smart cities

MDPI

Article

Smart Delivery Assignment through Machine Learning and the Hungarian Algorithm

Juan Pablo Vásconez [1,*,†], Elias Schotborgh [1,†], Ingrid Nicole Vásconez [2], Viviana Moya [3], Andrea Pilco [3], Oswaldo Menéndez [4], Robert Guamán-Rivera [5] and Leonardo Guevara [6]

[1] Faculty of Engineering, Universidad Andres Bello, Santiago 7500735, Chile; e.schotborghgonzlez@uandresbello.edu
[2] Centro de Biotecnología Dr. Daniel Alkalay Lowitt, Universidad Técnica Federico Santa María, Valparaíso 2390136, Chile; ingrid.vasconez@sansano.usm.cl
[3] Facultad de Ciencias Técnicas, Universidad Internacional Del Ecuador UIDE, Quito 170411, Ecuador; vimoyago@uide.edu.ec (V.M.); anpilcoat@uide.edu.ec (A.P.)
[4] Departamento de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Antofagasta 1249004, Chile; oswaldo.menendez@ucn.cl
[5] Institute of Engineering Sciences, Universidad de O'Higgins, Rancagua 2820000, Chile; robert.guaman@uoh.cl
[6] Lincoln Institute for Agri-Food Technology, University of Lincoln, Lincoln LN2 2LG, UK; lguevara@lincoln.ac.uk
* Correspondence: juan.vasconez@unab.cl
† Current address: Faculty of Engineering, Universidad Andres Bello, Santiago 7500735, Chile.

Abstract: Intelligent transportation and advanced mobility techniques focus on helping operators to efficiently manage navigation tasks in smart cities, enhancing cost efficiency, increasing security, and reducing costs. Although this field has seen significant advances in developing large-scale monitoring of smart cities, several challenges persist concerning the practical assignment of delivery personnel to customer orders. To address this issue, we propose an architecture to optimize the task assignment problem for delivery personnel. We propose the use of different cost functions obtained with deterministic and machine learning techniques. In particular, we compared the performance of linear and polynomial regression methods to construct different cost functions represented by matrices with orders and delivery people information. Then, we applied the Hungarian optimization algorithm to solve the assignment problem, which optimally assigns delivery personnel and orders. The results demonstrate that when used to estimate distance information, linear regression can reduce estimation errors by up to 568.52 km (1.51%) for our dataset compared to other methods. In contrast, polynomial regression proves effective in constructing a superior cost function based on time information, reducing estimation errors by up to 17,143.41 min (11.59%) compared to alternative methods. The proposed approach aims to enhance delivery personnel allocation within the delivery sector, thereby optimizing the efficiency of this process.

Keywords: smart delivery; machine learning; regression model; Hungarian optimization algorithm

check for updates

Citation: Vásconez, J.P.; Schotborgh, E.; Vásconez, I.N.; Moya, V.; Pilco, A.; Menéndez, O.; Guamán-Rivera, R.; Guevara, L. Smart Delivery Assignment through Machine Learning and the Hungarian Algorithm. *Smart Cities* **2024**, *7*, 1109–1125. https://doi.org/10.3390/smartcities7030047

Academic Editor: Pierluigi Siano

Received: 23 March 2024
Revised: 7 May 2024
Accepted: 9 May 2024
Published: 12 May 2024

1. Introduction

In the rapidly evolving landscape of smart transportation, e-commerce, and logistics, the efficient management of delivery operations remains an indispensable determinant of success for delivery businesses [1]. However, within this realm the convergence of smart transportation and mobility further complicates the equation. Smart delivery processes are confronted with a multitude of challenges, from the need to gather the appropriate data from users on location, traffic, and mobility through sensors and the Internet of Things (IoT) [2], to the complex task of managing inventory levels efficiently. Smart delivery tries to find an appropriate method to engage users in sharing this information in order to provide good solutions for large realistic instances along with a procedure that can be used in real environments [3], aiming to meet increasingly demanding customer expectations for fast and reliable service. However, optimizing delivery routes presents a formidable challenge, necessitating consideration of dynamic variables such as traffic patterns and delivery windows. Moreover, the seamless facilitation of communication and tracking throughout the delivery journey is essential for ensuring a smooth and transparent customer experience [4,5]. These multifaceted challenges collectively underscore the pressing need for innovative logistical solutions and strategies that harness the potential of smart transportation and mobility technologies to maintain competitiveness in the ever-evolving landscape of smart e-commerce and logistics.

In this context, using Artificial Intelligence (AI) techniques can present revolutionary opportunities to improve traditional approaches related to delivery, logistics, and supply chain management [6]. In particular, AI algorithms based on machine learning (ML), deep learning (DL), and reinforcement learning (RL) offer significant potential to enhance delivery processes and logistics. For example, ML algorithms can analyze vast amounts of historical delivery data to identify patterns, optimize routing, and predict demand more accurately, thereby improving the efficiency of delivery operations [6,7]. Moreover, DL models work well at processing complex data types such as images and text, enabling tasks such as automatic package sorting, vehicle recognition, and natural language processing for customer inquiries [8]. On the other hand, RL algorithms can optimize decision-making in dynamic environments by learning from interactions with the delivery environment, leading to more adaptive and responsive delivery strategies [9]. These AI techniques represent a possible solution to address delivery logistics challenges.

Optimization methods also play a crucial role in improving delivery problems and logistics by efficiently allocating resources, optimizing routes, and minimizing costs [10]. Techniques such as linear programming, integer programming, and metaheuristic algorithms enable businesses to address logistics and delivery challenges such as route optimization and inventory management [11]. By mathematically modeling delivery constraints and objectives, optimization methods can help businesses to make informed decisions, maximize resource utilization, and improve overall operational efficiency. These methods also facilitate dynamic adjustments to changing conditions, ensuring adaptive and responsive delivery strategies. Ultimately, optimization methods contribute to enhancing customer satisfaction, reducing delivery times, and achieving cost savings in logistics operations [12]. Before exploring the solution proposed in this work, it is essential to summarize some recent studies that have confronted the delivery problem and the application of artificial intelligence in the area. The following subsection presents the state of the art regarding this research.

1.1. Related Work

Over the past few years, task assignment approaches related to delivery problems have gained significant attention from various researchers. For example, in [13], an emulated food delivery service was modeled using a Markov decision process. This model was improved with Q-learning and DDQN using a rule-based policy to maximize revenue derived from served. The results showed that DDQN collects higher rewards compared to other algorithms. In another study presented in [14], the solution to the scheduling problem was addressed using real crowdsourced delivery platforms. The study proposed a machine learning method combining simulation optimization for offline training with a neural network. The obtained results presented a quality within 0.2–1.9% of a bespoke sample average approximation method while being several orders of magnitude faster regarding online solution generation. Furthermore, in [15] the authors presented an algorithm to optimize food delivery processes by minimizing delivery time in road networks. Their approach formulated the order assignment problem as a minimum-weight perfect matching task on a bipartite graph. They then employed the best-first search graph method to calculate the solution space efficiently. The authors used real data from Swiggy, the largest food-delivery company in India. This approach introduced novel concepts such as order batching and dynamic adaptation to vehicle locations to enhance the solution quality, and demonstrated its effectiveness in achieving a 30% reduction in delivery time. Other works have emphasized the importance of minimizing the delivery time for customer satisfaction [16]. The challenge in this respect involves order-to-vehicle assignment, order batching, and vehicle movement adaptation. To address this, the authors proposed mapping the vehicle assignment problem to minimum-weight perfect matching on a bipartite graph. To optimize efficiency, best-first search was utilized to construct a subgraph likely to contain the minimum matching. Using the angular distance, this algorithm enhances solution quality by considering graph batching and dynamic vehicle positions. Extensive experiments on real food-delivery data from metropolitan cities demonstrated that this approach presents substantial improvements over other strategies, including reductions in food delivery time and waiting time at restaurants as well as increased orders delivered per kilometer. In the work presented in [17], the authors addressed the challenge of online food delivery, focusing on efficiently allocating orders to drivers and route planning. For order assignment, a modified Kuhn–Hungarian (Munkres) algorithm is employed to optimize matching between orders and drivers, while a machine learning classification model using eXtreme Gradient Boosting (XGBoost) predicts order batching results to prevent inappropriate matches. Additionally, a rule-based route planning method generates viable routes for drivers. Experiments conducted on real datasets from Meituan demonstrated the effectiveness of the proposed algorithm in solving the online food delivery problem, validating the performance of the classification model and the overall efficiency. In addition, tp address the on-time performance in last-mile delivery services, the study of [18] designed a data-driven framework to model delivery performance and optimize order assignments to drivers. The driver's total delivery time was decomposed into uncertain service time at customer locations and predictable travel time on roads. A prediction model for delivery tour length was then developed, providing practical routing behaviors and accurate predictions; the results showed the advantages of data-driven order assignment models integrated with delivery tour prediction over classical vehicle routing problems.

Other studies have proposed solutions for areas outside the food delivery industry or simple courier management while applying the same principles and methods. For example, in [19], a two-echelon vehicle system was proposed for routing emergency mask deliveries during the COVID-19 pandemic. A hybrid machine learning and heuristic optimization method was proposed to address the delivery of medical masks. Deep learning combined with heuristic optimization was used to predict regional delivery demand, obtaining a reduced average weighted delivery time of up to 61%. In [20], the authors discussed an Autonomous Mobility-on-Demand system, considered an eco-friendly urban transportation service. It addresses the optimization of recharging, delivery, and repositioning tasks for shared autonomous electric vehicles through a multi-agent multi-task dynamic dispatching problem based on the Markov Decision Process. The decision-making process is divided into three subprocesses, each of which transformed into a mathematical problem; recharging and delivery task assignments are modeled as maximum weight matching problems of bipartite graphs, while repositioning task assignment is quantified as a maximum flow problem. Algorithms such as Kuhn–Hungarian and Edmond–Karp are employed to solve these problems and achieve optimal task allocation policies. In [21], a reward function balancing order income with trip satisfaction was proposed along with a state-value function estimated by a backpropagation deep neural network to assess the matching degree between vehicles and tasks. The results demonstrated significant improvements in total revenue, user waiting time reduction, and trip satisfaction through various optimization strategies, such as introducing task allocation repositioning and combining charging with task repositioning.

Tables A1–A3 in Appendix A summarize the bibliographic review showing the addressed problem, the authors' proposed solution, the disadvantages encountered, and the obtained results. Task assignment problems represent a broad field of research that has seen several developments with sundry results. In this context, the most common disadvantages are the number of variables needed to obtain proper results and the applicability and the scalability of the solution in environments outside of those analyzed during development. Due to the complexity and the number of factors present in urban environments, further investigation of possible solutions focusing on new methods of analysis and considering different sets of variables remains of interest, and has motivated this research.

1.2. Main Contribution

The motivation of this work was to use ML algorithms and the Hungarian optimization algorithm in a single architecture to find a feasible method to improve the assignment tasks between delivery people and orders. The synergy between ML and the Hungarian algorithm enabled us to improve delivery operations, which might have a positive impact on operational costs. The main contributions of this work can be summarized as follows:

- We build a dataset from the MAX Delivery company, headquartered in Barcelona, Spain. The dataset comprises 7707 order records. Each record contains details regarding the time and coordinates of delivery personnel assigned to specific customer orders.
- We use the Haversine formula to accurately compute the distances between delivery people and customer orders. This is essential for generating an assignment matrix to solve the optimization problem related to order allocation.
- We propose two different supervised machine learning methods to estimate delivery time and distance to each customer for each delivery person. This is crucial, as the dataset only contains specific data points for completed deliveries and creating the assignment matrix requires calculating potential delivery times for all delivery person–customer combinations.
- We use the Hungarian algorithm with the cost matrices obtained from the Haversine calculations, as well as the linear and polynomial regression methods. The Hungarian optimization algorithm solves the task assignment problem, which optimally assigns delivery people to customer orders. The algorithm efficiently determines the best possible assignments by considering the costs associated with each assignment and guarantees an optimal solution for this task.
- Finally, we compare the effectiveness of the Haversine calculations, linear regression, and polynomial regression techniques after applying the Hungarian optimization algorithm to solve the task assignment problem.

1.3. Outline

The remainder of this work is organized as follows: Section 2 introduces the materials and methods used in the study; Section 3 provides a detailed account of the experimental results; Section 4 addresses the analysis of the obtained results; finally, Section 5 offers concluding remarks and insights drawn from the research.

2. Materials and Methods

In this work, we present a smart optimization architecture for the assignment of delivery tasks, as shown in Figure 1. The proposed approach is based on ML algorithms and optimization methods to improve the delivery assignment task. To this end, a huge dataset associated with the MAX Delivery delivery company operating in Barcelona, Spain was constructed. Then, a cost function is proposed by analyzing different heuristics and ML algorithms. Two mathematical models, namely, linear regression and polynomial regression, are studied to model distance and time metrics. After estimating the cost functions, we solve the task assignment delivery problem using the Hungarian algorithm. Finally, we compare, evaluate, and present the task assignment delivery results.

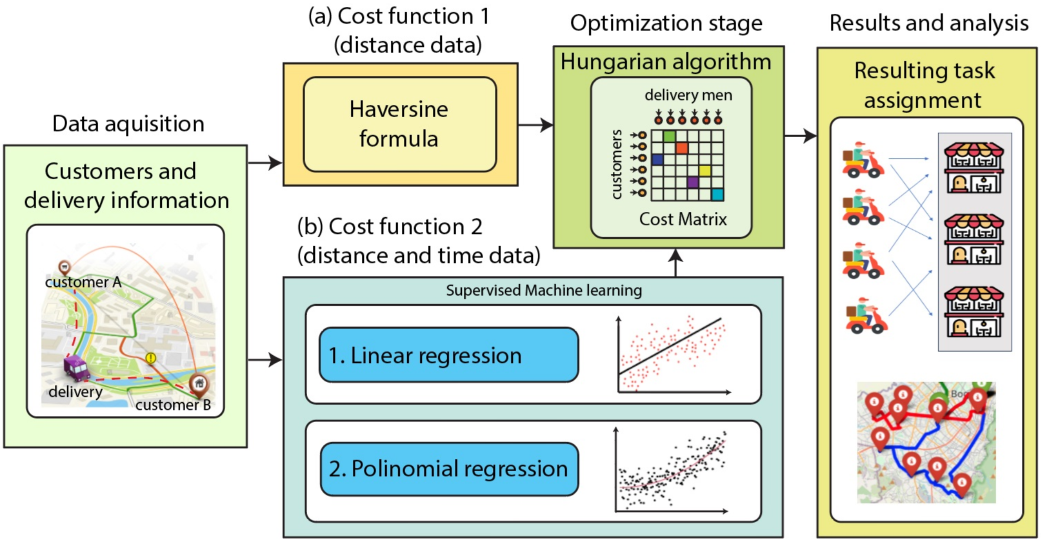


Figure 1. Proposed architecture for enhancing delivery assignments through supervised machine Learning regression techniques and the Hungarian algorithm.

2.1. Data Acquisition

For our data acquisition procedure, we used Python programming language to obtain and process information from MAX Delivery, a company headquartered in Barcelona, Spain. The dataset encompasses 7707 order records, each detailing the time and coordinates of delivery personnel assigned to specific customer orders. This includes information on both the delivery person and customer coordinates, along with the time taken for each delivery. The data analyzed in this study were collected between 1 January 2023 and 22 June 2023, enabling us to compile a dataset comprising 7707 successful delivery orders. To safeguard confidentiality, all consumer and delivery person names and specific details were anonymized.

It is important to note that the dataset provides information solely on completed deliveries and associated customer orders. However, to formulate a task assignment problem, we require a cost function represented by an assignment matrix. This matrix contains the potential cost for each delivery person to deliver to each customer. To construct this assignment matrix, we propose utilizing two distinct methods: the Haversine formula and supervised regression techniques. When employing the Haversine formula, we leverage the coordinates of each delivery person and customer to precisely compute the distance between their locations and thus complete the assignment matrix. It should be noted that while the Haversine formula accurately calculates the distance between points, determining the time is more complex due to the absence of speed, trajectory, and traffic information. Consequently, any time calculations are estimations derived from known information in our dataset using regression methods. Specifically, we utilized linear and polynomial regression to estimate the missing data in the assignment matrices of distance and time, respectively.

The objective of creating the proposed assignment matrices, whether using calculated or estimated data, is to optimally assign the nearest delivery person to each customer, thereby ensuring efficient service delivery, as elaborated in the subsequent sections.

2.2. Haversine Formula for Estimating Cost Matrix

The main objective of this process is to estimate the distance between the delivery people and the customer orders and to generate an assignment matrix to solve the optimization problem related to the task assignment problem. To accomplish this, we calculated the distances between the delivery people and the customer orders using the Haversine formula [22–24]. This formula utilizes the latitude and longitude information of each member of the delivery people's fleet and the current batch of customers in order to calculate the distance between each of them. The Haversine formula is based on spherical trigonometry and takes into account the curvature of the Earth. It involves several steps, such as applying the Haversine formula (using radians), which computes the central angle (θ) between the points using the differences in latitude and longitude, as indicated in Equation (1).

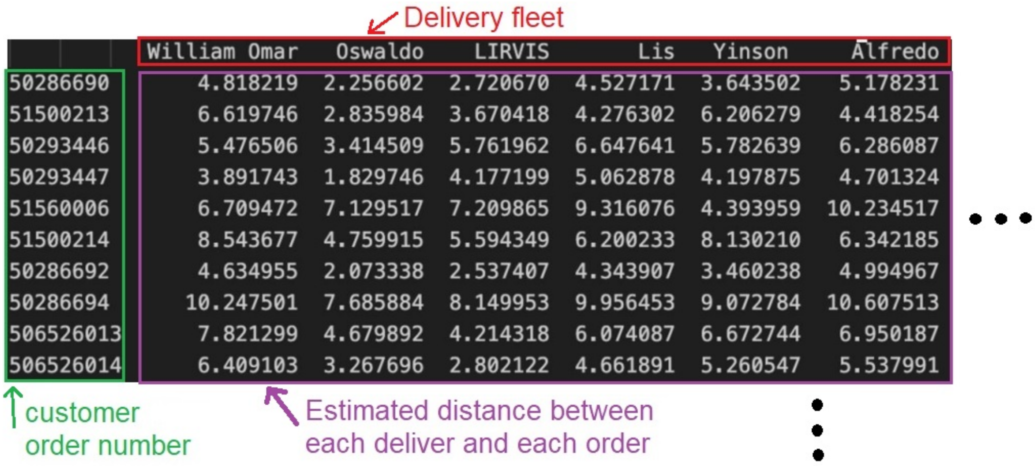
$$\mathrm{hav}(\theta) = \sin^2\left(\frac{\Delta\mathrm{lat}}{2}\right) + \cos(\mathrm{lat}_1) \cdot \cos(\mathrm{lat}_2) \cdot \sin^2\left(\frac{\Delta\mathrm{lon}}{2}\right) \tag{1}$$

The distance between the two points along the surface of the Earth is computed as

$$d = 2 \cdot R \cdot \mathrm{atan2}\left(\sqrt{\mathrm{hav}(\theta)}, \sqrt{1 - \mathrm{hav}(\theta)}\right), \tag{2}$$

where R is the Earth's radius, approximately 6371 km.

The calculated distances are then used to obtain the matrix that represents the cost function used to solve the task assignment problem, which is composed of the distance between each delivery man and each customer order, as illustrated in Figure 2. The matrix used during the experiments is composed of 7707 possible deliveries and 7707 customer orders.

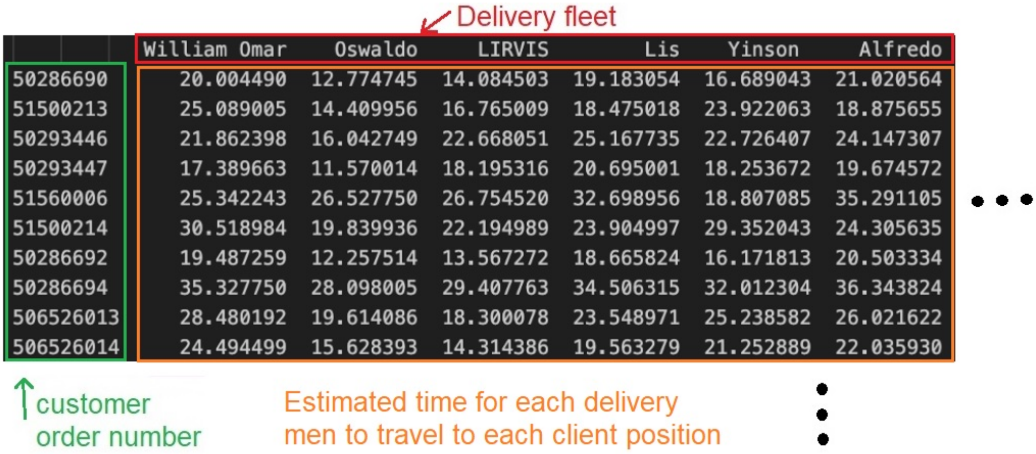


	William Omar	Oswaldo	LIRVIS	Lis	Yinson	Alfredo
50286690	4.818219	2.256602	2.720670	4.527171	3.643502	5.178231
51500213	6.619746	2.835984	3.670418	4.276302	6.206279	4.418254
50293446	5.476506	3.414509	5.761962	6.647641	5.782639	6.286087
50293447	3.891743	1.829746	4.177199	5.062878	4.197875	4.701324
51560006	6.709472	7.129517	7.209865	9.316076	4.393959	10.234517
51500214	8.543677	4.759915	5.594349	6.200233	8.130210	6.342185
50286692	4.634955	2.073338	2.537407	4.343907	3.460238	4.994967
50286694	10.247501	7.685884	8.149953	9.956453	9.072784	10.607513
506526013	7.821299	4.679892	4.214318	6.074087	6.672744	6.950187
506526014	6.409103	3.267696	2.802122	4.661891	5.260547	5.537991

Figure 2. Sample of the cost matrix that represents the distance between each delivery person and each customer order. The matrix we used is composed of 7707 possible deliveries and 7707 customer orders.

2.3. ML Algorithms to Determine the Cost Matrix

In addition to directly calculating distance with the Haversine formula, we propose the use of distance and time estimations to build the cost matrix to be optimized. As we only have specific data points for delivery persons who completed deliveries, and creating the assignment cost matrix requires calculating the potential distance and travel time for each delivery person to each customer, we utilize supervised regression-based learning models to estimate distance and delivery time. We employ two different methods to obtain an assignment matrix representing the cost function: linear regression and polynomial regression. An example of the cost matrix obtained for travel time estimation can be observed in Figure 3. We briefly explain each of the used regression methods below. It is worth mentioning that the matrix used during the experiments is composed of 7707 possible deliveries and 7707 customer orders.



	William Omar	Oswaldo	LIRVIS	Lis	Yinson	Alfredo
50286690	20.004490	12.774745	14.084503	19.183054	16.689043	21.020564
51500213	25.089005	14.409956	16.765009	18.475018	23.922063	18.875655
50293446	21.862398	16.042749	22.668051	25.167735	22.726407	24.147307
50293447	17.389663	11.570014	18.195316	20.695001	18.253672	19.674572
51560006	25.342243	26.527750	26.754520	32.698956	18.807085	35.291105
51500214	30.518984	19.839936	22.194989	23.904997	29.352043	24.305635
50286692	19.487259	12.257514	13.567272	18.665824	16.171813	20.503334
50286694	35.327750	28.098005	29.407763	34.506315	32.012304	36.343824
506526013	28.480192	19.614086	18.300078	23.548971	25.238582	26.021622
506526014	24.494499	15.628393	14.314386	19.563279	21.252889	22.035930

Figure 3. Sample of the cost matrix representing the estimated delivery time between each delivery person and each customer order. The time estimation was realized using the linear and polynomial regression models.

2.3.1. Linear Regression

Linear regression is a supervised learning method used to model the relationship between a dependent variable and one or more independent variables [25–27]. Linear regression finds the line or hyperplane of best fit that minimizes the error between the predicted values and the actual values in a dataset. To achieve this, various model parameters are iteratively updated to minimize a cost function, such as the Mean Square Error (MSE) [25–27]. Gradient descent can be used to update the parameters and minimize the cost function, as shown in Algorithm 1.

Algorithm 1 Gradient descent for solving linear regression [25–27]

Input: Training dataset with dependent variable y and independent variables $x_1, x_2, \ldots, x_n$
Initialize parameters $b_0, b_1, \ldots, b_n$ randomly for the linear regression model:
$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$
Choose a learning rate α
Set the maximum number of iterations N
Repeat N **iterations**:
  Compute predictions for all training examples:
  $h_\theta(x^{(i)}) = b_0 + b_1 x_1^{(i)} + b_2 x_2^{(i)} + \ldots + b_n x_n^{(i)}$
  Compute cost function that calculates the error between predicted and actual values:
  $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
  Update model parameters:
  $b_j := b_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$, for $j = 0, 1, \ldots, n$
  Check if N iterations were reached
Output: Model parameters $b_0, b_1, \ldots, b_n$ optimized for the dataset

Note that $J(\theta)$ represents the cost function to be optimized, m is the number of training examples, $h_\theta(x^{(i)})$ represents the predicted value for the i-th training example, $y^{(i)}$ is the actual value for the i-th training example, and α is the learning rate [25–27].

2.3.2. Polynomial Regression

Polynomial regression is a supervised learning method that extends linear regression to model the relationship between a dependent variable and one or more independent variables [27–29]. Instead of fitting a straight line or hyperplane, polynomial regression fits a best-fitting curve or surface to the data. As with linear regression, polynomial regression aims to minimize the error between the predicted values and the actual values in a dataset.

To achieve this, various model parameters are iteratively updated to minimize a cost function, such as the Mean Squared Error (MSE) [27–29]. We describe gradient descent in Algorithm 2 for polynomial regression, which can be used to update the parameters and minimize the cost function. This iterative optimization process allows polynomial regression to capture more complex relationships between variables than linear regression.

Algorithm 2 Gradient descent for polynomial regression [27–29]

Input: Training dataset with dependent variable y and independent variables $x_1, x_2, \ldots, x_n$
Initialize parameters $b_0, b_1, \ldots, b_n$ randomly for the polynomial regression model :
$y = b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n$
Choose a learning rate α
Set the maximum number of iterations N
Repeat N **iterations**:
  Compute predictions for all training examples:
  $h_\theta(x^{(i)}) = b_0 + b_1 x^{(i)} + b_2 (x^{(i)})^2 + \ldots + b_n (x^{(i)})^n$
  Compute cost function that calculates the error between predicted and actual values:
  $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
  Update model parameters:
  $b_j := b_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})(x^{(i)})^j$, for $j = 0, 1, \ldots, n$
  Check if N iterations were reached
Output: Model parameters $b_0, b_1, \ldots, b_n$ optimized for the dataset

Note that y represents the dependent variable, x is the independent variable, $b_0, b_1, b_2, \ldots, b_n$ are the coefficients of the polynomial model, and n is the degree of the polynomial. In this work, we propose the use of a quadratic polynomial regression (degree 2).

2.4. Hungarian Algorithm

To effectively assign orders to delivery people while maximizing assignment efficiency, we use the Hungarian algorithm [15,30,31]. This method is a widely used optimization algorithm for solving the assignment problem in bipartite graphs. It efficiently solves the problem of finding the optimal assignment of tasks to agents based on cost values [15,30,31]. The Hungarian algorithm is highly efficient, with a time complexity that makes it suitable for real-time and large-scale applications. Moreover, the algorithm guarantees an optimal solution to the assignment problem, ensuring that the total cost or benefit is minimized or maximized, depending on the objective function. The Hungarian algorithm operates on a cost matrix representing the costs or benefits of assigning each task to each agent. Through a series of matrix operations, including row and column reductions and assignments, the algorithm efficiently identifies the optimal assignment.

The algorithm requires a cost matrix representing the costs or benefits associated with assigning tasks to agents. This cost matrix is an $n \times n$ matrix, where n is the number of tasks or agents, as can be observed in Equation (3):

$$\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \tag{3}$$

where c_{ij} represents the cost of assigning task i to agent j.

The Hungarian algorithm starts by reducing each row of the cost matrix by subtracting the minimum cost in that row from all the elements in the row. This ensures that at least one zero is present in each row, as can be observed in Equation (4).

$$\text{Minimize each row: } \text{row}_i = \text{row}_i - \min(\text{row}_i) \tag{4}$$

Similarly, each column of the cost matrix is reduced by subtracting the minimum cost in that column from all the elements in the column. This ensures that at least one zero is present in each column, as can be observed in Equation (5).

$$\text{Minimize each column: } \text{column}_j = \text{column}_j - \min(\text{column}_j) \tag{5}$$

After row and column reductions, the algorithm creates an assignment matrix, initially filled with zeros, in which each zero represents an assignment of a task to an agent:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \tag{6}$$

where $a_{ij} = 1$ if task i is assigned to agent j and $a_{ij} = 0$.

The algorithm iterates through various steps of reducing rows and columns, updating the assignment matrix, and finding augmenting paths until optimality conditions are met, ensuring that no more zeros can be chosen without violating the assignment constraints [15,30,31]. This means that each row and each column will contain exactly one zero in the assignment matrix A. Finally, the Cost of Assignment $C(A)$ that represents the total cost of the assignment can be represented as stated in Equation (7).

$$C(A) = \sum_{i=1}^{n} c_{ij} \cdot a_{ij} \tag{7}$$

This equation represents the core operations of the Hungarian algorithm, which iteratively reduces costs, updates the assignment matrix, and finds the optimal assignment by satisfying the optimal conditions.

3. Results

In this section, we first evaluate the results of the linear and polynomial regression algorithms used to estimate the distance and delivery time of the delivery people, thereby completing the cost matrices. Subsequently, we present and compare the various outcomes obtained when solving optimization problems related to the assignment of delivery people to each potential customer. The results are presented below.

3.1. Regression Results

The experimental findings presented in Table 1 depict the performance of the proposed linear regression and polynomial regression techniques, which were trained to estimate distances and times that were then used to fill the cost matrix representing the cost function. In the case of distance estimation, polynomial regression outperforms linear regression, yielding a lower mean squared error (MSE) of 2.6 m^2 compared to 6.9 m^2 for linear regression. This translates to a substantially lower Root Mean Squared Error (RMSE) of 4.5 m for polynomial regression, indicating greater accuracy in predicting distances. Similarly, for time estimation, polynomial regression achieves superior results, with an MSE of 59.6 s^2 compared to 120.9 s^2 for linear regression. Consequently, the RMSE for time estimation with polynomial regression is notably lower at 7.7 s, showcasing its enhanced precision in predicting time durations compared to the RMSE of 11.0 s for linear regression. These results suggest that polynomial regression offers better predictive capabilities for both distance and time estimation tasks compared to linear regression, demonstrating its efficacy in capturing the underlying patterns and complexities of the proposed dataset.

Table 1. Mean squared error and root mean squared error regarding distance estimation.

	MSE	**RMSE**
Distance estimation - linear regression	6.9 m^2	20.3 m
Distance estimation - polynomial regression	2.6 m^2	4.5 m
Time estimation - linear regression	120.9 s^2	11.0 s
Time estimation - polynomial regression	59.6 s^2	7.7 s

3.2. Task Assignment Problem Results

The results presented in Table 2 show the performance of the proposed methods in estimating distances traveled by the delivery people after solving the task assignment problem with the Hungarian algorithm. The Hungarian method yields an average distance estimation of 4.55 km with a standard deviation of ±3.24 km, resulting in a total summation of 34,956.84 km. Interestingly, when combining linear regression with the Hungarian method, the average estimation remains consistent at 4.55 km, with a slightly reduced standard deviation of ±3.23 km, contributing to a marginally increased total summation of 34,957.74 km. Conversely, employing polynomial regression in conjunction with the Hungarian method leads to a slightly higher average distance estimation of 4.62 km accompanied by a larger standard deviation of ±4.49 km, resulting in a total summation of 35,526.26 km. These findings suggest that while both linear and polynomial regression exhibit similar average distance estimations when combined with the Hungarian method, polynomial regression introduces greater variability in distance estimations, as indicated by its higher standard deviation. Notably, considering that less distance is better, the Hungarian method and the linear regression combined with the Hungarian method emerge as the most effective approaches when distance estimation is used to solve the assignment problem for the proposed dataset.

Table 2. Comparison results of the proposed methods for solving the task assignment problem using distance metrics at the cost matrix.

Method	**Average (km)**	**Standard Deviation (km)**	**Summation of Total Distance (km)**
Hungarian	4.55	±3.24	34,956.84
Linear regression + Hungarian	4.55	±3.23	34,957.74
Polynomial regression + Hungarian	4.62	±4.49	35,526.26

As can be observed in Table 3, for the linear regression method the average time estimation is 19.24 min, accompanied by a standard deviation of ±9.12 min, resulting in a total summation of 147,891.23 min. Conversely, the polynomial regression method yields a notably lower average time estimation of 17.01 min along with a reduced standard deviation of ±5.25 min, contributing to a shorter total summation of 130,747.82 min. These results indicate that polynomial regression offers more precise estimations with less variability compared to linear regression when solving the task assignment problem with the Hungarian method.

Table 3. Comparison results of the proposed methods for solving the task assignment problem using time metrics at the cost matrix.

	Average (min)	Standard Deviation (min)	Summation of Total Time (min)
Linear Regression + Hungarian	19.24	±9.12	147,891.23
Polynomial Regression + Hungarian	17.01	±5.25	130,747.82

Finally, in Figures 4 and 5 we present a sample of the results of the automatic assignment of tasks using the Hungarian algorithm and the different methods of estimating the cost function. As shown in Figure 4, the use of distance to construct the cost matrix reveals that the best method is linear regression, with a total travel distance of 102.87 km. This is closely followed by the Haversine method at 103.89 km and polynomial regression at 109.14 km. Conversely, Figure 5 illustrates that when time is utilized to formulate the cost matrix, polynomial regression emerges as the best method, resulting in a total travel time of 376.06 min. In contrast, the linear regression method yields a total time of 424.87 min. To summarize, linear regression methods demonstrate superior performance for our dataset when distance estimation is utilized, whereas polynomial regression methods excel in scenarios where time estimation is paramount.

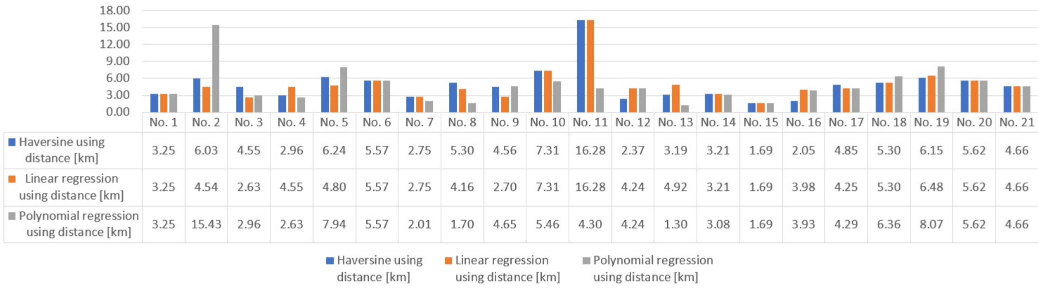


Figure 4. Distance calculation after optimization procedure for a sample of 21 delivery people. Total distance using Haversine method is 103.89 km. Total distance using linear regression is 102.87 km. Total distance using polynomial regression is 109.14 km.

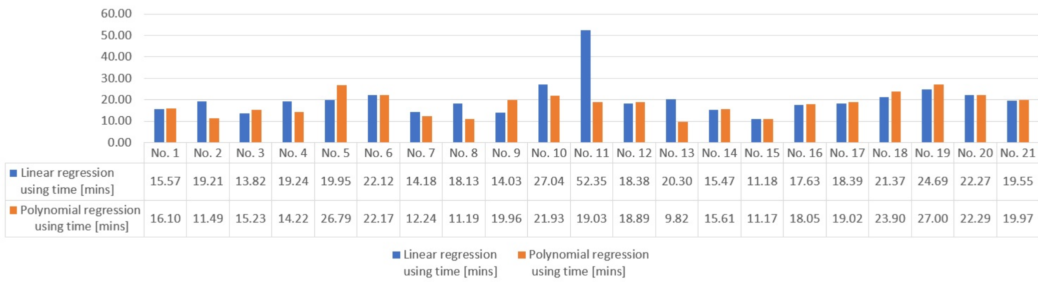


Figure 5. Time calculation after optimization procedure for a sample of 21 delivery people. Total time using linear regression is 424.87 min. Total time using linear regression is 376.06 min.

4. Discussion

In this section, we present a summary of the most important findings and future work insights related to the development of this work.

- During the implementation of regression algorithms, it can be seen that polynomial regression exceeds linear regression in distance and time estimation tasks. Considering the RMSE, the polynomial regression reaches errors of 4.5 m and 7.7 s, while the linear regression of only 20.3 m and 11 s. However, when applying regression methods together with the Hungarian algorithm it can be noted that the results are similar for all methods. However, linear regression in this case obtains better results, obtaining a total distance traveled from 34,957.74 km and the lowest standard deviation of 3.23 km. This indicates that even though polynomial regression presents less error during training, this does not necessarily imply better functioning when applying it when solving delivery assignment problems.

- The complexity of managing several orders for the automatic assignment of the delivery personnel is one of the inherent inefficiencies of the delivery process. To address this difficulty, it is necessary to consider different scenarios. For example, it should be considered in some way that different orders can be assigned to different distributors, and that the possibility of making an optimal delivery for a customer does not necessarily mean that the entire set of deliveries can be done optimally. Fortunately, optimal assignment algorithms such as the proposed Hungarian algorithm combined with machine learning methods can reduce cost functions to solve optimization problems optimally.
- In this work, we consider Max Delivery, a company headquartered in Barcelona, Spain, which delivers orders to customers within this country. This implies a limited scope to this region for the distances of the orders. However, the distances of orders within this country can be considerably high. For example, if the client is far from the starting point (restaurant), the delivery may need to travel very far to reach the client. From the point of view of developing an application, an economic compensation method for the delivery man, who could make several deliveries within the time it takes to make a distant delivery, should be considered. Another way to improve this process is to try to apply areas by zones to ensure that distributors do not have to travel too far from their home or work zone without receiving additional compensation.
- It is important to mention that consideration for future jobs can be provided as delivery companies prioritize security and packages during transport through several measures. These include comprehensive training programs for personnel, strict background verification, regular vehicle maintenance, and the use of advanced monitoring systems to monitor packages in real time. Secure delivery locations can be considered in the cost function to avoid damage and reduce robberies as well as cost to the nearby systems or security establishments in case of incidents.
- The delivery industry faces important ethical and environmental challenges, including transport carbon emissions, excessive waste, concerns about labor practices, data privacy problems, and community impacts such as traffic congestion and noise pollution. To address these challenges, in future works it would be possible to consider those parameters within the cost functions to be reduced for the assignment of deliveries. Additionally, it is important to consider that delivery companies might adopt sustainable practices such as the use of electric vehicles and recyclable containers. It is important to implement solid data protection measures to prevent vehicles with their orders from interceptions in the event that a cybersecurity problem is suffered in applications.
- Future work could consider issues related to customer satisfaction and behavior, as these directly affect purchases that are distributed via delivery. Late or damaged deliveries can lead to frustration and dissatisfaction, making the optimization of delivery processes to guarantee precision, speed, and reliability essential. Future work could investigate ways to estimate customer satisfaction based on the implementation of a system that generates efficient routes. In addition, flexible delivery options such as scheduled deliveries can be provided to improve customer satisfaction and optimize the allocation algorithm in a scheduled manner.
- Changes in technology and consumer behavior can have a deep impact on the delivery industry. The automatic assignment system could learn or adapt to these changes automatically. In this context, machine learning algorithms such as reinforcement learning could be analyzed in the future to evaluate their possible benefits to solving delivery problems adaptively in the time domain.

- In future works, the following considerations must be taken in order to predict and address possible delays in IA delivery. It is important to analyze large amounts of data from various sources, including historical delivery data, traffic patterns, weather forecasts, and road conditions. In this way, possible real-time delays can be forecast in order to dynamically adapt the delivery routes using AI algorithms together with optimization.
- The selection of monitoring technology for delivery and tracking of packages can significantly affect the cost function to be optimized by the task allocation algorithm. For example, bar scanning, radio frequency identification (RFID), and global positioning system (GPS) can offer real-time monitoring with different advantages and limitations. In future works, the strengths and weaknesses of the use of each of these technologies or the fusion of their information could be evaluated to consider them for application in an optimization algorithm for task assignment such as the one in this work.
- In future works, we will evaluate the use of an Open-Source Routing Machine (OSRM), which is a high-performance routing engine for various transportation tasks such as driving. This source of information could be very interesting in future works, as it can help to improve the cost matrix used to solve the task assignment problem. We will evaluate and compare the use of OSRM with the performance of the proposed approach.

5. Conclusions

In this work, we propose a smart delivery system to optimize the task assignment problem for delivery personnel with pending orders. The proposed approach utilizes the Hungarian algorithm and various cost functions obtained through deterministic and machine learning techniques. Linear and polynomial regression methods are compared to construct different cost functions represented by matrices with orders and delivery information. Empirical findings disclosed that linear regression models reduced estimation errors in distance estimation by up to 568.52 km (1.51%) compared to other methods, while polynomial regression models reduced errors in time estimation by up to 17,143.41 min (11.59%). The proposed smart delivery system aims to enhance delivery personnel allocation within the delivery sector, thereby optimizing efficiency.

Author Contributions: Conceptualization, J.P.V. and E.S.; Data curation, J.P.V., E.S. and V.M.; Formal analysis, J.P.V., E.S., I.N.V., A.P., O.M., R.G.-R. and L.G.; Funding acquisition, J.P.V. and I.N.V.; Investigation, J.P.V., E.S., I.N.V., V.M., A.P., O.M., R.G.-R. and L.G.; Methodology, J.P.V. and E.S.; Project administration, J.P.V.; Resources, J.P.V., E.S., I.N.V., V.M., A.P., O.M., R.G.-R. and L.G.; Supervision, J.P.V., I.N.V., O.M., R.G.-R. and L.G.; Visualization, J.P.V., E.S. and I.N.V.; Writing—original draft, J.P.V., E.S., I.N.V., V.M., A.P., O.M., R.G.-R. and L.G.; Writing—review and editing, J.P.V., I.N.V., V.M., A.P., O.M., R.G.-R. and L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by ANID under grant Fondecyt Iniciación 11240105 and 11241171.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: This work has been supported by ANID under grant Fondecyt Iniciación 11240105 and 11241171. The authors gratefully acknowledge the support provided by the Faculty of Engineering, Universidad Andres Bello, Santiago, Chile. The authors acknowledge that this article was partially generated by ChatGPT (powered by OpenAI's language model, GPT-3.5; http://openai.com). This model was primarily utilized to improve the writing quality, correct typos, and address language issues within this article, as the authors are not native English speakers. It is important to mention that the conceptualization, data curation and analysis, methodology, software development, formal analysis, conclusions, and final editing were all developed by the human authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

In Tables A1–A3, we summarize the bibliographic review showing the addressed problem, their proposed solution, the disadvantages encountered, and the obtained results.

Table A1. Comparative analysis among the state of the art—Part 1.

Ref.	Problem	Methodology Description	Disadvantages	Results
[13]	Food delivery service improved with Q-learning and DDQN using rule-based policy.	Emulated food delivery service was modeled using a Markov decision process. Q-learning and DDQN were employed to maximize revenue derived from served requests within a limited number of couriers over a period of time.	Limited scope of the state space, focusing on courier and order location information without incorporating additional relevant attributes, limiting generalizability to larger-scale operations.	DDQN algorithm collects more reward compared to other algorithms.
[14]	Scheduling problem in crowdsourced delivery platforms.	Machine learning method that combines simulation optimization for off-line training and a neural network. Real-world data provided by a crowdsourced delivery platform were used.	Complex implementation with difficult data requirements, scalability challenges and use of personal data.	Solution quality within 0.2–1.9% of a bespoke sample average approximation method, while being several orders of magnitude faster regarding online solution generation.
[19]	Two-echelon vehicle routing for emergency mask delivery during COVID-19.	A hybrid machine learning and heuristic optimization method was proposed to address the delivery of medical masks problem. Deep learning combined with heuristics optimization was used to predict regional delivery demand.	Not considering vehicle refueling or recharging, problematic for electric vehicles in short- and medium-distance delivery.	The average weighted delivery time was reduced up to 61%.
[18]	Enhancing on-time performance in last-mile delivery services.	A data-driven framework to model delivery performance and optimize order assignments to drivers. Total delivery time was decomposed into uncertain service time at customer locations and predictable travel time. A prediction model for delivery tour length was then developed.	Lacks discussion on potential limitations of implementing data-driven optimization strategies in last-mile delivery services.	Advantages of data-driven order assignment models integrated with delivery tour prediction over classical vehicle routing problems. Method Average mean square error (MSE): LASSO (0.314), Ridge regression (0.317), SVR (0.295), Random forest (0.304).

Table A2. Comparative analysis among among the state of the art—Part 2.

Ref.	Problem	Methodology Description	Disadvantages	Results
[5]	Optimization under uncertainty in the context of last-mile delivery and third-party logistics, concentrated on solving the variable cost and size bin packing.	Metaheuristic algorithms were used to optimize decision-making in logistics, and machine learning enhanced decisions by learning from data patterns, making predictions, and offering recommendations in logistics operations.	Limited discussion on the scalability and generalizability of the proposed machine learning optimization approach.	Progressive Hedging and Machine Learning approaches generate closely aligned first-stage solutions with minor variations in their outcomes.
[9]	Impact of the limited available resources in the meal delivery.	Implemented a Markov decision process and employed deep reinforcement learning with eight Deep Q-Networks (DQN) algorithms.	Exclusion of different characteristics of couriers, such as varying delivery speeds or behaviors, which may impact the real-world applicability.	Increasing the number of couriers in a delivery system results in higher rewards and fewer rejected orders.
[10]	Mixed-integer programming formulation for drone vehicle routing (VRPD) by assigning clients to drone-truck pairs.	An ant colony optimization (ACO) algorithm was developed.	Do not delve deeply into the real-world implementation challenges and regulatory hurdles of integrating drones into existing delivery systems.	The ACO algorithm outperformed classic VRP by achieving cost savings of more than 30% for large instances.
[12]	Enhance customer satisfaction	Employed the LSTM to predict future levels of customer satisfaction.	Do not address scalability challenges, implementation barriers, or real-world case studies to demonstrate the effectiveness of the proposed approach.	A smart contract was designed to provide compensation and/or refunds to customers when their satisfaction with the delivery services was low.
[15]	Assigning food orders to delivery vehicles to minimizes delivery time.	The algorithm FOODMATCH was designed to tackle vehicle assignment by treating it as a minimum weight perfect matching problem on a bipartite graph.	The generalizability of the findings to other regions or platforms may be limited.	Achieving a 30% reduction in delivery time.

Table A3. Comparative analysis among among the state of the art—Part 3.

Ref.	Problem	Methodology Description	Disadvantages	Results
[20]	Optimize recharging, delivering, and repositioning task assignments for electric vehicles.	Modeled as a multi-agent multi-task dynamic dispatching problem using a Markov Decision Process.	Lack of comparative analysis with existing methods could limit the assessment of the novelty and effectiveness.	Total revenue up by 33.2%; Task allocation repositioning raised total revenue by 50.0%; Re-estimated state value function boosted total revenue by 2.8%.

Table A3. *Cont.*

Ref.	Problem	Methodology Description	Disadvantages	Results
[21]	Income guarantees for delivery agents, minimize costs and ensure customer satisfaction.	The WORK4FOOD algorithm was designed and implemented, utilizing minimum weight bipartite matching and Gaussian process regression to assess the demand-supply dynamics.	Do not address the potential challenges or barriers to implementing WORK4FOOD in existing food delivery platforms.	Reduced platform costs by up to 25% compared to solutions like FOODMATCH and FAIRFOODY, achieving a balance between cost, delivery times, and fairness.
[16]	Delivery assignment based on order-to-vehicle, order batching, and vehicle movements.	Papping the vehicle assignment problem to minimum weight perfect matching. Best-first search utilized to construct a subgraph likely to contain the minimum matching.	Limited discussion on the environmental impact of increased food delivery services and vehicle usage.	Reduced food delivery time, waiting time at restaurants, and increased orders delivered per kilometer.
[17]	Efficient allocation of orders to drivers and route planning.	Modified Kuhn-Hungarian (Munkres) algorithm for orders-drivers matching. Machine learning to predict order batching. Plus, rule-based route planning for viable routes for drivers.	Lack of realistic constraints for the online food delivery problem, such as the uncertainty of travel time and dynamic arrival of orders.	Satisfying performance of the classification model on real datasets and effectiveness of the proposed algorithm for solving the OFDP.

References

1. Farooq, Q.; Fu, P.; Hao, Y.; Jonathan, T.; Zhang, Y. A review of management and importance of e-commerce implementation in service delivery of private express enterprises of China. *Sage Open* **2019**, *9*, 2158244018824194. [CrossRef]
2. Fadda, E.; Perboli, G.; Tadei, R. Customized multi-period stochastic assignment problem for social engagement and opportunistic IoT. *Comput. Oper. Res.* **2018**, *93*, 41–50. [CrossRef]
3. Fadda, E.; Perboli, G.; Tadei, R. A progressive hedging method for the optimization of social engagement and opportunistic IoT problems. *Eur. J. Oper. Res.* **2019**, *277*, 643–652. [CrossRef]
4. Giuffrida, N.; Fajardo-Calderin, J.; Masegosa, A.D.; Werner, F.; Steudter, M.; Pilla, F. Optimization and machine learning applied to last-mile logistics: A review. *Sustainability* **2022**, *14*, 5329. [CrossRef]
5. Bruni, M.E.; Fadda, E.; Fedorov, S.; Perboli, G. A machine learning optimization approach for last-mile delivery and third-party logistics. *Comput. Oper. Res.* **2023**, *157*, 106262. [CrossRef]
6. Reis, J.; Amorim, M.; Cohen, Y.; Rodrigues, M. Artificial intelligence in service delivery systems: A systematic literature review. In *Trends and Innovations in Information Systems and Technologies: Volume 1*; Springer: Cham, Switzerland, 2020; pp. 222–233.
7. Gursoy, D.; Chi, O.H.; Lu, L.; Nunkoo, R. Consumers acceptance of artificially intelligent (AI) device use in service delivery. *Int. J. Inf. Manag.* **2019**, *49*, 157–169. [CrossRef]
8. Adak, A.; Pradhan, B.; Shukla, N.; Alamri, A. Unboxing deep learning model of food delivery service reviews using explainable artificial intelligence (XAI) technique. *Foods* **2022**, *11*, 2019. [CrossRef]
9. Jahanshahi, H.; Bozanta, A.; Cevik, M.; Kavuk, E.M.; Tosun, A.; Sonuc, S.B.; Kosucu, B.; Başar, A. A deep reinforcement learning approach for the meal delivery problem. *Knowl.-Based Syst.* **2022**, *243*, 108489. [CrossRef]
10. Huang, S.H.; Huang, Y.H.; Blazquez, C.A.; Chen, C.Y. Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm. *Adv. Eng. Inform.* **2022**, *51*, 101536. [CrossRef]
11. Asih, A.M.S.; Sopha, B.M.; Kriptaniadewa, G. Comparison study of metaheuristics: Empirical application of delivery problems. *Int. J. Eng. Bus. Manag.* **2017**, *9*, 1847979017743603. [CrossRef]
12. Tian, Z.; Zhong, R.Y.; Vatankhah Barenji, A.; Wang, Y.; Li, Z.; Rong, Y. A blockchain-based evaluation approach for customer delivery satisfaction in sustainable urban logistics. *Int. J. Prod. Res.* **2021**, *59*, 2229–2249. [CrossRef]
13. Bozanta, A.; Cevik, M.; Kavaklioglu, C.; Kavuk, E.M.; Tosun, A.; Sonuc, S.B.; Duranel, A.; Basar, A. Courier routing and assignment for food delivery service using reinforcement learning. *Comput. Ind. Eng.* **2022**, *164*, 107871. [CrossRef]
14. Behrendt, A.; Savelsbergh, M.; Wang, H. A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. *Transp. Sci.* **2023**, *57*, 889–907. [CrossRef]

15. Joshi, M.; Singh, A.; Ranu, S.; Bagchi, A.; Karia, P.; Kala, P. Batching and matching for food delivery in dynamic road networks. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 2099–2104.
16. Joshi, M.; Singh, A.; Ranu, S.; Bagchi, A.; Karia, P.; Kala, P. FoodMatch: Batching and matching for food delivery in dynamic road networks. *ACM Trans. Spat. Algorithms Syst. (TSAS)* **2022**, *8*, 1–25. [CrossRef]
17. Wang, X.; Wang, L.; Wang, S.; Yu, Y.; Chen, J.f.; Zheng, J. Solving online food delivery problem via an effective hybrid algorithm with intelligent batching strategy. In *International Conference on Intelligent Computing*; Springer: Cham, Switzerland, 2021; pp. 340–354.
18. Liu, S.; He, L.; Shen, Z.J.M. Data-driven order assignment for last mile delivery. *SSRN Electron. J.* **2018**, *9*, 1–44. [CrossRef]
19. Chen, X.; Yan, H.F.; Zheng, Y.J.; Karatas, M. Integration of machine learning prediction and heuristic optimization for mask delivery in COVID-19. *Swarm Evol. Comput.* **2023**, *76*, 101208. [CrossRef] [PubMed]
20. Wang, N.; Guo, J. Multi-task dispatch of shared autonomous electric vehicles for Mobility-on-Demand services–combination of deep reinforcement learning and combinatorial optimization method. *Heliyon* **2022**, *8*, e11319. [CrossRef] [PubMed]
21. Nair, A.; Yadav, R.; Gupta, A.; Chakraborty, A.; Ranu, S.; Bagchi, A. Gigs with guarantees: Achieving fair wage for food delivery workers. *arXiv* **2022**, arXiv:2205.03530.
22. Robusto, C.C. The cosine-haversine formula. *Am. Math. Mon.* **1957**, *64*, 38–40. [CrossRef]
23. Basyir, M.; Nasir, M.; Suryati, S.; Mellyssa, W. Determination of nearest emergency service office using haversine formula based on android platform. *EMITTER Int. J. Eng. Technol.* **2017**, *5*, 270–278. [CrossRef]
24. Ashraf, S.; Saleem, S.; Ahmed, T.; Aslam, Z.; Shuaeeb, M. Iris and Foot based Sustainable Biometric Identification Approach. In Proceedings of the 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19 September 2020; pp. 1–6. [CrossRef]
25. Maulud, D.; Abdulazeez, A.M. A review on linear regression comprehensive in machine learning. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 140–147. [CrossRef]
26. Hope, T.M. Linear regression. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 67–81.
27. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*; Springer: New York, NY, USA, 2013; Volume 112.
28. Heiberger, R.M.; Neuwirth, E.; Heiberger, R.M.; Neuwirth, E. Polynomial regression. In *R Through Excel: A Spreadsheet Interface for Statistics, Data Analysis, and Graphics*; Springer: New York, NY, USA, 2009; pp. 269–284.
29. Ostertagová, E. Modelling using Polynomial Regression. *Procedia Eng.* **2012**, *48*, 500–506. [CrossRef]
30. Shah, K.; Reddy, P.; Vairamuthu, S. Improvement in Hungarian algorithm for assignment problem. In *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems: Proceedings of ICAEES 2014, Volume 1*; Springer: New Delhi, India, 2015; pp. 1–8.
31. Sanseverino, E.R.; Ngoc, T.N.; Cardinale, M.; Vigni, V.L.; Musso, D.; Romano, P.; Viola, F. Dynamic programming and Munkres algorithm for optimal photovoltaic arrays reconfiguration. *Sol. Energy* **2015**, *122*, 347–358. [CrossRef]