i_p - t^i_i\) respectively.

5. Experimental results

In this section, some computational results are presented using the integer linear solver CPLEX 12.3 on a iMac OS X with an Intel Core 2 Duo 3.06 GHz processor and 4 GB RAM. The computational effort to generate the valid inequalities for the SBTP MIP is negligible (less than one second) for the instances we consider. We compare several combinations of the proposed valid inequalities within the SBTP MIP in order to solve efficiently real SBTP instances.

5.1. Instances

We use the instance generation scheme proposed by Ibarra-Rojas & Rios-Solis (2012) as well as some information provided by a company of Monterrey’s transit network.

All the instances have the following common characteristics: a planning period of \(T = 240\) minutes; the frequency \(f^i\) for each line \(i\) is randomly generated in \([13,18]\); the travel time \(t^i_i\) from the depot to synchronization node \(b\) for each line \(i\) is randomly generated in \([20,60]\); the minimum (resp. maximum) waiting time for each synchronization node \(b\) is randomly generated in \([3,5]\) (resp. \([9,12]\)).
finally, the number of different pairs of lines to synchronize at each node \( b \) is randomly generated between 1 and 7.

The derived instances are grouped into 9 types depending on the following parameters: the number \( |I| \) of lines, the number \( |B| \) of synchronization nodes, and the flexibility parameter \( \delta^i \). This latter parameter is randomly generated so that the ratio between \( \delta^i \) and \( T_f^i \) belongs to a flexibility interval denoted by \([F_{min}, F_{max}]\). As it is highlighted by [barra-Rojas & Rios-Solis 2012], for a linear solver, the larger \( F_{min} \) is, the more intractable is the related instance using the basic SBTP MIP.

According to the planners of Monterrey’s transit network, it is reasonable to consider either \([10,20]\) or \([25,35]\) for \([F_{min}, F_{max}]\); and a number of lines 5 times the number of synchronization nodes. We randomly generate 10 instances for each of the 9 instance types (a total of 90 instances) to analyze the algorithm performance. The name of the instance types and their parameters are summarized in Table I. Notice that instances of type A9 have a very large number of synchronization nodes. In fact, we will use these non-realistic instances to show the limits of our solution approach.

| Inst. | \(|I|\) | \(|B|\) | \([F_{min}, F_{max}]\) in % |
|-------|-------|-------|--------------------------|
| A1    | 15    | 3     | \([10,20]\)              |
| A2    | 15    | 3     | \([25,35]\)              |
| A3    | 40    | 8     | \([10,20]\)              |
| A4    | 40    | 8     | \([25,35]\)              |
| A5    | 100   | 20    | \([10,20]\)              |
| A6    | 100   | 20    | \([25,35]\)              |
| A7    | 200   | 40    | \([10,20]\)              |
| A8    | 200   | 40    | \([25,35]\)              |
| A9    | 200   | 150   | \([25,35]\)              |

Table 1: Instance characteristics
5.2. Solving to optimality

Table 2 summarizes the experimental results obtained for different combinations of valid inequalities within a time limit of one hour. Preliminary experiments were conducted to identify the promising combinations of valid inequalities. The following 4 different combinations have been kept, that we will denote \([1)-(6)\), \([1)-(10)\), \([1)-(8)+lift\) and \([1)-(8)+lift_{20}\). Formulation \([1)-(6)\) corresponds to the basic SBTP MIP; \([1)-(10)\) to the SBTP MIP enhanced with synchronization and headway inequalities; \([1)-(8)+lift\) to the SBTP MIP with synchronization inequalities and all the lifted headway inequalities obtained using Algorithm 5. Finally \([1)-(8)+lift_{20}\) is similar to the latter formulation where only 20% of the generated lifted headway inequalities are included. Indeed, the inequalities maintained in the formulation correspond to those having the highest number of synchronization variables in their left hand sides. This ratio have been set after preliminary experiments: this value corresponds to the best compromise between the quality of the solution and the computation time.

For each of the 4 combinations, Table 2 provides the following entries:

\#Nodes : the average number of nodes in the branching tree,
Rel : the average value of the objective function at the root node,
Gap : the average relative error between the best obtained value and the best obtained lower bound,
CPU : the average computational time in seconds.

In the first block of rows of Table 2, we can see that the original formulation of SBTP is intractable since we obtain large gaps in one hour of computation time. Notice that no feasible solution can be found within this time limit for instances A9.

The second block of rows of Table 2 corresponds to the case where all the valid inequalities \([1)-(10)\) have been added to the original formulation. We can see that adding these inequalities to the original formulation leads to very good solutions within reasonable CPU times, the gaps are indeed significantly reduced.
| A1   | 383673.1  | 408.6  | 48.7% | 3600 | 1618.8 | 166.7  | 0%   | 15.3 | 1776.4 | 166.7  | 0%   | 23.8 | 3110.4 | 166.7  | 0%   | 45.4 |
| A2   | 190809.8  | 588.0  | 157.8%| 3600 | 28089.3| 153.0  | 0%   | 225  | 18412  | 152.9  | 0.1% | 365  | 25915.7 | 153.0  | 0.2% | 368.7 |
| A3   | 73774.3   | 946.3  | 70.5% | 3600 | 32667.8| 399.8  | 0.2% | 450.5| 12897.5| 399.7  | 0.2% | 512.1 | 19659.7 | 399.8  | 0.2% | 436.4 |
| A4   | 48458     | 1449.7 | 172.5%| 3600 | 33.6   | 385.3  | 0%   | 9.3  | 21.97  | 385.2  | 0%   | 34.4 | 0     | 385.2  | 0%   | 22   |
| A5   | 16855.9   | 2439.1 | 64.4% | 3600 | 142089.5| 1023.6 | 0.4% | 2851.7| 39601.4| 1023.5 | 0.6% | 3150.4| 91889.1| 1023.5 | 0.6% | 3254 |
| A6   | 14800.6   | 3585.6 | 183.7%| 3600 | 126.9  | 933.1  | 0%   | 56.3 | 376.2  | 932.8  | 0%   | 1068.5| 799.6 | 933.3  | 0%   | 354.2 |
| A7   | 644.3     | 5757.7 | 151.2%| 3600 | 30410.2| 1991.4 | 0%   | 1854.8| 5547.8 | 1991.2 | 0.2% | 3409.4| 16305.4| 1991.3 | 0.1% | 2915.4|
| A8   | 559.7     | 7233.8 | 277.3%| 3600 | 761.8  | 1876.3 | 0%   | 425.3| 327.1  | 1876.0 | 0.6% | 2120.3| 1133.4 | 1876.7 | 0.1% | 2039.6|
| A9   | 1.0       | -      | -     | 3600 | 1.0    | 4806.404| 48.83%| 3600 | 1.0    | -      | -     | 3600 | 1.0    | -     | -     | 3600 |

Table 2: Results for the instance types A1-A9 using CPLEX 12.3 and different combinations of valid inequalities.
in comparison with the ones obtained for the original formulation. Instances A1-A8 are almost all solved to optimality in less than 48 minutes. More precisely, all instances A1-A2, A4 and A6-A8 have been solved to optimality in less than 30 minutes and only few A3 and A5 instances have not been solved to optimality within one hour. The average number of nodes in the tree search decreases significantly as well as the average value of the objective function at the root node. We can notice that for the A9 instances, even if the gap is important, at least a feasible solution is found.

If we look at the results provided in Table 2 for the third block corresponding to the case where all the lifted headway inequalities are added to (1)-(8) formulation, we can see that the average values of the objective function at the root node are improved regarding the original formulation but are comparable to (1)-(10) with a greater computational effort. If we compare the results of the two last blocks of Table 2 corresponding to the formulation including lifted headway inequalities, we can notice that keeping 20% of the generated lifted inequalities leads to very good results both in terms of gaps and CPU times regarding (1)-(8)+lift formulation. Particularly, the gaps obtained for A7-A8 instances are better with lower computational times. We can see that a greater number of nodes are explored with no increase of CPU times leading to good solutions.

To sum up this first set of experiments, we can conclude that the valid inequalities proposed for the SBTP MIP are useful, they significantly improve the gaps and the values of the objective function at the root node. However, a huge computational effort is required. The lifted inequalities can also be useful to reach the same efficiency. In that case however, it can be tricky to find the relevant and adequate set of inequalities to maintain. To enhance the efficiency, a perspective of this work is to develop a Branch-and-Cut solving method for the SBTP.
6. Solving up to 3%

Table 3 provides a comparison between the best solution found using the (1)-(10) formulation with a stopping criterion of 3% of relative gap and a Multi-start Iterated Local Search (MILS) proposed by Ibarra-Rojas & Rios-Solis (2012) for the same problem. The idea of this MILS is to use constraint propagation shown in Algorithm 1 to define and explore the search space. In particular, constraint propagation is used to design randomized constructive algorithms and implement line departure times shifting within the feasible space to induce more synchronizations.

<table>
<thead>
<tr>
<th></th>
<th>(1)-(10)</th>
<th>MILS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gap</td>
<td>Time</td>
</tr>
<tr>
<td>A1</td>
<td>1.9%</td>
<td>1.7</td>
</tr>
<tr>
<td>A2</td>
<td>1.3%</td>
<td>3.0</td>
</tr>
<tr>
<td>A3</td>
<td>2.6%</td>
<td>33.5</td>
</tr>
<tr>
<td>A4</td>
<td>0.8%</td>
<td>4.5</td>
</tr>
<tr>
<td>A5</td>
<td>2.1%</td>
<td>41.9</td>
</tr>
<tr>
<td>A6</td>
<td>1.2%</td>
<td>18.6</td>
</tr>
<tr>
<td>A7</td>
<td>1.7%</td>
<td>91.2</td>
</tr>
<tr>
<td>A8</td>
<td>1.7%</td>
<td>85.5</td>
</tr>
</tbody>
</table>

Table 3: Results of solving instances A1-A8 implementing inequalities (1)-(10) with a stopping criterion of 3% of relative gap versus Multi-start Iterated Local Search (MILS) proposed by Ibarra-Rojas & Rios-Solis (2012).

To get the gap for the MILS implementation, we compute the deviation of the feasible solution obtained by MILS and the best upper bound found by CPLEX 12.3.

The exact approach based on (1)-(10) formulation leads to the best results for all instance types in terms of relative gaps. Although, MILS is more efficient considering the execution time, our exact approach reaches the stopping criterion in less than 2 minutes for most of the instances.
7. Conclusions

We define an exact solution approach for the NP-hard synchronization bus
timetabling problem. This problem determines regular spaced departure time
for all the trips of each line to allow well timed passenger transfers and reduce
congestion of buses belonging to different lines at common stops. The flexibility
in the SBTP given by headway time bounds (instead of a fixed headway time)
allows us to define different families of valid inequalities to tighten the SBTP
MIP.

Our solution approach is to strengthen the SBTP MIP using our proposed
valid inequalities and implement an integer linear solver. Numerical results
show that high quality solutions (optimal for most cases) can be found for large
instances of SBTP in a short time. Moreover, there is a fast convergence of our
approach to solutions with less than 3% of relative deviation from the optimal
solution in seconds. 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 inter-
esting research lines such as determining the dimension of the valid inequalities
proposed in this study. A natural improvement for this work is to develop a
polyhedral study along with a branch-and-cut approach to handle unsolvable
instances. Indeed, the lifting procedure that does not show significant empirical
results in this work may be of particular interest in a branch-and-cut.

Integration of SBTP with other subproblems of transit network planning
such as vehicle and crew scheduling is a challenging research area. Moreover,
the generalization of SBTP to cover the entire day instead of short planning
periods is needed to define accurate integrated approaches.
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.

References


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


