# Transit Network Timetabling Problem: A Case Study of the Transport Operator in Sinu (Colombia)

Andrés Rueda López[a,*], Juan Sebastian Agudelo Gomez[b]

[a] *Universidad de Los Andes, Bogotá, Colombia*
[b] *Universidad de Los Andes, Bogotá, Colombia*

---

**Abstract**

*Keywords:* Networks, Optimization, Transportation, Public transport systems, Transit Network Timetabling Problem

---

## 1. Introduction

Public transport systems (PTS) play an important role in the development of urban populations and have a significant influence on the quality of life for residents. In recent years, the need to develop more efficient PTS has grown, as it has become evident that the economic and environmental performance of cities can be improved by efficiently connecting resources and mass population mobility [1]. One of the main concerns in developing PTS and making accurate operational decisions is measuring the temporal and spatial coverage of the systems for the population. While spatial coverage or accessibility refers to ensuring that a large number of people have access to a transit station within a short distance, temporal coverage means that PTS are available when customers need the service [2].

The increasing demand for PTS, driven by the rapid growth of urban populations, implies that PTS planning becomes exponentially more complex. If this systems are unable to adapt rappidly and in an efficient way to those changes, it can result in an increase of waiting times or PTS congestions [3]. For this reason, it is essential that Public Transport Companies (PTC) allocate some of their resources to develop smart planning strategies that can quickly and accurately adapt to the needs of users.

A robust plan is the base for a PTC to satisfy the demand, give users the best possible experience, and guarantee a fair working policy for their drivers. A common transit operations planning process consists of four stages: first, designing the route network system; second, developing a timetable for each route; third, scheduling vehicles to match the timetable trips; and finally, crew scheduling [4].

This paper focuses on the second stage of the robust planning process by designing a bus timetable for one route of the *Transport Operator of Sinu (Colombia)*, dealing with what is known as a *Transit Network Timetabling Problem* (TNTP). This involves scheduling a timetable to operate over pre-planned routes, where all vehicles depart from and arrive at a single depot. This well-know problem can typically be solved in two possible ways, periodic timetabling represented as *trips/hour* and non-periodic timetabling where each trip has a fixed departure time [5].

The proposed solution should benefit the company by either reducing operational costs or increasing revenue through an efficient and replicable implementation across different routes. However, because the company provides a public service, it is equally important that the resulting solution guarantees temporal

*Corresponding author.

accessibility for users. Since users always have a preferred departure time, aligning the service with this preference in demand behavior will increase user satisfaction and should also be considered in the problem solution [6]. The common strategies to achieve a solution for this kind of problems involve exact and/or approximate methods.

Exact methods often formulate timetabling as an optimization problem. For complex combinatorial problems, classic optimization techniques such as column generation are commonly used. These methods enable the creation of variables without the need to explore the entire solution space directly, facilitating the generation of efficient schedules or shifts [7]. Additionally, given the mixed-integer nature of these problems, computational acceleration techniques such as branch-and-cut can be applied, utilizing both branching and cutting to enhance solution efficiency [8]. It has also been shown that these techniques are beneficial when applied to scheduling problems formulated in networks with resource constraints [9].

Approximation methods based on heuristics offer an alternative for reducing the complexity of solving transit network timetable problems (TNTP). Heuristics often frame TNTP as a graph problem, where nodes represent feasible departure times and arcs indicate possible sequences of these times, transforming the solution into a shortest path problem. In [10], a multi-objective label-correcting algorithm is proposed to minimize both user waiting time (service metric) and the number of empty seats (operations metric) by comparing near-optimal solutions using a Pareto strategy.

In recent years, *Reinforcement Learning* (RL) has emerged as a method to solve timetabling problems in real-time. RL treats timetabling as a Markov Decision Process (MDP), where future decisions depend only on the current state [11]. RL agents interact with simulated environments, receiving rewards for actions like departure timing. In [12] is proposed a reward function that aims to optimize bus utilization while minimizing waiting times, using a Deep Q-Network (DQN). However, RL agents may select infeasible actions, requiring a masking step to ensure that only valid actions are considered [13].

When dealing with large-scale scenarios, formulating and solving optimization problems can become challenging. In these cases, it is common to adopt a standard formulation. Most bus scheduling formulations aim to minimize user waiting times while adhering to resource constraints [14]. One formulation that follows this for a single-direction bus line is presented in [15]. There are several extensions to the standard formulation presented in [15], such as the one presented by [14], where the author introduces additional constraints, including vehicle availability for successive trips, vehicle capacities, and permitted dispatching headway variations.

This paper aims to model and optimize one bus line of the Transport Operator of Sinu (Colombia), using real-world data provided by the company. This article introduces a novel reformulation, replacing the MIP approach from authors like [15] with a network-based model. This reformulation leverages the computational efficiency of network optimization, particularly for integer problems, aiming to deliver scalable solutions that can be easily extended to larger transport networks, thus avoiding computational limitations [16].

## 2. Problem statement

Consider a route in the system represented as a directed graph $G = (\mathcal{N}, \mathcal{A}, \mathcal{T})$, where $\mathcal{N} = \{s, 1, \ldots, e\}$ is the set of nodes, $\mathcal{A} \subseteq \{(i, j) \mid i \in \mathcal{N}, j \in \mathcal{N}, i \neq j\}$ is the set of arcs, and $\mathcal{T} = \{0, \varphi, \ldots, t_{\max}\}$ is the set of time intervals, each of length $\varphi$. Let $P = \{p_1, p_2, \ldots, p_m\}$ be the set of buses available for the route, each with an identical capacity $k$. Every bus $p \in P$ must begin the route at the node $s \in \mathcal{N}$, and it may depart at any time $t \in \mathcal{T}$. Figure 4 exemplifies the main components described previously.

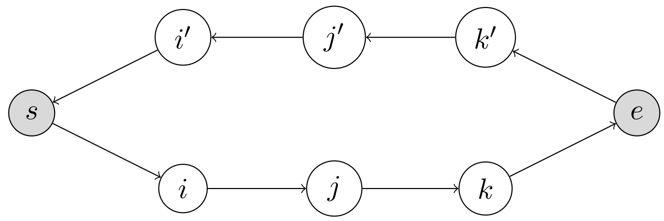


Figure 1: General form of the route

Moreover, a path $\Omega$ is defined as an ordered sequence of nodes, representing the order that each bus $p \in P$ must follow while traveling along $\mathcal{G}$. Each bus $p \in P$ must return to node $s \in \mathcal{N}$ by the end of the time horizon, under the assumption that the parking spot is only available at this node. This implies that every bus that begins the route must complete the entire cycle in $\mathcal{G}$ before finishing.

Let $y_{(i,j),p,t}$ be a binary variable that indicates whether arc $(i,j)$ is traversed by bus $p \in P$ at time $t \in \mathcal{T}$. Specifically, if $y_{(i,j),p,t} = 1$, it means that at time $t \in \mathcal{T}$, bus $p \in \mathcal{P}$ is at node $i \in \mathcal{N}$ and has begun its journey towards node $j \in \mathcal{N}$, allowing passengers at node $i \in \mathcal{N}$ to board. For simplicity, the boarding time of passengers is considered negligible.

The traversal time of arc $(i,j) \in \mathcal{A}$ is assumed to be deterministic and provided, which is then converted into time units $\varphi$ and denoted by $\tau_{i,j}$. Additionally, each node $i \in \mathcal{N}$ has an associated parameter $w_{i,t}$, which is assumed as a deterministic value and is an estimation of the number of passengers arriving at node $i$ at time $t$. By defining $W_{i,t}$ as the number of people at node $i \in \mathcal{N}$ at time $t \in \mathcal{T}$, it is possible to model the number of people for each arc at each time instant as:

$$W_{i,t} = w_{i,t} - \left[ \sum_{p \in P} (k - s_{i,p,t}) y_{(i,j),p,t} \right], \quad \forall (i,j) \in \mathcal{A},\, t \in \mathcal{T} \,|\, t = 0 \tag{1}$$

$$W_{i,t} = W_{i,(t-1)} + w_{i,t} - \left[ \sum_{p \in P} (k - s_{i,p,t}) y_{(i,j),p,t} \right], \quad \forall (i,j) \in \mathcal{A},\, t \in \mathcal{T} \,|\, t > 0 \tag{2}$$

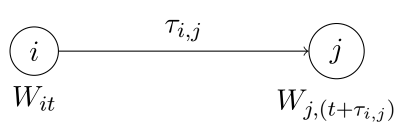


Figure 2: Atributes for each section of the route

Figure 2 summarizes the balance equations (2) and (1). Furthermore, $s_{i,p,t}$ represents the amount of people in the bus $p \in P$ at the time $t \in \mathcal{T}$ in the node $i \in \mathcal{N}$. To emulate the dynamics of getting on and off the bus at each of its stops along the route, let $u$ be a parameter that indicates the proportion of people in the bus that get off in each node, assuming that for every node this parameter is constant.

$$s_{i,p,t} = \lfloor s_{(i-1),p,(t-\tau_{i,i-1})} (1 - u y_{(i,j),p,t}) + (k - W_{i,t}) \rfloor \tag{3}$$

Because the amount of people that arrives to each node is out of control for the decision maker, what can be achieved is the amount of people that is waiting in each node. For this reason, the metric in the system, denoted by $Q$, represents the total number of people that is waiting for transportation and can be calculated as follows:

$$Q = \sum_{(i)\in\mathcal{N}} \sum_{t\in\mathcal{T}} W_{i,t} \tag{4}$$

Summarizing, the ~~timetabling problem~~ Transit Network Timetabling problem can be written as:

$$\arg \min_{y_{(i,j),p,t}} \sum_{(i)\in\mathcal{N}} \sum_{t\in\mathcal{T}} W_{i,t} \tag{5}$$

In the following section, the problem is translated into a mixed integer non-linear program and them translated into an equivalent mixed integer linear program (MILP).

### *2.1. Mixed Integer Non-linear Formulation*

A mixed integer non-linear program is proposed as an approximation to the problem above described. This approximation aims to generalize to any route without considering the interdependence that may exist between different routes. Consequently, a set of reformulation are proposed to hold linearity and transform the problem into a MILP.

#### *2.1.1. Nomenclature*

*Sets*

- $\mathcal{A}$: Set of arcs, $\mathcal{A} \subseteq \{(i,j)|i \in \mathcal{N}, j \in |, i \neq j\}$.
- $\mathcal{N}$: Set of nodes (or stations), $\mathcal{N} = \{0, 1, 2, \ldots, e\}$.
- $\mathcal{P}$: Set of buses, $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$.
- $\mathcal{T}$: Set of time intervals, $\mathcal{T} = \{0, \varphi, 2\varphi, \ldots, t_{\max}\}$.

*Parameters*

- $d_{i,t}$: Number of people arriving at node $i \in \mathcal{N}$ at time $t \in \mathcal{T}$ [Units].
- $I^0$: Initial charge of any bus [Units], with $I^0 = 0$.
- $K$: Max number of cycles that can be made by any bus.
- $k$: Capacity of any bus [Units].
- $M$: Big number.
- $\tau_{i,j}$: Travel time from node $i \in \mathcal{N}$ to node $j \in \mathcal{N}$ [Time units $\varphi$].
- $u$: Rate of decline at any node, $u \in [0, 1]$.
- $W_{i,t}$: Initial amount of people at node $i \in \mathcal{N}$ at time $t \in T$.

*Decision variables*

- $o_{i,t,p}$: Amount of people descending in the node $i \in \mathcal{N}$ at time $t \in \mathcal{T}$ from the bus $p \in \mathcal{P}$.
- $s_{t,p}$: Amount of people at time $t \in \mathcal{T}$ in the bus $p \in \mathcal{P}$.
- $w_{i,t,p}$: Amount of demand attended in the node $i \in \mathcal{N}$ at time $t \in \mathcal{T}$ by the bus $p \in \mathcal{P}$.
- $W_{i,t}$: Non-attended demand in the node $i \in \mathcal{N}$ at time $t \in T$.

- $x_{i,j,t,p} = \begin{cases} 1 & \text{if the bus } p \in \mathcal{P} \text{ starts the route along the arc } (i,j) \in \mathcal{A} \\ & \text{at time } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$

- $y_{i,j,t,p} = \begin{cases} 1 & \text{if the bus } p \in \mathcal{P} \text{ is travelling along the arc } (i,j) \in \mathcal{A} \\ & \text{at time } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$

- $z_{i,t,p} = \begin{cases} 1 & \text{if the bus } p \in \mathcal{P} \text{ is at node } i \in \mathcal{N} \\ & \text{at time } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$

*2.1.2. Objective Function*

The objective function in (6) is designed to minimize the cumulative waiting time for passengers across all nodes and time intervals. This function can be interpreted as an effort to enhance system efficiency by reducing delays for passengers. Thus, the objective function is:

$$\min \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} W_{i,t} \tag{6}$$

where $W_{i,t}$ represents the unmet demand, or the number of passengers waiting at node $i$ at time $t$. Minimizing this value optimizes the route schedule to ensure passengers spend as little time waiting as possible, thereby improving service quality and efficiency. In other words, by seeking to minimize unmet demand, the model aims to serve the maximum number of potential customers of the service. Note that the objective function does not consider any associated operating costs, as it is strongly assumed that the operation is economically viable. That is, it is assumed that the operating costs are offset by the bus fare. This assumption can be disregarded if relevant information is available; however, in this case, the goal is to consider a general model.

*2.1.3. Constraints*

$$\sum_{t \in T} x_{i,j,t,p} \leq K, \quad \forall (i,j) \in \mathcal{A}, p \in P | i = 0, j = i+1 \tag{7}$$

$$\sum_{t \in T} x_{i,j,t,p} = 0, \quad \forall p \in P(i,j) \in \mathcal{A} | i \neq 0, j \neq i+1 \tag{8}$$

$$\begin{aligned} y_{i,j,\delta,p} \leq x_{i,j,t,p}, \quad & \forall\, (i,j) \in \mathcal{A},\, t \in \mathcal{T},\, p \in \mathcal{P}, \\ & \delta \in \{t, t+1, \ldots, \min(t + \tau_{i,j}, |\mathcal{T}| + 1)\} \end{aligned} \tag{9}$$

~~Constraints (7)–(9) define the conditions for the initiation of the route and the activation of the arcs.~~
Equations (7) and (8) define the conditions for initiating each route and for activating specific arcs. Equation (7) ensures that each bus starts at the initial node $i = 0$, limiting the maximum number of route activations per bus to $K$. Minewhile, Equation (8) sets that for all nodes except the starting node $i = 0$, there are no initial route activations. These constraints allow each bus to start only at the origin node and maintain consistency in the initial setup.

The set of constraints (9) ensure that the travel decision variable $y_{i,j,\delta,p}$ aligns with the initiation of the trip along arc $(i,j)$. Where $\delta$ represents time steps ranging from $t$ to $t + \tau_{i,j}$, covering the duration of travel on the arc. This prevents the bus from being in multiple locations at once and ensures consistency between trip initiation and travel time along each arc.

$$z_{i,t,p} + z_{j,t,p} \geq y_{i,j,t,}, \quad \forall (i,j) \in \mathcal{T}, t \in \mathcal{T}, p \in \mathcal{P} \tag{10}$$

$$\sum_{t \in \mathcal{T}} z_{i,t,p} \leq 1, \quad \forall p \in \mathcal{P}, i \in \mathcal{N} | i \neq 0 \tag{11}$$

$$\sum_{n \in \mathcal{N}} z_{i,t,p} \leq 1, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} \tag{12}$$

$$z_{i,t,p} = z_{i,t-1,p} - \sum_{j \in \mathcal{N}} x_{i,j,t-1,p}, \forall p \in \mathcal{P}, t \in \mathcal{T} | i = 0 \tag{13}$$

$$z_{i,t,p} = \sum_{(k,i) \in \mathcal{A}} y_{k,i,t-1,p}, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} \tag{14}$$

$$z_{i,t,p} = 1, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} | i = 0, t = 0 \tag{15}$$

$$z_{i,t,p} = 0, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} | t = 0 \tag{16}$$

$$\sum_{t \in \mathcal{T}} z_{i,t,p} = 1, \quad \forall i \in \mathcal{N}, p \in \mathcal{P} | i = e \tag{17}$$

$$\sum_{(i,j) \in \mathcal{A}} x_{i,j,t,p} \leq z_{i,t,p}, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} \tag{18}$$

~~Constraints (10)–(17) define the relationships between the presence variable $z_{i,t,p}$ and the variables $x_{i,j,t,p}$ and $y_{i,j,t,p}$. These constraints model the presence of the bus at each node within the graph.~~

Constraints (10) to (18) define the relationship between the bus presence at nodes and the arcs traversed. These constraints manage bus location and movement as follows. Equation (10) links the presence variables $z_{i,t,p}$ and $z_{j,t,p}$ to the arc traversal variable $y_{i,j,t,p}$, ensuring that the bus is indeed at nodes $i$ and $j$ when traveling along arc $(i,j)$. Furthermore, (11) and (12) limit the presence of each bus to a single node at any time $t$ by enforcing that no bus appears at multiple nodes simultaneously. Follow, (13) dictates that if a bus departs from the initial node, its presence variable updates accordingly, simulating the departure from node $s$ and enabling it to enter the graph properly. Least but not last, (18) constrains the traversal variable $x_{i,j,t,p}$ based on the presence of the bus at node $i$, ensuring the bus departs only from nodes it is currently present at. These constraints provide a structured model for tracking bus location and arc traversal in real-time, ensuring efficient use of bus resources across the network.

$$s_{t,p} = s^0 + \sum_{n \in \mathcal{N}} w_{i,t,p} - \sum_{i \in \mathcal{N}} o_{i,t,p}, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} | t = 0 \tag{19}$$

$$s_{t,p} = s_{t-1,p} + \sum_{n \in \mathcal{N}} w_{i,t,p} - \sum_{i \in \mathcal{N}} o_{i,t,p}, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} | t > 0 \tag{20}$$

$$s_{t,p} \leq k, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} \tag{21}$$

~~Constraints (14) and (16) model the bus charge along the time horizon for each bus.~~

Equations (19)–(21) manage the on-bus passenger load $s_{t,p}$ as buses travel through the network. Equation (19) initializes the passenger load for each bus based on initial conditions, while (20) updates the load dynamically, considering new boardings and alightings at each time step. Finally, (21) enforces the maximum capacity constraint, ensuring that each bus does not exceed its specified limit $k$ at any time.

$$W_{i,t} = d_{i,t} - \sum_{p \in P} w_{i,t,p}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T} | t = 0 \tag{22}$$

$$W_{i,t} = W_{i,t-1} + d_{i,t} - \sum_{p \in P} w_{i,t,p}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T} | t > 0 \tag{23}$$

$$W_{i,t,p} \leq M z_{i,t,p}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} \tag{24}$$

$$W_{(i,t)} \geq w_{i,t,p}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T} p \in \mathcal{P} \tag{25}$$

$$\sum_{n \in \mathcal{N}} w_{i,t,p} \leq k - s^0, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} | t = 0 \tag{26}$$

$$\sum_{n \in \mathcal{N}} w_{i,t,p} \leq k - s_{t-1}, \quad \forall t \in \mathcal{T}, p \in \mathcal{P} | t = 0 \tag{27}$$

~~Constraints (17) through (22) model the non attended and new demand for each node.~~ Equations (22)–(23) express the unmet demand present at each node is modeled. Equation (22) models how the unmet demand at each node for the initial time step corresponds to the incoming demand at that moment, $d_{i,t}$, minus the demand served by the buses (note that this term is zero, but it is included for generality). Equation (23) generally models the amount of unmet demand at each time step and at each node, considering that people not served at that node in the previous time step $(t-1)$ remain waiting at time $t$ in the same node $i$. Equation (24) performs a binary activation using a large $M$ to ensure that demand can only be served if the binary variable $z_{i,t,p}$, which indicates the presence of a bus at that node and time, is activated. Equations (25) – (27) model the upper and lower bounds of the decision variable $w_{i,t,p}$, which represents the demand that each bus can serve, limiting it by its capacity, previous load, and potential demand to be loaded.

Note that in the previous section, the number of people at each node was modeled through Equation (1), which clearly indicated nonlinearity due to the product of the variable $y_{(i,j),t,p}$, representing whether the arc was traversed, and $s_{i,p,t}$, which indicated the load of each bus at each time and node. To linearize the model in its constraints, the previously discussed decomposition of multiple constraints was proposed, along with the incorporation of the decision variable $w_{i,t,p}$, which stores information about the demand served by each bus. This variable subsequently serves as a limit to model the bus load throughout the planning horizon.

$$o_{i,t,p} \leq s_{t-1,p}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} | t > 1 \tag{28}$$

$$o_{i,t,p} \leq k z_{i,t,p}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} | t > 1 \tag{29}$$

$$o_{i,t,p} \leq k(1 - z_{i,t,p}), \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} | i = 0, i = e, t > 1 \tag{30}$$

$$o_{i,t,p} = 0, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} | t = 0 \tag{31}$$

$$o_{i,t,p} \leq u_i s_{t-1,p} \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} \tag{32}$$

~~Where constraints (23) through (27) model the descent of the people from the buses in each node.~~ The constraints (28) through (32) model the descent of passengers from the buses at each node and time step. Constraint (28) ensures that the number of people descending at each node, time, and bus $o_{i,t,p}$ does not exceed the bus load $s_{t-1,p}$ from the previous time step, maintaining that only those already on the bus can disembark. Constraint (29) limits the number of people descending $o_{i,t,p}$ by the bus capacity $k$ and the presence indicator $z_{i,t,p}$, allowing passengers to disembark only when the bus is present at that node. Constraint (30) specifies that the number of passengers descending $o_{i,t,p}$ cannot exceed a predefined maximum proportion $k(1-z_{i,t,p})$, ensuring a controlled descent rate based on the bus's status at the node (i.e., entering or leaving). Constraint (31) enforces that no passengers disembark at the initial time step $t = 0$ by setting $o_{i,t,p} = 0$ for $t = 0$. Finally, constraint (32) sets an upper limit for the descent $o_{i,t,p}$ as a proportion $u$ of the previous load $s_{t-1,p}$, ensuring that the disembarkation rate is limited based on the initial load of the bus.

$$x_{i,j,t,p} \in \{0,1\}, \quad \forall (i,j) \in \mathcal{A}, t \in \mathcal{T}, p \in \mathcal{P} \tag{33}$$

$$y_{i,j,t,p} \in \{0,1\}, \quad \forall (i,j) \in \mathcal{A}, t \in \mathcal{T}, p \in \mathcal{P} \tag{34}$$

$$z_{i,t,p} \in \{0,1\}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} \tag{35}$$

$$w_{i,t,p} \in \mathbb{Z}^{+} \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} \tag{36}$$

$$s_{t,p} \in \mathbb{Z}^{+} \qquad \forall t \in \mathcal{T}, p \in \mathcal{P} \tag{37}$$

$$o_{i,t,p} \in \mathbb{Z}^{+} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} \tag{38}$$

$$W_{i,t} \in \mathbb{Z}^{+} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, p \in \mathcal{P} \tag{39}$$

~~Expressions (28)–(34) set the nature of the decision variables.~~ Equations (33), (34), and (35) express the binary nature of the variables $x_{i,j,t,p}$, $y_{i,j,t,p}$, and $z_{i,t,p}$, which serve to model whether the bus is traversing the arc $(i, j)$ or present at node $(i)$. On the other hand, Equations (37) to (39) indicate the positive integer nature of the variables that model the dynamics of loading and unloading passengers from the buses, along with the demand at each node.

## 3. Implementation for one line

### *3.0.1. Demand*

A small instance was built to test the previous model in a simple case. For this case, the parameters were generated using information of one line of the system, especially the route know as *Pradera 27*. The only information the company can provide corresponds to the estimated demand at each time instant. In this case, the demand was discretized into 15-minute intervals. A strong assumption made regarding this demand is that, based on the information provided by the company, it is necessary to assume that demand is uniformly distributed across each station along the route. Therefore, each node has the same demand distribution. Figure 3 shows the estimated demand for each node over the 64 intervals constructed for a day of operation. .

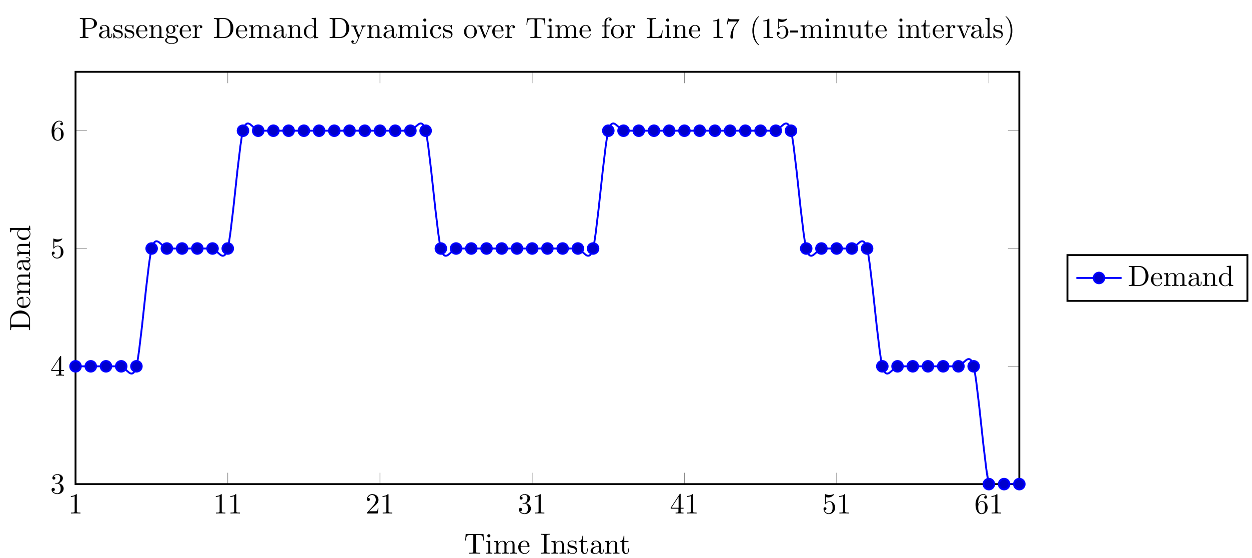


Figure 3: Demand Dynamics over Time for Line 17.

*3.0.2. Graph*

The graph illustrated below is organized with node $s$ as the starting point of the route, which progresses sequentially through a series of connected nodes and culminates at node 6, the designated endpoint. This structure provides a clear path from the origin to the destination, emphasizing the directional flow between nodes. The company does not provide any spatial information; therefore, an approximation was constructed to create a graph suitable for implementing the model. Note that the model can be adjusted to accommodate routes of any size. Figure 4 illustrates an example of how the route is represented as a graph.

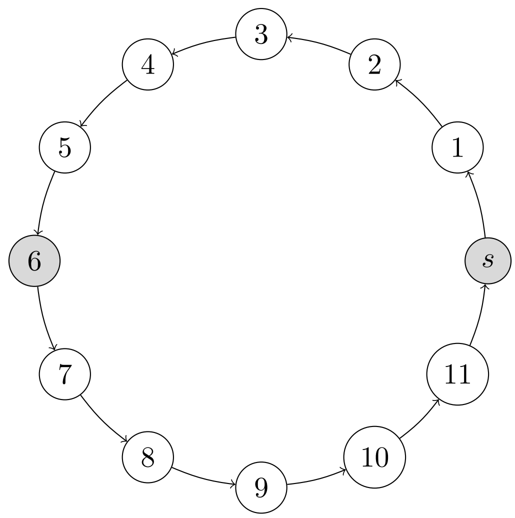


Figure 4: Example graph

The temporal distance between each pair of connected nodes $(i, j)$ within the set of arcs $\mathcal{A}$ is uniformly defined as $\tau_{i,j} = 1$, ensuring consistent transition times across all connections. Note that a value of $\tau_{i,j} = 1$ implies that the time to traverse any arc corresponds to 15 minutes, given that the time discretization is set to 15-minute intervals.

~~The demand at each node during different times is generated by summing up individual normal distributions, capturing the temporal variations and characteristics of demand effectively. This modeling approach allows us to reflect the fluctuating nature of demand as an approximation (see the ilustration for details).~~

*3.0.3. Other parameters*

Furthermore, the proportion of descent at each node is determined by a stochastic process, where each value is a uniformly distributed random number between 0 and 1. This method introduces an element of randomness into the model, influencing the dynamics and outcomes of the system in a probabilistically varied manner. Finally, the model was tested for 20 buses with equal capacity ($k = 50$) in set of 64 time intervals.

*3.0.4. Results*

The computational experiment was conducted on a device named IIIND-PIRYBKR3. The system is equipped with a 12th Generation Intel(R) Core(TM) i7-1265U processor with a base clock speed of 1.80 GHz, 16 GB of installed RAM (15.7 GB usable), and runs a 64-bit operating system based on the x64 architecture. The optimization model was implemented and solved using the PuLP library, a linear programming package in Python (version 3.11.5). The problem was solved using the Gurobi solver through the PuLP interface with specific settings: the relative optimality gap (`MIPGap`) was set to 0% and the time limit (`TimeLimit`) was set to 600 seconds. This configuration ensured that the model reached optimality or stopped after the defined time limit, depending on which condition was met first. The experiment was executed on the computational setup described above, providing adequate computational resources to handle complex optimization tasks.

Table 1: Bus Departure and Arrival Times for the First Arc

| **Bus Number** | **Departure Time** | **Arrival Time** | **Value** |
|---|---|---|---|
| 1 | 1 | 13 | 1.0 |
| 18 | 2 | 14 | 1.0 |
| 11 | 3 | 15 | 1.0 |
| 17 | 4 | 16 | 1.0 |
| 7 | 5 | 17 | 1.0 |
| 1 | 6 | 18 | 1.0 |
| 14 | 7 | 19 | 1.0 |
| 12 | 8 | 20 | 1.0 |
| 19 | 9 | 21 | 1.0 |
| 13 | 10 | 22 | 1.0 |
| 8 | 11 | 23 | 1.0 |
| 2 | 12 | 24 | 1.0 |
| 16 | 13 | 25 | 1.0 |
| 10 | 14 | 26 | 1.0 |
| 6 | 15 | 27 | 1.0 |
| 4 | 16 | 28 | 1.0 |
| 5 | 17 | 29 | 1.0 |
| 9 | 18 | 30 | 1.0 |
| 3 | 19 | 31 | 1.0 |

In Table 1, the departure times from the initial node and the subsequent returns to these node are presented. Given the simplicity of the assumptions, these times significantly influence the deterministic nature of the problem, thereby determining the total solution. Note that the solution provided by the optimizer under these conditions is trivial, as it dispatches buses at the start of each new time instant. In other words, the optimal solution for these conditions corresponds to sending a bus every 15 minutes. Furthermore, this model does not consider bus cycles, meaning that once a bus leaves the system, it does not re-enter. This is currently a simplification that could be refined to better approximate real operations. However, this initial instance only aims to evaluate the model's behavior in a real scenario, to later adapt and improve it. As presented in the following section, a modification to the objective function of the problem allows for better model performance.

However, the model demonstrated robustness in a real instance by achieving optimality in a runtime of 10 minutes and 36 seconds, which is an excellent performance considering that, in operational terms, it provides planning times that allow decision-makers to execute multiple instances without a strong time constraint. The objective function value is 52.508, which, as we will see later, can be improved by interpreting the problem differently. Additionally, the dynamics of the buses along the graph and the bus load behavior followed the expected real pattern, indicating a well-implemented model. The reader can visit the following link to visualize this instance on the graph, where each bus moves along the graph and displays its load at each time instant.

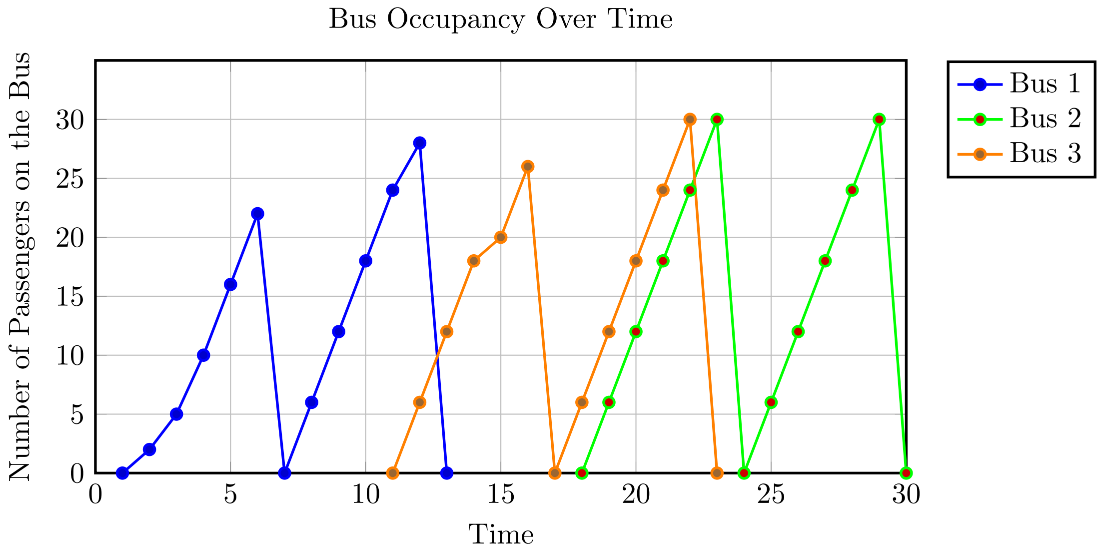


Figure 5: Occupancy over time for (3) buses

Figure 5 illustrates the occupancy levels for the 3 first buses in the float during the travel. Notably, the occupancy is zero at the beginning, midpoint, and end of the route, corresponding to the terminal nodes. This pattern highlights the nodes where each bus journey either starts or concludes.

Note that the bus load behavior indicates that its occupancy increases at each time instant, although the rate of increase varies according to the changing demand at each instant, as well as the bus's descent rate at each node. Additionally, observe that the bus never approaches its maximum capacity (which is $k = 50$), suggesting that bus space may be underutilized. This issue is addressed in the following section, where the objective function is modified to maximize bus occupancy, which in turn will improve the level of unmet demand.

### *3.1. Model aceleration and tunning*

For the next deadline...

## References


[1] J. Doe, Public transport systems and urban mobility, Urban Mobility Journal 10 (3) (2021) 123–135. doi:10.1000/xyz123.

[2] S. A. Mamun, N. E. Lownes, J. P. Osleeb, K. Bertolaccini, A method to define public transit opportunity space, Journal of Transport Geography 28 (2013) 144–154. doi:10.1016/j.jtrangeo.2012.12.007.

[3] M. A. Saif, M. M. Zefreh, A. Torok, Public transport accessibility: A literature review, Periodica Polytechnica Transportation Engineering 47 (1) (2019) 36–43. doi:10.3311/PPtr.12072.

[4] A. Ceder, Public transit planning and operation : modeling, practice and behavior, CRC Press, 2015.

[5] J. Mesa, F. Ortega, M. Pozo, Locating optimal timetables and vehicle schedules in a transit line, Annals of Operations Research 222 (2013) 439–455.

[6] A. de Palma, R. Lindsey, Optimal timetables for public transportation, Transportation Research Part B: Methodological 35 (2001) 789–813.

[7] G. Desaulniers, J. Desrosiers, M. M. Solomon (Eds.), Column Generation, Springer Science & Business Media, 2005.

[8] L. A. Wolsey, Integer Programming, 1st Edition, Wiley, 1998.

[9] F. B. Talbot, J. H. Patterson, An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems, Management Science 24 (11) (1978) 1163.

[10] S. Hassold, A. Ceder, Multiobjective approach to creating bus timetables with multiple vehicle types, Transportation Research Record 2276 (2012) 56–62.

[11] R. Bellman, A markovian decision process, Journal of Mathematics and Mechanics 6 (1957) 679–684.

[12] G. Ai, X. Zuo, G. Chen, B. Wu, Deep reinforcement learning based dynamic optimization of bus timetable, Applied Soft Computing 131 (109752) (2022).

[13] Y. Liu, X. Zuo, G. Ai, Y. Liu, A reinforcement learning-based approach for online bus scheduling, Knowledge-Based Systems 271 (110584) (2023).

[14] K. Gkiotsalitis, Bus scheduling considering trip-varying travel times, vehicle availability and capacity, IET Intelligent Transport Systems 14 (10) (2020) 1239–1246. doi:10.1049/iet-its.2019.0725.

[15] Y. Wang, D. Zhang, L. Hu, et al., A data-driven and optimal bus scheduling model with time-dependent traffic and demand, IEEE Transactions on Intelligent Transportation Systems 18 (9) (2017) 2443–2452. doi:10.1109/TITS.2017.2686015.

[16] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Upper Saddle River, New Jersey, 1993.