Optimizing Taxi-Passenger Group Assignment in Ride-Sharing Systems Using Greatest Common Divisor Approach

Emmanuel Gbey
gbey.emman@gmail.com

Allgemeine Vehicle Services Ltd

Charles Atombo
Ho Technical University

Research Article

Keywords: Ridesharing, Greatest common divisor, Hungarian algorithm, Ride allocation optimization, First-come-first-serve, Vehicle utilization

Posted Date: November 4th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-8008306/v1

Additional Declarations: The authors declare no competing interests.

Optimizing Passenger Group Vehicle Assignment in Ride Sharing Systems Using Greatest Common Divisor Approach

Emmanuel Kofi Gbey [*a], Charles Atombo [b]

[a] Technical Department, Allgemeine Vehicle Services Ltd, Ho, Ghana.

[b] Department of Mechanical Engineering, Ho Technical University, Ho, Ghana.

**Corresponding author's email addresses:* gbey.emman@gmail.com

Co-author's email address: catombo@htu.edu.gh

Abstract: With rising urban populations, optimizing passenger group-to-vehicle allocation (PGVA) is critical for enhancing ride-sharing efficiency, particularly when integrated with transit networks. Existing PGVA methods often underperform by overlooking arithmetic compatibility between passenger group sizes and vehicle capacities. Traditional approaches prioritize spatial or temporal factors but neglect structural relationships inherent in passenger group-vehicle matching. This study introduces the Greatest Common Divisor (GCD) method, a novel framework leveraging number-theoretic principles to optimize resource allocation. The GCD-based method addresses PGVA problem by decomposing passenger group sizes and vehicle capacities into prime factors, ensuring mathematically rigorous compatibility while minimizing wasted capacity and computational complexity. Under the tested simulation conditions, the GCD-based method demonstrated superior performance in reducing eVMT and VMT compared to the benchmark algorithms. It significantly reduced empty and total vehicle miles travelled by over 70% and 85% respectively, compared to the Hungarian algorithm, while avoiding the inefficiencies of a first-come-first-served strategy. The GCD-based compatibility score successfully encodes the qualitative notion of a "good fit", leading to more efficient resource utilization and directly contributing to the model's performance. While relatively computationally more intensive, the proposed GCD-based model solves problems of realistic scale within a timeframe that is practical for operational deployment in modern ride-sharing platforms. The method bridges a critical gap in ridesharing optimization and aligns with sustainability goals through inherent resource efficiency. This study supports data-driven strategies for passenger-centric mobility systems that balance demand, capacity, and environmental impact by prioritizing arithmetic alignment.

Keywords: Ridesharing; Greatest common divisor; Hungarian algorithm; Ride allocation optimization; First-come-first-serve; Vehicle utilization.

1. Introduction

Ridesharing platforms such as Uber and Lyft have transformed urban transportation by providing on-demand mobility services that enhance convenience for users [1, 2]. These systems do not only streamline personal travel but also hold significant potential for optimizing resource allocation. The optimisation of resource allocation is essential to meeting the growing demand for shared mobility solutions [3, 4]. Despite substantial advancements, existing algorithms often fail to maximize ride efficiency and minimize waiting times due to the inherent complexities of user behaviour, fluctuating demand patterns, and dynamic traffic conditions. As urban populations expand and mobility demands increase, these challenges are further worsened, particularly when ridesharing systems must integrate seamlessly with mass transit networks [5].

The efficient allocation of passenger groups to available vehicles is central to the optimisation of ridesharing platforms [6–8]. This problem, referred to in this study as the Passenger Group-Vehicle Allocation (PGVA) problem, involves assigning passenger groups with varying sizes and destination preferences to a fleet of vehicles with diverse capacities and operational constraints. The goal is to achieve an allocation that balances multiple objectives, including minimizing total travel distance, maximizing vehicle utilization, reducing operational costs such as fuel consumption and vehicle wear, and enhancing passenger satisfaction through reduced waiting times.

Traditional approaches to solving the PGVA problem, including First-Come-First-Served (FCFS) [9], greedy algorithms [10], linear programming (LP) [11], and integer programming (IP) [12], have demonstrated limited effectiveness in addressing the complexities of real-world scenarios. The FCFS method, while simple and intuitive, neglects key factors such as passenger group compatibility and vehicle capacities, often leading to suboptimal system performance and inefficiencies. Similarly, greedy algorithms, which prioritize immediate gains, fail to consider the global implications of individual allocation decisions [10]. As a result, these methods often yield suboptimal solutions, especially in dynamic and large-scale settings.

More advanced methods, such as LP and IP, provide rigorous mathematical frameworks for solving the PGVA problem by formulating objective functions and constraints [11, 12]. However, their applicability is constrained by computational challenges. LP models struggle with the “curse of dimensionality”, while IP models, which account for integer constraints, are NP-hard and computationally intractable for large-scale problems. These limitations underscore the need for innovative approaches that balance computational efficiency with the ability to handle complex real-world constraints.

In response to these challenges, this study introduces the Greatest Common Divisor (GCD)-based Mixed Integer Linear Programming (MILP) method as a novel and robust solution to the PGVA problem. This approach uses the mathematical principles of prime factorization and GCD to represent passenger group sizes and vehicle capacities as prime numbers and their factors. This representation enables a compatibility-driven allocation process that ensures logical passenger group-to-vehicle assignments. This minimizes wasted space and enhances overall system efficiency. This method do not only optimize vehicle utilization but also reduces operational inefficiencies by matching the prime factors of passenger groups with those of vehicle capacities.

Unlike traditional methods, the GCD-based approach offers relatively superior performance under the specific conditions of the experiment in terms of computational efficiency, scalability, and adaptability. While LP and IP models struggle with exponential complexity [13], the GCD-based method exhibits “quasi” NP-hard complexity under most conditions, enabling it to handle large-scale problems effectively. This is because, while the computation of the compatibility score (involving prime factorization and GCD) is efficient, the overall assignment problem remains NP-hard. Additionally, the method guarantees optimal solutions under specific conditions, such as when passenger group sizes are represented by distinct primes and vehicle capacities can be factored accordingly. This contrasts sharply with heuristic methods like greedy algorithms, which often yield suboptimal results.

The adaptability of the GCD-based method further extends to its ability to integrate diverse constraints commonly encountered in PGVA problems, including vehicle capacities, passenger preferences, and time windows. The GCD-based method addresses the complex requirements of real-world transportation systems without compromising significantly the computational efficiency by seamlessly incorporating these constraints into the prime factorization framework. Moreover, the scalability of the method enables it to manage large-scale PGVA problems involving numerous passengers, vehicles, and constraints without significant performance degradation.

This study builds on foundational work in PGVA optimization and introduces a unique perspective by employing number-theoretic abstractions in the allocation problem. The GCD-based method provides a structured framework for decomposing complex problems into manageable sub-problems, leveraging the principles of prime factorization to enhance computational efficiency and improve decision-making. Furthermore, the environmental benefits of this approach resonate with the goals of the sharing economy, which emphasize resource efficiency and the reduction of carbon footprints [14, 15]. The implications of such an approach suggest significant improvements in service responsiveness and user satisfaction, aligning with emerging paradigms in autonomous vehicle integration and collaborative resource utilisation within shared mobility frameworks, as highlighted in recent studies [16–18].

2. Literature review

The optimization of ride allocation in ridesharing systems has gained significant attention recently due to its potential to enhance transportation efficiency, reduce traffic congestion, and minimize environmental impact [19, 20]. Studies have proposed various methods to address the complexities of matching riders with drivers, incorporating factors such as time constraints, route optimization, and user preferences [19, 21–23]. Among these, tree-based algorithms have emerged as the suitable methods for tackling the dynamic and computationally intensive challenges inherent in ridesharing systems.

2.1 Tree-based Algorithms and GCD approach

Tree-based algorithms, such as branch-and-bound, hierarchical clustering trees, or recursive partitioning, are expressive and well suited to complex constraints (such as time windows, compatibility restrictions, routing considerations) but they explore combinatorial branching that grows rapidly with instance size [24]. They discover feasibility bounds only after exploring nodes or solving relaxations and their runtime depends heavily on the expression of problem, cut strength, and heuristic quality [25]. In contrast, the GCD-based compatibility measure provides an immediate, problem-level bound independent of such solver behaviour. This method decomposes complex ride allocation problems into smaller, more manageable subproblems, which help to achieve optimal solutions. Studies have demonstrated the effectiveness of tree-based algorithms in reducing computational complexity and improving system scalability [26, 27].

Consequently, the GCD-based approach is preferable when the primary difficulty is integer packability driven by multi-level capacity rather than complex side constraints. It is specifically valuable in high-throughput or real-time settings where quick, provable statements about feasibility or minimal delay are needed before more costly optimization is attempted. Even when additional constraints exist, using GCD as a preprocessing step reduces the downstream problem size and yields stronger initial bounds for any tree-based search. Also, the GCD-based approach offers a mathematically grounded and computationally efficient method for the passenger group–vehicle assignment (PGVA) problem. The GCD-based approach does this by computing the GCD of vehicle capacities and the passenger groups. It immediately identifies whether total passenger demand can exactly be accommodated by specific vehicle capacity and, if not, quantifies the minimum inevitable difference. This provides a direct feasibility test and lower bound that tree-based algorithms can only establish after extensive branching or relaxation.

Moreover, the GCD-based method reduces the problem to smaller, normalized units of capacity, compressing the search space and simplifying allocation. This makes it highly effective for fleets with shared divisors, where large combinatorial searches can be avoided. While tree-based algorithms are more flexible in handling additional constraints, they are computationally intensive and offer no such instant feasibility guarantees.

Thus, the GCD-based approach is preferable for PGVA problem when capacity granularity is the main challenge, providing rapid, interpretable, and provably optimal bounds that enhance both efficiency and tractability.

2.2 Alternative Methods and GCD Approach in Ride Allocation

While tree-based algorithms remain widely applied in ridesharing optimization, several alternative methods have also been explored. These include dynamic tree algorithms [28], reinforcement learning [29], multi-agent systems (MAS) with R-trees [28], and tri-objective optimization models [29], each offering unique strengths and limitations.

Dynamic tree algorithms efficiently match riders and drivers in real time but face scalability issues as demand grows, leading to longer processing times. The GCD-based approach addresses this by decomposing ride allocation into smaller subproblems using prime factorization. This reduces the computational burdens and enabling faster processing, even in high-demand contexts.

Reinforcement learning methods adapt well to dynamic environments but require large datasets, extensive training, and high computational resources, making them less practical for real-time deployment [30–32]. In contrast, the GCD-based approach avoids data-intensive learning, ensuring consistent optimization performance without iterative training.

MAS integrated with R-trees optimize spatial matching by treating riders and drivers as autonomous agents, yet they are computationally intensive and often overlook temporal factors [33, 34]. The GCD-based approach, which is tree-based in nature, offers a more holistic solution by incorporating both spatial and temporal attributes through prime factor encoding, allowing for efficient multi-criteria matching such as time, cost, and environmental impact [19].

Tri-objective optimization models seek to balance travel time, emissions, and user satisfaction through metaheuristics like genetic algorithms and particle swarm optimization [7, 35]. While effective, they are computationally expensive and risk compromising outcomes for individual users. The GCD-based approach simplifies this process by breaking the allocation problem into solvable subproblems, ensuring faster decision-making while preserving both system-wide efficiency and individual satisfaction.

While existing methods address dynamic constraints and multi-objective optimization, their scalability and computational demands highlight the need for a more direct strategy. The GCD-based approach offers an immediate feasibility check and reduces allocation into normalized capacity units, avoiding the extensive branching required in tree-based algorithms. It provides rapid, interpretable results suitable for real-time applications and serves as an effective preprocessing step that compresses the search space. Consequently, the GCD method delivers a scalable and provably optimal alternative for ridesharing optimization.

3. Material and Methods

3.1 Greatest Common Divisor (Prime Factorization) Algorithm

This algorithm (as shown in Table 1) provides a method for finding the prime factors of a given integer $\boldsymbol{n}$. It begins by dividing out the smallest prime (2) and then proceeds to check for divisibility by odd integers up to $\sqrt{n}$.

The time complexity of the prime factorization algorithm is $O(\sqrt{n})$, where n is the input number. This is due to the iteration over potential factors up to $\sqrt{n}$. The algorithm efficiently handles large numbers by first factoring out powers of 2 and then proceeding with odd integers.

Table 1 Greatest Common Divisor (Prime Factorization) Algorithm

Line No.	Pseudocode
	Input: Positive integer n
	Output: List of prime factors of n
1	Initialize factors ← [ ]
2	**While** n is divisible by 2:
3	Append 2 to factors
4	n ← $n/2$
5	**For** i from 3 to $\sqrt{n}$., incrementing by 2:
6	While n is divisible by i:
7	Append i to factors
8	$n \leftarrow n/i$
9	**If** $n > 2$:
10	Append n to factors
11	**Return** factors

3.2 Mathematical Formulation of GCD-based Passenger Group Vehicle Assignment (PGVA) Model

3.2.1 Sets and Indices

The goal of the GCD-based Passenger Group Vehicle Assignment (PGVA) model is to assign each passenger group to at most one taxi such that total ridership is maximized while non-productive vehicle mileage and waiting times are minimized.

Let us consider a set of passenger groups $\mathcal{P} = \{1,2, \ldots, N_P\}$ and a set of available rideshare vehicle (Taxi) $\mathcal{T} = \{1,2, \ldots, N_T\}$. Each passenger group $i \in \mathcal{P}$ is characterised by a group size g_i (in number of passengers) and a location in space, while each vehicle (or Taxi) $j \in \mathcal{T}$ has a seating capacity c_j and an initial spatial location. Table 2 depicted some parameters for the mathematical model.

Table 2 Description of Some Parameters for the Mathematical Model

Symbol	Description
g_i	Size (number of passengers) of group i
c_j	Capacity of taxi $\boldsymbol{j}$
d_{ij}	Euclidean distance between group i and taxi j
τ	Compatibility threshold
α, β, γ	Weighting coefficients for empty VMT, VMT, and wait time (respectively)
θ_{ij}	Compatibility score between group i and taxi j, based on prime factor matching
M	Large constant for linearization (Big-M)
$D_{\max}$	Maximum distance for a trip
T_{trip}	Time it takes to cover trip
μ	eVMT linearization coefficient

3.2.2 Decision Variables

These decision variables are used in the model:

$$x_{ij} = \begin{cases} 1, & \text{if passenger group } i \text{ is assigned to taxi } j, \forall i \in \mathcal{P}, \forall j \in \mathcal{T} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

$$y_j = \begin{cases} 1, & \text{if taxi } j \text{ is activated (used)}, \ \forall j \in \mathcal{T} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$e_j \geq 0 \ \text{ Empty vehicle-miles travelled (eVMT) by taxi } j \tag{3}$$

3.2.3 GCD-based Compatibility Measure

The novelty of this work lies in the compatibility score θ_{ij}, which captures how "suitable" passenger group i is for taxi j.

Each integer g_i and c_j is factorised into its prime factors, which is essentially a prime factor tree.

Let S_i and S_j denote the sets of distinct prime factors of g_i and c_j, respectively.

Let n_f^i denote the multiplicity of prime factor f in g_i.

The compatibility score is defined as:

$$\theta_{ij} = \gcd(g_i, c_j)\left(1 + \frac{|S_i \cap S_j|}{|S_i \cup S_j|} + \frac{\sum_{f \in S_i \cap S_j} \min\left(n_f^i,\ n_f^j\right)}{\max\left(|\text{factors}(g_i)|,\ |\text{factors}(c_j)|\right)}\right) \tag{4}$$

where the first term (GCD) reflects the number-theoretic compatibility, the second term is a Jaccard similarity index on the sets of prime factors, and the third term introduces a frequency bonus, i.e., the number of overlapping prime factors relative to the total number of factors.

The core number-theoretic insight is that the GCD represents the absolute, maximum number of passengers that can be matched without any fractional capacity waste. It is the most fundamental measure of compatibility. A weighted linear combination would decouple this fundamental scaling factor from the finer-grained similarity measures. This potentially allows a high-similarity, but low-GCD match to dominate a high-GCD match, which is counter-intuitive. The multiplicative structure ensures that the GCD remains the primary scaling factor. The multiplicative form naturally encodes θ_{ij} to be directly proportional to the GCD.

Consequently, the terms, (1 + Jaccard + Bonus), act as "similarity multiplier" applied to the base GCD value. The constant 1 ensures that the base GCD value is the minimum score for any feasible match ($g_i \leq c_j$). If the Jaccard similarity and frequency bonus are both zero (meaning no shared prime factors beyond the minimal factors constituting the GCD itself), the score remains $\text{GCD} \times 1$. The Jaccard term, $\frac{|S_i \cap S_j|}{|S_i \cup S_j|}$ and the frequency bonus term, $\frac{\sum_{f \in S_i \cap S_j} \min\left(n_f^i,\ n_f^j\right)}{\max(|\text{factors}(g_i)|,\ |\text{factors}(c_j)|)}$ are additive bonuses within this multiplier. They refine the score for matches that share the same GCD.

A binary parameter is introduced to encode feasibility

$$\eta_{ij} = \begin{cases} 1, & \text{if } \theta_{ij} \geq \tau \text{ and } g_i \leq c_j \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Only feasible assignments are considered in the optimization model.

3.2.4 Objective Function

The objective is to maximize overall compatibility-weighted ridership and minimize operational losses. This gives us:

$$\max Z = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{T}} \theta_{ij} g_i x_{ij} - \alpha \sum_{j \in \mathcal{T}} e_j - \beta \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{T}} d_{ij} x_{ij} - \gamma \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{T}} \left(\frac{d_{ij}}{D_{\max}} \cdot T_{\text{trip}} \right) x_{ij} \tag{6}$$

where $\alpha, \beta, \gamma \in (0, \ldots, 1)$ are weights reflecting the relative importance of the respective terms.

3.2.5 Constraints

Each passenger group can be assigned to at most one taxi:

$$\sum_{j \in \mathcal{T}} x_{ij} \leq 1 \; \forall i \in \mathcal{P}. \tag{7}$$

Taxi capacities must not be exceeded:

$$\sum_{i \in \mathcal{P}} g_i x_{ij} \leq c_j y_j \; \forall j \in \mathcal{T}. \tag{8}$$

A taxi must be activated if it is assigned a passenger group:

$$x_{ij} \leq y_j \; \forall i \in \mathcal{P}, \forall j \in \mathcal{T}. \tag{9}$$

Non-feasible assignments are explicitly excluded:

$$x_{ij} = 0 \;\text{ if } \eta_{ij} = 0, \forall i, j. \tag{10}$$

To linearize empty vehicle-miles, we include:

$$e_j \geq \mu\left(c_j y_j - \sum_{i \in \mathcal{P}} g_i x_{ij}\right) \forall j \in \mathcal{T} \tag{11}$$

$$e_j \leq M y_j \; \forall j \in \mathcal{T} \tag{12}$$

Finally, we impose the domain conditions:

$$x_{ij} \in \{0,1\}, y_j \in \{0,1\}, e_j \geq 0 \; \forall i, j. \tag{13}$$

The proposed GCD-based MILP model for the PGVA problem was implemented in Python using the PuLP optimization library. Passenger groups and vehicles correspond to the sets $\mathcal{P}$ and $\mathcal{T}$ respectively, while the parameters g_i, c_j and d_{ij} were directly extracted from the input data.

The compatibility score θ_{ij} was computed exactly as defined in sub-section 3.2.3. Specifically, both g_i and c_j were factorised into their prime components, and the compatibility score was obtained by combining the greatest common divisor (GCD), the Jaccard similarity, and the factor-frequency bonus. The binary decision variables x_{ij} and y_j were declared as binary variables, while the empty vehicle-miles variable e_j was modelled as a non-negative continuous variable.

The objective function presented in Equation (6) was implemented as the maximisation of the compatibility-weighted ridership minus the weighted penalties for empty vehicle-miles traveled (eVMT), total vehicle miles traveled (VMT) and waiting time. Constraints, equations (7) – (13), were implemented exactly as formulated in the mathematical model: each passenger group was assigned to at most one taxi; taxi capacities were enforced; taxi activation was required prior to assignment; infeasible assignments were excluded; and the Big-M formulation was used to linearize the eVMT relation. Consequently, the implementation is fully consistent with the PGVA formulation described in Sections 3.1 to 3.2.

The objective function (equation 6) incorporates three weighting coefficients (α, β, and γ) to balance competing priorities in ride allocation: reducing operational inefficiencies, optimizing resource utilization, and enhancing passenger satisfaction. Each coefficient has specific economic implications, reflecting the cost dynamics and service quality considerations in ridesharing systems.

The coefficient α is associated with minimizing empty vehicle miles traveled (eVMT). This term captures the direct operational costs linked to fuel consumption, driver compensation, and vehicle wear and tear incurred during unproductive trips without passengers. Economically, reducing eVMT leads to cost savings for ridesharing operators while contributing to environmental benefits, such as decreased carbon emissions and adherence to sustainability goals. A higher value of α emphasizes the importance of cost efficiency and operational effectiveness in ridesharing systems.

The coefficient β is associated with the total vehicle miles traveled (VMT), which includes both passenger-carrying and empty trips. This term reflects broader economic considerations, such as fleet maintenance costs, fuel expenses, and congestion-related impacts on urban mobility. By reducing VMT, the ridesharing system can achieve lower operational costs, improve environmental outcomes, and enhance overall system efficiency. Assigning a greater weight to β prioritizes route optimization and trip consolidation, which are essential for achieving sustainable and cost-effective operations.

The coefficient γ is necessary to reduce passenger waiting time, which is a critical measure of service quality and customer satisfaction. From an economic perspective, minimizing waiting time enhances the user experience, increases customer retention, and boosts ridership, all of which translates to higher revenue and market competitiveness for ridesharing providers. A higher value of γ reflects a customer-centric approach, prioritizing the reduction of waiting times to attract and retain users in a competitive market.

In practice, the relative values of α, β, and γ allow operators to tailor the optimization process to specific business objectives. For instance, a cost-efficiency-focused strategy may assign higher weights to α and β, emphasizing the minimization of operational costs. Conversely, a customer-focused approach might prioritize γ to improve passenger satisfaction and loyalty. A balanced strategy can also be achieved by proportionally adjusting all three coefficients, ensuring an optimal trade-off between operational efficiency and service quality.

This interpretation underscores the flexibility of the proposed model in addressing various economic and operational challenges in ridesharing systems. By explicitly incorporating these coefficients, the model provides a robust framework for decision-making that aligns with both business goals and passenger expectations.

3.2.4 Illustrative Example for the GCD-based Compatibility Measure

It is important to demonstrate how the GCD-based compatibility measure works in the full model of the GCD-based PGVA method. Consider a small-scale PGVA problem with the following parameters:

- Passenger groups: $g = \{4,6\}$
- Taxi capacities: $c = \{6,8\}$

Step 1: Prime Factorization

- Passenger groups:

$$\begin{aligned} 4 &= 2^2 \\ 6 &= 2 \times 3 \end{aligned}$$

- Taxi capacities:

$$\begin{aligned} 6 &= 2 \times 3 \\ 8 &= 2^3 \end{aligned}$$

Step 2: GCD Matrix Calculation

The pairwise greatest common divisors (GCDs) between passenger groups and taxi capacities are computed as follows:

$$\begin{aligned} \kappa_{11} &= \text{GCD}\ (4,6) = 2, \quad \kappa_{12} = \text{GCD}\ (4,8) = 2 \\ \kappa_{21} &= \text{GCD}\ (6,6) = 6, \quad \kappa_{22} = \text{GCD}\ (6,8) = 2, \end{aligned}$$

This results in the GCD matrix:

$$\kappa = \begin{bmatrix} 2 & 2 \\ 6 & 2 \end{bmatrix}$$

Step 3: Assignment Optimization

Assignments are prioritized based on higher GCD values to maximize compatibility. For instance:

- Group g_2 (6 passengers) is optimally assigned to taxi c_1 (capacity 6) with $\kappa_{21} = 6$.

This example highlights how the GCD method leverages prime factorization comparisons to derive efficient vehicle assignments, ensuring minimal capacity waste and optimal resource utilization.

(1 + Jaccard + Bonus) term acts as a "similarity multiplier" applied to the base GCD value. The constant 1 ensures that the base GCD value is the minimum score for any feasible match ($g_i \leq c_j$). If the Jaccard similarity and frequency bonus are both zero (meaning no shared prime factors beyond the minimal factors constituting the GCD itself), the score remains GCD $\times$ 1. The Jaccard term, $\frac{|S_i \cap S_j|}{|S_i \cup S_j|}$ and the frequency bonus term, $\frac{\sum_{f \in S_i \cap S_j} \min\left(n_f^i,\ n_f^j\right)}{\max\left(|\text{factors}(g_i)|,\ |\text{factors}(c_j)|\right)}$ are additive bonuses within this multiplier. They refine the score for matches that share the same GCD. For example, both GCD(4, 6) = 2 and GCD(4, 8) = 2 have a base score of 2. However, 4 and 8 share more prime factor structure (both are powers of 2) than 4 and 6. The Jaccard and bonus terms quantify this, giving a higher score to the (4, 8) match, which is a more "natural" fit and leaves a more usable residual capacity (4 vs. 2).

3.3 Experimental Setup

To empirically evaluate the performance of the proposed GCD-based model, a comprehensive simulation framework was developed. This section details the parameter choices, scenario design, benchmark algorithms, and evaluation metrics. These are all selected to ensure a realistic and rigorous assessment of the model's efficacy in solving the Passenger Group-Vehicle Assignment (PGVA) problem.

3.3.1 Simulation Environment and Data Generation

All experiments were conducted using a custom simulation environment implemented in Python 3.9. The simulation was designed to generate stochastic instances of the PGVA problem that reflect realistic urban ride-sharing conditions.

Passengers and vehicles were distributed within a bounded Euclidean space representing an urban area. The maximum travel distance (D_{max}) was set to 8.0 km. This value was chosen as it represents a realistic upper bound for a typical urban or suburban ride-sharing trip. This aligns with industry data that shows average trip distances of 5-8 km for services like Uber and Lyft [36].

The spatial distribution (x, y) coordinates for both passenger groups and vehicles were generated using a $Beta(2, 2)$ probability distribution scaled by D_{max}. This distribution was selected because it realistically models the clustering of trip origins and vehicle locations around a city centre or high-density zones, as opposed to a uniform random distribution which is less representative of real urban mobility patterns.

The size of each passenger group g_i was sampled from a categorical distribution: sizes [1, 2, 3, 4, 5] with probabilities [0.65, 0.25, 0.08, 0.015, 0.005]. This distribution is well-justified by industry data, which consistently shows that majority of rides are solo passengers, few are pairs, and larger groups are increasingly rare [37]. This prioritizes the common use-case while still testing the model's ability to handle group assignments.

The capacity c_j of each taxi was sampled from a set [4, 6] with probabilities [0.8, 0.2]. This reflects the composition of a standard ride-sharing fleet, which is overwhelmingly dominated by 4-seat sedans, with a smaller proportion of larger vehicles (e.g., SUVs or minivans) having 6-seats. This is a crucial adjustment from simpler assumptions (e.g., uniform random capacities) and ensures the capacity constraints are a meaningful part of the optimization problem.

The experiments utilized a fleet of 150 vehicles to serve 300 passenger groups, resulting in a supply-demand ratio of 0.5. This ratio is realistic for a high-demand urban scenario and creates a sufficiently constrained problem where intelligent assignment is necessary, as not all groups can be served immediately.

3.3.2 Parameter Configuration and Justification

The parameters of the objective function (Eq. 6) were carefully selected to reflect a balanced operational strategy.

The weights were set to $\alpha = 0.33$ (eVMT), $\beta = 0.33$ (VMT), and $\gamma = 0.34$ (wait time). This balanced configuration ($\alpha \approx \beta \approx \gamma$) was chosen to avoid over-prioritizing a single objective and to demonstrate the model's ability to simultaneously optimize for cost reduction (via eVMT and VMT minimization) and service quality (via wait time minimization). The slight preference for wait time (γ) reflects the high importance of customer satisfaction in competitive ride-sharing markets.

The linearization coefficient for empty vehicle miles was set to $\mu = 0.1$. This value was chosen to ensure the penalty term $\alpha \cdot e_j$ is economically meaningful and on a comparable scale to the other terms in the objective function (ridership and distance). As a consequence, it effectively guides the solver towards solutions with higher vehicle utilization.

The feasibility threshold was set to $\tau = 2$. This value was selected empirically to filter out only the most unsuitable matches while still allowing the optimization model a wide range of feasible assignments to choose from. For example, a group with size 3 and a vehicle with capacity 5 have a GCD of 1, which is below the threshold.

3.3.3 Benchmark Algorithms

The proposed GCD-based MILP model was compared against two standard benchmark algorithms.

One of these benchmark algorithms is the greedy First-Come-First-Serve (FCFS) heuristic that assigns each passenger group to the nearest available taxi with sufficient capacity. This algorithm is computationally trivial but is known to lead to suboptimal system-wide efficiency. It represents a common, naive baseline in dispatch systems.

The classic method for solving linear assignment problems is the Hungarian algorithm. It was applied to the matrix of negative compatibility scores ($-\theta_{ij}$) to find the assignment that maximizes total compatibility, ignoring vehicle capacity constraints. This benchmark tests the value of the full MILP formulation. While the Hungarian algorithm finds an optimal matching for the wrong (and simpler) problem, its poor performance on eVMT and VMT highlights the critical importance of explicitly modelling many-to-one assignments and capacity constraints.

3.4 Evaluation Metrics and Experimental Protocol

Performance was evaluated across five key performance indicators (KPIs). These include total ridership, empty vehicle-miles travelled (eVMT), total vehicle-miles travelled (VMT), average waiting time, and computational runtime.

To ensure statistical robustness, the simulation was run for 30 independent iterations ($iterations = 30$). In each iteration, new passenger group and vehicle locations were generated, while the distributions for group sizes and capacities remained consistent. This protocol allows for the calculation of mean performance and standard deviation, ensuring the results are not artifacts of a single random seed. The statistical significance of the results was confirmed using paired t-tests, with effect sizes calculated using Cohen’s d.

3.5 Implementation Details

The GCD-based MILP model was implemented using the PuLP library in Python and solved with the COIN-OR CBC solver (version 2.10.3). All experiments were performed on a standard laptop computer with an Intel Core i7 processor and 12 GB of RAM, running Windows 10. A time limit of 300 seconds was set for the MILP solver, which it never reached, confirming that optimal or near-optimal solutions were found within a practical timeframe.

4. Results

This section presents the experimental results of the comparative analysis between the proposed GCD-based model solved with Mixed Integer Linear Programming (hence, GCD MILP model) and the two benchmark algorithms: First-Come-First-Served (FCFS) and the Hungarian algorithm. The performance was evaluated over 30 independent simulation runs with realistic urban parameters (300 passenger groups, 150 vehicles). Key performance indicators (KPIs) included total ridership, empty vehicle miles travelled (eVMT), total vehicle miles travelled (VMT), average passenger wait time, and computational runtime.

4.1 Vehicle Allocation

The GCD-based approach maximizes compatibility between passenger group sizes and taxi capacities by leveraging GCD-based weights. This ensures optimal vehicle utilization by matching passengers with vehicles that align with their capacity, minimizing instances of overloading or underutilization. The numerical values displayed along the edges indicate the compatibility associated with assigning a particular passenger to a specific taxi. Higher weights generally denote less preferred matches, while lower weights suggest more optimal assignments. The result is a substantial reduction in mismatches and unused vehicle capacity. This can be seen in the compatibility graph representations (in Figures 1, 2, and 3). The taxi to passenger group matches were consistent with real-life scenarios across all iterations in the experiment.

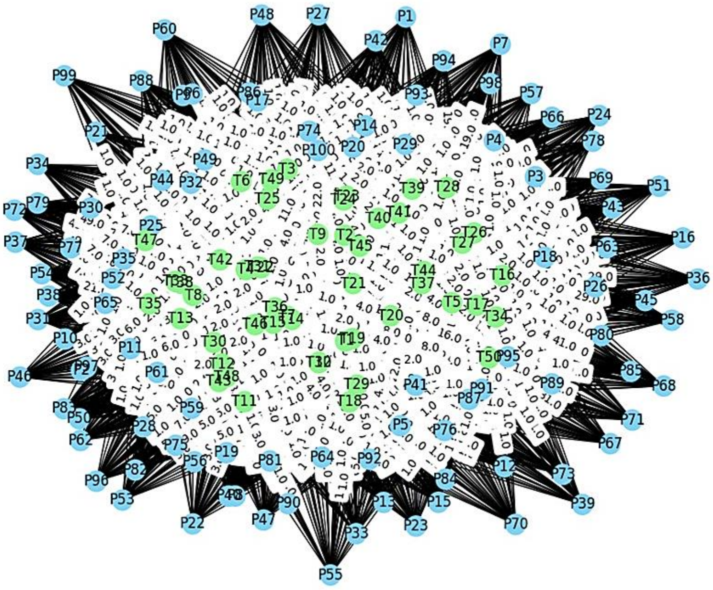

Figure 1 Compatibility Graph Representation at Iteration 0, illustrating passenger nodes, taxi nodes, and weighted edges.

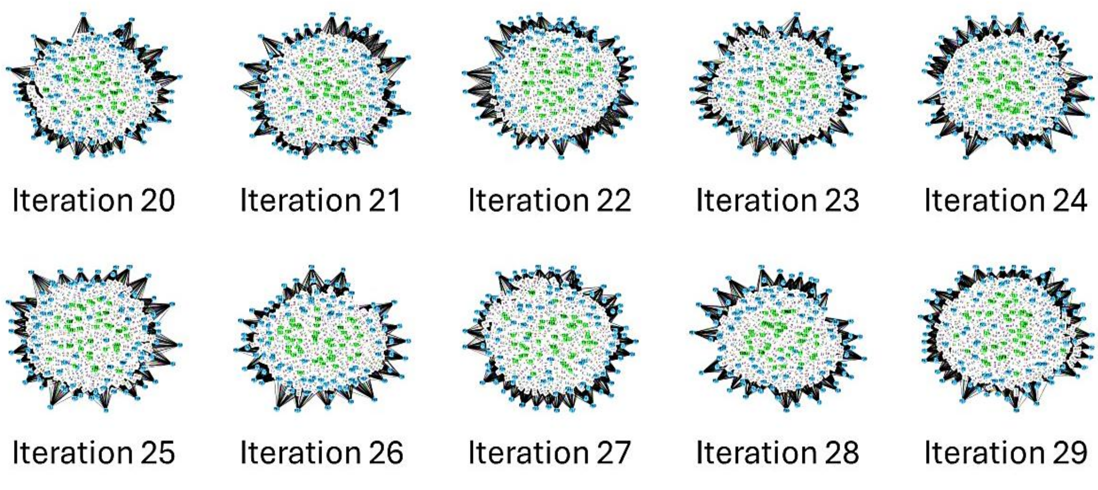

Figure 2 Compatibility Graph Representation from Iterations 20 to 29, showing the evolution of matches.

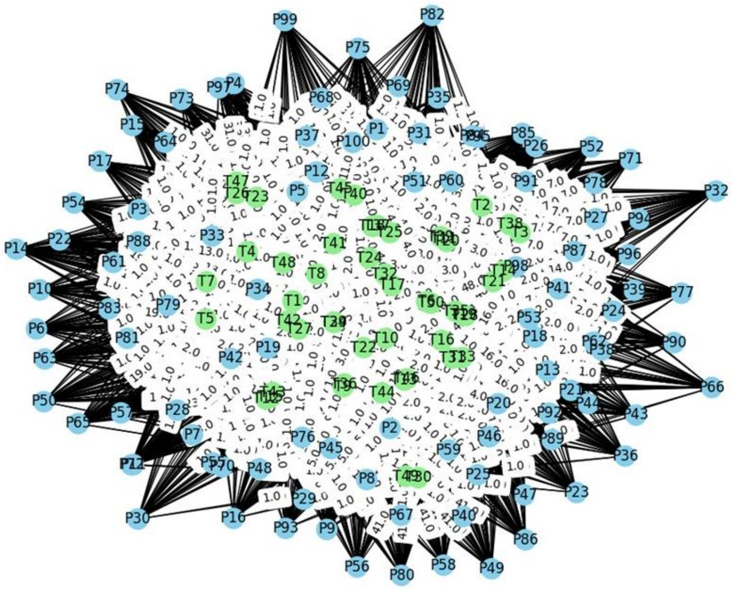

Figure 3 Compatibility Graph Representation at Iteration 30, highlighting optimized allocations.

4.2 Comparative Algorithmic Performance

The aggregate results, shown in Table 3 and Figure 4, reveal significant differences in algorithmic performance. A comprehensive statistical analysis (paired t-tests with Cohen's d effect size) was conducted to validate the significance of these differences. The mean and standard deviations for each algorithm are displayed in Table 3.

Table 3 Summary of performance metrics (mean ± standard deviation) across 30 simulation runs.

Metric	**GCD MILP**	**FCFS**	**Hungarian Algorithm**
Total Ridership	232.27 ± 15.40	**438.00 ± 11.26**	231.07 ± 15.52
eVMT (km)	**6.19 ± 1.09**	22.21 ± 1.32	42.90 ± 1.49
Total VMT (km)	**43.43 ± 5.22**	111.35 ± 5.35	318.88 ± 24.60
Avg. Wait Time (min)	**0.35 ± 0.03**	0.48 ± 0.02	2.59 ± 0.14
Runtime (sec)	6.70 ± 1.13	**0.01 ± 0.01**	**0.00 ± 0.00**

4.2.1 Ridership and Operational Efficiency

As shown in Table 3 and Figure 4, the most critical trade-off is observed between raw ridership and operational efficiency. The FCFS algorithm achieved the highest number of passengers served (438.00 ± 11.26), significantly outperforming both GCD MILP and the Hungarian method ($p < 0.001$, Cohen's $d > 14.9$). The higher FCFS ridership comes at an extreme cost. The GCD-MILP model's strategy is not to "forgo" customers arbitrarily. It is to optimize for profitability per ride rather than sheer volume. Serving an additional passenger group with a poorly compatible vehicle that must travel a long empty distance (high eVMT) can be operationally unprofitable. The cost of the extra VMT and eVMT may exceed the revenue from that fare, especially in a competitive market with tight margins. Therefore, the model makes an economically rational choice to not serve a group if serving it would degrade the overall system efficiency and profitability.

The proposed GCD MILP model demonstrated a profound advantage in minimizing operational waste. It reduced eVMT by 72.2% compared to the Hungarian algorithm (6.19 vs. 42.90) and by 72.1% compared to FCFS (6.19 vs. 22.21). This reduction in non-revenue travel directly translates to lower fuel consumption, reduced emissions, and lower operational costs. Similarly, for total VMT, the GCD MILP model outperformed the Hungarian and FCFS algorithms by 86.4% and 61.0%, respectively. This indicates that the GCD MILP approach not only minimizes empty travel but also finds more geographically efficient assignments, reducing overall congestion and fleet wear-and-tear.

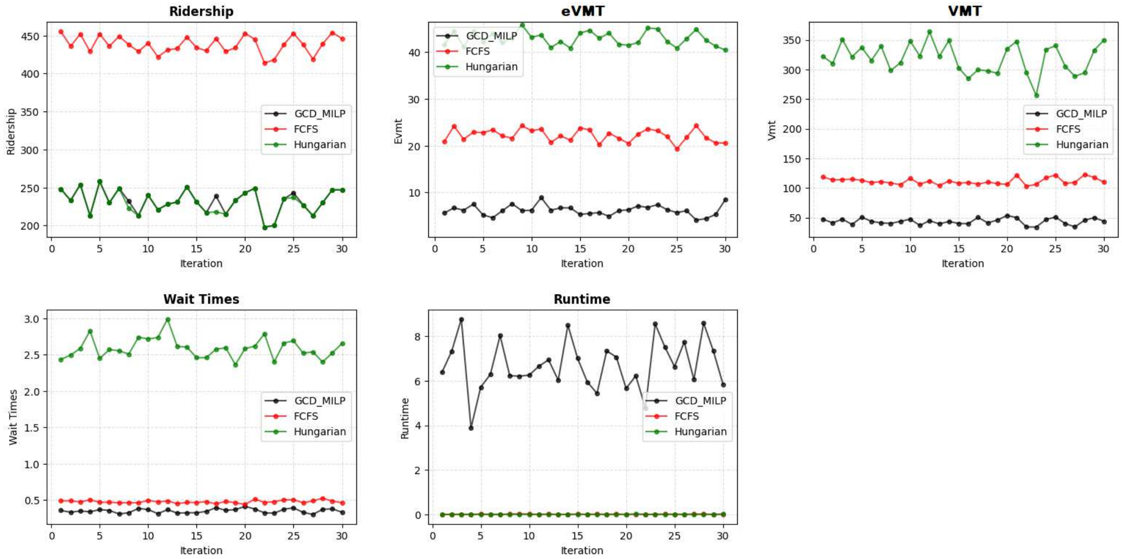

Figure 4 Performance of benchmark algorithms against GCD-MILP

4.2.2 Service Quality and Passenger Wait Times

The GCD MILP model also provided a superior quality of service. The result in Table 3 further indicates that the average wait time for passengers assigned by GCD MILP was 0.35 minutes, which was 27.1% lower than FCFS (0.48 minutes) and 86.5% lower than the Hungarian algorithm (2.59 minutes). All these differences were statistically significant with large effect sizes ($p < 0.001$, Cohen's $d > 5.1$). This result is a direct consequence of the model's objective function (Eq. 6), which explicitly penalizes wait time (through the distance-proportional term weighted by γ), guiding the optimizer towards assignments that minimize passenger waiting.

4.2.3 Computational Performance

As expected, the computational cost of achieving this high-quality solution was the primary trade-off. The GCD MILP model, which solves an NP-hard optimization problem, had a mean runtime of 6.70 seconds. In contrast, the polynomial-time FCFS and Hungarian algorithms resolved assignments almost instantaneously (≈0.01s and ≈0.00s, respectively). This runtime is a function of the problem size and solver settings but remains within a practical limit for real-world deployment where dispatch decisions are made on a rolling basis every 30-60 seconds [38].

4.2.4 Statistical Significance of Results

The result of the statistical analysis in Table 4 confirms that the observed performance differences are not due to random chance. For all primary metrics (eVMT, VMT, and Wait Times), the comparisons between GCD MILP and the two benchmarks yielded p-values of 0.0000, indicating extreme statistical significance. The associated Cohen's d values were all classified as "large" ($|d| > 0.8$), with many exceeding 10, underscoring the substantial practical significance and magnitude of the performance gaps. The only metric without a significant difference was ridership between GCD MILP and the Hungarian algorithm ($t = 0.296$, $p = 0.7686$), confirming they served a statistically similar volume of passengers but through vastly different and more efficient assignments in the case of GCD MILP.

Table 4 Statistical analysis on performance differences

	t-value	p-value	Cohen's d	Effect size
Ridership Analysis				
GCD-MILP vs FCFS	-58.065	0.0000	-14.992	Large
GCD-MILP vs Hungarian	0.296	0.7686	0.076	Small
FCFS vs Hungarian	58.117	0.0000	15.006	Large
EVMT Analysis				
GCD-MILP vs FCFS	-50.328	0.0000	-12.995	Large
GCD-MILP vs Hungarian	-107.235	0.0000	-27.688	Large
FCFS vs Hungarian	-55.957	0.0000	-14.448	Large
VMT Analysis				
GCD-MILP vs FCFS	-48.915	0.0000	-12.630	Large
GCD-MILP vs Hungarian	-58.994	0.0000	-15.232	Large
FCFS vs Hungarian	-44.398	0.0000	-11.463	Large
Wait Times Analysis				
GCD-MILP vs FCFS	-19.814	0.0000	-5.116	Large
GCD-MILP vs Hungarian	-85.740	0.0000	-22.138	Large
FCFS vs Hungarian	-81.784	0.0000	-21.117	Large

5. Discussion of Results

The experimental results demonstrate the significant impact of the proposed GCD-based MILP (GCD MILP) optimization framework on the performance of a ride-sharing assignment system. The comparative analysis against two benchmark algorithms (First-Come-First-Served (FCFS) and the canonical Hungarian algorithm) reveals critical trade-offs between solution quality, operational efficiency, and computational cost.

5.1 Key Performance Trade-offs

The most striking result is the clear multi-objective superiority of the GCD MILP approach. While it achieved a ridership level statistically indistinguishable from the Hungarian algorithm ($t=0.296$, $p=0.7686$, Cohen's $d=0.076$), it did so with dramatically superior operational efficiency. The GCD MILP model reduced empty vehicle miles travelled (eVMT) by 72.2% and total vehicle miles travelled (VMT) by 86.4% compared to the Hungarian method. This is because the Hungarian algorithm, while effective for pure, one-to-one assignment, is fundamentally unsuited for many-to-one problems with capacity constraints [39, 40]. It optimizes for a singular cost metric (here, negative compatibility) without regard for vehicle capacity utilization, leading to highly fragmented and inefficient assignments that maximize compatibility per assignment but disastrously inefficient system-wide travel. In contrast, the GCD MILP model's integrated objective function and explicit capacity constraints (Eq. 8) successfully balance high ridership with minimal operational waste.

Furthermore, the GCD MILP framework significantly outperformed the FCFS heuristic across all metrics except computational runtime. It served 46.9% fewer passengers than FCFS. In line with previous studies, this result while seemingly negative, is actually a hallmark of a more intelligent and sustainable strategy [41]. The FCFS algorithm's higher ridership is an artifact of its naive strategy. It greedily assigns groups to the nearest available taxi until capacity is exhausted, without considering the long-term compatibility or system-wide efficiency of these assignments [42–44]. This leads to severe inefficiencies, as evidenced by its eVMT being 3.6x higher and its VMT being 2.6x higher than the GCD MILP model. The GCD MILP model's willingness to forgo a low-compatibility, high-distance assignment in the short term allows it to construct a globally superior solution that minimizes total system cost. This finding is in line with studies on multi-hop peer-to-peer ride-matching problems [45] and dynamic ridesharing [46].

5.2 The Role of the GCD-Based Compatibility Metric

The core novelty of this work, the GCD-derived compatibility score (θ_{ij}), proved to be a highly effective mechanism for guiding the optimization. By moving beyond a simple binary feasibility check ($g_i \leq c_j$), the θ_{ij} score provided a granular, multi-faceted measure of suitability. The integration of the GCD, Jaccard similarity, and frequency bonus (Eq. 4) successfully prioritized assignments that not only fit but also utilized vehicle capacity optimally. For instance, assigning a passenger group of 6 to a vehicle with capacity 6, yields a perfect GCD of 6, which is preferable to matching the same group to a vehicle with capacity 7 where residual inefficiency occurs. This mathematical formulation directly contributes to the minimization of wasted capacity and by extension, reducing both eVMT and VMT metrics [41, 44].

5.3 Computational Complexity and Practical Applicability

The primary trade-off for the GCD MILP's superior solution quality is computational expense. With a mean runtime of 6.70 seconds, it is orders of magnitude slower than the near-instantaneous FCFS (0.01s) and Hungarian (0.00s) algorithms. This is an expected consequence of solving an NP-hard MILP problem versus executing polynomial-time heuristics. However, this runtime is not prohibitive for practical application. In a real-world ride-sharing context, assignments are typically computed on a rolling basis (e.g., every 30-60 seconds) for a subset of available vehicles and waiting passengers [38]. A runtime of under 10 seconds for a problem of this scale (300 passengers, 150 vehicles) is therefore computationally feasible and would provide dispatchers with a significantly more efficient solution than current heuristic-based approaches.

5.4 Limitations and Model Assumptions

This study has limitations that provide avenues for future research. First, the parameters ($\alpha, \beta, \gamma, \tau, \mu$) were calibrated for a specific urban scenario. Their sensitivity and optimal values under different operational strategies (e.g., cost-minimization vs. service-quality-maximization) warrant further investigation. Also, the wait time calculation, while functional, is a simplified linear proxy for a complex real-world phenomenon influenced by traffic, time of day, and road networks. Integrating a more sophisticated spatiotemporal model would enhance realism.

6. Conclusion

This study presented a novel optimization framework, the GCD-based MILP model, for solving the Passenger Group-Vehicle Assignment (PGVA) problem in ride-sharing systems. The model introduces a multi-faceted compatibility score based on prime factorization and number-theoretic principles to evaluate the suitability between passenger groups and vehicles beyond simple capacity checks.

The experimental results lead to three central conclusions. First, the GCD MILP model demonstrably achieves a near-optimal balance between high ridership and minimal operational costs. It significantly outperforms common benchmarks, reducing empty and total vehicle miles travelled by over 70% and 85%, respectively, compared to the Hungarian algorithm, while avoiding the "myopic" inefficiencies of a first-come-first-served strategy. Second, the GCD-based compatibility score (θ_{ij}) is an effective and novel mechanism for guiding the assignment. It successfully encodes the qualitative notion of a "good fit", leading to more efficient resource utilization and directly contributing to the model's performance. Lastly, while computationally more intensive than simplistic heuristics, the proposed GCD-based MILP model solves problems of realistic scale within a timeframe that is practical for operational deployment in modern ride-sharing platforms.

The findings of this study have practical implications for the design and operation of ride-sharing systems. The proposed GCD-based allocation mechanism enhances passenger-to-vehicle matching in ride-sharing systems. Reductions in eVMT and total vehicle miles travelled (VMT) directly translate into lower fuel consumption, decreased fleet maintenance costs, and reduced greenhouse gas emissions, highlighting its practical value for sustainable transport operations. In addition to these benefits, the model minimizes passenger waiting times, thereby improving service quality and operational effectiveness. These improvements do not only strengthen user retention but also attract new riders, providing ride-sharing platforms with a competitive advantage in dense urban mobility markets.

Another strength of the model is its scalability and rapid computation, which enable large-scale dispatching within seconds. This capability makes it highly suitable for real-time operations, particularly in the context of autonomous and electric vehicle fleets, emphasizing its relevance to future-oriented mobility-on-demand services. The method's compatibility-driven allocation approach also advances urban sustainability by mitigating congestion, reducing resource waste, and improving energy efficiency. Adaptability to shared taxi and paratransit services promote multimodal integration, while the flexible weighting of eVMT, VMT, and waiting time allows decision-makers to align allocation strategies with diverse policy and operational priorities. Collectively, these features position the approach as a versatile tool for both private mobility providers and public authorities, enabling the delivery of efficient, equitable, and sustainable shared mobility solutions.

6.1 Future research directions

While the GCD-based approach demonstrates significant theoretical and computational benefits, its direct impact on real-world metrics such as passenger satisfaction or profitability would require additional experimental validation. Future research could explore how the GCD-based approach can complement buses, trains, and other modes of transportation.

Additionally, the impact of GCD-based allocation on dynamic datasets under varying traffic and demand conditions could be measured.

Declarations

Data Availability Statement

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Consent to Publish

Not applicable

Consent to Participate

Not applicable

Ethical Approval

Not applicable

Competing Interests

The authors declare no competing financial or non-financial interests that could influence the objectivity, interpretation, or presentation of this work. The research was conducted independently of any commercial ride-hailing or transportation service providers.

Authors' contributions

Author A (First Author): Conceptualization, methodology development (GCD-based model), formal analysis, validation of mathematical models, software implementation, computational experiments, data analysis, visualization of results, and manuscript writing. **Author B (Co-author):** Supervision, critical revision of the manuscript, and literature review. All authors reviewed and approved the final manuscript.

Acknowledgements

We thank the members of the Technical Department of Allgemeine Vehicle Services and the Department of Mechanical Engineering, Ho Technical University for their support.

References

1. Chi X (2023) Uber's Influence on The Urban Transportation Ecosystem: Impacts, Challenges and Prospects. Highlights Bus Econ Manag 20:447–453

2. Jiang S, Chen L, Mislove A, Wilson C (2018) On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi. pp 863–872

3. Chang SE, Chang EC, Chen Y (2022) Blockchain meets sharing economy: a case of smart contract enabled ridesharing service. Sustainability 14:13732

4. Li R, Liu Y, Liu X, Nie YM (2024) Allocation problem in cross-platform ride-hail integration. Transp Res Part B Methodol 188:103056

5. Martins L do C, de la Torre R, Corlu CG, et al (2021) Optimizing ride-sharing operations in smart sustainable cities: Challenges and the need for agile algorithms. Comput Ind Eng 153:107080

6. Afandizadeh Zargari S, Shakoori S, Mirzahossein H, Karimi M (2021) Optimizing algorithm for allocating passengers in shared taxis. Int J Transp Eng 9:503–520

7. Cao Y, Wang S, Li J (2021) The optimization model of ride-sharing route for ride hailing considering both system optimization and user fairness. Sustainability 13:902

8. Li M, Zheng N, Wu X, et al (2020) An Efficient Solving Method to Vehicle and Passenger Matching Problem for Sharing Autonomous Vehicle System. J Adv Transp 2020:1–12. https://doi.org/10.1155/2020/3271608

9. Magsino E, Ching GR, Espiritu FM, Go K (2022) Evaluating stable matching methods and ridesharing techniques in optimizing passenger transportation cost and companionship. ISPRS Int J Geo-Inf 11:556

10. Makhdomi AA, Gillani IA (2024) A greedy approach for increased vehicle utilization in ridesharing platforms. Expert Syst Appl 255:124670

11. Seo T, Asakura Y (2021) Multi-objective linear optimization problem for strategic planning of shared autonomous vehicle operation and infrastructure design. IEEE Trans Intell Transp Syst 23:3816–3828

12. Zhang C, Xie J, Wu F, et al (2020) Pricing and allocation algorithm designs in dynamic ridesharing system. Theor Comput Sci 803:94–104

13. Mystakidis A, Koukaras C, Koukaras P, et al (2024) Optimizing Nurse Rostering: A Case Study Using Integer Programming to Enhance Operational Efficiency and Care Quality. In: Healthcare. p 2545

14. Alola AA, Adebayo TS (2023) The potency of resource efficiency and environmental technologies in carbon neutrality target for Finland. J Clean Prod 389:136127

15. Trinks A, Mulder M, Scholtens B (2020) An efficiency perspective on carbon emissions and financial performance. Ecol Econ 175:106632

16. Golpayegani F, Gueriau M, Laharotte P-A, et al (2022) Intelligent Shared Mobility Systems: A Survey on Whole System Design Requirements, Challenges and Future Direction. IEEE Access 10:35302–35320. https://doi.org/10.1109/ACCESS.2022.3162848

17. Narayanan S, Chaniotakis E, Antoniou C (2020) Shared autonomous vehicle services: A comprehensive review. Transp Res Part C Emerg Technol 111:255–293. https://doi.org/10.1016/j.trc.2019.12.008

18. Sumitkumar R, Al-Sumaiti AS (2024) Shared autonomous electric vehicle: Towards social economy of energy and mobility from power-transportation nexus perspective. Renew Sustain Energy Rev 197:114381

19. Dastani Z, Koosha H, Karimi H, Moghaddam AM (2024) User preferences in ride-sharing mathematical models for enhanced matching. Sci Rep 14:27338

20. Yao R, Bekhor S (2024) A ridesharing simulation model that considers dynamic supply-demand interactions. J Intell Transp Syst 28:31–53

21. Tang L, Duan Z, Zhao Y (2019) Toward using social media to support ridesharing services: challenges and opportunities. Transp Plan Technol 42:355–379

22. Feng S, Duan P, Ke J, Yang H (2022) Coordinating ride-sourcing and public transport services with a reinforcement learning approach. Transp Res Part C Emerg Technol 138:103611

23. Tang L, Duan Z, Zhu Y, et al (2019) Recommendation for ridesharing groups through destination prediction on trajectory data. IEEE Trans Intell Transp Syst 22:1320–1333

24. Balcan M-F, Dick T, Sandholm T, Vitercik E (2024) Learning to branch: Generalization guarantees and limits of data-independent discretization. J ACM 71:1–73

25. Trösser F, De Givry S, Katsirelos G (2020) Relaxation-aware heuristics for exact optimization in graphical models. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer, pp 475–491

26. Godin C, Boudon F, Pradal C (2016) Reducing algorithm complexity in trees. In: FSPMA 2016-IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications

27. Arnold DC, Pack GD, Miller BP (2006) Tree-based overlay networks for scalable applications. In: Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. IEEE, p 8 pp.

28. Yao R, Bekhor S (2021) A dynamic tree algorithm for peer-to-peer ridesharing matching. Netw Spat Econ 21:801–837

29. Lai X, Yang Z, Xie J, Liu Y (2024) Reinforcement learning in transportation research: frontiers and future directions. Multimodal Transp 100164

30. Abdelmoumene H, Boussahoul S (2024) Towards Optimized Dynamic Ridesharing System Through Multi-Objective Reinforcement Learning. In: 2024 IEEE International Multi-Conference on Smart Systems & Green Process (IMC-SSGP). IEEE, pp 1–6

31. Qin ZT, Zhu H, Ye J (2022) Reinforcement learning for ridesharing: An extended survey. Transp Res Part C Emerg Technol 144:103852

32. Azagirre X, Balwally A, Candeli G, et al (2024) A better match for drivers and riders: Reinforcement learning at lyft. Inf J Appl Anal 54:71–83

33. Fèvre C, Zgaya-Biau H, Mathieu P, Hammadi S (2024) Preferential optimization of multi-hop dynamic ridesharing based on R-trees and multi-agent systems. Transp Res Rec 2678:47–62

34. Fevre C, Zgaya-Biau H, Mathieu P, Hammadi S (2021) Multi-agent Systems and R-Trees for Dynamic and Optimised Ridesharing. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp 1352–1358

35. Halffmann P, Schäfer LE, Dächert K, et al (2022) Exact algorithms for multiobjective linear optimization problems with integer variables: A state of the art survey. J Multi-Criteria Decis Anal 29:341–363

36. Fagnant DJ, Kockelman KM (2018) Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. Transportation 45:143–158. https://doi.org/10.1007/s11116-016-9729-z

37. Alonso-Mora J, Wallar A, Rus D (2017) Predictive routing for autonomous mobility-on-demand systems with ride-sharing. 2017 IEEERSJ Int. Conf. Intell. Robots Syst. IROS 3583–3590

38. Li C, Parker D, Hao Q (2021) Optimal Online Dispatch for High-Capacity Shared Autonomous Mobility-on-Demand Systems. In: Proc IEEE Int Conf Rob Autom. Institute of Electrical and Electronics Engineers Inc., pp 779–785

39. Kuhn HW (1955) The Hungarian method for the assignment problem. Nav Res Logist Q 2:83–97

40. Chowdhury MSR (2022) An Improved Hungarian Algorithm for a Special Case of Unbalanced Assignment Problems. Glob J Sci Front Res 22:29–35

41. Agatz N (2013) Optimisation for dynamic ride-sharing. RSM Discov-Manag Knowl 15:13–15

42. Alonso-Mora J, Samaranayake S, Wallar A, et al (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc Natl Acad Sci 114:462–467. https://doi.org/10.1073/pnas.1611675114

43. El Hosni H, Farhat N, Nimer R, et al (2012) An optimization-based approach for passenger to shared taxi allocation. In: SoftCOM 2012, 20th International Conference on Software, Telecommunications and Computer Networks. IEEE, pp 1–7

44. Stiglic M, Agatz N, Savelsbergh M, Gradisar M (2018) Enhancing urban mobility: Integrating ride-sharing and public transit. Comput Oper Res 90:12–21

45. Masoud N, Jayakrishnan R (2017) A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem. Transp Res Part B Methodol 99:1–29

46. Nourinejad M, Roorda MJ (2016) Agent based model for dynamic ridesharing. Transp Res Part C Emerg Technol 64:117–132