EVALUATING MRT SCHEDULING IN DHAKA: A DYNAMIC, DATA-DRIVEN APPROACH USING GENETIC ALGORITHM AND DEEP LEARNING METHODS

Kazi Azmain Awsaf *†[1], A.H.M. Fuad †[2], Arnab Raha †[3]

[1] *Student, Department of Civil Engineering, Bangladesh University of Engineering and Technology, Bangladesh, e-mail: awsafsami@gmail.com*
[2] *Student, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh, e-mail: ahmfuad01@gmail.com*
[3] *Student, Department of Civil Engineering, Bangladesh University of Engineering and Technology, Bangladesh, e-mail: arnab.raha@outlook.com*

***Corresponding Author †These authors contributed equally to this work.**

ABSTRACT

In rapidly expanding cities such as Dhaka, effective metro scheduling is essential to enhance service quality and reduce operational inefficiencies. Currently, it operates on a fixed schedule. This fixed timetable is ineffective as it does not adjust according to the number of individuals present at the station at any given time, particularly during peak hours. This may result in increased overcrowding, extended wait times for passengers, and diminished resource efficiency. This paper presents a dynamic scheduling framework designed to enhance the efficiency of Dhaka's MRT operations. The architecture integrates deep learning for demand forecasting, real-time passenger monitoring, and heuristic optimization. An LSTM model, trained on historical data, is employed to forecast the number of passengers expected to arrive in the near future. Simultaneously, real-time inflow assessments from RFID and CCTV systems are continuously monitored to identify any anomalous demand trends. An optimizer employs a Genetic Algorithm (GA) to identify the optimal scheduling system and allocates the subsequent headway for the metro trains. The objective is to reduce inefficiencies for both passengers and operators. The proposed dynamic schedule is evaluated against the existing fixed timetable using synthetic demand scenarios. The cost of passenger wait times, operational expenses, and service reliability are all critical performance metrics. Preliminary findings indicate that the dynamic timetable has improved conditions for passengers and operational efficiency. The optimized system reduces wasted work hours and positively impacts the environment by consuming less energy. The proposed framework is a feasible and scalable method to enhance urban rail operations and facilitates the development of intelligent transportation systems in Dhaka.

Keywords: *Headway Optimization, Genetic Algorithm, Computer Vision, Urban Transit, Sustainable Transport*

1. INTRODUCTION

Urban rail transit systems (MRT/Metro) are regarded as vital components of infrastructure that accelerate urbanization and serve as significant catalysts for regional economic development in metropolitan areas. The quality of service in public transportation is a crucial factor influencing passenger satisfaction, mode transition, and sustainable urban mobility. It is essential to objectively assess service quality to enhance operational metrics such as MRT headway (Eboli & Mazzulla, 2012). The need for MRT in Dhaka is increasing daily, rendering the enhancement of urban rail transit a paramount necessity. Numerous studies have examined timetable design through mathematical programming and simulation-based models (e.g., Ceder, 1984; Koutsopoulos et al., 1985; Niu & Zhou, 2013), yet optimizing headway remains one of the most effective methods to equilibrate passenger wait time and operational costs in high-frequency urban rail systems (Zhu et al., 2015). This project aims to provide a simple mathematical model and an efficient solution approach to minimize on-platform waiting time for passengers and operating costs of the metro. One method to achieve this is to regulate the distance between trains. Reducing headways, hence enhancing service frequency, results in increased passenger satisfaction. Conversely, the railway companies seek to reduce their costs by extending the duration of their routes (Hassannayebi et al., 2017). Until now, rail planners have manually created train schedules, resulting in less flexibility and efficiency in rail operations. The volume of passengers utilizing public transit systems fluctuates continuously, contingent upon peak or off-peak hours and during anomalies. The service frequency decreases during off-peak periods and subsequently increases during peak periods. The headway must be adjusted according to the number of users. The simplicity of the Dhaka MRT facilitates the simulation of passenger movement. This is due to its linear architecture, absence of interchanges, and consistent patterns of passenger movement, which facilitate demand prediction and timetable planning. Consequently, Dhaka MRT serves as an excellent venue for evaluating headway optimization models with little structural constraints. Due to the structural simplicity of the MRT, a basic evolutionary algorithm can rapidly identify optimal headways for the entire day. Heuristic approaches like as Genetic Algorithms are essential for identifying near-optimal solutions with a manageable level of computational resources, as headway optimization is an NP-hard problem (Zhao & Zeng, 2006). Our approach utilizes real-time passenger volume data from computer vision (CV) and RFID channels, in contrast to previous models that rely on static demand assumptions. This dynamic input enables the system to swiftly adapt to fluctuations in demand by re-optimizing headway. This ensures operational stability over time and facilitates prompt reactions during unexpected surges. Our approach utilizes real-time passenger volume data derived from computer vision (CV) and RFID channels, in contrast to previous models that relied on static demand assumptions. This dynamic input enables the system to promptly adjust to fluctuations in demand by re-optimizing headway. This ensures the system remains responsive in the short term amid abrupt surges and steady in the long run. This integrated and demand-responsive architecture enhances passenger experience and operational efficiency while setting a benchmark for intelligent urban transportation design in rapidly growing cities such as Dhaka. It is important to note that this study serves as a simulation-based concept validation. While the proposed framework is designed to integrate real-time CCTV and RFID data streams to get the real-time passenger flow, the current evaluation utilizes synthetic demand scenarios generated from historical data to validate the algorithmic approach. Full hardware integration and field testing are reserved for potential future deployment.

2. PROBLEM STATEMENT

The paper focuses on a one-direction metro with n stations denoted by $s_1, s_2, \dots, s_n$. Trains depart the depot and visit stations $s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_n$, where s_1 and s_n indicates the start terminal and end terminal respectively. Passenger arrivals at stations are time-varying and observed or predicted by a LSTM module using historical passenger data as arrival rates $\lambda_i(t)$. We want a dynamic dispatch schedule (headways) $H = [h_1, h_2, \dots, h_K]$ over a planning horizon so as to minimize passenger waiting and operating cost, while respecting capacity and safety constraints. This is solved via a Genetic Algorithm (GA) that evaluates candidate schedules using a discrete-time/discrete-event passenger-flow simulation. The following notations and parameters are used throughout the paper.

Indices and sets are as follows:

i : index of station $i \in \{1, 2, \dots, n\}$,
k : index of trains (dispatches) in planning horizon $k \in \{1, 2, \dots, K\}$,
h : index of headway $h \in \{1, 2, \dots, K\}$,
t : time that is continuous (minutes) or discretized at train events.

Parameters are as follows:

C : train capacity (passengers),
K : plan for the number of next train dispatches so the horizon covers the near-term decision,
$\lambda_i(t)$: arrival rate at station i at continuous time t (passengers/min). We assume $\lambda_i(t)$ known or estimated for the planning horizon,
h_{min} : prespecified minimum interval between two consecutive trains at the same station (e.g., 3 minutes),
h_{max} : prespecified maximum interval between two consecutive trains at the same station (e.g., 8 minutes),
h_k : headway between train $k-1$ and train k (minutes). h_k is time from "now" to next dispatch. $h_k \in [h_{min}, h_{max}]$,
d_i : travel time from station s_i to s_{i+1} (minutes),
t^0 : current time,
T_k : departure time of train k; $T_k = t^0 + \sum_{r=1}^{k} h_r$,
$p_{r,i}$: OD probability that a passenger boarding at r alights at station i (so, $p_{r,i} = 0$ for $i \leq r$, and $\sum_{j=r+1}^{n} p_{r,i} = 1$),
c_t : fixed cost per trip/dispatch of a train,
c_d : distance/time proportional cost.

State variables are as follows:

$Q_i^{(k)}$: queue size (waiting passengers) at station i just before arrival of train k,
$A_i^{(k)}$: arrivals between train $k-1$ and train k at station i,
$B_i^{(k)}$: number of passengers boarded at s_i onto train k,
$\mathrm{R_{k,i}}$: remaining occupancy when train k reaches station i,
$O_{k,i}^{in}$: occupancy of train k when it arrives at station i (passengers on board),
$D_i^{(k)}$: passengers alighting at s_i from train k,
$W_i^{(k)}$: waiting time of passengers between train $k-1$ and k at station i.

Scalar functions used in Genetic Algorithm are as follows:

$\bar{W}(H)$: average waiting time (minutes/passenger) for passengers served within horizon T,
$P(H)$: penalty for total number of passengers left unserved (overflow) at end of planning horizon,
$J(H)$: scalar objective function,
$C_{op}(H)$: operating cost estimate.

The assumptions made throughout the paper are explained as follows:

Assumption 1: The metro system consists of n stations aligned in a single one-way corridor. Trains stop at all stations, boarding is First Come First Serve (FCFS) at station queue.

Assumption 2: All trains are homogeneous in configuration, having identical capacity, speed, dwell characteristics, and acceleration or deceleration profiles. Capacity C of the train is strict. The number of passengers boarding at any station cannot exceed the remaining available capacity of the train.

Assumption 3: Arrivals between two consecutive trains are approximated using the predicted arrival rate $\lambda_i(t) \cdot h_k$ obtained from LSTM based on historical data. Real-time observations from CV and RFID systems are used only for anomaly detection and creating reactive headways for demand control, not for determining $\lambda_i(t)$ during regular GA optimization. A Poisson process approximation may be adopted to represent stochastic arrival behavior when required.

Assumption 4: Each candidate headway configuration $H = [h_1, h_2, \dots, h_K]$ is evaluated over a fixed planning horizon, $T = \sum_{k=1}^{K} h_k$. This represents the total duration of service covered by K successive headways. Within this horizon, arrival rates $\lambda_i(t)$ are assumed stationary (i.e., approximately constant) so that headway performance can be evaluated deterministically.

2.1 Model Formulation

This paper aims to suggest a dynamic dispatch schedule H which is best understood as the vector of train headways. The model simulates train-by-train. Let train index k progress and $Q_i^{(1)}$ is known continuously through CV. For station i, passenger arrival between trains $k-1$ and k is:

$$A_i^{(k)} = \int_{T_{k-1}}^{T_k} \lambda_i(t)dt \tag{1}$$

If the passenger arrival rate, $\lambda_i(t)$ is assumed to be approximately constant over $[T_{k-1}, T_k]$ or the planning horizon, then for station i, equation (1) becomes:

$$A_i^{(k)} \approx \lambda_i \cdot h_k \tag{2}$$

The queue update before boarding at train k:

$$Q_i^{(k)} = Q_i^{(k-1)} - B_i^{(k-1)} + A_i^{(k)} \tag{3}$$

But since $B_i^{(k-1)}$ is boarding at previous train, the above reduces to the standard event update implemented sequentially.

$$R_{k,i} = C - O_{k,i}^{in} \tag{4}$$

$$B_i^{(k)} = min\left(Q_i^{(k)}, R_{k,i}\right) \tag{5}$$

After boarding:

$$R_{k,i+1} = R_{k,i} - B_i^{(k)} \tag{6}$$

An origin-destination distribution $p_{r\to i}$, derived from historical data, is used, where the number of alighting passengers at station i from train k equals sum over origin stations $r < i$ of boarded passengers who had $j = i$. Then the expected number of passengers alighting at station i from train k is:

$$D_i^{(k)} = \sum_{r=1}^{i-1} B_r^{(k)} \cdot p_{r,i} \tag{7}$$

Occupancy update:

$$O_{k,i+1}^{in} = O_{k,i}^{in} - D_i^{(k)} + B_i^{(k)} \tag{8}$$

$O_{k,1}^{in}$ is usually 0 if the train starts empty from the depot. If arrivals are assumed uniform within headway, average waiting time for passengers who arrive between train $k-1$ and k and get served by train k is approximately $h_k/2$. But because of capacity limitations and left-behind passengers, exact waiting times require tracking delays: for each passenger boarded on train k who arrived at time

$t_{arr}, waiting = T_k - t_{arr}$. In aggregate, for deterministic arrival rate, total waiting time contributed by interval $(T_k, T_{k-1}]$ is approximately:

$$W_i^{(k)} = \int_0^{h_k} \lambda_i(T_{k-1} + u) \cdot (h_k - u) du \tag{9}$$

Here, the total passenger waiting time for a station during a headway interval is computed by integrating the instantaneous arrival rate multiplied by the residual waiting time of each infinitesimal arrival. Equivalently, letting the local time origin $u = 0$ at the departure of train $k-1$, the waiting contribution of arrivals at $u \in [0, h_k]$ is $\lambda_i(T_{k-1} + u) \cdot (h_k - u)$. If $\lambda_i(t)$ is approximately constant over the interval, $\lambda_i(T_{k-1} + u) = \lambda_i$ and equation (9) evaluates to:

$$\int_0^{h_k} \lambda_i(T_{k-1} + u) \cdot (h_k - u) du = \lambda_i \left[h_k u - \frac{1}{2} u^2 \right]_0^{h_k} = \lambda_i \cdot \frac{1}{2} h_k^2$$

$$or, \qquad W_i^{(k)} = \frac{1}{2} \lambda_i \cdot h_k^2 \tag{10}$$

If train fare from station i to j is denoted by $f_{i,j}$, total expected revenue over the planning horizon for schedule H:

$$R(H) = \sum_{k=1}^{K} \sum_{i=1}^{n} B_i^{(k)} \left(\sum_{j=i+1}^{n} p_{i,j} \cdot f_{i,j} \right) \tag{11}$$

The scalar objective, $J(H)$ is to be minimized. Passenger waiting and operating cost are standard performance indicators in metro scheduling. The penalty term discourages excessive queues and ensures service quality, while the revenue term encourages efficient use of train capacity. Let H be a candidate headway vector. It will be evaluated with the simulation above to obtain $J(H)$:

$$\bar{W}(H) = \frac{\sum_{k=1}^{K} \sum_{i=1}^{n} \left(Q_i^{(k-1)} \cdot h_k + W_i^{(k)} \right)}{\sum_{k,i} B_i^{(k)}} \tag{12}$$

The term $Q_i^{(k-1)} \cdot h_k$ captures the waiting of leftover passengers who remain in the queue for the full headway. The integral term $W_i^{(k)}$ captures the waiting of new arrivals during the interval $(T_k, T_{k-1}]$.

$$P(H) = \gamma \sum_{k=1}^{K} \sum_{i=1}^{n} max \left(0, Q_i^{(k)} - Q_{thres} \right)^2 \tag{13}$$

$$C_{op}(H) = c_t \cdot K + c_d \cdot K \cdot \sum_{i=1}^{n} d_i \tag{14}$$

$$J(H) = w_w \cdot \bar{W}(H) + w_p \cdot P(H) + w_o \cdot C_{op}(H) - w_r R(H) \tag{15}$$

Here, w_w : weight for passenger waiting (service quality),
w_o : weight for operating cost (economy),
w_p : penalty weight for leftover passengers or overcrowding,
w_r : weight for revenue which measures the usability of train's capacity.

Some constraints to ensure better service quality and safety are as follows:

$$h_{min} \leq h_k \leq h_{max} \tag{16}$$

$$0 \leq B_i^{(k)} \leq min\left(Q_i^{(k)}, R_{k,i}\right) \tag{17}$$

$$\underset{i}{max}\, Q_i^{(k)} \leq Q_{max} \tag{18}$$

3. METHODOLOGY

This section describes the proposed real-time dynamic scheduling framework. The goal is to generate an adaptive dispatch schedule that updates periodically using predicted passenger arrivals and responds immediately to sudden demand surges. The approach integrates LSTM-based passenger forecasting, real-time anomaly detection, and a Genetic Algorithm (GA) that optimizes short-term headways.

The system operates over the metro service day by repeatedly solving a short-term scheduling problem. The whole service period is divided into some small time blocks. The system generates headways for each time block or planning horizon. At the beginning of each planning horizon T_H , the system predicts near-term passenger arrival rates using an LSTM model trained on historical entry data. Then it runs a GA that optimizes the headway vector $H = [h_1, h_2, \dots, h_K]$ for the upcoming horizon. Finally, the resulting headway schedule is deployed in real time. In parallel, a continuously running anomaly detector known as reactive module compares real-time station inflows against the recent LSTM prediction. When abnormal crowding is detected, the regular execution is interrupted, and a short reactive optimization is triggered using the real-time observed demand.

Passenger arrival rates are strongly time-dependent, influenced by commuting patterns, weekly cycles, and special events. A Long Short-Term Memory (LSTM) neural network is used to forecast near-term passenger arrivals because it effectively captures temporal dependencies, nonlinearities, and non-stationary behavior in time-series demand data. LSTMs naturally model seasonality and peak-hour variations, making them suitable for short rolling-horizon operational control. The model takes a multivariate time-series of historical inflow rates λ_i across all $n = 16$ stations. A sliding window of 60 minutes, discretized into 5-minute intervals (12-time steps), is used as input. The network consists of two stacked LSTM layers with 128 and 64 hidden units, respectively, followed by a dropout layer (rate 0.2) to improve generalization. A fully connected output layer produces predicted arrival rates for the next 12-time steps at all stations, yielding $\lambda_i^{hist}(t_0)$, which is assumed constant over the planning horizon. The model is trained using Mean Squared Error (MSE) loss and the Adam optimizer (learning rate 0.001). Training is performed for up to 100 epochs with early stopping (patience = 10), achieving a Mean Absolute Percentage Error (MAPE) below 8% on validation data. At each planning epoch t_0, the LSTM produces predicted arrival rates, $\lambda_i^{hist}(t_0)$. These arrival rates are treated as constant (or slowly varying) over the panning horizon T_H, consistent with the model assumptions.

For each planning horizon, the regular GA takes the predicted demand $\lambda_i^{hist}(t_0)$, as input and constructs a feasible headway vector: $H(t_0) = [h_1, h_2, \dots, h_K]$, $h_k \in [h_{min}, h_{max}]$. Each candidate H is evaluated through the discrete-event simulation described in problem statement. The GA minimizes the scalar objective $J(H)$. The output is the optimized schedule applied for the next horizon unless preempted by an anomaly.

While the regular plan is being executed, a parallel module monitors real-time station inflows. Let $\lambda_i^{obs}(t)$ be the short-window estimate of arrivals derived from CCTV-based crowd counting and RFID entries. An anomaly is detected when:

$$z_i(t) = \frac{\lambda_i^{obs}(t) - \lambda_i^{hist}(t_0)}{\sigma_i} \tag{19}$$

exceeds a threshold z_{thres}, where σ_i is the historical standard deviation for that time of day. The threshold z_{thres} is crucial for adjusting the reactive module's sensitivity to differences between actual and expected passenger arrivals. Given the stable statistical patterns of passenger inflows during normal conditions, a z-score criterion is used to identify significant deviations. In this research, z_{thres} is empirically chosen within the typical range of 2–3, allowing a balance between responsiveness and stability. A lower threshold may lead to frequent unnecessary schedule changes, whereas a higher threshold can delay necessary actions during unexpected demand spikes. The chosen threshold aims to trigger optimization only for significant crowding events while maintaining normal operations during regular demand variations.

If $\max_i |z_i(t)| > z_{thres}$, a significant deviation from expected demand is assumed, such as a sudden surge at a station. When an anomaly is detected during the execution of a regular schedule, the ongoing schedule is preempted. A short reactive horizon T_{react} is initiated. A reactive GA uses the real-time observed arrival rates $\lambda_i^{obs}(t)$, not historical prediction. The resulting plan, $H_{react}(t)$ is executed immediately. After completing the reactive horizon, the system returns to the regular planning cycle. This mechanism ensures that service frequency increases rapidly during crowding while avoiding excessive short-term fluctuations.

At all times, the system operates under one of the two modes:

I. **Regular mode** (history-driven, LSTM-based optimization)
II. **Reactive mode** (real-time observation-driven optimization)

System time is advanced after completing a regular horizon as $t_0 \leftarrow t_0 + T_H$ and after completing a reactive horizon: $t_0 \leftarrow t + T_{react}$. The flowchart in figure 1 illustrates the full control logic.

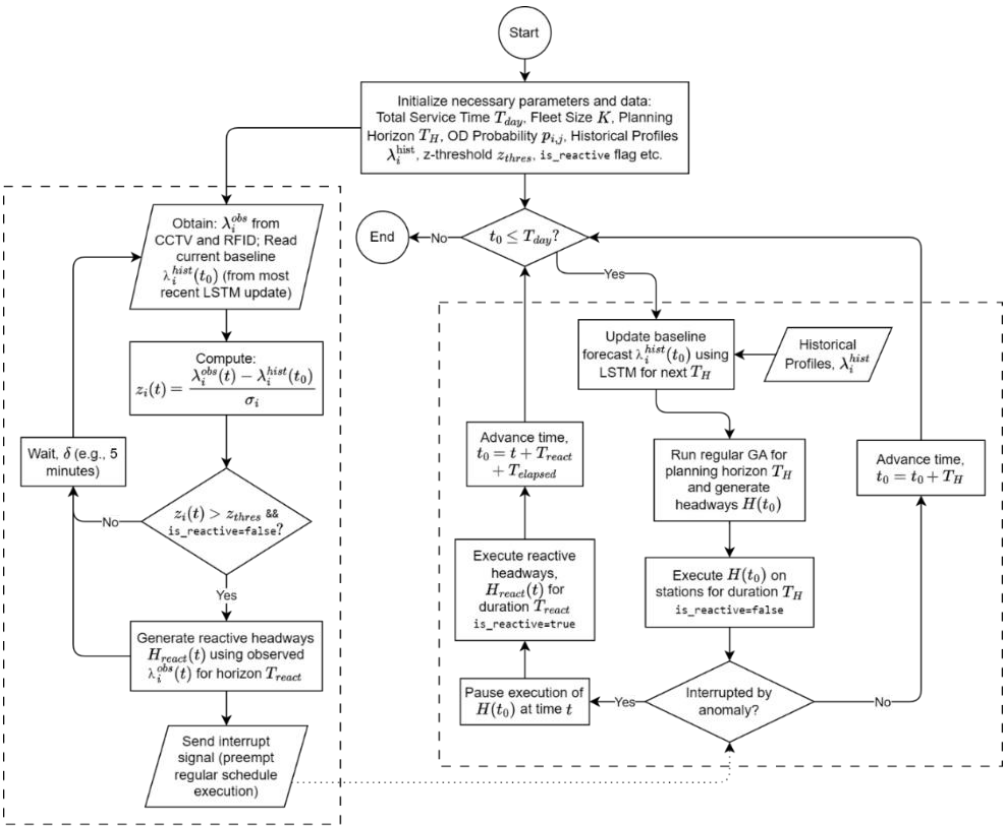

Figure 1: Full control logic of the whole system

The methodology integrates deep learning (LSTM) for short-term arrival forecasting, Real-time sensing (CCTV+RFID) for anomaly detection, Heuristic optimization (Genetic Algorithm) for headway

scheduling, simulation-based evaluation for realistic modeling of passenger and train dynamics and revenue, service quality, and operating cost metrics for multi-criteria evaluation. Together, these components enable a real-time, data-driven dispatching framework that adapts to changing passenger conditions while maintaining cost-service balance. The framework integrates seamlessly with existing metro systems. Passenger inflow is estimated using CCTV and RFID ticketing already in place at Dhaka MRT stations. LSTM forecasting and Genetic Algorithm optimization run on a central server with standard GPU/CPU resources, while the reactive module uses short rolling windows for near real-time adjustments. Communication relies on current operational networks, so no extra hardware is required beyond routine monitoring and control.

4. NUMERICAL EXAMPLE

To test the viability of the proposed Rolling Horizon control framework, a high fidelity simulation environment was developed that tries to replicate the operational characteristics of Dhaka MRT Line-6. Due to the unavailability of granular individual-level passenger logs, a stochastic dataset was synthesized calibrating against the aggregate flow pattern found in the MRT Origin-Destination (OD) report that shows the OD pattern for the month of July 2025 for each station. The report was obtained from DMTCL authority. The data generation process converts the static monthly ridership aggregates into a dynamic minute by minute passenger arrival log for simulation and evaluation.

The simulation covers the active corridor, from Uttara Uttar (S_1) to Motijheel (S_{16}) consisting of $n = 16$ stations. The spatial demand is taken from the OD-Matrix, $P_{i,j}$ where each value represents the total monthly passenger count travelling from station i to station j. The matrix was generated from historical smart card transaction aggreagates.

Table 1: OD report obtained from authority

O/D	S01 UN	S02 UC	S03 US	S04 PL	S05 M11	S06 M10	S07 KP	S08 SP	S09 AG	S10 BS	S11 FG	S12 KB	S13 SB	S14 DU	S15 BS	S16 MJ	Total
S01UN	-	2.6	3.0	53.6	73.5	151.7	31.8	39.0	87.9	17.1	51.5	71.9	54.8	59.1	125.0	129.5	952.1
S02UC	3.0	-	0.4	20.4	18.7	43.4	6.5	9.2	19.6	5.6	16.6	19.7	10.6	13.5	20.3	25.3	232.7
S03US	2.9	0.5	-	10.0	12.0	17.1	4.3	4.1	6.4	1.7	4.7	5.6	2.4	3.7	4.8	5.3	85.5
S04PL	57.6	20.5	9.3	-	4.0	40.0	30.5	28.5	64.1	13.0	47.1	65.5	33.5	38.9	71.5	61.8	585.9
S05M11	83.9	21.1	12.0	4.4	-	12.2	29.0	46.7	57.1	13.1	55.3	66.7	52.3	46.9	87.5	80.1	668.2
S06M10	143.7	38.4	12.4	29.6	8.8	-	9.3	28.5	67.4	23.7	92.8	127.2	76.6	93.9	182.1	183.9	1118.2
S07KP	37.3	7.9	3.9	30.3	26.0	11.2	-	1.7	15.2	10.4	33.7	45.0	30.0	26.1	52.9	42.3	373.8
S08SP	43.2	9.8	4.3	28.9	42.4	35.8	1.7	-	8.2	10.7	56.7	72.1	40.7	40.7	66.7	64.6	526.5
S09AG	86.8	18.4	5.1	55.0	48.3	76.7	13.5	8.1	-	3.8	27.3	73.3	50.6	61.1	134.5	132.6	795.0
S10BS	19.0	6.7	1.6	15.4	15.2	35.3	11.8	11.5	3.7	-	4.3	15.1	20.2	26.2	26.9	40.1	253.0
S11FG	50.6	16.0	3.9	41.4	47.5	98.6	30.1	54.2	30.6	4.9	-	17.3	38.7	73.4	98.3	155.1	760.5
S12KB	69.6	19.4	4.6	59.2	58.0	137.8	38.0	62.9	69.5	13.9	11.6	-	19.9	53.8	119.5	140.1	877.8
S13SB	58.4	11.1	2.2	31.1	47.2	86.7	27.7	37.8	54.4	23.2	36.6	22.1	-	4.8	31.3	102.5	577.2
S14DU	61.1	14.1	3.0	37.1	42.2	101.9	23.0	38.0	62.3	26.0	65.7	53.1	4.4	-	11.2	60.3	603.3
S15BS	106.1	18.9	4.0	56.0	68.1	177.2	42.0	55.4	127.6	27.0	80.6	113.4	26.9	12.0	-	18.7	933.9
S16MJ	143.5	29.4	5.1	62.7	82.8	215.6	40.9	68.2	145.4	40.8	144.9	146.2	92.7	58.0	16.1	-	1292.5
Total	966.7	234.9	74.7	535.1	594.7	1241.2	340.0	493.7	819.5	234.9	729.3	914.3	554.2	612.2	1048.6	1242.2	10636.1

Although individual passenger trajectories were not available, the synthesized demand reproduces key operational characteristics of Dhaka MRT Line-6, including station-wise boarding dominance, directional imbalance, and peak-hour concentration. The resulting peak loads and average per-station arrivals fall within ranges reported by DMTCL operational summaries, ensuring realism of the simulation environment. To simulate the temporal variation of urban commuting (i.e. the peak hour and off peak hour), the daily demand was distributed across the operational horizon of each day, based on service intensity. The operational horizon, T_{day} is divided into M time windows ($M = 5$) each characterized by a specific headway h_w. The headways for each time window is shown in Table 2.

Table 2: List of Headways during each time windows

Starting Time	Ending Time	Headways, h_w (minutes)
7:00	7:30	8
7:30	11:30	10
11:30	2:30	8
2:30	8:30	10
8:30	9:30	8

The passenger arrival rate $\rho_{i,j}$ (passengers/minute) for an OD pair at time t is derived as a function of the daily demand and the current service frequency. The generation algorithm is modeled as:

$$\rho_{i,j}(t) = \frac{P_{i,j}}{\kappa \cdot h_w(t)} \tag{20}$$

$P_{i,j}$: the total monthly passenger volume from $i\ to\ j$,
$h_w(t)$: the scheduled headway,
κ : normalizing constant representing approximately the total number of trips each way per day ($\kappa = T_{window} = 102$).

A discrete-event generator was implemented to generate individual passenger records. For every $t \in [0, T_{day}]$ and every OD pair (i, j).

I. **Probability Calculation:** The arrival probability $P_{arrival} = \rho_{i,j}(t)$ is calculated.
II. **Integer Sampling:** To handle fractional flow rates, the algorithm separates the deterministic component from the stochastic component $N_{pax} = \lfloor P_{arrival} \rfloor + Bernoulli(P_{arrival} - \lfloor P_{arrival} \rfloor)$. Where $Bernoulli(p)$ returns 1 with probability p and 0 otherwise.
III. **Log Generation:** For each generated passenger, a unique record is created containing a hashed ID. Origin (S_i), Destination (S_j), Entry Time (t_{entry}), and a projected Exit Time (t_{exit}) based on fixed inter-station travel time is logged into a .csv file.

4.1 Predictive Modelling with Deep Learning (LSTM): In order to feed the regular mode with a passenger flow, a Long Short-Term Memory (LSTM) network was utilized to forecast the baseline passenger flow $\lambda_i^{hist}(t)$, for the next planning horizon. The raw simulation was first aggregated into 5-minute time steps, converting discrete passenger events into a multivariate time-series tensor. The model architecture uses a sliding window approach, processing the past 60 minutes of data through two LSTM layers to predict inflow rates for the next 12-time steps. Trained with the Adam optimizer on an 80/20 data split, the model achieved a Mean Absolute Percentage Error (MAPE) of less than 8%, validating its reliability as the system's historical baseline.

4.2 Optimization Results with Different Number of Maximum Fleet Size Per Hour: Fleet size per hour directly determines achievable headway flexibility and operating cost. Evaluating system performance under varying fleet availability allows assessment of how efficiently the proposed dynamic framework utilizes limited rolling stock compared to static dispatching. The Dynamic Schedule was evaluated against a static schedule with different number of maximum fleet size per hour, namely $K = 5$ to $K = 10$. For each value of K, the system simulates the whole operational horizon for both dynamic and static schedules and gives us a headway vector. The static schedule dispatches each train after $60/K$ minutes, while the dynamic scheduling dynamically generates a maximum of K number of headways ranging from h_{min} to h_{max} after choosing the optimal fleet size for each planning horizon. An "alighting" noise or localized demand surge is introduced at 300th minute at Agargaon station with a multiplier of 3 which lasts for 120 minutes to simulate real life anomaly. The parameters used in the study are shown in table 3, 4 and the obtained results are shown in table 5:

Table 3: Parameters used in numerical examples

T_H	T_{react}	K	h_{max}	h_{min}	C	z_{thres}	d_i	c_t	c_d	Pop-Size	Max Generation	Mutation Rate	Crossover Rate
60	10-15	5-10	15	3	2184	2.5	2	1680	3.5	20	80	0.2	0.8

Table 4: Objective Function Weights

$\mathbf{w_w}$	w_o	w_p	w_r
2.5	50	0.25	1×10^(-5)

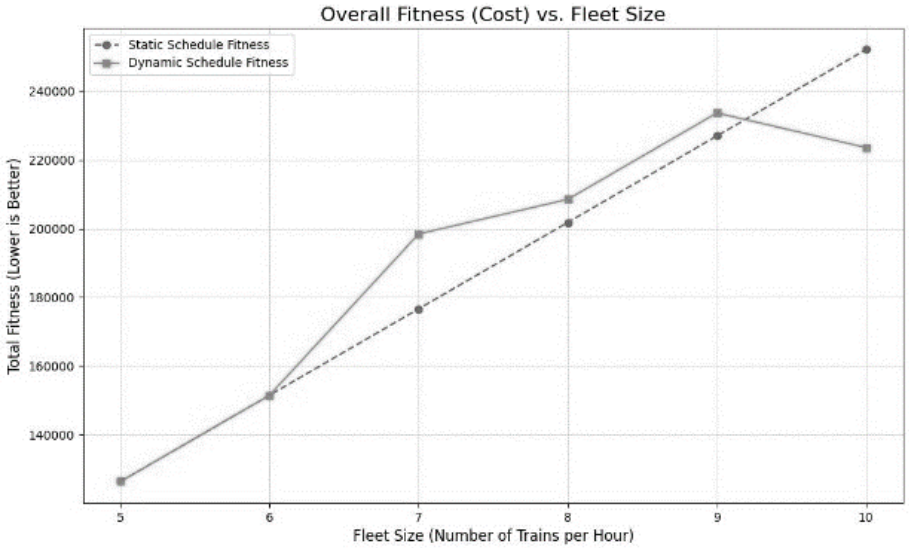

Figure 2: Overall Fitness vs. Maximum Fleet Size comparison

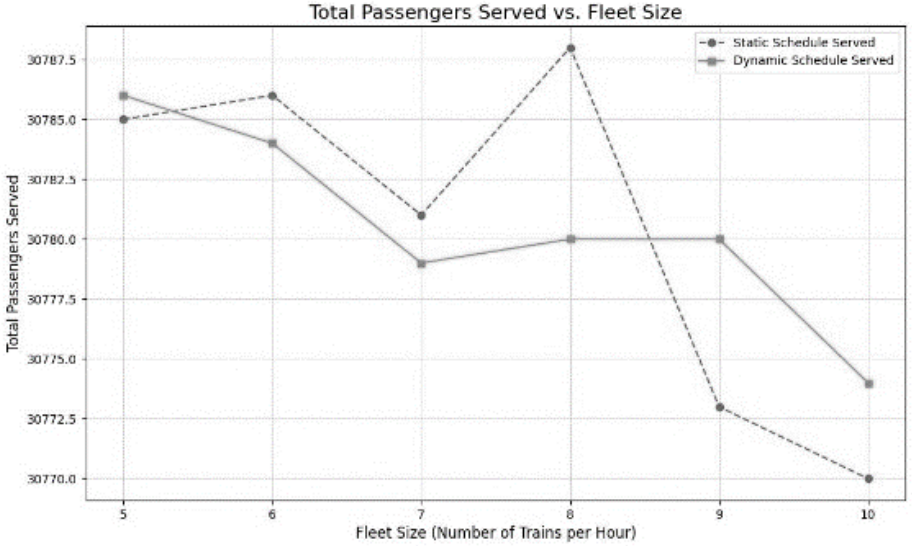

Figure 3: Total Passengers Served vs. Fleet size

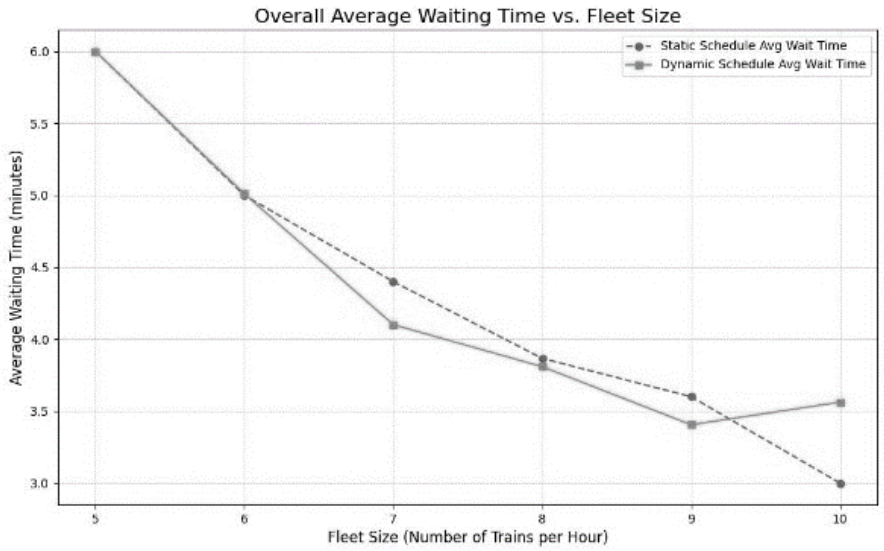

Figure 4: Comparison of overall average waiting time and fleet size

Table 5: Observed results

Fleet Size	Static Fitness	Dynamic Fitness	Static Served	Dynamic Served	Static Avg. Wait	Dynamic Avg. Wait
5	126315	126315	30785	30785	6	6
6	151462	151462	30786	30785	5	5
7	176631	195096	30781	30776	4.4	4.13
8	201803	205162	30788	30784	3.86	3.85
9	226989	233699	30773	30776	3.6	3.42
10	252157	225303	30770	30775	3	3.5

At small fleet sizes ($K \leq 7$), dynamic scheduling sometimes favors immediate crowd relief over long-term efficiency, which can raise overall costs compared to static dispatching. As fleet size grows, rolling-horizon optimization gains flexibility, lowering costs and wait times. Once availability reaches about $K = 9 - 10$, improvements plateau, with waiting times and passenger service levels showing little further benefit.

5. CONCLUSION

From the simulation results, it is observed that under the static scheduling approach, the total fitness value (representing operational cost and service penalties) increases approximately linearly with fleet size, while the average waiting time decreases in a similarly linear manner. In contrast, the dynamic scheduling framework exhibits a markedly different behavior: Initially, the dynamic model matches the fitness and waiting time of static scheduling, occasionally scoring slightly worse. This occurs because the GA is a complex heuristic non-linear search rather than a simple mathematical formulation, causing it to occasionally converge on local optima that are slightly less efficient than the idealized static case. However, as the fleet size increases, both the waiting time and the fitness value plateau. This plateau indicates a point of diminishing returns, beyond which deploying additional trains no longer yields meaningful improvements in service quality. This occurs because, in static scheduling, increasing the fleet size beyond a certain threshold causes operational costs to outweigh the reduction in passenger waiting costs. In contrast, dynamic scheduling optimizes fleet size based on passenger flow, preventing additional deployments when operational costs become excessive. Furthermore, the results show that the static schedule requires a substantially larger fleet to match the service performance achieved by the dynamic schedule, while the proposed framework maintains efficiency by avoiding unnecessary dispatches once demand is adequately met. Overall, the proposed framework maintains operational

efficiency and demand responsiveness by balancing service quality improvements against operational cost.

DECLARATION OF USE OF AI

During the preparation of this work, the authors used Quillbot, ChatGPT, and Gemini to polish the writing and improve the readability of the manuscript. Additionally, AI tools were utilized in the research methodology; specifically, GitHub Copilot was used to assist in generating and optimizing the computational code. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

REFERENCES

Ceder, A. (1984). Bus frequency determination using passenger count data. *Transportation Research Part A: General*, *18*(5-6), 439-453.

Cui, Y., Yu, Q., & Wang, C. (2022). Headway Optimisation for Metro Lines Based on Timetable Simulation and Simulated Annealing. *Journal of Advanced Transportation*, *2022*(1), 7035214.

Eboli, L., & Mazzulla, G. (2012). Performance indicators for an objective measure of public transport service quality.

Hassannayebi, E., Zegordi, S. H., Amin-Naseri, M. R., & Yaghini, M. (2018). Optimizing headways for urban rail transit services using adaptive particle swarm algorithms. *Public Transport*, *10*(1), 23-62.

Koutsopoulos, H. N., & Wilson, N. H. M. (1985). DETERMINATION OF HEADWAYS AS A FUNCTION OF TIME VARYING CHARACTERISTICS ON A TRANSIT NETWORK. FROM THE BOOK COMPUTER SCHEDULING OF PUBLIC TRANSPORT 2.

Niu, H., & Zhang, M. (2012). An Optimization to Schedule Train Operations with Phase-Regular Framework for Intercity Rail Lines. *Discrete dynamics in nature and society*, *2012*(1), 549374.

Niu, H., & Zhou, X. (2013). Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. *Transportation Research Part C: Emerging Technologies*, *36*, 212-230.

Zhao, F., & Zeng, X. (2006). Optimization of transit network layout and headway with a combined genetic algorithm and simulated annealing method. *Engineering Optimization*, *38*(6), 701-722.

Zhu, Y. T., Mao, B. H., Liu, L., & Li, M. G. (2015). Timetable design for urban rail line with capacity constraints. *Discrete Dynamics in Nature and Society*, *2015*(1), 429219.