

Article

Optimal Route Crowd-Shipping System for Sustainable Rapid Delivery: Algorithm, Simulation, and Feasibility Evaluation

Lior Aronshtam , Benny Sand , Tammar Shrot , Ruth Cohen, Chaya Levin and Hadassa Daltrophe 

Department of Software Engineering, Shamoon College of Engineering (SCE), Ashdod 77245, Israel; bnayasa@sce.ac.il (B.S.); tammash@sce.ac.il (T.S.); hadasda1@sce.ac.il (H.D.)

* Correspondence: liorar1@sce.ac.il

Abstract: Delivery systems are ubiquitous in today's economy. However, those systems usually operate through purpose-built vehicles, which are inefficient, expensive, and highly harmful to the environment. We propose an optimal route crowd-shipping (OR-CS) system, a delivery service based on crowd transportation. The OR-CS system utilizes service points (SPs) and occasional couriers (OCs) to transfer deliveries. Senders drop packages at SPs, while receivers collect them from different SPs. The system is based on a new algorithm that assigns to each package an optimal route. The route is chosen individually for the package and is personally tailored according to the sender's preferences and to the predefined routes of the OCs. To assess the real-life feasibility of the system, we developed a general simulator that emulates a city environment with authentic service points specifically selected based on desired attributes. The routes of OCs and the origins and destinations of packages were generated using a random process that differed between simulations. The results indicate that the system can be implemented and utilized. In addition, it yields positive results when the number of OCs surpasses a minimum threshold, which is feasible in most cities, given existing traffic loads. By adopting OR-CS, we can expect lowered delivery costs, reduced traffic congestion, and enhanced environmental sustainability.

Keywords: crowd shipping; delivery; public mobility; occasional courier; sustainable packages transit; traffic congestion



Citation: Aronshtam, L.; Sand, B.; Shrot, T.; Cohen, R.; Levin, C.; Daltrophe, H. Optimal Route Crowd-Shipping System for Sustainable Rapid Delivery: Algorithm, Simulation, and Feasibility Evaluation. *Future Transp.* **2024**, *4*, 1–14. <https://doi.org/10.3390/futuretransp4010001>

Academic Editors: Ouri E. Wolfson, Kum Fai Yuen and Shunde Yin

Received: 19 September 2023
Revised: 6 December 2023
Accepted: 25 December 2023
Published: 27 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today's economy relies heavily on the ability to deliver products to consumers [1]. Over \$5 trillion was made in global eCommerce sales in 2022, and it is expected that sales will reach \$6.31 trillion in 2023 (<https://www.insiderintelligence.com/content/worldwide-e-commerce-forecast-update-2022>, accessed on 16 November 2023). However, delivery systems such as those used by USPS, FedEx, UPS, DHL, or smaller local companies are usually built around vehicles tailored to this purpose. Furthermore, those vehicles are typically not loaded to their full capacity—on average, they are less than 50% full. Additionally, trucks may drive back empty after making a delivery [2]. This situation increases traffic congestion, which is harmful to the environment and to the population. Concurrently, traffic contains other vehicles passing the delivery route for various reasons. The drivers of these vehicles might be interested in serving as occasional couriers (OC) to deliver the packages for a reward. This approach, known as crowd-shipping, will reduce traffic congestion, cause a substantial reduction in carbon emissions and environmental impact, and decrease transit costs.

The utilization of crowd-shipping can be done in several ways; one of them is using a network of service points (SPs) (e.g., gas stations, local groceries or automated parcel machines). In detail, the deliveries are made by OCs who input their route into the system. The OCs move packages from one SP to another along their predefined route. The sender dispatches a package in an SP, and the receiver ultimately picks it up at an SP close to the agreed destination.

Various studies and implementations have recently been proposed for crowd-shipping delivery systems to solve diversity challenges in various fields, including technology, engineering, security, economy, marketing, and others. This manuscript focuses on the core algorithmic problem of route planning optimization.

Several crowd-shipping delivery systems that optimize the routes for packages were offered [3–5]. These systems consider factors such as time or distance and optimize them for all packages accordingly. However, from the sender's point of view, the optimal route for a package relies on different parameters, each of which has a different priority. One is the *time* it takes for the package to arrive at its destination. The second is the *distance* that the package travels, which affects the price of the delivery (for example [4]). Another parameter is the *amount of OCs* that are involved in the delivery, since as this amount increases, so does the probability of the package getting damaged or lost. Each sender has different preferences, and the prioritization of these parameters varies accordingly. In this paper, we establish a personalized algorithm that finds the optimal route for each package according to the sender's prioritization and its strictness. In addition, we implement the algorithm into an optimal route crowd-shipping (OR-CS) system. To the best of our knowledge, this is the first system that considers each sender's desires individually.

To establish a proof of feasibility, we implemented a general simulator that can be run in any city. We focused on a real-world city, Ashdod, and chose a set of SPs according to a predefined set of attributes. A set of OCs was generated, each with a corresponding route and departure time. The routes and times were generated using a random model. The travel time of the different OCs was calculated while considering the expected time. The random model varied between the different simulations. Every simulation also consisted of a set of deliveries (i.e., origin and destination SP and time) that were generated using a random model.

Running several different simulations yielded promising results, indicating that the OR-CS is feasible and can be effectively used in real-world settings. The simulations have provided evidence that the system can be successfully utilized with positive results. These findings highlight the potential for practical implementation and suggest that the OR-CS can perform effectively, enhancing the sustainability of logistics by reducing the number of vehicles on the roads, and thus should become part of future transportation.

To summarize, the main contributions of this paper are (i) Creating a new crowd-shipping system with a personalized algorithm that finds an optimal route for each package according to the sender's preference. (ii) Building a general simulator to check the system's performance. (iii) Running simulations on the city of Ashdod and analyzing the results.

The rest of the paper is built as follows: Section 2 reviews recent studies in this field. Section 3 describes the new algorithm and the OR-CS system. Section 4 describes the general simulator we built, and Section 5 shows the results of the simulation run on the city of Ashdod. Finally, Section 6 has a discussion and conclusions.

2. Literature Review

The remarkable triumph in the last decade of sharing platforms like Uber, Lyft, and Airbnb, which operate without actually possessing their assets; instead, they facilitate connections between owners and users. This has further fueled the desire to apply the sharing economy concept to delivery services [6].

Recent studies report on the willingness of consumers from developing countries [7] to use the pick-up point structures and the viability of implementing them. Case studies conducted in Greece [3] and Italy [8] demonstrate that crowd-shipping and public transport crowd-shipping can alleviate the challenges associated with last-mile deliveries, significantly enhancing the system's overall performance. Further, a literature review [9] that related the factors within mobile applications that encourage sustainable choices found that a well-designed approach incorporating financial incentives and environmental consciousness could boost public transport usage. Alternatively, the shipping mode may include electric bicycles [10,11] as a solution for urban last-mile deliveries.

The optimization aspects of crowd-shipping were studied from different perspectives. Ref. [12] focuses on the problem of matching crowd-shipping supply and demand, presenting a procedure that maximizes the number of matched shipments while considering the employees' minimum expected earnings per time unit, this problem was also considered in [13]. In [14], the authors proposed mechanisms to achieve outcome prediction in crowd-sourcing, while the authors of [15] suggest agile optimization to handle problems related to routing time and distance or to reducing energy consumption. In [16], they address energy optimization and constraints using linear programming. In [17], a closely related problem is considered, and the goal is to optimize total travel time and costs.

The ultimate crowd-shipping solution to deliver packages is based on OCs, as assumed in many studies [5,18,19]. However, employing company couriers as a backup is also suggested, so there is a need to refer to reducing the costs of company messengers [20–23]. In [3], the system assigns each package to one OC. The system in [5] assigns the packages to several OCs; however, the route is determined not according to the OC's routs. In [4], the system also assigns the packages to several OCs; it tries to maximize profit while fulfilling constraints using mixed integer programming.

Relating to the SPs, some studies [20,21] assume carriage directly between origin and destination, whereas others [5,18,23] suggest that different drivers would unload the package at an SP along the route, and there it would be stored until the next driver loads it. Alternatively, the drivers meet at intermediate destinations at synchronized times to pass the package from hand to hand [22,24]. Choosing the locations of the SPs is close to choosing the locations of centers in different supply chains (for example [25]).

What sets our work apart is its unique approach, providing a flexible solution that caters to the preferences set by the package sender. Namely, depending on the prioritization of time distance and amount of OCs, the algorithm shown in this paper finds an optimal route for the package. Moreover, we showcase the feasibility of this solution by presenting a simulator capable of testing diverse scenarios using real-world data.

3. The OR-CS System Description

The OR-CS system's goal is to use OCs to transfer packages between SPs along the delivery routes. The system contains a set $P = \{p_1, \dots, p_n\}$ of n SPs that are chosen at the initialization from a predefined region. A location is considered an SP if it has an appropriate place to park, load, and unload OC vehicles safely. Places that fit these criteria include gas stations and parking lots. Obviously, the set P can change from time to time.

The OR-CS system has known time intervals; sometimes, an interval may extend to a day or even more. Each individual who wishes to be an OC can register in the OR-CS before the beginning of the time interval. During the registration process, the OC must declare the start time and route associated with the future journey (i.e., the OC registrant must identify, order, and sequence of SPs that will be included in the journey). Namely, the SPs that will be visited on the journey and their order. The OR-CS calculates the approximate time of arrival to each SP. Hence, for each time interval, the system also contains a set $C = \{c_1, \dots, c_m\}$ of m OCs, a set $R = \{r_{c_1}, \dots, r_{c_m}\}$ of m routes for each $c_i \in C$, and a set $T = \{t_{c_1}, \dots, t_{c_m}\}$ of starting times for each OC.

The OR-CS also has a defined bound on the maximal time allowed for a package to be delivered, denoted by U_t . A package sender allocates an origin SP and a destination SP for the package and the time the package will be placed in the origin SP, t_o . Each sender must prioritize the following parameters concerning the package route from its origin SP to its destination SP: (i) *Time*: the amount of time it takes the package to travel the route. (ii) *Distance*: the length of the route. (iii) *#OC*: the total number of OCs that carried the package through the route. The motivation for taking the third parameter into account is the assumption that as the number of OCs that deliver a package increases, so does the probability of the package being damaged or lost. The sender also chooses how strict the prioritization should be.

At the beginning of each time interval, the OR-CS creates a TIMEEXPANDEDGRAPH. For every new package during this time interval, the OR-CS defines different weights over the edges of the TIMEEXPANDEDGRAPH, according to the sender's preferences. Using the weighted TIMEEXPANDEDGRAPH, the OR-CS runs an algorithm that chooses the route the package will follow and the OCs that will deliver it. Following, we describe how we build the TIMEEXPANDEDGRAPH (Algorithm 1) and how the optimal route is found (Algorithm 2) for a given package.

3.1. The TIMEEXPANDEDGRAPH

A TIMEEXPANDEDGRAPH $G = (V, E)$ is a directed graph of a type suggested in [26]. Each vertex $v \in V$ in the graph is a triplet $v = (c, p, t)$ in which $c \in C$ is an OC, $p \in P$ is a specific SP, and t is a time. A triplet is a vertex in the graph iff OC c reaches SP p in time t .

The following definitions are used to describe the set of edges. The LOADINGTIME is the maximum time it takes to load or unload a package. For each OC, c , $route(c)$ represents the entire series of vertices of the form, (c, p_i, t_i) , ordered by time.

The set of edges contains two types: (i) *Station edges*: these edges represent the fact that a certain package that was dropped off in an SP can be later picked up by some OC that reaches this SP. Formally, there exists a station edge from vertex (c, p, t) to vertex (c', p', t') iff $p = p'$ and $t' \geq t + \text{LOADINGTIME}$. (ii) *Route edges*: an edge between a vertex and its successor in the route. Notice that route edges connect only vertices associated with the same OC (see an example of the edges in Figure 1).

Algorithm 1: Build TIMEEXPANDEDGRAPH

Input: $P = \{p_1, \dots, p_n\}$ a set of n SPs, $C = \{c_1, \dots, c_m\}$ a set of m OCs

$R = \{r_{c_1}, \dots, r_{c_m}\}$ a set of m routes for each $c_i \in C$

s.t. $r_{c_i} = \{p_j, \dots, p_k\}$ where $p_j, \dots, p_k \in P$

and a set $T = \{t_{c_1}, \dots, t_{c_m}\}$ of starting times for each OC.

Output: TIMEEXPANDEDGRAPH $G = (V, E)$

1. $V \leftarrow \emptyset, E \leftarrow \emptyset$

2. **For each** $r_{c_i} \in R$:

(a) **For each** $p \in r_{c_i}$:

$t \leftarrow$ calculate arriving time of c_i to p

$v \leftarrow$ a new **vertex** $v = (c_i, p, t)$

$V \leftarrow V \cup \{v\}$

(b) **For each** $p_j \in r_{c_i}$:

$e \leftarrow$ a new **route edge** $(c_i, p_j, t) \rightsquigarrow (c_i, p_{j+1}, t')$

$E \leftarrow E \cup \{e\}$

3. **For each** $v_i = (c, p, t), v_j = (c', p', t') \in V$:

(a) **If** $p = p' \ \& \ t' \geq t + \text{LOADINGTIME}$

$e \leftarrow$ a new **station edge** $(c_i, p_j, t) \rightsquigarrow (c_i, p_{j+1}, t')$

$E \leftarrow E \cup \{e\}$

Algorithm 2: Optimal Route

- Input:** A TIMEEXPANDEDGRAPH $G = (V, E)$, p_o, p_d, t_o
a prioritization $\langle x, y, z \rangle$ and upper bounds: U_x, U_y, U_z , accordingly
strictness parameters α, β .
- Output:** A route for the package
1. $\delta(z) \leftarrow 1$
 2. $\delta(y) \leftarrow U_z - \beta$
 3. $\delta(x) \leftarrow \delta(y)(U_y - \alpha)$
 4. **For each** station edge $e = ((c, p, t), (c', p', t')) \in E$:
 $w(e) \leftarrow \delta(\#OC) + \delta(\text{time}) \cdot (t' - t)$
 5. **For each** route edge $e = ((c, p, t), (c, p', t')) \in E$:
 $w(e) \leftarrow \delta(\text{time}) \cdot (t' - t) + \delta(\text{distance}) \cdot \text{dist}(p, p')$
 6. $O \leftarrow$ new **vertex**, $D \leftarrow$ new **vertex**
 7. $V' \leftarrow V \cup \{O, D\}$ $E' \leftarrow E$
 8. **For each** $v = (c, p, t) \in V$:
 - (a) **If** $p = p_o$ & $t_o < t < t_o + U_t$
 $e \leftarrow$ a new **edge** $O \rightsquigarrow (c, p, t)$
 $w(e) \leftarrow \delta(\text{time}) \cdot (t - t_o)$
 $E' \leftarrow E \cup \{e\}$
 - (b) **If** $p = p_d$ & $t_o < t < t_o + U_t$
 $e \leftarrow$ a new **edge** $(c, p, t) \rightsquigarrow D$
 $w(e) \leftarrow 0$
 $E' \leftarrow E \cup \{e\}$
 9. Apply Dijkstra's algorithm on $G' = (V', E')$ with O as the source node.
 10. Return the route found for vertex D

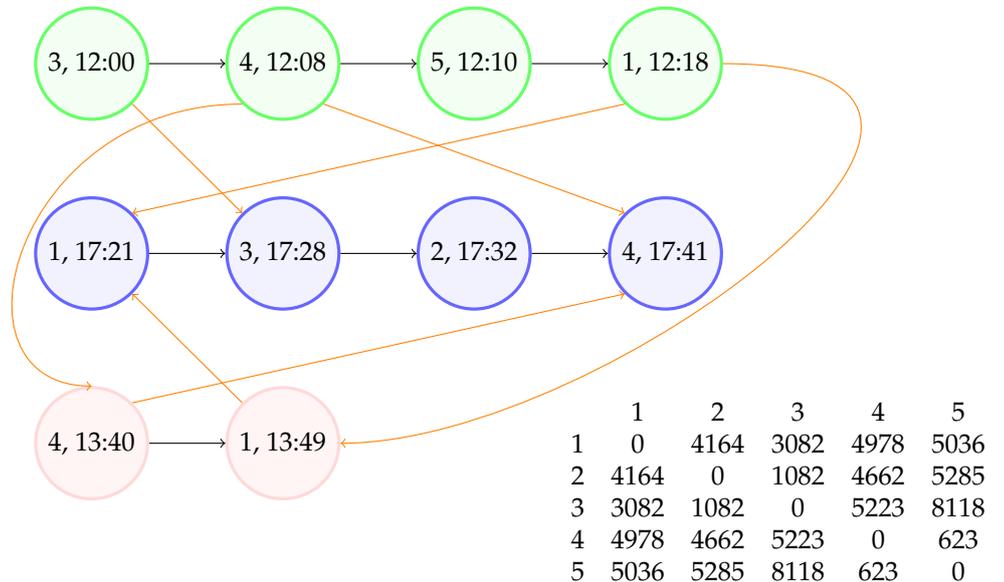


Figure 1. Example of a TIMEEXPANDEDGRAPH. The graph presents a scenario with five SPs $\{1, 2, 3, 4, 5\}$ and three drivers c_1 (marked green), c_2 (marked blue) and c_3 (marked pink). c_1 has a route $3 \rightsquigarrow 4 \rightsquigarrow 5 \rightsquigarrow 1$, which starts at 12:00 p.m. c_2 has a route $1 \rightsquigarrow 3 \rightsquigarrow 2 \rightsquigarrow 4$, which starts at 5:20 p.m. c_3 has a route $4 \rightsquigarrow 1$, which starts at 13:40 p.m. The distance matrix between the SP appears to the graph's right. The orange edges describe the *Station edges* while the black edges describe the *Route edges*.

3.2. The Optimal Route Algorithm

Given the TIMEEXPANDEDGRAPH, and a new package to deliver, the sender must choose what variables are most important regarding delivering the package: delivering time, overall distance traveled of the delivery, or number of OCs used to carry the package. The sender defines the prioritization order $\langle x, y, z \rangle$ (six possible permutations exist) and the strictness parameters α, β .

Recall that U_t denotes the upper bound over time, as described above. Let U_d denote the upper bound of a considerable distance for a delivery route of a package. Let $U_{oc} \leq m$ denote an upper bound of the number of OCs involved in the delivery of a package.

To establish the weight function, the unit used to measure each parameter must be normalized with the units of the other parameters. Namely, $\delta(\text{time})$ represents the weight that is added to the total weight by one unit of time, $\delta(\text{distance})$ represents the weight that is added by one unit of distance, and $\delta(\#\text{OC})$ the weight added by one OC.

To preserve a prioritization, $\langle x, y, z \rangle$, the following equations must hold:

$$\frac{\delta(x)}{\delta(y)} = U_y - \alpha \quad (1)$$

$$\frac{\delta(y)}{\delta(z)} = U_z - \beta \quad (2)$$

where $0 \leq \alpha \leq U_y, 0 \leq \beta \leq U_z$ are parameters that represent the strictness of the prioritization. In detail, if α and β equal 0, the prioritization is very strict; if these values are higher, prioritization is less strict.

To better understand, look at the following example: The upper bounds are: $U_t = 400$, $U_{oc} = 100$ and $U_d = 5000$. The prioritization given by the sender is in the following order: $\langle x, y, z \rangle = \langle \text{time}, \#\text{OC}, \text{distance} \rangle$, (i.e., $U_x = 400, U_y = 100, U_z = 5000$). If the sender defines α to be 0 then

$$\delta(\text{time}) = \delta(\#\text{OC}) \cdot 100.$$

Hence, any decrease in the time, even a small one, will be preferable even if it results in a considerable increase in the number of OCs. However, if we let $\alpha = 96$ then

$$\delta(\text{time}) = \delta(\#\text{OC}) \cdot 100 - \delta(\#\text{OC}) \cdot 96 = \delta(\#\text{OC}) \cdot 4$$

Namely, a decrease in a time unit is preferable only if it increases no more than 4 OCs. The weight of a station edge from (c, p, t) to (c', p, t') is

$$\delta(\#\text{OC}) + \delta(\text{time}) \cdot (t' - t).$$

The weight of a route edge from (c, p_i, t_i) to (c, p_{i+1}, t_{i+1}) is

$$\delta(\text{time}) \cdot (t_{i+1} - t_i) + \delta(\text{distance}) \cdot |p_{i+1} - p_i|,$$

where $|p_{i+1} - p_i|$ is the length of the shortest route between two SPs.

To determine the best route of a package in the weighted TIMEEXPANDEDGRAPH, we first add two special nodes to the graph—an origin node, O , and a destination node, D —as follows: let p_o be the origin SP of a package, p_d its destination SP, and t the time the package arrived at the origin SP. The origin node will have an edge to every node of the form $(_, p_o, t')$, where $t < t' < t + U_t$, and the weight of the edge is $\delta(\text{time}) \cdot (t' - t)$. The destination node has edges going into it from every node of the form $(_, p_d, t')$ where $t < t' < t + U_t$ and the weight of the edge is 0.

After adding the two nodes, we adjust the weights to comply with the sender's priorities. Finally, we apply Dijkstra's algorithm [27] to the graph with the origin node as the source node. The route the algorithm outputs from the origin node to the destination node will be the route of the package (See an example in Figure 2).

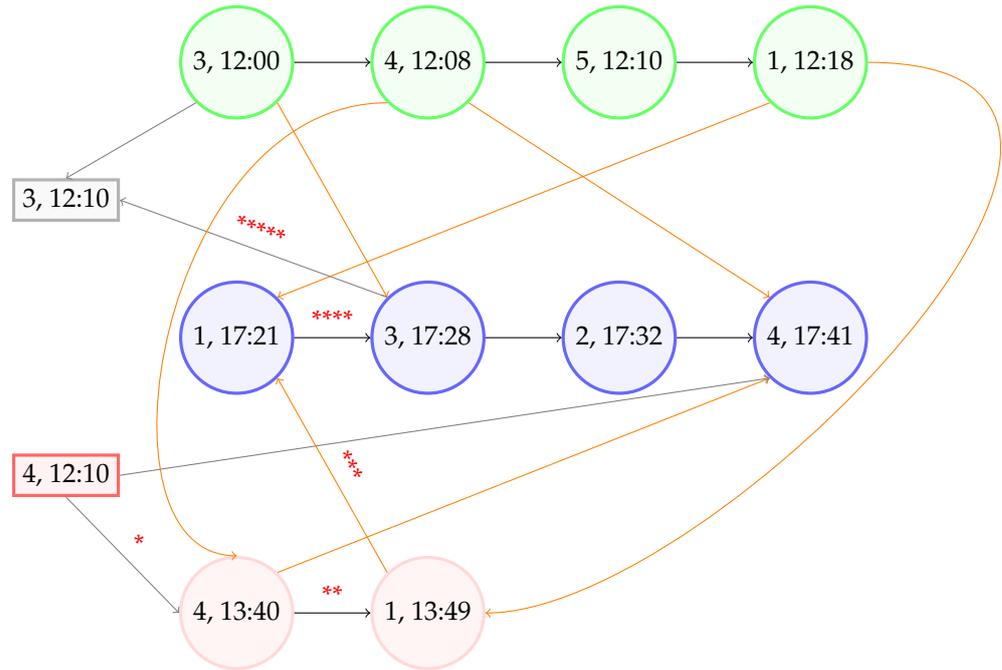


Figure 2. TIMEEXPANDEDGRAPH for a given package. Given the TIMEEXPANDEDGRAPH presented in Figure 1, and a package arriving at the origin SP: $p_o = 4$ at time $t_o = 12:10$, with destination SP: $p_d = 1$, and prioritization: $\langle distance, \#OC, time \rangle$. The rectangular vertices indicate the origin (red) and the destination (gray), while the gray edges are their respective temporary edges. The resulting route (output by Algorithm 2) is marked with (*) over the relevant edges.

4. Simulator

We implement the OR-CS system as a simulator to evaluate its feasibility (Our commitment to the open-source paradigm is exemplified by the availability of our codebase, accessible at [28], fostering collaborative knowledge sharing and community involvement). The simulator is a web app with an interface to feed a road map of the related region, chosen SPs within, and traffic load information for each route segment. The simulator allows tuning the number of drivers and parcels and displays the simulation results on a map at various speeds. In addition, the user should define the prioritization (time, driver, or distance) for each package (see Figure 3).

The simulations were conducted in the Israeli city of Ashdod. The simulator is fed with real traffic data gathered using Waze’s API [29]. The traffic data used can be found here [30]. The population of Ashdod is approximately 225,000 with a density of 4791 people per km^2 . To select suitable SPs, we evaluated locations based on their ability to provide safe parking and facilitate loading and unloading. This criterion included locations such as gas stations and parking lots. The SPs were chosen using commonly employed Geographic Information System (GIS) tools, with additional SPs being manually selected. In total, 70 SPs were designated for this purpose, as illustrated in Figure 4.

There was a need to calculate both the distance between every pair of SPs and the time it takes to travel from one to another. The travel time varies depending on the traffic, which usually depends on the time of day. Hence, the day was divided into 15 intervals that share the same estimated traffic load (In the simulator, the intervals parameter is tuneable). The estimated travel time was calculated for each interval using Waze’s API [29]. This information was retrieved and stored before running the simulations.



Figure 3. The interface of the simulator.

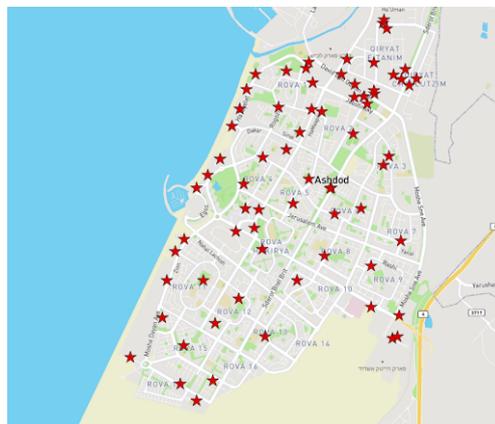


Figure 4. A map of Ashdod, the stars are the SPs chosen for the simulator.

There was also a need to create random routes for the OCs. For simplicity, the origin and destination points were chosen randomly from the set of SPs (the distribution varied between the simulation as described in Section 5). The optimal route for this journey, which does not consider the SPs, was retrieved via the Mapbox directions API. The next stage was finding SPs close to the route. To do so, first, all SPs that have an aerial separation distance of less than $10^{-3} - 3 \cdot 10^{-4}$ from the route were marked. Using a traveling salesperson optimization algorithm, with the origin, destination, and marked SPs as input, a new route was chosen (see an example in Figure 5).

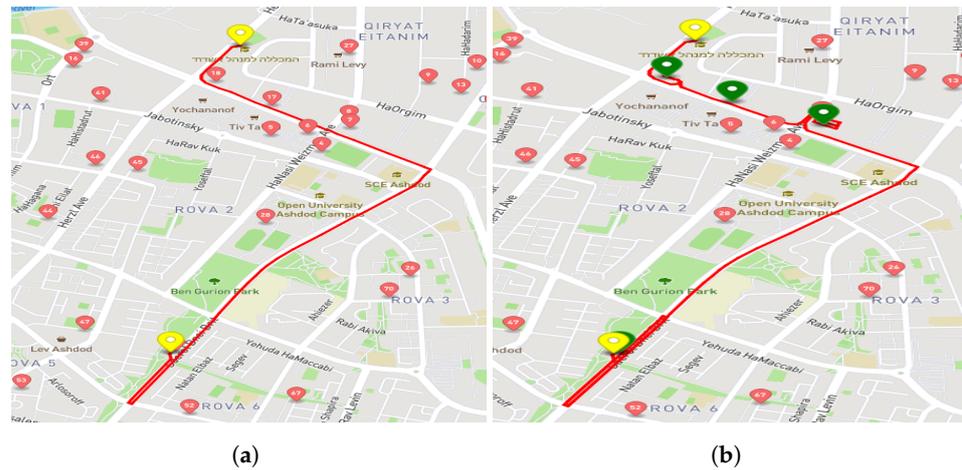


Figure 5. An example of the route the simulator created for an OC, after choosing the origin and destination SP randomly. (a) The optimal route between the origin and destination SPs. (b) The route connecting the origin and destination SPs after adding possible intermediate SPs.

5. Results

The model’s performance was assessed across various parameters through a series of experiments. In each set of experiments with fixed parameter values, a total of 50 random trials were conducted.

Initially, we evaluated the model’s performance by generating random routes, origin, and destination SPs, all with a uniform probability distribution. Our criteria for success were delivering a package within 24 h. As illustrated in Figure 6, the model exhibited a low success rate when dealing with a thousand packages while employing 100 OCs. However, when utilizing 500 OCs, the success rate significantly improved, reaching 90%. This trend was consistent even with smaller quantities of packages.

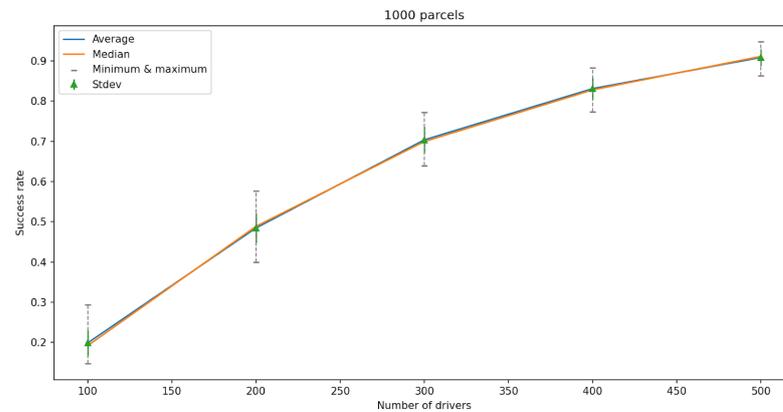


Figure 6. Success rates of delivering packages as a function of the number of OCs.

Interestingly, a reduction in the number of packages did not lead to a significant change in the required number of OCs to achieve a high success rate, as depicted in Figure 7.

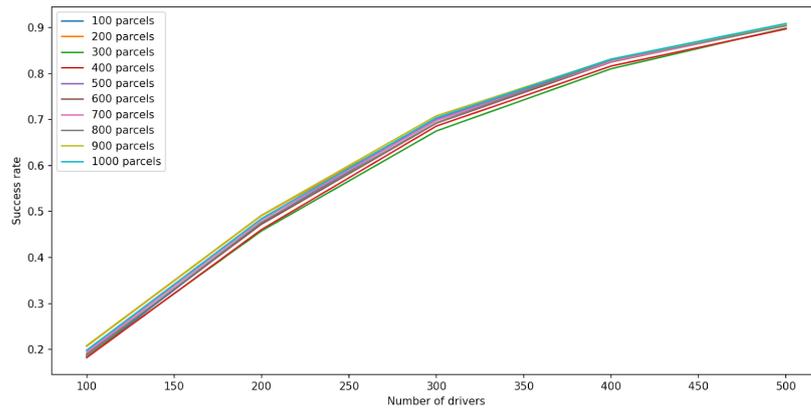


Figure 7. Success rates of delivering packages as a function of the number of OCs for different amounts of packages.

Furthermore, we explored the model’s performance in scenarios where OC routes were not generated uniformly. Specifically, for certain OCs, we exclusively selected origin or destination stations (or both) from a subset of 14 northern SPs out of the total 100 SPs (refer to Figure 8). The results of this analysis, conducted with 300 OCs and 1000 packages, are presented in Figure 9.

The findings indicate that using a uniform distribution yields the best results. Nevertheless, it’s intriguing to note that even when approximately 60% of both the origin and destination stations are exclusively chosen from the northern SPs, the success rate still surpasses the 50% mark.

This observation shows the model’s capability to deliver satisfactory results even when faced with a non-uniform distribution of origins and destinations.

We also investigated whether there is a correlation between the delivery price and the number of packages. To answer this question, we conducted a simulation with 300 OCs, where the priority of each package was randomly assigned using a uniform distribution from a set of six possible priorities (all possible prioritization of time, distance, and amount of OCs). For each package amount (ranging between 100 to 1000 in intervals of 100), we conducted 50 simulations and calculated their average. The results in Figure 10 indicate that until the number of packages reaches a threshold of around 600 there is a decrease in the average cost as the number of packages increases, and from this threshold, the average cost stays the same. However, the maximum cost does not show a similar decrease.

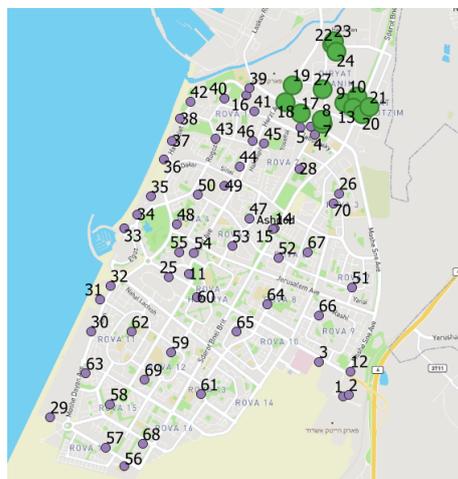


Figure 8. The 14 SPs located in the north are represented as green dots, while purple dots represent the remaining SPs.

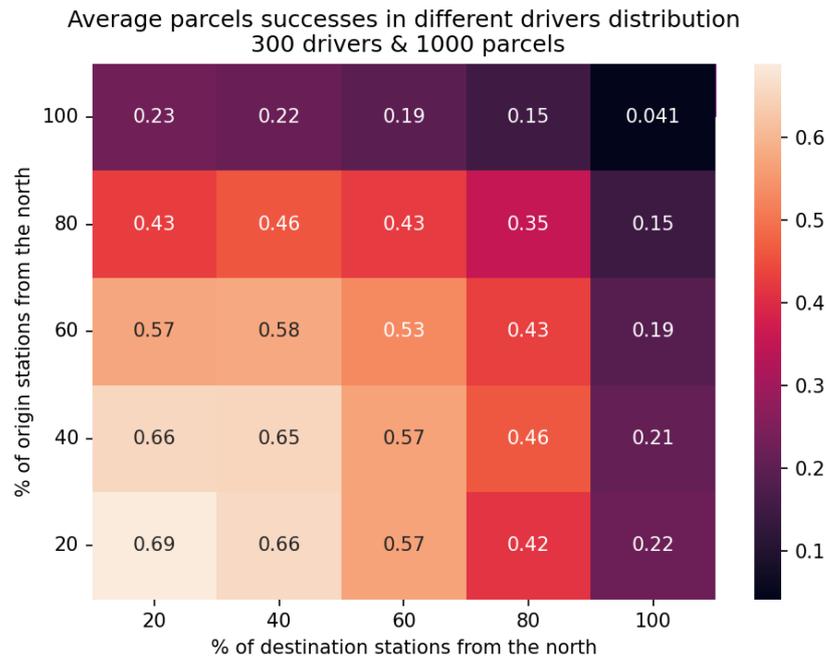


Figure 9. Success rates of delivering packages as a function of the number of OCs for different amounts of packages.

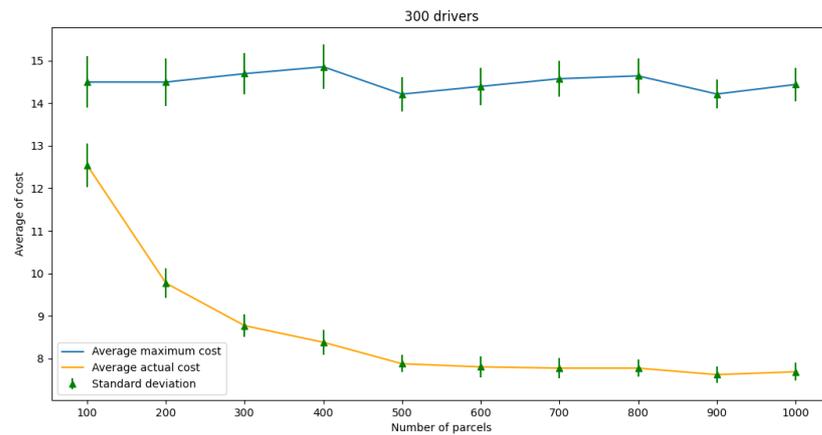


Figure 10. The effect of the number of packages on the payment of each package. The graph presents the average and the standard deviation of the maximum and the actual costs.

6. Discussion and Conclusions

Wasted resources in the transit system have become a growing concern in today’s world. The inefficient use of vehicles, leading to empty or underutilized vehicles, causes unnecessary fuel consumption, increased emissions, waste of resources, and financial burdens. Additionally, the lack of proper synchronization and coordination between different transit vehicles often leads to longer travel times, redundant routes, and overlapping services. This inefficiency wastes time and energy and contributes to congestion on roads. Moreover, the lack of technology integration prevents the optimization of routes and capacity management.

This work examines the possibility of addressing these issues by implementing a smart crowd transit solution that can reduce waste in the transit system by utilizing drivers who are already on the road as occasional carriers of goods. We believe such a system can improve resource utilization and be more sustainable.

In this study, we accomplished several significant tasks. Initially, we developed an optimal package routing algorithm. Within this algorithm, we introduced the concept of a

preference function, recognizing that time alone is not the sole parameter to consider. We introduced two significant additional parameters: travel distance and the number of OCs. Importantly, our algorithm is designed to accommodate the inclusion of more parameters in the future. Furthermore, we implemented the algorithm into a fully working system, OR-CS, and ensured that each sender's preference function is individually customized. This means that every sender benefits from a personalized adjustment to their route selection method, enhancing the overall effectiveness of the routing process.

Later, we created a general simulator that can check the performance of the OR-CS system on every city with any number of OCs and packages. Lastly, we conducted multiple simulations to assess the feasibility of the system. These simulations demonstrated the system's validity and ability to achieve the desired goal successfully. Namely, the simulations revealed that the system proved effective once the number of participating OCs surpassed a minimal threshold. It is worth noting that the threshold required was only 500 OCs, which is a relatively small number considering the average traffic load in medium or larger cities. As a result, achieving such numbers in a real-world setting should not pose a significant challenge. These findings highlight the potential for widespread adoption and successful system implementation in practical scenarios. Another interesting result is that the average cost per package decreases as the number of packages in the system increases. From a threshold of around 600 the average cost is steady and relatively low.

Currently, most transport deliveries are conducted by human-driven vehicles. The future of transportation could mean a less human approach, such as that supported by electric, self-driving cars, or drones. Although the OR-CS uses conservative means of transportation, it can also be used when using autonomous vehicles. The same vehicle can be efficiently used for driving people and multiple deliveries. Hence, a system that enables the utilization of vehicles for efficient deliveries will be needed.

The simulator described in this paper was built in a generic way that can be applied to every city. It would be interesting to apply it to different cities and see how the OR-CS performs under different cities' topologies, sizes, and traffic congestion.

In future work, we aim to enhance the applicability of our research to real-world settings by addressing several important aspects that were simplified in our current study. Specifically, focuses on accommodating the limited space available in SPs and OCs' vehicles and developing robust mechanisms to handle situations where drivers may change their minds after committing to picking up a package or deviating from their designated routes. Additionally, it will be beneficial to examine the practical way to combine OR-CS with automated parcel machines.

The proposed OR-CS system has several risks to be considered in future work. These include concerns about the security and trustworthiness of occasional couriers, liability in case of accidents or losses, potential privacy issues, variations in service quality, user adoption resistance, scalability limitations, regulatory and compliance hurdles, uncertain environmental impact, and safety concerns related to untrained or unvetted occasional couriers. Successful implementation will require careful planning, legal compliance, insurance coverage, background checks, and user education to mitigate these risks and ensure the system's viability.

These enhancements will enable the crowd-shipping system to operate seamlessly and effectively in noisy and dynamic real-world environments, offering practical solutions to the challenges faced in the field of logistics and transportation.

Author Contributions: Conceptualization, L.A., R.C. and C.L.; methodology, L.A., B.S., R.C., C.L., T.S. and H.D.; software, R.C. and C.L.; validation, R.C. and C.L.; formal analysis, L.A., B.S., R.C., C.L., T.S. and H.D.; investigation, L.A., B.S., R.C., C.L., T.S. and H.D.; resources, R.C. and C.L.; data curation, R.C. and C.L.; writing—original draft preparation, L.A., T.S. and H.D.; writing—review and editing, L.A., T.S. and H.D.; visualization, L.A., T.S. and H.D.; supervision, L.A., B.S., T.S. and H.D.; project administration, L.A. and B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The data is available for use and can be found in [30].

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

References

- Lafkihi, M.; Pan, S.; Ballot, E. Freight transportation service procurement: A literature review and future research opportunities in omnichannel E-commerce. *Transp. Res. Part E Logist. Transp. Rev.* **2019**, *125*, 348–365. [CrossRef]
- Belien, J.; Boute, R.; Creemers, S.; De Bruecker, P.; Gijbrecchts, J.; Padilla Tinoco, S.V.; Verheyen, W. Collaborative shipping: Logistics in the sharing economy. *ORMS Today* **2017**, *44*, 20–23.
- Karakikes, I.; Nathanail, E. Assessing the Impacts of Crowdshipping Using Public Transport: A Case Study in a Middle-Sized Greek City. *Future Transp.* **2022**, *2*, 55–81. [CrossRef]
- Liang, X.; Yang, H.; Wu, Q. Online crowdsourced delivery for urban parcels using private cars under time-dependent travel times. *Comput. Ind. Eng.* **2022**, *174*, 108807. [CrossRef]
- Hong, H.; Li, X.; He, D.; Zhang, Y.; Wang, M. Crowdsourcing incentives for multi-hop urban parcel delivery network. *IEEE Access* **2019**, *7*, 26268–26277. [CrossRef]
- Tsay, S.P.; Accuardi, Z.; Schaller, B.; Hovenkotter, K. *Private Mobility, Public Interest: How Public Agencies Can Work with Emerging Mobility Providers*; TransitCenter: New York, NY, USA, 2016.
- Guarino Neto, L.; Geraldo Vidal Vieira, J. An investigation of consumer intention to use pick-up point services for last-mile distribution in a developing country. *J. Retail. Consum. Serv.* **2023**, *74*, 103425. [CrossRef]
- Gatta, V.; Marcucci, E.; Nigro, M.; Patella, S.; Serafini, S. Public Transport-Based Crowdshipping for Sustainable City Logistics: Assessing Economic and Environmental Impacts. *Sustainability* **2018**, *11*, 145. [CrossRef]
- Casquero, D.; Monzon, A.; García, M.; Martínez, O. Key Elements of Mobility Apps for Improving Urban Travel Patterns: A Literature Review. *Future Transp.* **2022**, *2*, 1–23. [CrossRef]
- Papaioannou, E.; Iliopoulou, C.; Kepaptsoglou, K. Last-Mile Logistics Network Design under E-Cargo Bikes. *Future Transp.* **2023**, *3*, 403–416. [CrossRef]
- Nascimento, C.d.O.L.; Oliveira, L.K.d. Analysing the Potential of Performing Last-Mile Bicycle Deliveries: The Perspective of Brazilian Logistics Operators. *Future Transp.* **2023**, *3*, 296–310. [CrossRef]
- Boysen, N.; Emde, S.; Schwerdfeger, S. Crowdshipping by employees of distribution centers: Optimization approaches for matching supply and demand. *Eur. J. Oper. Res.* **2022**, *296*, 539–556. [CrossRef]
- Tapia, R.J.; Kourouniotti, I.; Thoen, S.; de Bok, M.; Tavasszy, L. A disaggregate model of passenger-freight matching in crowdshipping services. *Transp. Res. Part A Policy Pract.* **2023**, *169*, 103587. [CrossRef]
- Bruns, R.; Dötterl, J.; Dunkel, J.; Ossowski, S. Evaluating Collaborative and Autonomous Agents in Data-Stream-Supported Coordination of Mobile Crowdsourcing. *Sensors* **2023**, *23*, 614. [CrossRef] [PubMed]
- Ghorbanioskalaei, E.; Herrera, E.M.; Ammouriouva, M.; Juan, A.A. On the Use of Agile Optimization for Efficient Energy Consumption in Smart Cities's Transportation and Mobility. *Future Transp.* **2022**, *2*, 868–885. [CrossRef]
- Macrina, G.; Guerriero, F. The Green Vehicle Routing Problem with Occasional Drivers. In *New Trends in Emerging Complex Real Life Problems: ODS, Taormina, Italy, 10–13 September 2018*; Daniele, P., Scrimali, L., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 357–366. [CrossRef]
- Khanchehzharrin, S.; Shahmizad, M.; Mahdavi, I.; Mahdavi-Amiri, N.; Ghasemi, P. A model for the time dependent vehicle routing problem with time windows under traffic conditions with intelligent travel times. *RAIRO-Oper. Res.* **2021**, *55*, 2203–2222. [CrossRef]
- Horner, H.; Pazour, J.; Mitchell, J.E. Optimizing driver menus under stochastic selection behavior for ridesharing and crowdsourced delivery. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *153*, 102419. [CrossRef]
- Raviv, T.; Tenzer, E.Z. Crowd-Shipping of Small Parcels in a Physical Internet. 2018. Available online: <http://www.eng.tau.ac.il/~talraviv/Publications/> (accessed on 16 November 2023)
- Archetti, C.; Guerriero, F.; Macrina, G. The online vehicle routing problem with occasional drivers. *Comput. Oper. Res.* **2021**, *127*, 105144. [CrossRef]
- Arslan, A.M.; Agatz, N.; Kroon, L.; Zuidwijk, R. Crowdsourced delivery—A dynamic pickup and delivery problem with ad hoc drivers. *Transp. Sci.* **2019**, *53*, 222–235. [CrossRef]
- Chen, W.; Mes, M.; Schutten, M. Multi-hop driver-parcel matching problem with time windows. *Flex. Serv. Manuf. J.* **2018**, *30*, 517–553. [CrossRef]
- Yıldız, B. Express package routing problem with occasional couriers. *Transp. Res. Part C Emerg. Technol.* **2021**, *123*, 102994. [CrossRef]
- Treiblmaier, H.; Mirkovski, K.; Lowry, P.B.; Zacharia, Z.G. The physical internet as a new supply chain paradigm: A systematic literature review and a comprehensive framework. *Int. J. Logist. Manag.* **2020**, *31*, 239–287. [CrossRef]

25. Goodarzian, F.; Ghasemi, P.; Gonzalez, E.D.S.; Tirkolae, E.B. A sustainable-circular citrus closed-loop supply chain configuration: Pareto-based algorithms. *J. Environ. Manag.* **2023**, *328*, 116892. [[CrossRef](#)] [[PubMed](#)]
26. Orenstein, I.; Raviv, T. Reliable Parcel Routing Policy in a Physical Internet. In Proceedings of the Triennial Symposium on Transportation Analysis (TRISTAN), Hamilton Island, Australia, 17–21 June 2019; pp. 375–379. Available online: <https://tristanconference.org/system/tristan-x/documents/TRISTAN-X-Program.pdf> (accessed on 16 November 2023).
27. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [[CrossRef](#)]
28. Ruth, C.; Chaya, L. Smart Shipping System. 2023. Available online: <https://github.com/Mussil/Shipping/tree/main> (accessed on 16 November 2023).
29. Waze. Waz'e API Tools for Developers. Available online: <https://developers.google.com/waze> (accessed on 16 November 2023).
30. Roth, C.; Chaya, L. Real Traffic Data from the City of Ashdod. 2023. Available online: <https://drive.google.com/file/d/1RJJ0rKxgqwkL5IWes1Qn07FTfRzmwhT9/view?usp=sharing> (accessed on 16 November 2023).

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.