Optimizing Passenger Waiting Time on Istanbul M2 Metro Line Using Genetic Algorithms

Berker Eriş
Department of Artificial Intelligence and Data Engineering
Faculty of Computer and Informatics, ITU
erisb22@itu.edu.tr
150220315

Enes Yemenli
Department of Artificial Intelligence and Data Engineering
Faculty of Computer and Informatics, ITU
yemenli22@itu.edu.tr
150220311

***Abstract*—This paper presents an optimization-based study that minimizes passenger waiting times on the Istanbul M2 metro line, Yenikapı-Hacıosman direction. Due to the rapid population growth and existing public transportation problems in Istanbul, an efficient train schedule is the main goal. We designed a simulation that simulates the boarding time of passengers and the departure of trains using real metro data from the Istanbul Metropolitan Municipality. We used a genetic algorithm to solve this problem, which is nonlinear due to discrete departure times and capacity limits. The optimized schedule reduced the average waiting time from 3.65 minutes to 3.23 minutes.**

I. Introduction

The population growth in Istanbul has resulted in the population doubling from 8 million to 16 million since 2000. The city's transportation infrastructure began to fail to meet the increasing demand and problems began to arise. One of these problems was the increase in waiting time for passengers. Our model was designed to minimize the average waiting time of passengers on Istanbul's busiest metro line, M2, on Yenikapı-Hacıosman direction, by regulating the departure times of the trains, using the data of passengers using this metro line on September 4, 2024 provided by IBB Open Data Platform.

II. Problem Description

Passengers are subjected to long waiting times during peak hours in the current train schedule. There are unnecessary services during low demand hours. Considering the inconsistency and incompleteness of the data regarding the "Number of trains" and "Number of drivers" constraints mentioned in the Proposal, we had to use the total number of trips per day as a constraint instead of these two constraints. Thus, we tried to keep the total number of trips in a day as 196, the same as the current schedule.

We wanted to increase the fidelity of the optimization result by keeping the number of trips constant. The fixed number of trips increased the applicability of our result without additional resources (extra vehicles or drivers required). In order to increase the applicability in the real world, we added constraints such as the earliest and latest time for the first and last trains, a maximum of 15 minutes between consecutive trips, and no two trips can depart at the same time, instead of the constraints that were infeasible due to data limitations. With these constraints, our model was compatible with the existing infrastructure and expectations of the Istanbul metro system.

A. Objective Function

Let P be the set of all passengers and S be the set of all scheduled trips. Each passenger $p \in P$ has a known entrance time e_p and will board the first train with sufficient capacity after e_p. The objective is:

$$\min \sum_{p \in P} (b_p - e_p) \tag{1}$$

where b_p is the boarding time of passenger p.

B. Constraints

Let $T = \{t_1, t_2, \ldots, t_{196}\}$ be the list of departure times from Yenikapı(our initial station), where t_i is the departure time of the i-th train.

- **Capacity Constraint:** Ensures that the number of simultaneous passengers on any train does not exceed the physical limit of the train.

$$\forall i : \quad \text{Number of simultaneous passengers on a train } i \leq 950 \tag{2}$$

- **Time Window Constraint:** Guarantees that the train schedule starts at 06:00 and ends before or at 23:45.

$$06{:}00 = t_1 < t_2 < \cdots < t_{196} \leq 23{:}59 \tag{3}$$

- **Max Gap Constraint:** Limits the maximum time between any two consecutive train departures to 15 minutes to prevent long waiting periods.

$$\forall i \in \{1, \ldots, 195\} : \quad t_{i+1} - t_i \leq 15 \text{ minutes} \tag{4}$$

- **Uniqueness Constraint:** Ensures that no two trains depart at the exact same time.

$$\forall i \neq j : \quad t_i \neq t_j \tag{5}$$

III. Dataset and Preprocessing

We use datasets from the *Istanbul Metropolitan Municipality Open Data Portal*, including:

- Metro Istanbul Timetable Web Service
- Hourly Passenger Boarding Data
- Metro Vehicle Specifications

The raw dataset(Figure1) only contains the entrance time intervals and boarding stations of the passengers who boarded the M2 metro line on **September 4, 2024**.

transition_date	transition_hour	line_name	station_poi_desc_cd
2024-09-04	10	M2	HACIOSMAN
2024-09-04	10	M2	HACIOSMAN
2024-09-04	10	M2	HALIC

Fig. 1. `binen_yolcu`: Columns: transition_date (day), transition_hour (hour block), station_poi_desc_cd (boarding station name).

The dataset did not include passengers' destination stations. To overcome this, we developed a heuristic method based on directional travel patterns: passengers boarding in the morning are more likely to get off at stations with high evening drop-off rates, and those boarding in the evening tend to alight at stations with high morning drop-off rates. These destination estimates were computed using a time-dependent weighted probability distribution, where the weights were derived from hourly boarding frequencies at each station(Figure2)

transition_date	transition_hour	line_name	station_poi_desc_cd	destination	direction
2024-09-04	10:00:00 - 10:59:59	M2	HACIOSMAN	GAYRETTEPE	2
2024-09-04	10:00:00 - 10:59:59	M2	HACIOSMAN	YENIKAPI	2
2024-09-04	10:00:00 - 10:59:59	M2	HALIC	OSMANBEY	1

Fig. 2. `binen_yolcu_direction_updated`: Columns: includes destination (predicted stop), direction (1 = Yenikapı to Hacıosman, 2 = Hacıosman to Yenikapı).

After estimating the direction each passenger will go (Yenikapı or Hacıosman), we filtered the dataset to only those who went from Yenikapı to Hacıosman.

There were no exact minute-based entry times required for the simulation. Entrance times are recorded in broad hourly ranges, such as 08:00:00 - 08:59:59, which lacks the minute-level detail required for accurate simulation. To overcome this problem, we obtained a clearer structure by cubic interpolation.

For each hourly interval, we analyzed the overall boarding distribution and used cubic spline interpolation to model the minute-level distribution of passengers within the hour. We smoothed the existing step function using cubic interpolation(Figure3) and assigned an entrance time to each passenger with minute precision. Knowing the exact minutes of boarding was crucial to controlling capacity and creating a continuous and realistic flow of passengers throughout the day. All optimization modeling and evaluations were applied on this filtered dataset(Figure4).

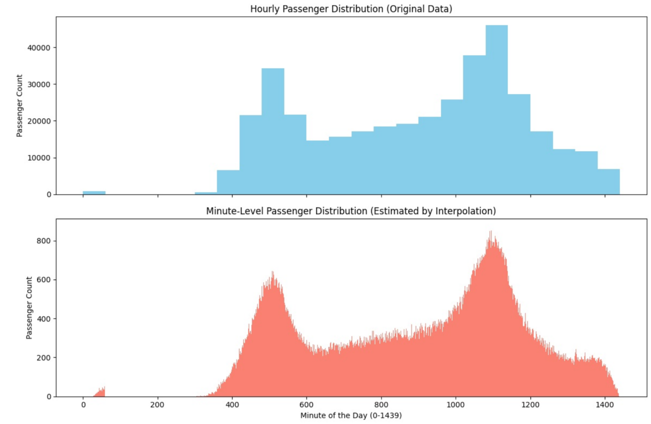

Fig. 3. Cubic interpolation of hourly boarding data to obtain minute-level probability distribution.

passenger_id	boarding_station	destination_station	enterance_time	boarding_station_id	destination_station_id
38698	YENIKAPI	GAYRETTEPE	10:24:00	0	7
38699	OSMANBEY	ITU	10:24:00	5	11
38700	VEZNECILER	4.LEVENT	10:24:00	1	9

Fig. 4. `PASSENGER_LIST.csv`: Columns: passenger_id, boarding_station, destination_station, entrance_time, boarding_station_id, destination_station_id.

To measure the efficiency of the current metro schedule, we used official timetable data(Figure5) for September 4, 2024 provided by Istanbul Metropolitan Municipality. Each passenger was matched to the first available train after entrance and simulated. Waiting time is calculated for each passenger(Figure6). This simulation gave us an average waiting time of 3.65 minutes as a reference point to evaluate the effectiveness of the optimization model.

sefer_id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	00:00:00	00:03:00	00:05:00	00:07:00	00:09:00	00:11:00	00:13:00	00:15:00	00:17:00	00:19:00	00:21:00	00:23:00	00:25:00	00:27:00	00:29:00
1	05:57:00	06:00:00	06:02:00	06:04:00	06:06:00	06:08:00	06:10:00	06:12:00	06:14:00	06:16:00	06:18:00	06:20:00	06:22:00	06:24:00	06:26:00
2	06:04:00	06:07:00	06:09:00	06:11:00	06:13:00	06:15:00	06:17:00	06:19:00	06:21:00	06:23:00	06:25:00	06:27:00	06:29:00	06:31:00	06:33:00
3	06:12:00	06:15:00	06:17:00	06:19:00	06:21:00	06:23:00	06:25:00	06:27:00	06:29:00	06:31:00	06:33:00	06:35:00	06:37:00	06:39:00	06:41:00
4	06:20:00	06:23:00	06:25:00	06:27:00	06:29:00	06:31:00	06:33:00	06:35:00	06:37:00	06:39:00	06:41:00	06:43:00	06:45:00	06:47:00	06:49:00
5	06:27:00	06:30:00	06:32:00	06:34:00	06:36:00	06:38:00	06:40:00	06:42:00	06:44:00	06:46:00	06:48:00	06:50:00	06:52:00	06:54:00	06:56:00
6	06:35:00	06:38:00	06:40:00	06:42:00	06:44:00	06:46:00	06:48:00	06:50:00	06:52:00	06:54:00	06:56:00	06:58:00	07:00:00	07:02:00	07:04:00
7	06:43:00	06:46:00	06:48:00	06:50:00	06:52:00	06:54:00	06:56:00	06:58:00	07:00:00	07:02:00	07:04:00	07:06:00	07:08:00	07:10:00	07:12:00
8	06:50:00	06:53:00	06:55:00	06:57:00	06:59:00	07:01:00	07:03:00	07:05:00	07:07:00	07:09:00	07:11:00	07:13:00	07:15:00	07:17:00	07:19:00
9	06:58:00	07:01:00	07:03:00	07:05:00	07:07:00	07:09:00	07:11:00	07:13:00	07:15:00	07:17:00	07:19:00	07:21:00	07:23:00	07:25:00	07:27:00
10	07:02:00	07:05:00	07:07:00	07:09:00	07:11:00	07:13:00	07:15:00	07:17:00	07:19:00	07:21:00	07:23:00	07:25:00	07:27:00	07:29:00	07:31:00

Fig. 5. `SEFER_LIST.csv`: Columns: stations , Rows: train's arrival time for stations. Based on the real timetable

passenger_id	boarding_station	destination_station	enterance_time	boarding_station_id	destination_station_id	sefer_id	boarding_time	wait_duration_min
147437	OSMANBEY	SISLI	19:04:00	5	6	158	19:14:00	10.0
147438	OSMANBEY	GAYRETTEPE	19:04:00	5	7	158	19:14:00	10.0
147439	SISHANE	LEVENT	19:04:00	3	8	157	19:06:00	2.0

Fig. 6. `UPDATED_PASSENGER_LIST.csv`: Columns: adds sefer_id (assigned trip), boarding_time, wait_duration_min. Based on the real timetable.

IV. METHODOLOGY

We employ a Genetic Algorithm (GA) to optimize departure times of trains, with the goal of minimizing average waiting time of passengers. The steps of the algorit are as follows:

A. Individual Representation

Each individual in the population is a feasible timetable co sisting of 196 train departures from 06:00 to 23:59. Departi times are computed using passenger demand distributions.

B. Fitness Function

The fitness of an individual is evaluated by simulating passenger boarding behavior. The mean of all passenger waiting times (difference between actual boarding time and entrance time) is taken as the fitness value:

$$f = \frac{1}{|P|} \sum_{p \in P} (b_p - e_p)$$

C. Initial Population

An initial population of 30 individuals is created using demand-based heuristics. Each individual's departure times are distributed proportionally to the number of passengers entering in each hour.

D. Genetic Operators

- **Crossover:** A single-point crossover is performed around a randomly chosen index.
- **Mutation:** Each gap between train departures may be increased or decreased by up to 3 minutes, bounded by [1, 15] minutes.
- **Selection:** Top 5 individuals are chosen with elitism ($e = 3$) and random parent selection from the top 5.

E. Hyperparameters

- Population size: 30
- Number of generations: 40
- Mutation rate: 0.15
- Elitism count: 3

F. Performance

The best individual discovered yielded an average waiting time of **3.23 minutes**, compared to the original **3.65 minutes**. Over 40 generations, both average and best fitness steadily improved (Figure7).

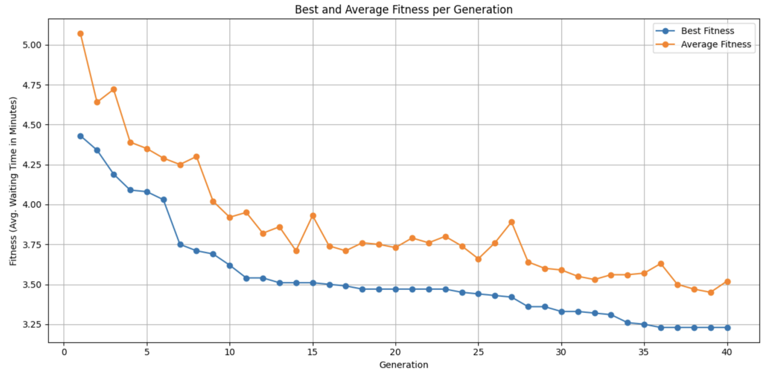

Fig. 7. Best fitness score and average fitness score values per generation.

V. RESULTS

To measure the success of our optimization model, we compared the average passenger waiting times between the existing schedule and the optimized schedule(Figure8) produced by the Genetic Algorithm.

sefer_id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	06:00:00	06:03:00	06:05:00	06:07:00	06:09:00	06:11:00	06:13:00	06:15:00	06:17:00	06:19:00	06:21:00	06:23:00	06:25:00	06:27:00	06:29:00
1	06:15:00	06:18:00	06:20:00	06:22:00	06:24:00	06:26:00	06:28:00	06:30:00	06:32:00	06:34:00	06:36:00	06:38:00	06:40:00	06:42:00	06:44:00
2	06:26:00	06:29:00	06:31:00	06:33:00	06:35:00	06:37:00	06:39:00	06:41:00	06:43:00	06:45:00	06:47:00	06:49:00	06:51:00	06:53:00	06:55:00
3	06:36:00	06:39:00	06:41:00	06:43:00	06:45:00	06:47:00	06:49:00	06:51:00	06:53:00	06:55:00	06:57:00	06:59:00	07:01:00	07:03:00	07:05:00
4	06:39:00	06:42:00	06:44:00	06:46:00	06:48:00	06:50:00	06:52:00	06:54:00	06:56:00	06:58:00	07:00:00	07:02:00	07:04:00	07:06:00	07:08:00
5	06:40:00	06:43:00	06:45:00	06:47:00	06:49:00	06:51:00	06:53:00	06:55:00	06:57:00	06:59:00	07:01:00	07:03:00	07:05:00	07:07:00	07:09:00
6	06:55:00	06:58:00	07:00:00	07:02:00	07:04:00	07:06:00	07:08:00	07:10:00	07:12:00	07:14:00	07:16:00	07:18:00	07:20:00	07:22:00	07:24:00
7	07:03:00	07:06:00	07:08:00	07:10:00	07:12:00	07:14:00	07:16:00	07:18:00	07:20:00	07:22:00	07:24:00	07:26:00	07:28:00	07:30:00	07:32:00
8	07:04:00	07:07:00	07:09:00	07:11:00	07:13:00	07:15:00	07:17:00	07:19:00	07:21:00	07:23:00	07:25:00	07:27:00	07:29:00	07:31:00	07:33:00
9	07:14:00	07:17:00	07:19:00	07:21:00	07:23:00	07:25:00	07:27:00	07:29:00	07:31:00	07:33:00	07:35:00	07:37:00	07:39:00	07:41:00	07:43:00
10	07:19:00	07:22:00	07:24:00	07:26:00	07:28:00	07:30:00	07:32:00	07:34:00	07:36:00	07:38:00	07:40:00	07:42:00	07:44:00	07:46:00	07:48:00

Fig. 8. `OPTIMIZED_SEFER_LIST.csv`: Same format as Figure 6, generated by the Genetic Algorithm.

The average waiting time under the current train schedule was calculated as **3.65 minutes**. After optimization, this value decreased to **3.23 minutes**, representing an improvement of approximately **11.5%**. In addition to reducing the mean, the variance of waiting times also decreased, resulting in a more balanced(Figure9) and equitable passenger experience. Figure10 and Figure11 illustrate the pre-optimization and post-optimization distribution clearly.

passenger_id	boarding_station	destination_station	enterance_time	boarding_station_id	destination_station_id	sefer_id	boarding_time	wait_duration_min
147437	OSMANBEY	SISLI	19:04:00	5	6	153	19:08:00	4.0
147438	OSMANBEY	GAYRETTEPE	19:04:00	5	7	153	19:08:00	4.0
147439	SISHANE	LEVENT	19:04:00	3	8	153	19:04:00	0.0

Fig. 9. `OPTIMIZED_PASSENGER_LIST.csv`: Same columns as Figure 5, but based on optimized schedule output from the Genetic Algorithm.

We also found that the optimization was particularly effective during evening rush hours, as revealed by hourly waiting time comparisons(Figure10 and Figure11). These are periods when the current static schedule fails to adapt to rapid demand fluctuations, causing excessive delays. The optimized schedule dynamically aligns better with demand surges, leading to a more efficient and responsive system.

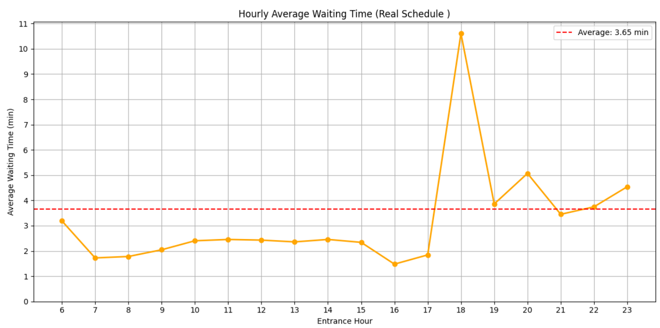

Fig. 10. Distribution of passenger waiting times before and after optimization.

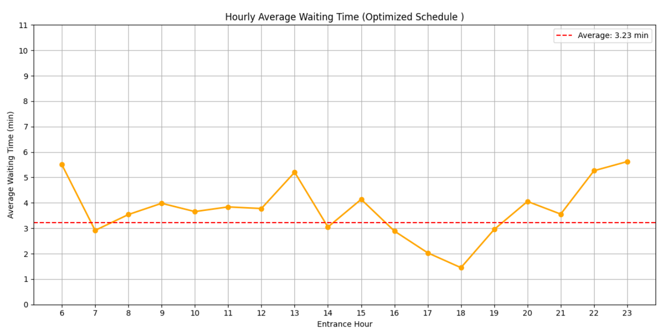

Fig. 11. Distribution of passenger waiting times after optimization. Lower variance and mean is observed after optimization.

Overall, these results indicate that heuristic methods like Genetic Algorithms can significantly improve service quality without requiring additional physical resources.

VI. Conclusion

This study demonstrates the effectiveness of heuristic methods such as genetic algorithms in solving inefficient scheduling problems in public transportation systems. We combined real passenger data with individual passenger behaviors and adjusted train departure times according to demand hours. As a result of our simulation, we observed a significant increase in efficiency. Future studies can apply this study to other metro lines and make better optimizations with clearer data.

Acknowledgment

We thank the Istanbul Metropolitan Municipality for providing open access to the metro datasets that made this study possible.

References

[1] Istanbul Metropolitan Municipality Open Data Portal. Available: https://data.ibb.gov.tr