

Delivery Management System based on Vehicles Monitoring and a Machine-learning Mechanism

Guillaume Habault
IMT Atlantique,
SRCD, Rennes, FRANCE,
guillaume.habault@imt-atlantique.fr

Yuya Taniguchi, Naoaki Yamanaka
Keio University,
Yagami Campus, Yamanaka Laboratory, Yokohama, JAPAN,
taniguchi@yamanaka.ics.keio.ac.jp, yamanaka@ics.keio.ac.jp

Abstract—The continuously growing online shopping is increasing the number of attended home deliveries. The last-mile delivery plays an important role in online shopping satisfaction and especially for food deliveries.

This paper focuses on food delivery retailers and particularly investigates the possibility to enhance deliveries using information and data knowledge. In fact, in addition to optimize and to share delivery routes, delivery vehicles could be monitored in order to always maintain shortest delivery delays. We propose in this paper a delivery management architecture targeting these principles. This system is composed of several core mechanisms that should keep delivery delays to a minimum while maintaining low service times. A proof-of-concept of this delivery management system has been developed using Electric Scooters, smartphones and several algorithms. It demonstrates how this architecture could work in a food delivery scenario.

Index Terms—Vehicle monitoring, Machine-learning, Last-mile delivery, Database architecture, Delivery Management System.

I. INTRODUCTION

The delivery business is undergoing a rapid change as various items can now be delivered to your doorstep in a single tap of your mobile phone [1]. Customers' behavior slightly turns toward more food deliveries and is not anymore limited to "fast-food" delivery. For instance, in Japan several companies deliver meal kits (ingredients and associated recipes) to help out busy households but also the growing number of seniors.

Some platforms ride along this downward trend and offer access to several restaurants in order to offer better services. They appear as food delivery "brokers", interconnecting restaurants with customers as well as taking care of deliveries.

The delivery to the final destination is known as the Last-Mile Delivery (LMD). Its principal objective is to deliver items as fast as possible and directly to the customer's home. In particular, food deliveries have to be delivered quickly to preserve their freshness. The LMD fulfillment is very important as it is one of the main factor of the overall customers satisfaction when using delivery services.

Last-mile deliveries could be more efficient and automated by using new systems and technologies. Several researches are conducted to optimize the LMDs problem. Especially, the Vehicle Routing Problem (VRP) aims at reducing the number of deliveries and vehicles required for a set of deliveries [2]. However, VRP solutions might not be optimal for stochastic problems [3]. Recent approaches investigate the ability to use automated vehicles in order to improve efficiency. For

instance, Murray et al. [4] studied how drones could be used along with non-automated delivery vehicles.

Another possible solution to reduce costs associated to delivery is to collaborate with other delivery systems. Indeed, collaboration on the LMD [5] has shown that the number of carbon emission as well as the number of required delivery vehicles could be reduced by 25%. As a result, it will be possible to share delivery among different types of goods. Therefore, it is not surprising that well-known delivery "brokers" started other delivery services. For example, *Amazon* is already delivering food items; and *Uber* has its own parcel (*UberRush*) as well as food (*UberEats*) delivery services. Using citizen workers is dependent on their availability, which may introduce delays, but it can help retailers both optimize LMD and reduce the overall cost [6].

In this paper, we study the case of a food delivery broker. We propose a monitoring architecture in order to always provide optimal sharing and routes to delivery drivers. A proof-of-concept of the proposed architecture has been implemented to support this work. Therefore, this paper investigates how common Information and Communication Technology (ICT) techniques such as route optimization and Machine Learning (ML), can help delivery retailers optimize their systems. To the best of our knowledge, such ML enhancements in the food delivery domain have not been studied previously.

The remainder of this paper is organized as follows. Section II describes the context and the scenario envisioned during this work. It also explains the main reason for using vehicles monitoring and ML techniques. A proof-of-concept, presented in Section III, has been implemented in order to support and demonstrate the feasibility of such a system. Finally, Section IV concludes the paper and lists perspectives to be investigated in the future.

II. CONTEXT AND OBJECTIVE

A. General Scenario Envisioned

We consider the case of a complete delivery "broker". In the rest of the document, such a broker will be identified as the Delivery Management System (DMS). As depicted in Figure 1, it makes it possible to interconnect both food shops, retailers and customers. Moreover, it has its own pool of delivery drivers and vehicles. In the rest of the paper and as illustrated in the Figure 1, we only consider scooter as delivery vehicles.

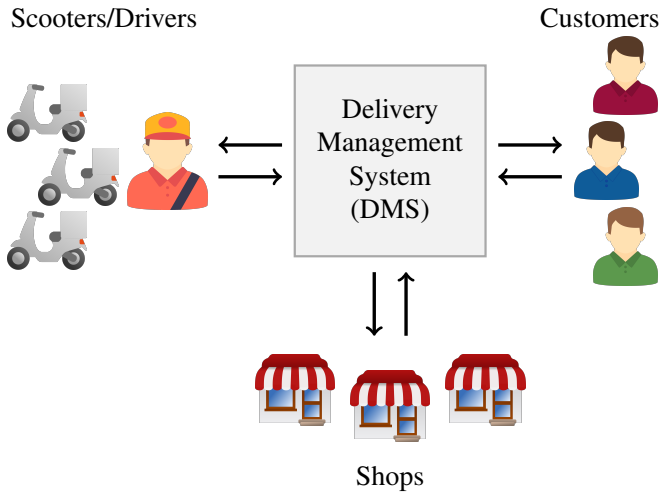


Fig. 1. Representation of a Delivery Management System.

As a matter of fact, these vehicles, due to their size and maneuverability, offer better delivery access. In addition, with the proper optimization mechanisms, smaller delivery vehicles may be used without increasing overall costs [7]. Furthermore, they will reduce travel time reliability in urban areas where congestion and space limit the access of larger vehicles such as trucks. In addition, using Electric Scooter (ES) will reduce the overall carbon footprint of food deliveries.

Drivers pick up items at a shop and deliver them directly to the customer's doorstep. They spent some time serving customers (parking, walking to the door, etc.). Each driver will carry a smartphone. Among other things, it will help the DMS follow the status of the delivery as well as communicate with the driver. This smartphone will inform the driver about the next destination and an optimal route toward it. Simultaneously, it will monitor the delivery status (current position and speed) at a periodic interval.

B. Objective: Reducing Delivery Delays

The objective of this study is to reduce delivery delays as well as improve the overall delivery efficiency.

As mentioned in [8]–[11], ICTs can be deployed to monitor real-time road contexts (e.g. vehicles, traffic jams, weather, construction) in order to support optimal routing. Similar mechanisms can be used to reduce travel time during deliveries. In fact, current route planning systems are made for large vehicles such as cars and trucks. Therefore, monitoring delivery scooters could greatly improve their route planning. Analyzing the monitored data with ML mechanisms could help DMSs learn when traffic peaks occur and which roads to avoid. This knowledge should enable DMSs to precisely plan the delivery travel time and prevent reckless driving.

Sometimes slowdowns could be noticed without specific known traffic congestion for a given period. They might be caused by external phenomena (e.g. weather). As a consequence, the monitored data from ESs could be coupled with external information in order to have better results. And with

appropriate learning mechanisms, the system will be able to predict any reduction in speed, which will result in obtaining more precise routes.

However, this paper will not focus on both the efficiency and the performance evaluation of these mechanisms. In fact, as of this writing, we only have a couple of ESs. As a consequence, we cannot collect sufficient vehicles data in order to use and fully test ML techniques for delivery. Nevertheless, based on results from aforementioned works, food LMDs should also benefit from ML techniques. This is the reason why this paper will concentrate on the architecture set up to allow such data analysis as well as its preliminary results. Either way, the proposed system should help DMSs keep to a minimum delivery travel times.

III. DELIVERY MANAGEMENT SYSTEM PROTOTYPE

A. Core Mechanisms

A DMS prototype has been implemented in order to prove the feasibility of this concept. As illustrated in Figure 1, a DMS offers customers the possibility to order from different shops. For now, these shops and associated orders are fully simulated. Our simulation system enables us to generate orders in different areas (dense, rural, etc.) and with different parameters. As shows in Figure 2, the time at which orders are placed follows a custom law. It ensures that the order placement respects both meal times and shops opening hours.

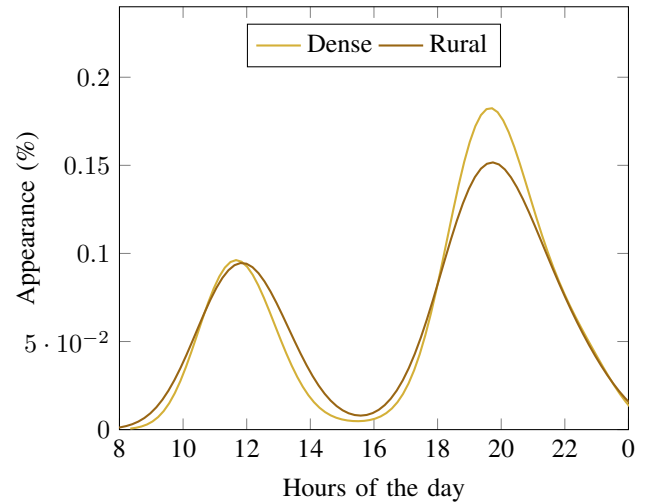


Fig. 2. Distribution of orders placement and reserved delivery.

The prototype a DMS is composed of five core mechanisms as depicted in Figure 3:

- 1) a communication system, enabling DMS to interact with others and to collect information;
- 2) a ML algorithm, using historical data to predict different information;
- 3) a sharing algorithm, determining sharing possibilities;
- 4) a route algorithm, computing optimal delivery routes; and
- 5) a view system, allowing managers to consult the overall system status (incoming orders and prediction, past as well as on-going deliveries).

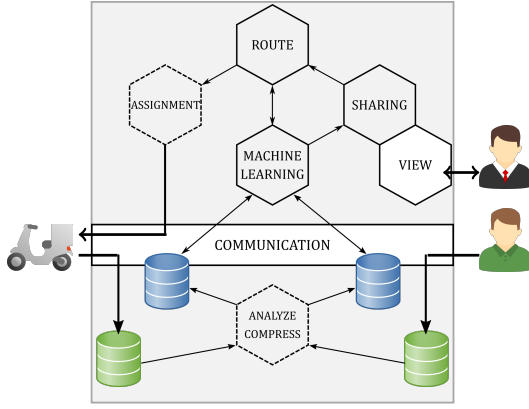


Fig. 3. Illustration of the Delivery Management System prototype.

B. Communication System

DMSs need to assign deliveries to their drivers as well as collect information regarding these deliveries for learning purposes. Thus, a bi-directional communication between a DMS and its drivers is required. In our prototype, we used ESs from *Adviva*, but these vehicles are not equipped with communication systems. Therefore, an *Android* smartphone is used to act as a gateway between the driver/ES and the DMS. An application has been developed both to realize such communications and to collect Controller Area Network (CAN) data. Communications between the DMS and the *Android* application use the HTTP protocol with REST APIs over Wi-Fi or cellular networks. Therefore, drivers can receive, via the application interface, computed routes information from the DMS. In parallel, when on delivery, the device collects information about the ES such as current speed and acceleration status. We choose a real-time monitoring system compared to an event-based one as it is important to determine the travel time per road segment. Therefore, in our custom monitoring system, every second, the average state of each monitored data is stored by the application. The real-time position is given by the built-in GPS of the smartphone as ESs are not equipped with one.

The rate at which the smartphone sends the delivery information is controlled by the DMS. Based on similar work [12], the default sending interval has been set to 1 minute. As a result, the application sends 60 delivery information every minute. This trade-off enables us to limit the amount of data transmitted on the network while still allowing the DMS to analyze delivery status in almost real-time.

Figure 4 shows the prototype view of the application. It gives information regarding the connection status: (i) Bluetooth, for the communication with the ES; (ii) Wi-Fi and Cellular networks, for the communication with the DMS; and (iii) the DMS availability. It also shows both the reception status of the CAN data and the transmission status of monitoring packets. Finally, the map at the bottom displays to the driver the computed optimal route.

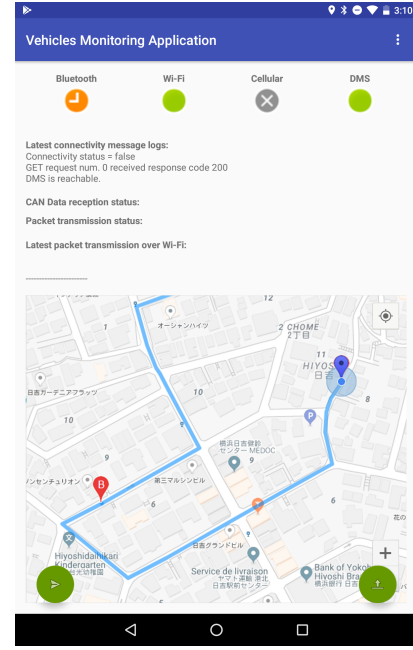


Fig. 4. Preview of the *Android* application.

C. Machine-Learning Algorithm and Data Storage

The monitored data received from ESs are used at different moments. In the first phase, when vehicles are “on delivery”, their data are stored in a “temporary” Database (DB) (represented in green in the Figure 3). These data are analyzed in “real-time” by the DMS in order to detect any slowdown in the delivery and if possible determine alternate routes. In a second phase, when a delivery is completed, the raw data from the temporary DB will be further examined before being compressed and stored in a “permanent” DB (illustrated in blue in the Figure 3). This simple storage hierarchy, as suggested by Andrew Brust¹, enables DMSs to use the monitored data based on their natures and to only archive relevant data. Furthermore, according to Dave Graham, monitored data can be used at different periods, therefore compression is required and temporal ones should be favored. A compression similar to the one used in the video “image-to-image prediction” is deployed in our prototype. In fact, the system will determine a set of moments for each delivery. For these moments, all the monitored information will be stored. However, for the information in between two moments, only the difference with the previous monitored one will be stored. This mechanism makes it possible to reduce the amount of stored data, while maintaining the same precision.

Orders information are stored in the same way. Non-delivered orders are stored in a temporary DB, while completed orders are archived in a permanent DB. Data from the two permanent DBs will be used by the ML algorithm to estimate both incoming food orders and travel times for different road segments at different hours. A time-based ML

¹<https://it.toolbox.com/blogs/erpdesk/six-tips-for-storing-iot-data-082916>

mechanism similar to the one used in [13], [14] will be deployed. Indeed, both incoming orders and traffic congestion depend on the day and the hour of the day. For instance, based on the monitored vehicle speed, the ML algorithm can estimate for each day of the week and each hour of the day, the time required for an ES to travel a given road segment. Therefore, with such information learned from on-field travel, the route algorithm will be able to determine the fastest route for future deliveries at both a given day and a given hour.

As a result, the ML mechanism should be able to learn and predict information on a timely scale.

D. Sharing and Route Algorithms

As mentioned previously, the food delivery is a specific case of the LMD. Contrary to classic freight deliveries, the orders are mostly not planned (i.e. stochastic arrivals) and should be delivered within the next half hour. Nevertheless, it might be possible to share these stochastic deliveries with other planned and pending deliveries. The sharing algorithm enables the DMS to check whether different orders can be gathered in common deliveries. Such a mechanism could help a DMS maximize its delivery capacity while limiting the traveled distance with an empty trunk.

The sharing algorithm outputs a list of way points to “visit” for each delivery. And for each list, the route algorithm has to determine an optimal delivery route. If no data have been recorded for given road segments, *Google Maps APIs* are used to obtain an estimation of the travel time at a given hour. Otherwise, it will be estimated by the ML mechanism with historical delivery data. Based on these travel times, the algorithm establishes the fastest route passing by each way point. Finally, a function assigns deliveries to available drivers.

E. Management View

Last but not least, a management view system has been developed so that managers can visualize the status of the DMS. These view pages enable them to see past and incoming orders as well as to trace past and on-going deliveries. In addition, the decision made by the different core mechanisms can be adapted or refined if necessary.

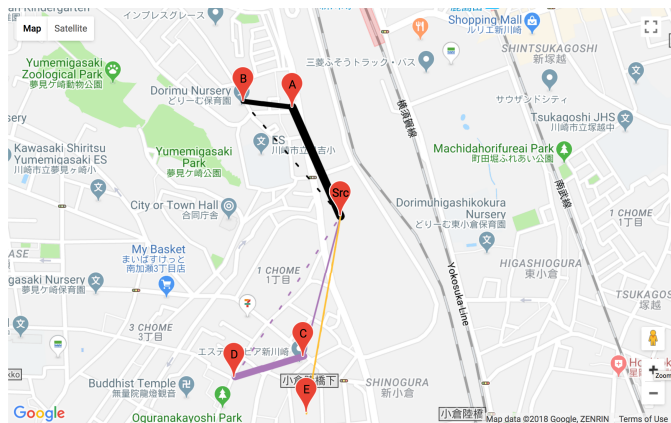


Fig. 5. Preview of the sharing view.

Figure 5 presents the sharing view of this prototype. It displays the visualization of some received orders along with their association into delivery paths. The sharing algorithm uses at the same time the order position and the expected delivery time to estimate such sharing opportunities. At the given moment of the Figure 5, the order *A* could share its delivery with the order *B*. Readers might think that the order *E* could share its delivery with orders *C* and *D*, but it is not possible. In fact, even though their positions are close, their delivery times are too far from each other. Each time a new order is placed, the sharing algorithm re-computes all sharing opportunities within the pending orders in order to determine an optimal solution.



Fig. 6. Preview of the tracing view.

Figure 6 shows the vehicle’s tracing system. Each delivery is completely monitored and with this view, managers can have the delivery information of any past or on-going delivery. They can especially visualize road segments where slowdowns occurred (displayed in purple on the Figure). In addition, they have general information on the delivery such as the average and maximum speed of the ES. In a future version, they will be able to directly see the service time of each delivery stop.

This proof-of-concept is consequently operational. More data is now required in order to further investigate the benefits of this prototype. The performance of the sharing mechanism could be studied with our order generation system. However, real delivery data are required to test and to evaluate both the ML mechanism and the database architecture efficiency.

IV. CONCLUSION AND PERSPECTIVES

This paper presents a delivery management system architecture and its proof-of-concept implementation. This work relies on information and data knowledge techniques. It enables retailers to minimize the delivery travel time along with optimizing the delivery efficiency. This system allows them to monitor the delivery vehicles and to efficiently store corresponding data. These data are then used with a machine-learning mechanism in order to estimate relevant information. With such an algorithm, they could forecast on hourly basis both the travel time of different road segments and incoming orders. In addition, a sharing algorithm can determine any sharing opportunities among different orders. This delivery management system will reduce the total number of required deliveries and thus, will increase the delivery capacity.

In the future, the performance of this prototype and its algorithms should be studied. On one hand, the travel time obtained with the ML mechanism should be compared to the travel time provided by only *Google Maps APIs*. This study should determine the gain offered by such learning techniques and if any, the associated cost. However, it is of crucial importance to collect several delivery data for this study.

On the other hand, the efficiency of the sharing mechanism can be investigated. Food deliveries due to their stochastic arrivals are more complex than other logistics or freight deliveries. Moreover, their delivery times are shorter due to food constraints such as the freshness. Therefore, this study could unveil if solutions proposed for vehicle routing problem and used to optimize freight deliveries will be relevant for such scenarios.

Finally, this prototype could also be used to test other type of optimization solutions such as smoothing order arrivals with coupons or incentives.

ACKNOWLEDGMENT

This work was partially supported by a mobility scholarship of the author at Keio University in the frame of the Erasmus Mundus Action 2 Project TEAM Technologies for Information and Communication Technologies, funded by the European Commission. This publication reflects the view only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

REFERENCES

- [1] C. Hirschberg, A. Rajko, T. Schumacher, and M. Wrulich. (2016) The changing market for food delivery. [Online]. Available: <https://www.mckinsey.com/industries/high-tech/our-insights/the-changing-market-for-food-delivery>
- [2] W. Zheng, X. Hu, and A. Z. Zeng, "A two-stage solution procedure for food delivery decisions in cities with circular transportation networks," in *2007 IEEE International Conference on Service Operations and Logistics, and Informatics*, Aug 2007, pp. 1–6.
- [3] U. Janjarassuk and R. Masuchun, "Cost and reliability analyses for the vehicle routing problem," in *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, June 2017, pp. 338–341.
- [4] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.trc.2015.03.005>
- [5] A. Muñoz-Villamizar, J. R. Montoya-Torres, and C. A. Vega-Mejía, "Non-collaborative versus collaborative last-mile delivery in urban systems with stochastic demands," *Procedia CIRP*, vol. 30, pp. 263–268, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.procir.2015.02.147>
- [6] Y. Wang, D. Zhang, Q. Liu, F. Shen, and L. H. Lee, "Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 93, pp. 279–293, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.trre.2016.06.002>
- [7] A. Conway, P. Fatisson, and P. Eickemeyer, "Urban micro-consolidation and last mile goods delivery by freight-tricycle in Manhattan: Opportunities and challenges," *TRB 2012 Annual Meeting. Washington, D.C., USA*, 2011.
- [8] B. Fleischmann, S. Gnutzmann, and E. Sandvoß, "Dynamic vehicle routing based on online traffic information," *Transportation Science*, vol. 38, no. 4, pp. 420–433, Nov. 2004. [Online]. Available: <http://dx.doi.org/10.1287/trsc.1030.0074>
- [9] J. N. O. Fernandes, "A real-time embedded system for monitoring of cargo vehicles, using controller area network (can)," *IEEE Latin America Transactions*, vol. 14, no. 3, pp. 1086–1092, March 2016.
- [10] A. T. Akabane, R. W. Pazzi, E. R. M. Madeira, and L. A. Villas, "Modeling and prediction of vehicle routes based on hidden markov model," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sept 2017, pp. 1–5.
- [11] R. K. Megalingam, V. Mohan, A. Mohanan, P. Leons, and R. Shooja, "Wireless sensor network for vehicle speed monitoring and traffic routing system," in *2010 International Conference on Mechanical and Electrical Technology*, Sept 2010, pp. 631–635.
- [12] G. Habault, M. Lefrançois, F. Lemercier, N. Montavont, P. Chatzimisios, and G. Z. Papadopoulos, "Monitoring traffic optimization in a smart grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3246–3255, Dec 2017.
- [13] G. Habault, J. Hursti, and J.-M. Bonnin, "Defining a distributed architecture for smart energy aware systems," in *Complex Systems Design & Management*, G. Fanmuy, E. Goubault, D. Krob, and F. Stephan, Eds. Cham: Springer International Publishing, 2017, pp. 83–94.
- [14] M. Mao, Y. Wang, Y. Yue, and L. Chang, "Multi-time scale forecast for schedulable capacity of evs based on big data and machine learning," in *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, Oct 2017, pp. 1425–1431.