Article

Not peer-reviewed version

Smart Delivery Assignment Problem through Machine Learning and the Munkres Algorithm

Juan Pablo Vásconez [*] , Elias Schotborgh , Ingrid Nicole Vásconez , Viviana Moya , Andrea Pilco , Oswaldo Menéndez , Robert Guamán-Rivera , Leonardo Guevara

Posted Date: 28 March 2024

doi: 10.20944/preprints202403.1703.v1

Keywords: Smart delivery; Machine learning; Regression model; Munkres optimization algorithm

Article

Smart Delivery Assignment through Machine Learning and the Munkres 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] Leonardo Guevara [6]

[1] Faculty of Engineering, Universidad Andres Bello, Santiago, Chile; e.schotborghgonzlez@uandresbello.edu (E.S.)
[2] Federico Santa Maria Technical University, Valparaíso, Chile; ingrid.vasconez@sansano.usm.cl (I.N.V.)
[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] Universidad Católica del Norte, Departamento de Ingeniería de Sistemas y Computación, Antofagasta 1249004, Chile; oswaldo.menendez@ucn.cl
[5] Institute of Engineering Sciences, Universidad de O'Higgins; robert.guaman@uoh.cl
[6] Lincoln Institute for Agri-food Technology, University of Lincoln, Lincoln, UK; lguevara@lincoln.ac.uk
[*] Correspondence: juan.vasconez@unab.cl
[†] Current address: Faculty of Engineering, Universidad Andres Bello, Antonio Varas 810, 7500735 Providencia, Santiago, Chile.

Abstract: In the realm of smart transportation and mobility, delivery organizations are continuously refining their strategies, harnessing resources and advanced analytical techniques to streamline operations, aiming to achieve optimal solutions in terms of both efficiency and cost-effectiveness. However, a persistent challenge faced by these delivery companies revolves around the effective assignment of delivery personnel to customer orders, often grappling with suboptimal assignments that lead to inefficiencies and delays. To address this issue, we propose an architecture that uses the Munkres algorithm to optimize the task assignment problem for delivery personnel with pending orders, employing different cost functions obtained with deterministic and machine learning techniques. We compared the performance of linear and polynomial regression methods to construct different cost functions represented by matrices with orders and delivery people information. Subsequently, we applied the Munkres optimization algorithm to solve the assignment problem, which optimally assigns delivery people and orders. The results demonstrate that linear regression, used to estimate distance information, can reduce estimation errors by up to 568.52 km (1.51%) for our dataset compared to other methods. Conversely, polynomial regression proves effective in constructing a superior cost function based on time information, reducing estimation errors by up to 17,143.41 minutes (11.59%) compared to alternative methods. The proposed approach aims to enhance delivery personnel allocation within the delivery sector, optimizing the efficiency of this process.

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

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, spanning from the imperative to meet increasingly demanding customer expectations for swift and reliable service to the intricate task of managing inventory levels efficiently. Additionally, the optimization of 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 [2,3]. 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, the use of Artificial Intelligence (AI) techniques can represent revolutionary opportunities to improve traditional approaches related to delivery, logistics, and supply chain management [4]. 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 [4,5]. Moreover, DL models work well at processing complex data types like images and text, enabling tasks such as automatic package sorting, vehicle recognition, and natural language processing for customer inquiries [6]. 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 [7]. 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 [8]. 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 [9]. By mathematically modeling delivery constraints and objectives, optimization methods can help businesses 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 [10].

Over the past few years, task assignment approaches related to delivery problems have gained significant attention from various researchers. For example, in cite [11], the authors present an algorithm aimed at optimizing food delivery processes by minimizing delivery time in road networks. This approach formulates an order assignment problem as a minimum weight perfect matching task on a bipartite graph. Then, the authors employ the best-first search graph method to efficiently calculate the solution space. 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 demonstrates its effectiveness in achieving a 30% reduction in delivery time. Other works emphasize the importance of minimizing the delivery time for customer satisfaction [12]. The challenge in this respect involves stages such as order-to-vehicle assignment, order batching, and adapting to vehicle movements. To address this, the authors introduced a solution mapping the vehicle assignment problem to minimum weight perfect matching on a bipartite graph. To optimize efficiency, best-first search is utilized to construct a subgraph likely to contain the minimum matching. The algorithm enhances solution quality by considering graph batching and dynamic vehicle positions using angular distance. Extensive experiments on real food-delivery data from metropolitan cities demonstrate this approach presents substantial improvements over other strategies, including reduced food delivery time, waiting time at restaurants, and increased orders delivered per kilometer. In the work presented in [13], the authors address the challenge of online food delivery focusing on the efficient allocation of orders to drivers and route planning. For order assignment, a modified Kuhn-Munkres (Munkres) algorithm was 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 demonstrate 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 [14], the authors discuss an Autonomous Mobility-on-Demand system, which was 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 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-Munkres and Edmond-Karp were employed to solve these problems and achieve optimal task allocation policies. In [15], a reward function balancing order income with trip satisfaction was proposed along with a state-value function estimated by Back Propagation-Deep Neural Network to assess the matching degree between vehicles and tasks. The results demonstrate 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.

Based on the bibliographic review presented above, it can be observed that task assignment problems are a broad field of research, which has had several developments with sundry results. In this context, the motivation of this work is to use machine learning and the Munkres optimization algorithm in a single architecture to find a feasible method to improve the assignment tasks between delivery people and orders. The synergy between Machine Learning and the Munkres Algorithm enabled us to improve delivery operations, which might have a positive impact on operational costs. Our article contributions can be summarized as follows.

- We build a dataset from 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 used the Haversine formula to accurately calculate 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 techniques to estimate delivery time and distance for each delivery person to each customer. This is key because 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 used the Munkres Algorithm with the cost matrices obtained from the Haversine calculations, as well as the linear and polynomial regression methods. The Munkres 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 compared the effectiveness of the Haversine calculations, linear regression, and polynomial regression techniques after applying the Munkres optimization algorithm to solve the task assignment problem.

2. Materials and Methods

In this work, we propose the architecture presented in Figure 1, which is based on machine learning and optimization methods to improve the delivery assignment task. For this, we first create a dataset based on delivery company information called MAX Delivery from Barcelona, Spain. Then, to create a cost function to be optimized, we compared heuristics and Machine Learning methods based on Linear Regression and Polynomial Regression to model distance and time metrics. Once we estimate the cost functions, we solve the task assignment delivery problem by using the Munkres Algorithm (also known as the Hungarian method). Finally, we compare, evaluate, and present the task assignment delivery results.

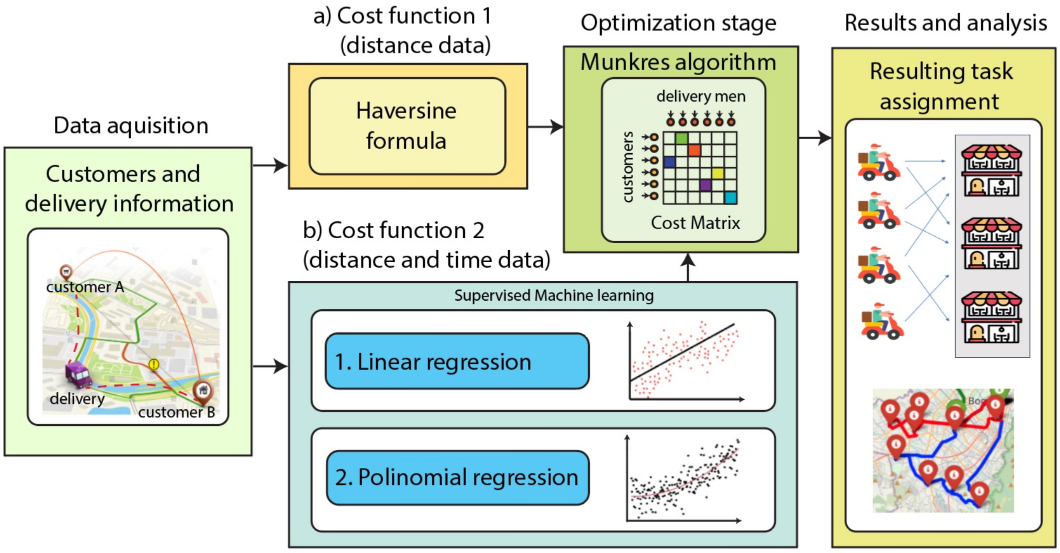


Figure 1. Proposed architecture for enhancing delivery assignments through Supervised Machine Learning regression techniques and the Munkres algorithm.

2.1. Data Acquisition

For our data acquisition procedure, we sourced 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 was collected between January 1, 2023, and June 22, 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's important to note that our 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 time is more complex due to the absence of speed, trajectory, or traffic information. Consequently, any time calculations are estimations are derived from known information in our dataset using regression methods. Specifically, we utilize linear and polynomial regression to estimate the missing data in the assignment matrix 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 subsequent sections.

2.2. Haversine Formula to Estimate Cost Matrix

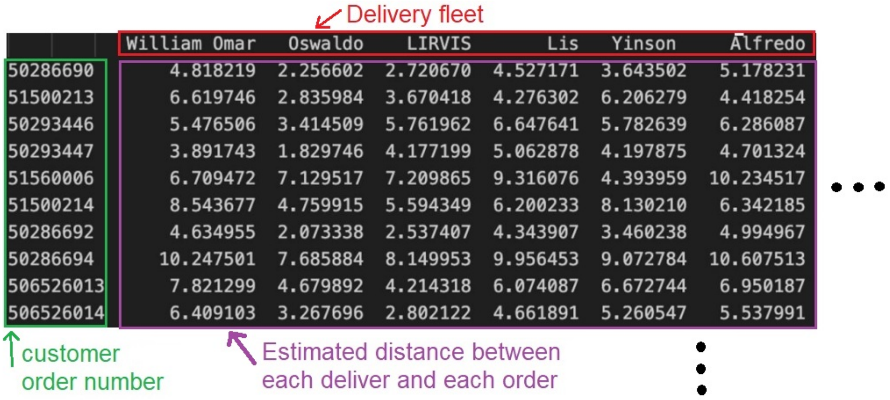
The main objective of this process is to estimate the distance between the delivery people and the customer orders and generate an assignment matrix to solve the optimization problem related to the task assignment problem. To accomplish this, we calculated the distances between the delivery people and the customer orders by using the Haversine formula [16,17]. This formula utilizes the latitude and longitude information of each member of the delivery people's fleet and the current batch of customers 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.

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

Then, we calculate the distance d by using Equation 2 using the central angle and the Earth's radius (R), typically taken as approximately 6,371 kilometers (or 3,959 miles).

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

The resulting value for d represents the distance between the two points along the surface of the Earth. We used this method because it is commonly used in geographic applications and navigation systems to accurately measure distances between locations. 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.

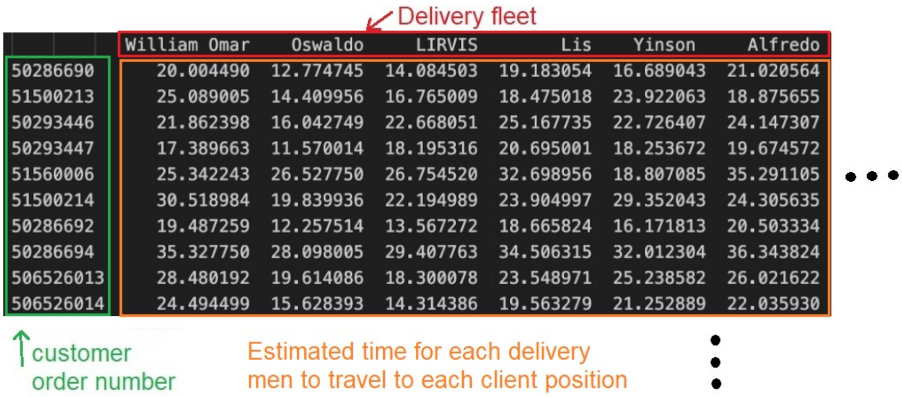


	William Omar	Oswaldo	LIRVIS	Lis	Yinson	Alfredo
50286690	4.818219	2.256602	2.720670	4.527171	3.643502	5.178231
51500213	6.619746	2.835984	3.670418	4.276302	6.206279	4.418254
50293446	5.476506	3.414509	5.761962	6.647641	5.782639	6.286087
50293447	3.891743	1.829746	4.177199	5.062878	4.197875	4.701324
51560006	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 used is composed of 7707 possible deliveries and 7707 customer orders.

2.3. Machine Learning to Estimate 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. Since 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 as follows. 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
```

Delivery fleet
customer order number
Estimated time for each delivery men to travel to each client position

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

2.3.1. Linear Regression

Linear regression is a supervised machine learning method used to model the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the input variables and the output [18,19]. The Equation 3 represents a multiple linear regression with n independent variables.

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n \tag{3}$$

where y is the dependent variable, $x_1, x_2, \ldots, x_n$ are the independent variables, and $b_1, b_2, \ldots, b_n$ are the coefficients.

In the context of linear regression, the primary goal is to find the best-fitting line (or hyperplane in higher dimensions) that minimizes the error between the predicted values and the actual values in the dataset [18,19]. This is often done by minimizing a cost function, such as the Mean Squared Error (MSE), which quantifies the difference between the predicted values and the actual values [18,19]. The cost function equation measures the error between predicted and actual values as defined in Equation 4.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{4}$$

where $J(\theta)$ represent the cost function to be optimized, m is the number of training examples, $h_\theta(x^{(i)})$ represent the predicted value for the i-th training example, and $y^{(i)}$ is the actual value for the i-th training example.

Finally, Gradient descent is used to update the parameters to minimize the cost function $J(\theta)$. Gradient descent is a powerful optimization algorithm that effectively addresses the optimization task in linear regression by efficiently searching for the optimal parameters that minimize the cost function $J(\theta)$ [18,19]. The equation that represents the update rule of the Gradient descent algorithm can be observed in Equation 5.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{5}$$

where α is the learning rate, and the expression $\frac{\partial}{\partial \theta_j} J(\theta)$ represents the partial derivative of the cost function concerning the j-th parameter. This equation iteratively updates the parameters θ_j in the direction of the steepest descent of the cost function. The gradient $\frac{\partial}{\partial \theta_j} J(\theta)$ indicates the direction and magnitude of the change needed to minimize the cost function. The learning rate α scales this change to control the step size.

2.3.2. Polynomial Regression

Polynomial regression is a form of regression in which the relationship between the independent variables x and the dependent variable y is modeled as an n-degree polynomial [20,21]. Polynomial regression allows for a more flexible model than linear regression, as it can capture non-linear relationships between variables. However, higher-degree polynomials can lead to overfitting if not carefully controlled. The general form of a polynomial regression equation is:

$$y = b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n \tag{6}$$

where, y is the dependent variable, x is the independent variable, $b_0, b_1, b_2, \ldots, b_n$ are the coefficients, and n is the degree of the polynomial. In this work, we propose the use of a quadratic polynomial regression (degree 2) as can be observed in Equation 7.

$$y = b_0 + b_1 x + b_2 x^2 \tag{7}$$

In polynomial regression, as in linear regression, the aim remains to minimize the cost function, typically the MSE (see Equation 4), through gradient descent (see Equation 5). However, polynomial regression introduces higher-degree polynomial terms into the hypothesis function, leading to a more complex equation for both the cost function and the gradient descent update rule. Despite these differences, the core principles of minimizing the cost function iteratively using gradient descent remain the same as indicated in Equations 4 and 5, with the mentioned adjustments to accommodate the polynomial features in the hypothesis function.

2.4. Munkres Algorithm

To effectively assign orders to delivery people while maximizing assignment efficiency, we propose the use of the Munkres Algorithm [11,22,23]. This method is also known as the Hungarian algorithm, and it 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 [11,22,23]. The Munkres 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 Munkres 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 8.

$$\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \tag{8}$$

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

The Munkres 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 11.

$$\text{Minimize each row: } \text{row}_i = \text{row}_i - \min(\text{row}_i) \tag{9}$$

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 10.

$$\text{Minimize each column: } \text{column}_j = \text{column}_j - \min(\text{column}_j) \tag{10}$$

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

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

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 [11,22,23]. 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 12.

$$C(A) = \sum_{i=1}^{n} c_{ij} \cdot a_{ij} \tag{12}$$

These equations represent the core operations of the Munkres Algorithm, which iteratively reduces costs, updates the assignment matrix, and finds the optimal assignment by satisfying the optimality 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, thus 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 results 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 used to fill the cost matrix that represents the cost function. In the case of distance estimation, polynomial regression outperforms linear regression, yielding a lower mean squared error (MSE) of 2.6 compared to 6.9 for linear regression. This translates to a substantially lower Root Mean Squared Error (RMSE) of 4.5 for polynomial regression, indicating greater accuracy in predicting distances. Similarly, for time estimation, polynomial regression achieves superior results with an MSE of 59.6 compared to 120.9 for linear regression. Consequently, the RMSE for time estimation with polynomial regression is notably lower at 7.7, showcasing its enhanced precision in predicting time durations compared to the RMSE of 11.0 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 (MSE) and Root Mean squared error (RMSE)

	Mean squared error (MSE)	Root Mean squared error (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 detail the performance of the proposed methods in estimating distances traveled by the delivery people after solving the task assignment problem with the Munkres algorithm. The Munkres method yields an average distance estimation of 4.55 kilometers, with a standard deviation of 3.24 kilometers, resulting in a total summation of 34956.84 kilometers. Interestingly, when combining linear regression with the Munkres method, the average estimation remains consistent at 4.55 kilometers, with a slightly reduced standard deviation of 3.23 kilometers, contributing to a marginally increased total summation of 34957.74 kilometers. Conversely, employing polynomial regression in conjunction with the Munkres method leads to a slightly higher average distance estimation of 4.62 kilometers, accompanied by a larger standard deviation of 4.49 kilometers, resulting in a total summation of 35526.26 kilometers. These findings suggest that, while both linear and polynomial regression exhibit similar average distance estimations when combined with the Munkres method, polynomial regression introduces greater variability in distance estimations, as indicated by its higher standard deviation. Notably, considering that less distance is better, the Munkres method and the linear regression combined with the Munkres 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 to solve the task assignment problem by using distance metrics at the cost matrix.

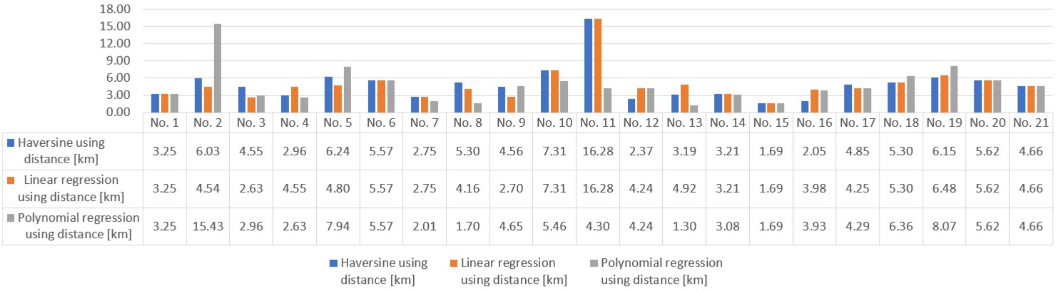
Method	Average [km]	Standard Deviation [km]	Summation of total distance [km]
Munkres	4.55	3.24	34956.84
Linear regression + Munkres	4.55	3.23	34957.74
Polynomial regression + Munkres	4.62	4.49	35526.26

As can be observed in Table 3, for the Linear Regression method the average time estimation is 19.24 minutes, accompanied by a standard deviation of 9.12 minutes, resulting in a total summation of 147891.23 minutes. Conversely, the Polynomial Regression method yields a notably lower average time estimation of 17.01 minutes, with a reduced standard deviation of 5.25 minutes, contributing to a shorter total summation of 130747.82 minutes. These results indicate that Polynomial Regression offers more precise estimations with less variability compared to Linear Regression when we solve the task assignment problem with the Munkres method.

Table 3. Comparison results of the proposed methods to solve the task assignment problem by using distance metrics at the cost matrix.

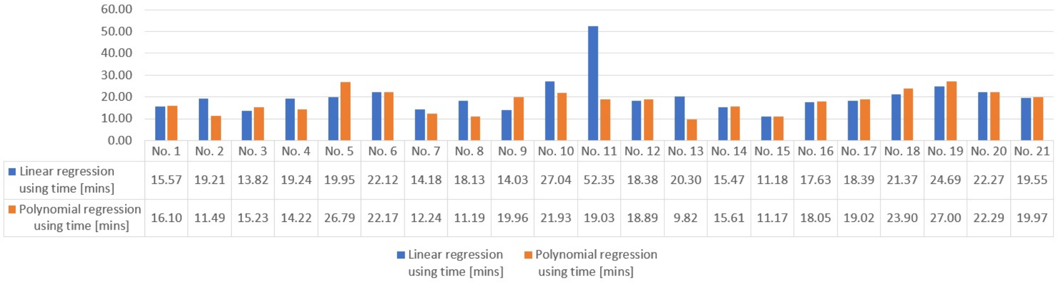
	Average [minutes]	Standard Deviation [minutes]	Summation of total time [minutes]
Linear Regression + Munkres	19.24	9.12	147891.23
Polynomial Regression + Munkres	17.01	5.25	130747.82

Finally, we present a sample of the results of the automatic assignment of tasks using the Munkres algorithm and the different methods of estimating the cost function in Figures 4 and 5. 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 Harveysine 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. In this case, 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 when distance estimation is utilized for our dataset, whereas polynomial regression methods excel in scenarios where time estimation is paramount.



	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10	No. 11	No. 12	No. 13	No. 14	No. 15	No. 16	No. 17	No. 18	No. 19	No. 20	No. 21
■ Haversine using distance [km]	3.25	6.03	4.55	2.96	6.24	5.57	2.75	5.30	4.56	7.31	16.28	2.37	3.19	3.21	1.69	2.05	4.85	5.30	6.15	5.62	4.66
■ Linear regression using distance [km]	3.25	4.54	2.63	4.55	4.80	5.57	2.75	4.16	2.70	7.31	16.28	4.24	4.92	3.21	1.69	3.98	4.25	5.30	6.48	5.62	4.66
■ Polynomial regression using distance [km]	3.25	15.43	2.96	2.63	7.94	5.57	2.01	1.70	4.65	5.46	4.30	4.24	1.30	3.08	1.69	3.93	4.29	6.36	8.07	5.62	4.66

Figure 4. Distance calculation after optimization procedure for a sample of 21 delivery people. a) Total Munkres distance 103.89 [km], b) Total Munkres distance 102.87 [km], c) Total Munkres distance 109.14 [km].



	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10	No. 11	No. 12	No. 13	No. 14	No. 15	No. 16	No. 17	No. 18	No. 19	No. 20	No. 21
■ Linear regression using time [mins]	15.57	19.21	13.82	19.24	19.95	22.12	14.18	18.13	14.03	27.04	52.35	18.38	20.30	15.47	11.18	17.63	18.39	21.37	24.69	22.27	19.55
■ Polynomial regression using time [mins]	16.10	11.49	15.23	14.22	26.79	22.17	12.24	11.19	19.96	21.93	19.03	18.89	9.82	15.61	11.17	18.05	19.02	23.90	27.00	22.29	19.97

Figure 5. Time calculation after optimization procedure for a sample of 21 delivery people. a) Total time using linear regression of 424.87 [min], b) Total time using linear regression of 376.06 [min].

4. 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 Munkres 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 people information. Results show that linear regression reduces estimation errors in the distance by up to 568.52 km (1.51%) compared to other methods, while polynomial regression reduces errors in time estimation by up to 17,143.41 minutes (11.59%). The proposed smart delivery system aims to enhance delivery personnel allocation within the delivery sector, optimizing efficiency.

Author Contributions: Conceptualization, Juan Vásconez and Elias Schotborgh; Data curation, Juan Vásconez, Elias Schotborgh and Viviana Moya; Formal analysis, Juan Vásconez, Elias Schotborgh, Ingrid Vásconez, Andrea Pilco, Oswaldo Menéndez, Robert Guamán-Rivera and Leonardo Guevara; Funding acquisition, Juan Vásconez and Ingrid Vásconez; Investigation, Juan Vásconez, Elias Schotborgh, Ingrid Vásconez, Viviana Moya, Andrea Pilco, Oswaldo Menéndez, Robert Guamán-Rivera and Leonardo Guevara; Methodology, Juan Vásconez and Elias Schotborgh; Project administration, Juan Vásconez; Resources, Juan Vásconez, Elias Schotborgh, Ingrid Vásconez, Viviana Moya, Andrea Pilco, Oswaldo Menéndez, Robert Guamán-Rivera and Leonardo Guevara; Supervision, Juan Vásconez, Ingrid Vásconez, Oswaldo Menéndez, Robert Guamán-Rivera and Leonardo Guevara; Visualization, Juan Vásconez, Elias Schotborgh and Ingrid Vásconez; Writing – original draft, Juan Vásconez, Elias Schotborgh, Ingrid Vásconez, Viviana Moya, Andrea Pilco, Oswaldo Menéndez, Robert Guamán-Rivera and Leonardo Guevara; Writing – review & editing, Juan Vásconez, Ingrid Vásconez, Viviana Moya, Andrea Pilco, Oswaldo Menéndez, Robert Guamán-Rivera and Leonardo Guevara.

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.

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.
2. 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.
3. Bruni, M.E.; Fadda, E.; Fedorov, S.; Perboli, G. A machine learning optimization approach for last-mile delivery and third-party logistics. *Computers & Operations Research* **2023**, *157*, 106262.
4. Reis, J.; Amorim, M.; Cohen, Y.; Rodrigues, M. Artificial intelligence in service delivery systems: a systematic literature review. *Trends and Innovations in Information Systems and Technologies: Volume 1 8* **2020**, pp. 222–233.
5. Gursoy, D.; Chi, O.H.; Lu, L.; Nunkoo, R. Consumers acceptance of artificially intelligent (AI) device use in service delivery. *International Journal of Information Management* **2019**, *49*, 157–169.
6. 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.
7. 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. *Knowledge-Based Systems* **2022**, *243*, 108489.
8. 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. *Advanced Engineering Informatics* **2022**, *51*, 101536.
9. Asih, A.M.S.; Sopha, B.M.; Kriptaniadewa, G. Comparison study of metaheuristics: Empirical application of delivery problems. *International Journal of Engineering Business Management* **2017**, *9*, 1847979017743603.
10. 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. *International Journal of Production Research* **2021**, *59*, 2229–2249.
11. Joshi, M.; Singh, A.; Ranu, S.; Bagchi, A.; Karia, P.; Kala, P. Batching and matching for food delivery in dynamic road networks. 2021 IEEE 37th international conference on data engineering (ICDE). IEEE, 2021, pp. 2099–2104.
12. Joshi, M.; Singh, A.; Ranu, S.; Bagchi, A.; Karia, P.; Kala, P. FoodMatch: Batching and matching for food delivery in dynamic road networks. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* **2022**, *8*, 1–25.
13. 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. International Conference on Intelligent Computing. Springer, 2021, pp. 340–354.
14. 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*.
15. Nair, A.; Yadav, R.; Gupta, A.; Chakraborty, A.; Ranu, S.; Bagchi, A. Gigs with guarantees: Achieving fair wage for food delivery workers. *arXiv preprint arXiv:2205.03530* **2022**.
16. Robusto, C.C. The cosine-haversine formula. *The American Mathematical Monthly* **1957**, *64*, 38–40.

17. Basyir, M.; Nasir, M.; Suryati, S.; Mellyssa, W. Determination of nearest emergency service office using haversine formula based on android platform. *EMITTER International Journal of Engineering Technology* **2017**, *5*, 270–278.
18. Maulud, D.; Abdulazeez, A.M. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends* **2020**, *1*, 140–147.
19. Hope, T.M. Linear regression. In *Machine Learning*; Elsevier, 2020; pp. 67–81.
20. Heiberger, R.M.; Neuwirth, E.; Heiberger, R.M.; Neuwirth, E. Polynomial regression. *R Through Excel: A Spreadsheet Interface for Statistics, Data Analysis, and Graphics* **2009**, pp. 269–284.
21. Avery, M. Literature review for local polynomial regression. *Unpublished manuscript* **2013**.
22. Shah, K.; Reddy, P.; Vairamuthu, S. Improvement in Hungarian algorithm for assignment problem. Artificial Intelligence and Evolutionary Algorithms in Engineering Systems: Proceedings of ICAEES 2014, Volume 1. Springer, 2015, pp. 1–8.
23. 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. *Solar Energy* **2015**, *122*, 347–358.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.