Optimization of Bicycle Allocation in Urban Rental Systems to Minimize Customer Dissatisfaction

Ömer Faruk San
Artificial Intelligence and Data Science
Computer and Informatics
san22@itu.edu.tr
150220307

Mustafa Kerem Bulut
Artificial Intelligence and Data Science
Computer and Informatics)
bulut22@itu.edu.tr
150220303

Abdullah Vefa Yankın
Artificial Intelligence and Data Science
Computer and Informatics
yankin22@itu.edu.tr
150220318

***Abstract*—Urban bike-sharing systems often suffer from inefficiencies caused by uneven bicycle distribution—some stations are empty while others overflow—leading to user dissatisfaction. This study presents a data-driven Linear Programming (LP) model to optimize the initial allocation of bicycles across stations. Using real-world data from the Konya Metropolitan Municipality, we estimate net flow patterns and model overflow and underflow as dissatisfaction costs.**

We evaluate several strategies, including cost minimization and maximizing operational continuity. The proposed approach, while focused on bike-sharing, is applicable to other shared mobility systems like e-scooters or portable battery rentals, offering valuable insights for urban mobility planning.

***Index Terms*—Optimization, Bike-Sharing, Linear Programming, Urban Mobility, Resource Allocation.**

I. Introduction

Bike-sharing systems have become an essential component of urban transportation in many smart cities, offering a sustainable and flexible alternative to traditional travel modes. Despite their benefits, these systems frequently encounter operational inefficiencies—most notably, the uneven distribution of bicycles across docking stations over time. When users are unable to find a bike to rent or a space to return one, the result is frustration and reduced service reliability.

This project addresses the problem of optimizing bicycle allocation to minimize such dissatisfaction. Rather than focusing solely on daily rebalancing, our approach considers long-term patterns in rental and return behavior to compute optimal initial allocations for extended time horizons, such as weeks or months. The goal is to determine how many bikes should be placed at each station in a way that reduces future overflow (too many bikes) and underflow (not enough bikes) events, while adhering to capacity constraints.

To solve this, we develop several Linear Programming (LP) models based on historical usage data from the Konya Metropolitan Municipality's bike-sharing system. These models incorporate both cost-based objectives—such as minimizing user dissatisfaction—and alternative strategies that aim to maximize the duration of uninterrupted operation. The framework also accounts for network layout by implicitly assuming that users may walk to nearby stations when necessary.

While the primary focus is on bicycles, the proposed methodology is adaptable to other shared mobility services, including electric scooters and portable battery rentals. As such, it offers a scalable and data-driven tool for enhancing the efficiency and user experience of smart city transportation systems.

II. Problem Definition

In urban bike-sharing systems, customer dissatisfaction is primarily due to two critical failure scenarios:

- **Stockout Events:** No bikes available at a station when a user attempts to rent.
- **Overflow Events:** No empty slots available at a station when a user attempts to return a bike.

To address these inefficiencies, we aim to optimize the initial daily allocation of bikes across stations in a way that minimizes cumulative customer dissatisfaction. Our model operates on aggregated **daily net flow** values rather than hourly data, as hourly flow was found to be too sparse and noisy for stable optimization.

The bike flow for each station is defined as:

$$a_n = I_n + \Delta_n$$

where I_n is the initial number of bikes at station n, and Δ_n is the projected net bike flow over the time horizon (e.g., one month).

Two slack variables are introduced:

$$x_n = \max(0, a_n - C_n) \quad \text{(overflow)}$$
$$y_n = \max(0, -a_n) \quad \text{(underflow)}$$

A weighted dissatisfaction function is defined as:

$$z_n = \alpha \cdot x_n + \beta \cdot y_n$$

with α and β being tunable cost weights (e.g., $\alpha = 1$, $\beta = 1.5$ to penalize underflows more heavily).

The total cost to be minimized:

$$\text{Cost} = \sum_{n=1}^{N} z_n$$

The optimization problem is subject to:

- Capacity constraints: $0 \leq I_n \leq C_n$
- Global bike count constraint: $\sum I_n = B_{\text{total}}$

Additionally, the project explores **alternative objective functions**, such as maximizing operational duration or ensuring long-term balance. These variants are modeled via linear programming as well and are detailed in later sections of this report.

III. Dataset Description

Our project utilizes data provided by the Konya Metropolitan Municipality through the ULASAV Open Data Platform. Two main datasets are used:

- **Bicycle Usage Data (2022–2023):** [1] Contains detailed records of individual bike rentals, including timestamps for rental start and end events as well as the corresponding station codes.
- **Station Metadata:** [2] Provides information on the location and maximum capacity of each docking station in the network.

A. Data Cleaning and Preprocessing

The raw data required significant preprocessing due to various quality issues, including character encoding errors, missing values, and inconsistent station identifiers. The following steps were performed:

1) **Character Correction:** Common encoding artifacts (e.g., “Ã¶” in place of “ö”) were fixed by applying a mapping of incorrect to correct Turkish characters.
2) **Data Consolidation:** Monthly Excel files were merged into a unified dataset, and only relevant fields such as timestamps and station codes were retained.
3) **Station Code Filtering:** Records were filtered to include only those with valid station codes starting with “KON.” Unofficial or inconsistent station identifiers such as “KON 2005” and “KON 3000” were excluded.
4) **Datetime Normalization:** Rental times were parsed into standardized ‘datetime‘ objects and truncated at both the hourly and daily levels to support time-based aggregation.
5) **Flow Aggregation:** For each station, the average number of daily rentals and returns was computed. The net flow was calculated as:

$$\text{avg\_flow}_n = \text{avg\_returns}_n - \text{avg\_rentals}_n$$

B. Justification for Using Daily Flow

As shown in Figure 1, hourly rental and return activity is generally low and sparse, which leads to unstable behavior when used directly in optimization. To handle this, we used daily-aggregated flows as the main unit of analysis.

C. Station Dynamics and Imbalance Patterns

To understand structural imbalances across the system, we visualized the average daily net flow at each station (Figure 2) and compared it to station capacities (Figure 3). These plots reveal that certain stations consistently experience excessive demand or supply, leading to persistent underflow or overflow pressures.

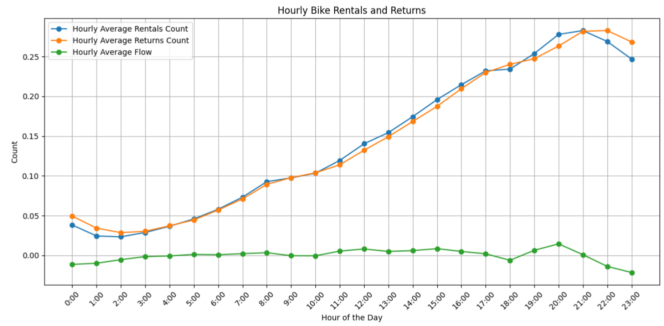

Fig. 1: Hourly average rentals, returns, and net flow per station.

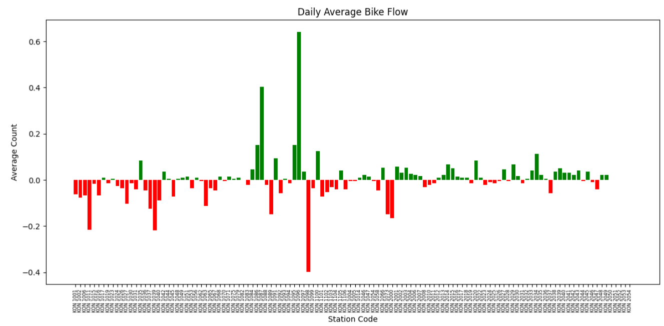

Fig. 2: Daily average net bike flow by station. Green: net inflow, Red: net outflow.

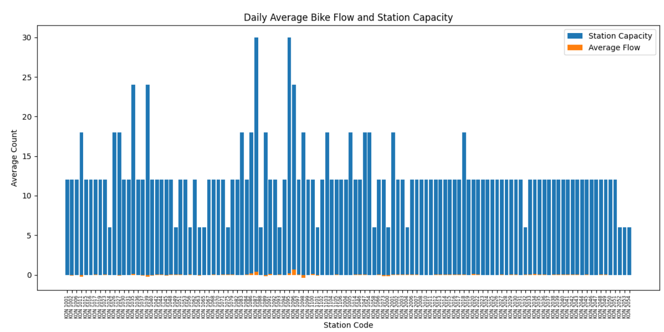

Fig. 3: Comparison of station capacities and average daily net flow.

D. Extended Horizon Patterns

To capture broader temporal trends and assess long-term imbalances, weekly aggregated flows were also computed (Figure 4). These trends indicate that certain stations show sustained mismatches between supply and demand, supporting the case for optimization on a monthly basis.

E. Final Processed Dataset

The final processed dataset includes:

- Daily and weekly average rental, return, and net flow statistics for each station.
- Validated station capacities and metadata.
- Cleaned transaction records spanning two full years.

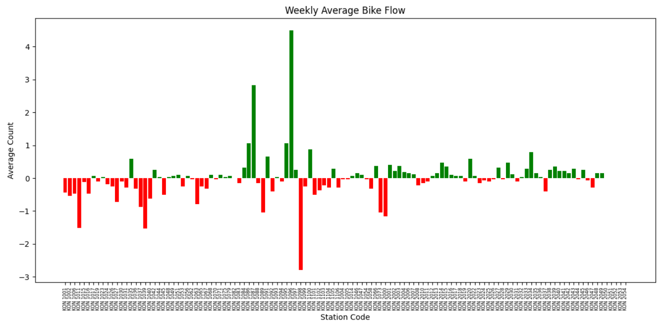

Fig. 4: Weekly average bike flow across all stations.

This refined dataset serves as the basis for constructing and solving the optimization model introduced in the following sections.

IV. Modeling and Formulation

Our objective is to determine optimal initial bicycle allocations to minimize system-wide dissatisfaction and ensure continuous, balanced operation. To address this, we formulated four variants of Linear Programming (LP) models using different objective functions and structural constraints.

A. Decision Variables and Parameters

Let the following denote key variables and inputs:

- I_n: Number of bikes initially placed at station n
- C_n: Capacity of station n
- Δ_n: Projected net flow at station n (e.g., over 30 days)
- x_n: Overflow (bikes exceeding capacity after Δ_n flow)
- y_n: Underflow (bike shortage after Δ_n flow)
- D_n: Deviation of I_n from half-capacity (used in balancing models)
- t: Operational duration (days) until first failure (overflow or underflow)

B. Model 1: Dissatisfaction Minimization

This baseline model minimizes a weighted cost of dissatisfaction due to overflow and underflow. The net final count after flow is:

$$a_n = I_n + \Delta_n$$

Slack variables model constraint violations:

$$x_n = \max(0, a_n - C_n)$$
$$y_n = \max(0, -a_n)$$

The objective function is:

$$\text{minimize} \quad \sum_n \alpha x_n + \beta y_n$$

with $\alpha = 1.0$ and $\beta = 1.5$.

Constraints:

$$0 \le I_n \le C_n, \quad \forall n$$
$$x_n \ge a_n - C_n, \quad y_n \ge -a_n$$
$$\sum_n I_n = B_{\text{total}}$$

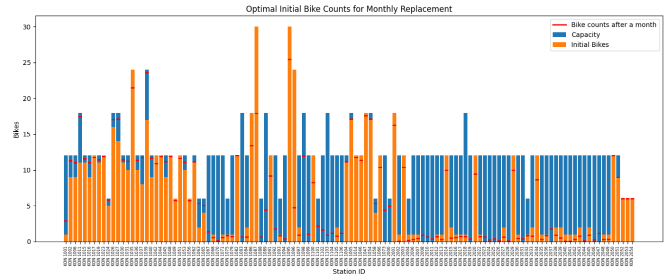

Fig. 5: Optimized bike allocation (monthly horizon, dissatisfaction minimization).

C. Model 2: Maximizing Operational Time

This formulation seeks to extend the system's serviceable period by allocating bikes such that stations stay within operational range for as long as possible:

$$\text{maximize} \quad t$$

Here, t represents the number of days the system can function without any station exceeding capacity (overflow) or falling below zero bikes (underflow), assuming constant average net flow. This model de-emphasizes direct dissatisfaction and instead focuses on robustness and long-term operability.

Based on the solution, the system achieves a minimum failure time of:

$$t = 36 \text{ days}$$

meaning all stations remain operational for at least 36 days under the optimized allocation.

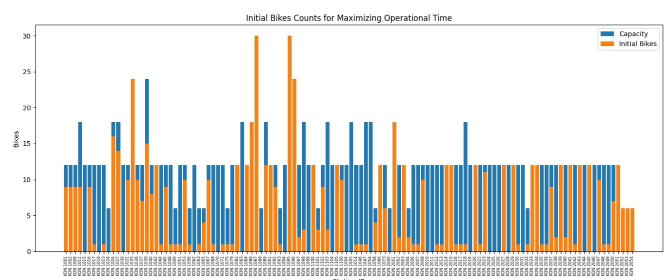

Fig. 6: Initial allocations maximizing estimated operational time ($t = 36$).

D. Model 3: Maximize Days Without Failure with Balanced Allocation

This model introduces a continuous variable t representing the number of days the system should function without any station hitting overflow or underflow. The objective is:

$$\text{maximize} \quad t - \sum_n (\lambda D_n + \alpha x_n + \beta y_n)$$

Where:

- $\lambda = 0.001$ is a small weight penalizing deviation from balanced capacity
- x_n, y_n are overflow/underflow values as before

- D_n measures how far I_n is from $\frac{1}{2}C_n$

Flow is scaled by time:

$$a_n = I_n + t \cdot \Delta_n$$

Constraints are the same with added deviation terms:

$$|I_n - 0.5C_n| \leq D_n, \quad \forall n$$
$$x_n \geq a_n - C_n, \quad y_n \geq -a_n$$

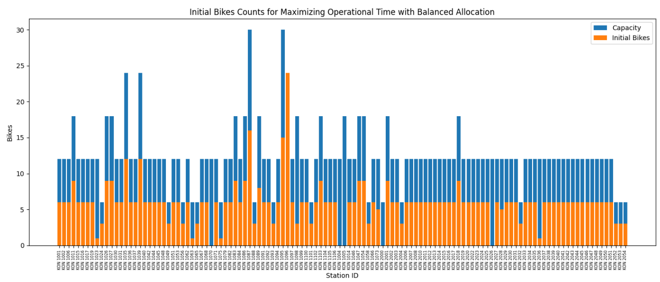

Fig. 7: Bike allocation maximizing days with deviation-balanced strategy.

E. Model 4: Operational Time Maximization with Minimum Bike Constraint

This variant builds upon Model 3 by adding a hard constraint:

$$I_n \geq 2 \quad \forall n$$

This ensures each station has at least two bikes initially, making the solution more applicable to real-world conditions.

The objective and constraints mirror Model 3, with t being maximized subject to deviation, overflow/underflow, and the minimum bike limit. Here, t represents the number of days the system can operate without any station experiencing overflow or underflow.

Based on the results, the system achieves:

$$t = 24 \text{ days}$$

indicating all stations can remain operational for at least 24 days under this more practical constraint.

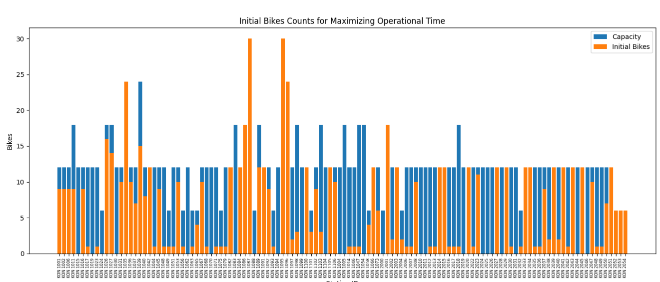

Fig. 8: Optimized allocation with minimum 2-bike constraint and operational time maximization ($t = 24$).

F. Interpretation of t

In Models 3 and 4, t represents the number of days before any station is expected to fail, assuming constant average flow. It is computed analytically as:

$$t = \min_n \begin{cases} \frac{C_n - I_n}{\Delta_n}, & \Delta_n > 0 \\ \frac{I_n}{|\Delta_n|}, & \Delta_n < 0 \\ \infty, & \Delta_n = 0 \end{cases}$$

This allows the planner to estimate system survivability over time and trade off between robustness and fairness.

V. Solution and Results

A. Implementation Environment

The optimization models were implemented in Python, utilizing the `PuLP` library to define and solve linear programs. The CBC solver, integrated with PuLP, was used to obtain solutions. Data preprocessing, aggregation, and visualization were performed using `pandas`, `matplotlib`, and `seaborn`.

All models were run on a standard personal computer, and the LP formulations converged within seconds, reflecting the tractability of the problem despite involving integer and continuous variables across dozens of stations.

B. Results Summary

Key findings from each model are summarized below:

- **Dissatisfaction Minimization:** Provided allocations that reduced immediate overflow and underflow costs, but often left some stations vulnerable after a few days of usage.
- **Operational Time Maximization:** Produced allocations that extended the system's survival time by avoiding early failures, especially for high-variance stations.
- **Balanced Allocation with t Maximization:** Balanced initial bike counts around half-capacity while maximizing t, resulting in smoother long-term behavior and fairness across stations.
- **Minimum Bike-Constrained Model:** Added robustness by ensuring all stations began with at least 2 bikes, avoiding zero-allocation risks while still achieving high operational durations.

Figure 5 through Figure 8 visually compare the initial bike allocations across these strategies.

C. Trade-off Analysis

Models prioritizing immediate cost minimization tend to concentrate resources at historically problematic stations. In contrast, models that maximize operational time distribute bikes more evenly, sacrificing some short-term optimality for long-term resilience.

Balancing these strategies offers planners a flexible toolset. Depending on whether the system prioritizes fairness, robustness, or minimizing customer complaints, a different LP formulation may be preferred.

VI. Conclusion and Future Work

In this study, we handled a key issue faced by modern bike-sharing systems: how to distribute bikes across stations in a way that minimizes user dissatisfaction and enhances overall efficiency. By analyzing historical rental data from the Konya Metropolitan Municipality, we estimated average net flows at each station and developed four linear programming (LP) models to test different optimization strategies.

Each model was designed to capture different aspects of system performance. The first focused on minimizing dissatisfaction by penalizing shortages and overflows. The others introduced new goals—such as maximizing the number of operational days and ensuring a more balanced or equitable distribution of bicycles. These variations allowed us to explore the trade-offs between fairness, robustness, and operational efficiency.

Our results showed that models emphasizing long-term system survivability and balanced allocations tended to produce more resilient and equitable outcomes. In particular, ensuring a minimum level of service at each station proved important for maintaining overall system reliability.

Future Work

While our models offer promising solutions, there are several ways this work could be extended in the future:

- **Dynamic Redistribution:** Introducing rebalancing actions throughout the day or modeling multiple time periods could help the system adapt to changing demand patterns more effectively.
- **Stochastic Demand Modeling:** Considering the uncertainty and variability in user behavior by using probabilistic or robust optimization approaches could improve the system's performance under real-world conditions.
- **User Behavior Integration:** Including factors like users' willingness to walk to nearby stations when their preferred station is empty could make the model outcomes more realistic and user-centric.
- **Multi-objective Optimization:** Exploring solutions that simultaneously balance cost, fairness, and robustness could offer policymakers a range of trade-offs, possibly through Pareto-efficient frontiers.

Overall, the methodology presented here can be adapted to other shared mobility systems such as e-scooters or portable battery rentals. It also lays a foundation for broader smart city applications where effective resource allocation is crucial.

References

[1] Konya Metropolitan Municipality. *Kiralık Bisiklet Kullanım Verileri*. ULASAV Açık Veri Platformu, 2025. Available at: https://ulasav.csb.gov.tr/dataset/42-kiralik-bisiklet-kullanim-verileri

[2] Konya Metropolitan Municipality. *Paylaşımlı Kiralık Bisiklet İstasyonları Konumları*. ULASAV Açık Veri Platformu, 2025. Available at: https://ulasav.csb.gov.tr/dataset/42-paylasimli-kiralik-bisiklet-istasyonlari-konumlari/resource/2fd88ce8-2e7c-4aa5-917a-0f19ae08b845