Otto-von-Guericke University Magdeburg

Faculty of Economics and Management

Type of paper

Seminar Paper

Title

Scheduling Last-Mile Deliveries with Truck-Based Autonomous Robots

Submitted in partial fulfilment of the requirements for the degree of

Master of Science (M.Sc.)

in

Operations Research And Business Analytics

Author: Hamzauddin Siddiqui

Matriculation Number: 249341

Email: hamzauddin.siddiqui@st.ovgu.de

Examiner:

Prof. Dr. Janis S. Neufeld

Chair of Operations Management

Supervisor:

Hendrik Busmann

22 January 2025

Abstract

There have been attempts to solve the problem of delivery in congested urban areas in an innovative manner. One such solution involves the integration of autonomous robots with the traditional truck-based delivery systems. In this idea, a truck is loaded with cargo for multiple customers at a depot and travels to city centers. The truck carries the parcels and the autonomous robots both, and each robot can be loaded with packages for a particular customer. These robots are then deployed and navigate independently to their assigned customer, complete the delivery and then return to the depot on their own, without any need for any other additional human input. The truck, after deploying the robots, carries on to other drop off points to deploy more robots, until they no longer have any more robots in their back, at which point they can return to a depot to reload the robots and repeat the cycle, until all deliveries are completed. This seminar paper evaluates the viability of this innovative solution by developing the algorithms that optimize the truck's route between the depot and the drop off points with the objective of minimizing late deliveries. We present the mathematical model of the problem, analyze the computational complexity, and compare appropriate solution methodologies.

Contents

List of Tables

List of Figures

List of Abbreviations

ADR	Autonomous Delivery Robot
HTP	Hitchcock Transportation Problem
MIP	Mixed-Integer Programming
TBRD	Truck-Based Robot Delivery
TSP	Traveling Salesman Problem
TTRP	Truck and Trailer Routing Problem
VRP	Vehicle Routing Problem

List of Symbols

C	Set of customers
D	Set of drop-off points
D_k	Deadline of customer k
K	The truck's maximum loading capacity for robots
l_j	Continuous: amount of robots on board at the departure from location j
P	Set of all locations ($C \cup D \cup R$)
R	Set of robot depots
$s_{i,j}$	Binary: 1, if location j is the direct successor of location i; 0, otherwise
t_j	Continuous: arrival time at location j
U	Set of customers receiving late deliveries
v_i	Location of ith stop
w_k	Weight of customer k
$x_{j,k}$	Binary: 1, if customer k is supplied from location j; 0, otherwise
γ, γ^e	Initial position, final (dummy) position of the truck
z_k	Binary: 1, if customer k is supplied late; 0, otherwise
δ	Number of robots initially loaded on the truck
$\vartheta^t_{v,v'}$, $\vartheta^r_{v,v'}$	Travel time of truck and robots between two points v, v'
π	Solution (sequence of tuples)
π_0	Set of tuples before first robot depot visit
π_i	Set of tuples after ith robot depot visit
σ_j	Stop number where robot for customer j is launched
ϕ_j	Delivery time of customer j

1. Introduction

The modern transportation landscape has evolved significantly due to environmental concerns, safety issues, and urban congestion. In 2024, congestion cost the U.S. over four billion hours and $74 billion in lost productivity (Fernandez, 2025). This has spurred innovation, with various solutions addressing these challenges. Electric vehicles have reduced emissions but fail to resolve the core issue of congestion. Drone-based delivery systems (Murray & Chu, 2015) offer small-scale alternatives but are hindered in dense urban areas by legal restrictions and limited payload capacity. Autonomous delivery robots, designed to navigate sidewalks and pedestrian zones, have emerged as a promising solution to urban congestion and last-mile delivery inefficiencies.

Based on the proposal by Boysen et al. (2018), this paper examines a truck-based autonomous robot delivery system, where trucks transport robots from centralized depots to urban centers. Robots are deployed to deliver parcels and return to decentralized depots for collection by the truck, enabling it to continue its route efficiently. This approach is especially beneficial for time-sensitive businesses like HelloFresh, where maintaining freshness is critical. By minimizing vehicle idle time and delivery delays, this system addresses key issues in urban logistics.

The study focuses on overcoming last-mile delivery challenges by developing and evaluating scheduling procedures for this system. Mirroring the Boysen et al. (2018) study, its main objective is to create an efficient scheduling algorithm that reduces delivery delays. To achieve this, the study is structured as follows: first, it reviews literature on urban logistics and autonomous delivery systems. It then presents a mathematical formulation of the scheduling problem and develops both exact and heuristic solution methods. Comprehensive computational experiments evaluate the system's performance. Finally, the paper discusses the implications of its findings and identifies future research directions to advance this innovative delivery approach.

2. Literature Review

Urban logistics has experienced rapid change due to technological advancements and shifting consumer expectations. Traditional transportation methods have been thoroughly reviewed (Speranza, 2018; Pelletier et al., 2016; Savelsbergh & Van Woensel, 2016), but autonomous delivery technologies bring unique challenges that demand new theoretical frameworks and practical approaches.

The evolution of last-mile delivery solutions reflects this shift. The Truck and Trailer Routing Problem (TTRP), an extension of the Vehicle Routing Problem (VRP), introduced essential principles of coordinated vehicle operations by addressing scenarios where trucks with trailers serviced some customers, while others required truck-only service (Lin et al., 2009). However, it did not consider the independent operation of autonomous delivery units.

Aerial delivery systems emerged as a more flexible alternative, leveraging autonomous units launched from mobile platforms. These systems showed potential in improving efficiency (Agatz et al., 2018; Murray & Chu, 2015; Wang et al., 2017), but their urban applicability was constrained by payload limits, regulations, and operational challenges in dense environments.

In response, ground-based autonomous delivery robots (ADRs) have gained prominence. They provided reliable, contactless delivery during the COVID-19 pandemic (Chen et al., 2021), emphasizing consistent operation, higher payload capacity, and reliability in diverse urban conditions. Recent innovations, such as smart collection points integrating package handling and energy management, further enhance sustainability (Bachi et al., 2024).

Classical routing problems provide theoretical underpinnings for these systems but also reveal critical gaps. The traveling repairman problem addressed cumulative service time minimization but lacked solutions for coordinating multiple autonomous units (Nucamendi-Guillén et al., 2016; Heilporn et al., 2010; Gouveia & Voss, 1995). The covering salesman problem optimized partial node visitation under coverage constraints but overlooked dynamic scheduling (Current & Schilling, 1989). Similarly, vehicle routing problems with synchronization offered models for coordinating vehicles but assumed fixed vehicle relationships (Drexl, 2012).

Modern research builds on these foundations to tackle new challenges. Studies validate ADRs' effectiveness in urban settings, despite navigation and weather-related issues (Muller et al., 2024), and highlight their benefits, though implementation remains largely theoretical (Alverhed et al., 2024).

This research takes a similar approach to Boysen et al. (2018) approach by assessing the practical viability of truck-based robot delivery (TBRD). By systematically testing its components and simplifying the problem while retaining essential characteristics, this study establishes a baseline proof of concept. This foundation can guide future research into more complex scenarios and theoretical extensions.

3. Problem Definition

The Truck-based Robot Delivery (TBRD) addresses the optimization of last-mile delivery operations using a combined system of trucks and autonomous robots. Consider a single truck operating from an initial position γ with δ robots onboard. While these parameters typically represent the starting depot configuration, they can also be used to denote the truck's current status during a route if real-time replanning becomes necessary due to unforeseen events like unexpected congestion.

Each customer $j \in C$ has a deadline d_j which represents the latest acceptable delivery time, and a weight w_j that defines the penalty for late delivery. The problem operates on a set $P = C \cup D \cup R$ comprising:

- Customer locations (C): Points where customers receive deliveries, accessible exclusively by robots.
- Drop-off points (D): Designated positions where the truck can stop to load and launch robots.
- Robot depots (R): Facilities where the truck can both launch robots and replenish its robot capacity up to K units.

Given the differential in velocities between truck and robots, we define:

- $\vartheta^t_{v,v'}$: Travel time for the truck between any points $v,v' \in P$.
- $\vartheta^r_{v,v'}$: Travel time for robots between any points $v,v' \in P$.

A solution π to TBRD consists of a sequence of tuples (v,L), where:

- $v \in D \cup R \cup \{\gamma\}$ represents a location.
- $L \subseteq C$ defines the subset of customers served by robots launched from location v.

For a solution π to be feasible, it must satisfy:

- Complete Coverage: The union of all customer subsets L in π must equal C, ensuring all customers receive service.
- Unique Service: For any two unique tuples (v,L) and (v',L') in π, $L \cap L' = \emptyset$, guaranteeing each customer is served exactly once.

- Initial Position: The first tuple in π must start with $v = \gamma$, establishing the truck's initial position.
- Initial Capacity: For launches before the first robot depot visit (set π_0), the total number of robots launched cannot exceed δ.
- Ongoing Capacity: After each robot depot visit i (set π_i), the number of robots launched before the next depot visit cannot exceed K.

For a customer $j \in C$, the delivery time φ_j is calculated as the sum of truck travel times to the launch point plus the robot travel time to the customer. More precisely, if $\sigma(j)$ represents the stop number where j's robot is launched, and v_i is the location of the i-th stop: $\varphi_j = \Sigma_{i=1}^{\sigma(j)-1} \vartheta^t_{vi,vi+1} + \vartheta^r_{v\sigma(j),j}$.

Let U be the set of customers receiving late deliveries ($\varphi_j > d_j$). The objective is to minimize the weighted sum of late deliveries: $Z(\pi) = \Sigma_{j \in U} w_j$.

The model operates within a framework defined by several key assumptions to maintain computational tractability while preserving practical relevance. The truck's role is strictly limited to serving as a mobile robot launcher, reflecting real-world efficiency considerations in congested urban areas. At robot depots, unlimited robot availability is assumed - a reasonable approximation for well-managed operations where robot availability aligns with truck schedules. All travel times are considered deterministic to establish baseline performance, though real implementations would incorporate appropriate time buffers. Customer availability is assumed to align with delivery windows, supported by modern scheduling applications, while truck stop times at depots and drop-off points are considered negligible relative to travel durations. These modeling choices, while idealized, allow evaluation of fundamental TBRD viability while maintaining analytical tractability.

3.1. Computational Complexity

The NP-hard complexity of TBRD can be established through a reduction from the Traveling Salesman Problem (TSP). This reduction is constructed by transforming a TSP instance into a TBRD instance in the following manner: the TSP's starting city is designated as the truck's initial position, and for each remaining city, a drop-off point is created at its location with a customer location placed immediately adjacent to it, establishing a zero-distance relationship. The truck is equipped with n-1 robots (where n represents the number of cities), and robot velocity is set to be significantly lower than the truck's speed. All customers are assigned an identical deadline T, corresponding to the TSP's tour length limit. The validity of this transformation is ensured by several key properties: the extremely slow robot speed necessitates that each customer must be served from their adjacent drop-off point, as service from any other location would inevitably result in late delivery. This constraint, combined with the requirement that each customer must receive service, forces the truck to visit each drop-off point exactly once. Consequently, the truck's route becomes equivalent to a TSP tour through the cities, and a feasible TBRD solution exists if and only if there exists a valid TSP tour meeting the length constraint T.

This transformation demonstrates that solving TBRD efficiently would imply an efficient solution to TSP, which is known to be NP-hard (Garey & Johnson, 1979), thus establishing TBRD's computational complexity. This intractability necessitates a two-pronged solution approach: an exact method using mixed-integer programming for smaller instances to establish performance benchmarks, and a heuristic approach capable of handling larger, real-world scenarios efficiently.

3.2. Mixed Integer Programming Model

Table 1: Notation for MIP Model

Symbol	Description
C	Set of customers
D	Set of drop-off points
R	Set of robot depots
d_k	Deadline of customer k
w_k	Weight of customer k
δ	Number of robots initially loaded on the truck
γ, γ^e	Initial position, final (dummy) position of the truck
K	The truck's maximum loading capacity for robots
$\vartheta^t_{v,v'}$, $\vartheta^r_{v,v'}$	Travel time of truck and robots between two points v, v'
l_j	Continuous variables: amount of robots on board at the departure from location j
$s_{i,j}$	Binary variables: 1, if location j is the direct successor of location i; 0, otherwise
t_j	Continuous variables: arrival time at location j
$x_{j,k}$	Binary variables: 1, if customer k is supplied from location j; 0, otherwise
z_k	Binary variables: 1, if customer k is supplied late; 0, otherwise

Using the notation in Table 1 (Boysen et al., 2018), the mixed integer programming formulation of the problem would be structured as follows:

$$\text{Minimize } F(L,S,T,X,Z) = \sum_{k\in C} z_k \cdot w_k \qquad [1]$$

$$\sum_{j\in D\cup R\cup\{\gamma\}} x_{j,k} = 1 \qquad \forall k \in C \qquad [2]$$

$$l_\gamma \leq \delta - \sum_{k\in C} x_{\gamma,k} \qquad [3]$$

$$l_j \leq K + M \cdot (1 - s_{i,j}) - \sum_{k\in C} x_{j,k} \qquad \forall\, i \in R;\, j \in D \qquad [4]$$

$$l_j \leq l_i + M \cdot (1 - s_{i,j}) - \sum_{k\in C} x_{j,k} \qquad \forall\, i \in D \cup \{\gamma\};\, j \in D \setminus \{i\} \qquad [5]$$

$$t_\gamma = 0 \qquad [6]$$

$$t_j \geq t_i - M \cdot (1 - s_{i,j}) + \vartheta^t_{i,j} \qquad \forall\, i \in D \cup R \cup \{\gamma\};\, j \in D \cup R \setminus \{i\} \qquad [7]$$

$$M \cdot z_k \geq t_j - M \cdot (1 - x_{j,k}) + \vartheta^r_{j,k} - d_k \qquad \forall\, j \in D \cup R \cup \{\gamma\};\, k \in C \qquad [8]$$

$$\sum_{j \in D \cup R \cup \{\gamma^e\}} s_{\gamma,j} \leq 1 \qquad [9]$$

$$\sum_{i \in D \cup R \cup \{\gamma^e\} \setminus \{j\}} s_{j,i} = \sum_{i \in D \cup R \cup \{\gamma\} \setminus \{j\}} s_{i,j} \qquad [10]$$

$$\sum_{k \in C} x_{j,k} \leq M \cdot \sum_{i \in D \cup R \cup \{\gamma\} \setminus \{j\}} s_{i,j} \qquad \forall\, j \in D \cup R \qquad [11]$$

$$\sum_{k \in C} x_{j,k} \geq \sum_{i \in D \cup R \cup \{\gamma\} \setminus \{j\}} s_{i,j} \qquad \forall\, j \in D \qquad [12]$$

$$\sum_{k \in C} x_{j,k} + \sum_{i \in D} s_{j,i} \geq \sum_{i \in D \cup R \cup \{\gamma\} \setminus \{j\}} s_{i,j} \qquad \forall\, j \in R \qquad [13]$$

$$s_{i,j} \in \{0, 1\} \qquad \forall\, i \in D \cup R \cup \{\gamma\};\, j \in D \cup R \cup \{\gamma^e\} \setminus \{i\} \qquad [14]$$

$$x_{i,k},\, z_k \in \{0, 1\} \qquad \forall\, i \in D \cup R \cup \{\gamma\};\, k \in C \qquad [15]$$

$$l_j \geq 0 \qquad \forall\, j \in D \cup \{\gamma\} \qquad [16]$$

[1] represents the main goal of the TBRD problem which is to create such a route for the truck, as well as a robot launching schedule that it optimises for minimising the weighted number of late deliveries. [2] shows that for each customer k in set C, exactly one location j (which can be either a drop-off point, robot depot, or the initial location γ) must be selected as the launch point. [3] says initial robot load should not exceed the starting number of robots (δ) minus those deployed at the start position. [4] shows robot loading capacity when moving from a robot depot to a drop-off point, considering the maximum capacity K. [5] is for robot quantity consistency between consecutive drop-off points. Starting time is set to 0 and is tracked in [6] and [7] shows that If we go from point i to point j, we can't arrive at j before we left i plus the time it takes to travel between them. [8] does late delivery detection whereas [9] ensures the truck starts its route from at most one location. For sub-tour elimination, [10] ensures that for every location the truck visits (except start and end), the number of times we enter must equal the number of times we exit. [11] ensures robots are only launched from locations the truck visits and [12] forces drop-off points to be used if visited. [13] is for controlling robot depot visits - must either launch robots or connect to drop-off points. [14] and [15] are binary variables for route segments (s), assignments (x), and late deliveries (z) whereas [16] shows non-negative continuous variables for robot loads (l).

4. Solution Methodology

Table 2: Parameter values for Instance Generation

Symbol	Description	Small	Large
w	Side length of main square	2 km	5 km
$\lvert R \rvert$	Number of robot depots	4	16
$\lvert D \rvert$	Number of drop-off points	6	30
$\lvert C \rvert$	Number of customers	6	40
$[\rho_{min}, \rho_{max}]$	Deadline factor interval (named {tight, wide})	([2, 4]; [3, 5])	([4, 12]; [4, 15])
$[w_{min}, w_{max}]$	Weights interval (named {homo, hetero})		([1, 1]; [1, 3])
K	Truck's robot capacity	2	8
	Speed truck/robot	30 kmph / 5 kmph	

4.1. Instance Generation

To evaluate our solution approaches, we generate two distinct sets of test instances, as proposed by Boysen et al. (2018) as well: a small dataset intended for proving optimality using exact solvers, and a large dataset representing real-world scenarios. The instance generation process begins with layout creation, where we first establish a square area with side length w, divided into a grid with cells measuring 1/6 kilometers. Robot depots are strategically placed in an equidistant pattern across this grid to ensure balanced coverage of the service area. Following depot placement, drop-off points and customer locations are randomly scattered across the grid, with care taken to prevent any overlapping positions. The truck's initial position γ is then randomly selected from either a depot or drop-off point ($\gamma \in D \cup R$), with an initial robot count δ equal to the truck's capacity K.

For distance and travel time calculations, we employ the Euclidean distances to calculate distances between all points. Based on this and given truck and robot speeds, we calculate travel times $\vartheta^t_{v,v'}$ and $\vartheta^r_{v,v'}$ between any two points $v,v' \in P$. This approach ensures the triangle inequality holds for each mode of transportation independently.

Customer deadlines are generated through a systematic process that accounts for minimum possible service times much like Boysen et al. (2018). For each customer $k \in C$, we first calculate the minimum possible travel time $\vartheta_{min}(k)$ as the shortest combination of truck travel to any depot or drop-off point plus robot travel to the customer: $\vartheta_{min}(k) = \min_{j \in D \cup R} \{\vartheta^t_{\gamma,j} + \vartheta^r_{j,k}\}$. Using this baseline, we generate a random number $r_k \in (0,1]$, then set the customer's deadline as $d_k = \vartheta_{min}(k) \cdot (\rho_{min} + (\rho_{max} - \rho_{min}) \cdot r_k)$. Customer weights, which represent the penalty for late delivery, are assigned by randomly selecting values w_k from the interval $[w_{min}, w_{max}]$ using a uniform distribution.

To ensure robust testing, we create various combinations of deadlines and weights using a full factorial approach. This methodology results in comprehensive test sets of 100 small instances and 100 large instances, with all parameters controlled by the values specified in Table 2. This ensures consistent and comparable test scenarios across both datasets.

These generated instances serve as input for evaluating two distinct solution approaches: an exact solution method using the mixed-integer programming model (TBRD-MIP) implemented in Gurobi, and an efficient heuristic procedure. While the small instances are used to assess solution quality by comparing the heuristic results against optimal solutions obtained from Gurobi, the large instances demonstrate the practical applicability of our heuristic approach in real-world scenarios where exact solutions become computationally intractable.

4.2. Gurobi Solution

The implementation of TBRD-MIP in Gurobi requires several essential preprocessing steps. When the truck's starting point γ is also a drop-off point, TBRD-MIP can be used directly. However, if γ is a robot depot, we must first remove all customers $k \in C$ who can be supplied in time from the current depot (where $\vartheta^r_{\gamma,k} \leq d_k$), set $\delta = K$ to ensure the truck is filled to capacity, and enforce that customers are not served from the starting point by setting $\Sigma_{k\in C}\, x_{\gamma,k} = 0$.

The model must also account for multiple visits to locations through location duplication. For drop-off points, $|C|/K + 1$ duplicates are required, based on the scenario where all deliveries are completed from a single point. For robot depots, the maximum number of visits is bounded by $|C|(K+1)/2$, considering the max requirement of $(K+1)/2$ robots at each drop-off point.

To minimize the number of duplicates and improve model efficiency, we remove customers $k \in C$ who cannot be served in time, identified by $\min\{\vartheta^r_{\gamma,k}, \min_{j\in D\cup R}|\vartheta^t_{\gamma,j} + \vartheta^r_{j,k}\} > d_k$. Additionally, for each drop-off point $j \in D$ with a set of customers $\bar{C} \subseteq C$ that cannot be supplied in time ($\vartheta^t_{\gamma,j} + \vartheta^r_{j,k} > d_k$, $k \in \bar{C}$), we create only $|C\backslash\bar{C}|/K + 1$ duplicates.

Despite these preprocessing optimizations, TBRD-MIP is expected to encounter computational limitations with large-scale instances typical of real-world scenarios. This computational challenge is a direct consequence of the problem's NP-hard nature, making the development of efficient heuristic solutions particularly important for practical applications.

3.3. Heuristic Solution

The heuristic solution approach consists of two major components: a multi-start local search procedure that generates and improves truck routes and the Hitchcock Transportation Problem (HTP) for optimizing robot assignments given a fixed truck route.

We employ a multi-start local search procedure combined with an efficient Hitchcock Transportation Problem (HTP) optimization for robot assignments (Boysen et al. 2018). The procedure begins by generating initial solutions using two priority rules: PR1 and PR2. PR1 focuses on maximizing satisfiable customers, where at each step we determine $C^t_{vn} = \{k \in C \backslash \bar{C} \mid t + \vartheta^r_{vn,k} \leq d_k\}$ and move to locations that maximize the number of customers that can be served in time. PR2 operates based on urgency, first assigning each customer to their nearest facility ($\tilde{\imath}_k = \arg\min_{i \in D \cup R}\{\vartheta^r_{i,k}\}$) and then prioritizing locations based on the smallest positive timeframe needed to reach designated customers.

These initial solutions may be infeasible as they ignore capacity constraints - either no depot is visited, or too many customers are allocated to the same subtour. To address this, we employ a two-step improvement process. First, we attempt to improve the initial solutions π_{PR1} and π_{PR2} by applying the HTP optimization. Then we generate solution pools PR1 and PR2 through strategic depot insertions at positions where $v_{n-1}, v_n \in D$, considering only the depot that causes minimal additional distance at each potential insertion point. These pools are evaluated in parallel with $(\kappa_{rs})/2$ iterations per pool (chosen to balance solution quality with computational efficiency), using a local search procedure with four neighborhood structures: random position removal (preserving depot feasibility), random location insertion, position swap, and strategic depot insertion before drop-off points - each selected to address specific routing challenges while maintaining solution feasibility.

For each truck route generated during this process, we optimize robot assignments by transforming the problem into a Hitchcock Transportation Problem (HTP). This transformation essentially creates a matching system between robot launch points (suppliers) and deliveries (customers). Given a static truck route v where v_i denotes the i-th stop, we define two types of suppliers: robot depots R(v) that can serve any number of customers (supply $a_i = |C|$), and sequences of drop-off points $S(v) = S_k = v_{k1},\ldots,v_{ki}k$ that are limited by the truck's robot capacity (supply $a_k = K$, or δ for the first subtour if $\gamma \in R$). On the demand side, we have each customer requiring exactly one delivery (demand $b_j = 1$), plus a dummy customer that acts as a balancing mechanism for unused capacity (demand $b_0 = \Sigma_{i=1}^{|R(v)|+|S(v)|} a_i - |C|$).

The effectiveness of each potential match is determined by its "transportation cost": zero if delivery from that location would be on time ($\Sigma_{l=1}^{r(i)-1} \vartheta^t_{vl,vl+1} + \vartheta^r_{vr(i),j} \le d_j$ for depot-customer pairs, or $\min_{i=k1,\dots,kik}\{\Sigma_{l=1}^{i-1} \vartheta^t_{vl,vl+1} + \vartheta^r_{vi,j}\} \le d_j$ for subtour-customer pairs), the customer's priority weight w_j if delivery would be late, and zero for any assignment to the dummy customer. This transformation is particularly valuable because it converts our complex assignment problem into a well-studied optimization problem that can be solved efficiently in polynomial time, allowing us to quickly determine optimal robot assignments as we evaluate different truck routes during our search process.

The overall procedure continues iteratively, with each generated solution v evaluated using HTP(v), and improvements accepted as new starting points until reaching the specified iteration limits κ_{rs} and $(\kappa_{rs})/2$.

5. Results

Our computational experiments were conducted to evaluate the performance of both solution approaches: the mixed-integer programming model (TBRD-MIP) solved using Gurobi, and our heuristic solution procedure. All computations were executed on a PC with an Intel Core processor (8 x 2.8 gigahertz), running Windows 11, with implementations in Python and using Gurobi version 12.0.

Table 3: Performance of Small Dataset

		Gurobi		Heuristic	
Deadlines	Weights	Opt	Sec	Opt	Sec
Tight	Homo	7	0.11	25	0.45
	Hetero	7	0.09	25	0.36
Wide	Homo	18	0.11	24	0.13
	Hetero	18	0.12	24	0.14
Total/Average		50	0.1075	98	0.27

Table 4: Performance of Large Dataset

		Gurobi		Heuristic	
Deadlines	Weights	Opt	Sec	Opt	Sec
Tight	Homo	12	4.7	6	1.21
	Hetero	13	5.44	8	1.26
Wide	Homo	15	8.06	9	0.3
	Hetero	13	7.82	11	0.33
Total/Average		53	6.505	34	0.775

For the small dataset, our results demonstrate interesting performance characteristics of both approaches. Gurobi found optimal solutions in approximately 50% of test cases with remarkably fast computation times, averaging just 0.11 seconds per instance. Our heuristic procedure showed exceptional performance on this dataset, finding optimal solutions in about 98% of test cases while maintaining reasonable computational efficiency at 0.27 seconds per instance.

For the large dataset, Gurobi found optimal solutions in 53% of cases, requiring an average of 6.5 seconds, while the heuristic achieved 34% with a faster 0.775-second average. This indicates the heuristic's suitability for larger, complex instances that mimic real-world scenarios. Gurobi's stronger performance on larger datasets, though counterintuitive, stems from effective preprocessing and tighter bounds, albeit at higher computational costs.

Non-optimal results highlight key challenges. Instances with tightly clustered deadlines often became infeasible, especially with high robot capacity utilization. Widely dispersed delivery points requiring complex routing frequently exceeded time limits. These patterns suggest that real-world systems must balance delivery deadlines and depot placement to ensure efficiency. While the heuristic is more efficient for large-scale instances, its performance relies heavily on the spatial and temporal distribution of delivery requirements.

While our results align with the general direction of findings from Boysen et al. (2018), we observed two notable differences, despite very similar solution methodologies. First, Gurobi showed better performance in finding optimal solutions for the large dataset compared to the small dataset, contrary to what might be expected. Second, our average runtimes were significantly lower across all 16 test combinations (with each combination running 25 times). These differences might be attributed to several factors:

- Hardware differences (our 8 x 2.8 gigahertz setup versus their 4 x 4.0 gigahertz)
- Different Gurobi versions (12.0 versus 7.0.2)
- Implementation language differences (Python versus C#)

Overall, these results suggest that our heuristic approach provides a practical and efficient solution method for the TBRD problem, particularly for larger instances where exact methods may become computationally prohibitive.

6. Sensitivity Analysis

Given that our computational results closely align with those of Boysen et al. (2018), particularly in terms of solution quality and runtime patterns, we can draw insights from their comprehensive sensitivity analysis to understand practical implications of TBRD systems. Their findings provide valuable complementary evidence for parameter impacts in real-world implementations, though future work would benefit from independent validation in specific operational contexts. Their study investigated four critical parameters on a 4 km² grid.

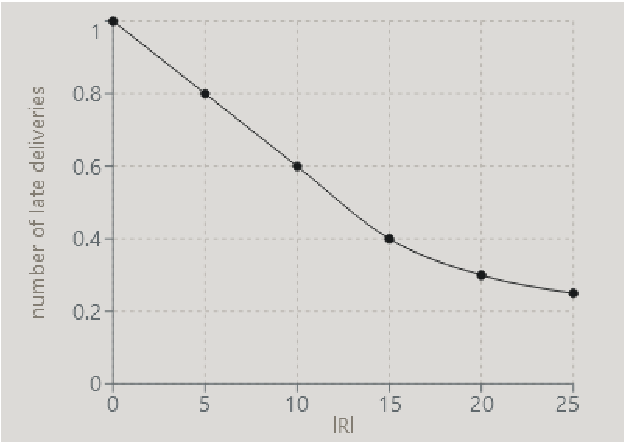

Figure 1: Impact of Network Density of Depots

Boysen et al. (2018) first analyzed depot network density. Their findings demonstrated that while increasing depot density improved delivery performance, the marginal benefits diminished with each additional depot. Their experiments showed that in the 4 km² test area, adding more than 5 depots provided negligible performance improvements as shown in figure 1. This is particularly relevant since trucks must periodically visit robot depots for replenishment when their onboard robot count falls below required levels.

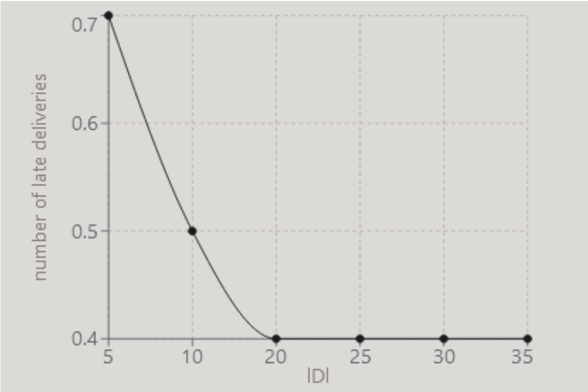

Figure 2: Impact of Drop-Off Points

Their second parameter focused on drop-off point density, a critical factor given that real-world implementations would require designated safe zones for robot loading and launching operations. Their analysis, as shown in figure 2, revealed that, similar to depot density, increasing the number of drop-off points showed diminishing marginal returns, with performance improvements plateauing beyond 1.5 drop-off points per km².

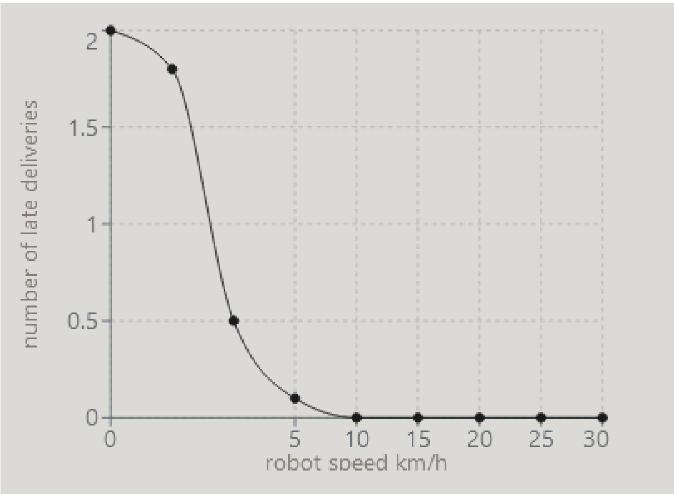

Figure 3: Impact of Robot Speed

In examining robot velocity, their study used 5 km/h as the base configuration and explored variations within safety constraints. Their results showed that a modest increase from 5 to 6 km/h yielded a dramatic 75% improvement in delivery performance, as shown in figure 3, though any further speed increases produced minimal additional benefits.

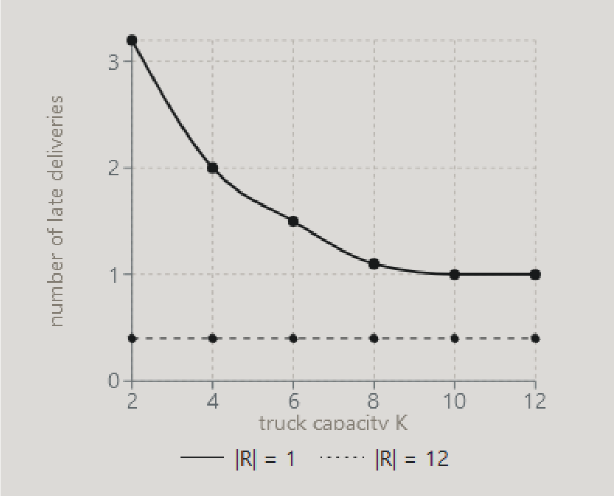

Figure 4: Impact of Truck Capacity

Finally, their examination of truck capacity K revealed interesting interactions with depot density, also shown in figure 4. Their analysis showed that in scenarios with high depot density ($\rho_d > 0.75$ depots/km²), variations in truck capacity had minimal impact on performance. However, with sparse depot networks ($\rho_d \approx 0.06$ depots/km²), truck capacity became a critical factor, though showing diminishing returns beyond K = 5 robots. This finding highlights the importance of balanced system design, as infrastructure decisions directly influence optimal vehicle specifications.

7. Conclusion

This study explored a novel last-mile delivery system using autonomous robots launched from trucks, focusing on scheduling truck routes and robot launches to minimize delays. A mathematical model was developed, and its computational complexity was analyzed alongside an efficient heuristic solution.

Key findings showed that the heuristic achieved optimal solutions in 98% of small instances while remaining computationally efficient. For larger, more realistic instances, the optimal rate dropped to 34%, but computation times were significantly faster than exact methods. Sensitivity analysis highlighted diminishing returns for depot density beyond five depots in a 4 km^2 area and revealed a critical robot speed threshold, where increasing speed from 5 to 6 km/h improved delivery performance by 75%.

Future research should address three areas: dynamic strategies for robot availability during peak periods, the influence of urban infrastructure on depot placement and routing, and the integration of truck-based robot delivery (TBRD) with traditional logistics systems during transitional phases.

While TBRD shows theoretical promise, practical implementation will require careful navigation of infrastructure, regulatory, and operational challenges. Encouraging results from the heuristic suggest TBRD can effectively tackle urban congestion and delivery delays, especially in dense cities where traditional methods are increasingly strained.

Appendix A: Link to the Python Model

Description	Link
Gurobi	https://drive.google.com/file/d/1s8bv_DlrCi8i2wq3L93NAFBdWvBHU816/view?usp=drive_link
Heuristic	https://drive.google.com/file/d/11ZnGQn9yjbvyXzH10HLWQHMS4C_LdcYN/view?usp=drive_link

References

Alverhed, Elin & Hellgren, Simon & Isaksson, Hanna & Olsson, Lisa & Palmqvist, Hanna & Flodén, Jonas. (2024). Autonomous last-mile delivery robots: a literature review. European Transport Research Review. 16. 10.1186/s12544-023-00629-7.

Agatz, N. , Bouman, P. , & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone . Transportation Science.

Bachi, V., Graffione, S., Pasquale, C., & Siri, S. (2024). A planning algorithm for dispatching and smart charging operations of autonomous delivery robots in pedestrian areas. 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), 633–638. https://doi.org/10.1109/case59546.2024.10711444

Boysen, N., 1, Schwerdfeger, S., & Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. In European Journal of Operational Research, European Journal of Operational Research (Vol. 271, pp. 1085–1099). https://doi.org/10.1016/j.ejor.2018.05.058

Chen, C., Demir, E., Huang, Y., & Qiu, R. (2021). The adoption of self-driving delivery robots in last mile logistics. Transportation research part E: logistics and transportation review, 146, 102214

Current, J. R., & Schilling, D. A. (1989). The covering salesman problem. Transportation science, 23(3), 208-213.

Drexl, M. (2012). Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. Transportation Science, 46(3), 297-316.

Fernandez, C. (2025, January 7). U.S. drivers lost 43 hours—a full work week—to traffic in 2024: “If you’re driving in New York, it’s brutal,” expert says. CNBC. https://www.cnbc.com/2025/01/07/inrix-most-congested-us-cities.html

Garey, M. R., & Johnson, D. S. (1979). Computers and intractability (Vol. 174). San Francisco: freeman.

Gouveia, L., & Voß, S. (1995). A classification of formulations for the (time-dependent) traveling salesman problem. European Journal of Operational Research, 83(1), 69-82.

Heilporn, G., Cordeau, J. F., & Laporte, G. (2010). The delivery man problem with time windows. Discrete Optimization, 7(4), 269-282.

Lin, S.-W., Yu, V. F., & Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. Computers & Operations Research, 36(5), 1683–1692. https://doi.org/10.1016/j.cor.2008.04.005

Müller, L., Pang, H., Ohnemus, L., Winter, M., & Furmans, K. (2024). Design and Real-World application of a flexible mobile robot system for urban logistics. 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), 1–8. https://doi.org/10.1109/etfa61755.2024.10710927

Murray, C. C., & Chu, A. G. (2015). The flying sidekick travelling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies, 54, 86–109. https://doi.org/10.1016/j.trc.2015.03.005

Pelletier, S., Jabali, O., & Laporte, G. (2016). 50th Anniversary Invited Article—Goods Distribution with Electric Vehicles: Review and Research Perspectives. Transportation Science, 50(1), 3–22. https://doi.org/10.1287/trsc.2015.0646

Savelsbergh, M., & Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. Transportation science, 50(2), 579-590.

Nucamendi-Guillén, S., Martínez-Salazar, I., Angel-Bello, F., & Moreno-Vega, J. M. (2016). A mixed integer formulation and an efficient metaheuristic procedure for the k-travelling repairmen problem. Journal of the Operational Research Society, 67(8), 1121-1134.

Speranza, M. G. (2018). Trends in transportation and logistics. European journal of operational research, 264(3), 830-836.

Wang, X. , Poikonen, S. , & Golden, B. (2017). The vehicle routing problem with drones: Several worst-case results. Optimization Letters, 11 , 679–697 .

Declaration of Authenticity

I, Hamzauddin Siddiqui, declare

- that I completed the present Thesis independently and without any unauthorized assistance
- that I did not use any aids other than those explicitly approved in advance and specified by me
- that that I have identified parts of the work whose wording or meaning has been taken from other documents (this also includes internet sources and AI-based tools), stating the source.
- AI tools have been used merely to the extent of making the tone of the text formal and academic and correcting potential grammatical mistakes and
- that I have not yet submitted this work in the same or similar form to any other examination authority.

I agree,

- that my work can be checked for plagiarism using software. I am aware that plagiarism constitutes serious academic misconduct, and that deceit will be punished in accordance with the study and examination regulations that apply to me.

I am aware,

- that in case of deceit the work will be officially declared “insufficient (5.0)”
- that I bear full responsibility for incorrect or distorted content generated by AI, incorrect references, violations of data protection and copyright law, or plagiarism.

My intellectual performance was consistently at the forefront, and I always managed the process myself.

Hamzauddin Siddiqui/ Magdeburg, 21 Jan 2025

(Signature) / (Place, Date)