A Blockchain-Based System for the Last-Mile Delivery

Francesca Guerriero[1] [a], Edoardo Scalzo[1] [b], Rodolfo Pietro Calabrò[2] and Giuseppe Scarfò[3]
[1]Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende (CS) 87036, Italy
[2]VeChain Foundation San Marino Srl, Milano (MI), Italy
[3]NTT Data Italia SPA, Rende (CS) 87036, Italy

Keywords: Sharing Economy, Occasional Drivers, Vehicle Routing Problem, Blockchain.

Abstract: This paper describes the application of the blockchain technology in the last-mile delivery. The attention is focused on the Vehicle Routing Problem with Occasional Drivers and Time Windows constraints. The main aim is to develop a strategy, which uses the blockchain as a support to achieve a partially decentralized delivery system. An analysis of the advantages obtained by using such a technology, in terms of economic sustainability, is carried out on small-size test instances. The proposed system is implemented and tested by using three different EVM-compatible blockchains. The collected experimental results have shown that the implemented Polygon and VeChain blockchain systems allow to achieve good performance in terms of sustainability and seems to have good prospects in terms of scalability.

1 INTRODUCTION

Online shopping has experienced a relevant growth over the last years. This tendency has raised increasing interest in developing new innovative strategies to improve customers' online shopping experience and to reduce delivery times. Thus, an increase in pressure on the efficiency and effectiveness of last-mile delivery has been observed.

For this reason, the management of same-day/last-mile delivery process is one of the most essential activities for shipper companies. To pursue successful and efficient aims in this context, the larger e-retailers have begun to investigate atypical and unusual last-mile delivery solutions. One of the innovative approaches suggested by companies' research teams is crowd-shipping. The basic principle of crowd-shipping is to incorporate the fundamentals of sharing economy into the transportation process, by assigning some deliveries to regular people (referred to as occasional drivers (ODs)), that are traveling to their destination. For a small compensation, ODs will share the empty space in their own vehicles to deliver packages, while deviating from their usual route. In general, in a classic Vehicle Routing Problem with Occasional Drivers (VRPOD) system, both the routing of a fleet of company vehicles and a fleet of vehicles belonging to ODs are managed in the same approach. In this paper, we propose a delivery system consisting only of ODs; therefore, we do not examine and process the routes of company vehicles. In this scenario, an OD route consists of starting from the shipper depot, picking-up the packages and making deliveries to assigned customers, and then ending at a destination location. Another important consequence of the increase in requests for online purchases is the low benefit that the delivery system of large e-retailers can grant to small-medium shops, especially for deliveries to be made in a restricted local area.

The objective of our work is to solve the aforementioned problems through the implementation of a driver system, managed entirely in a decentralized way. To this aim, we have decided to use a relatively new technology, that is the blockchain. Blockchain is a kind of distributed software system, that handles the main issues of certification, data integrity, security, data ownership and privacy. All these properties are satisfied by some particular architecture systems that use the following main components: decentralization and peer-to-peer network, cryptography, consensus and immutability. Everything is immutable because when data are written to the blockchain, they cannot be changed. Indeed, the cryptography guarantees data integrity. Furthermore, modifying data is practically impossible; indeed, nowadays the cryptography used in these systems is unbreakable. All data and information are validated by a group of nodes (consensus

[a] https://orcid.org/0000-0002-3887-1317
[b] https://orcid.org/0000-0003-0970-6264

Guerriero, F., Scalzo, E., Calabrò, R. and Scarfò, G.
A Blockchain-Based System for the Last-Mile Delivery.
DOI: 10.5220/0011886200003396
In *Proceedings of the 12th International Conference on Operations Research and Enterprise Systems (ICORES 2023)*, pages 237-244
ISBN: 978-989-758-627-9; ISSN: 2184-4372
Copyright © 2023 by SCITEPRESS – Science and Technology Publications, Lda. Under CC license (CC BY-NC-ND 4.0)

and decentralization). All the nodes represent the core of the system: they are the owners of all data, and all processes of validation. These nodes validate everything, but, before validating data and information, even if this is not mandatory, they need to reach a consensus about it. Indeed, in a decentralized system each node can be considered as an state machine independent from the others ((Nakamoto, 2008), (Calabrò et al., 2021)).

This paper represents the first attempt to use the blockchain technology in the ODs scenario, to manage and preserve some scores, which are used to assign, in a completely transparent way, deliveries to the available ODs.

It is assumed that shippers and ODs can register on a dedicated crowd-shipping blockchain-based online platform. In this paper, the shipper term is used to refer, in addition to the classic company that deals with shipping goods, also to a small-medium shop that sells its products online and requires drivers to make the deliveries for logistical reasons or for meeting specific customer needs. By logging on this platform, shippers can public a Delivery Box (DB), i.e., a set of deliveries with the minimum information necessary for the ODs to make a decision. The ODs can accept the most convenient requests, receive all the information to perform the deliveries and gain a compensation.

The selection of the delivery assignments is made by an algorithm guided by three scores, that keep track of the punctuality in packages collection, the punctuality in delivery and in the completion of all deliveries, respectively. The scores are initially set to default values and are updated after each delivery. The score updating schemes are not studied in this work, and for this reason we consider, for the computational experiments, that each OD makes all deliveries on time.

The remainder of this paper is structured as follows. In Sect. 2, we review the scientific literature related to the problem under investigation, i.e., the Vehicle Routing Problem with Occasional Drivers and Time Windows (VRPODTW), and to the blockchain technology with specific reference to the context of the sharing-economy. In Sect. 3, we present the VRPODTW, while in Sect. 4 we give details about the support of the blockchain technology. In Sect. 5 we report the computational results and in Sect. 6 we summarize our conclusions and outline possible future work along this research line.

2 STATE OF THE ART

Numerous relevant studies that discuss the advantages and disadvantages of crowd-shipping can be found in the scientific literature ((Alnaggar et al., 2021)). (Archetti et al., 2016) were the first to take into account ODs when addressing the vehicle routing problem. The authors proposed a hybrid Variable Neighborhood Search (VNS) and a Tabu Search to address the problem. The researchers pointed out the importance and difficulties in developing suitable compensation schemes. (Macrina et al., 2017) studied a variant of the problem addressed in (Archetti et al., 2016), where time windows for customers and ODs are taken into account. The authors also examined a system with multiple deliveries for ODs and also took into account a split delivery policy scheme. They discussed on the advantages of these crowd-shipping versions in a variety of contexts and scenarios. To solve the VRPODTW with multiple deliveries, (Macrina et al., 2020a) proposed a VNS solution and tested the effectiveness of this heuristic on some benchmark instances of (Solomon, 1987). (Festa et al.,) presented for the first time an application of a variant of Biased Random-Key Genetic Algorithm with a restart strategy on the VRPODTW. They built a decoder function to convert the chromosomes into OD paths starting from a check on the compatibility of time windows and travel times. Transshipment nodes were added to the VRPODTW with multiple deliveries by (Macrina et al., 2020b). These nodes are intermediate locations that are served by company drivers and are closer to the delivery area than the central depot. They showed the benefits of this specific scenario as a particular case of a two-echelon VRP. (Di Puglia Pugliese et al., 2021) introduced occasional depots in the VRPOD. These depots are considered to be places where owners decide to share storage space for a small reward. Therefore, just as it happens for ODs, also in this context the owners of the intermediate depots are common people. The goal is to minimize the routing cost and the activation cost for the use of occasional depots. They grouped the ODs into two sets based on the tasks they undertook and the remuneration policy and proposed a mixed-integer programming model to mathematically represent the problem under study. The possible advantages of using intermediate transfer points to enable driver deliveries are also examined by (Sampaio et al., 2020). The authors underline the benefits of adopting this system when the pickup and delivery sites are very distant and the time windows of the ODs are narrow.

Other variants of the VRPOD consider the specific scenarios where the relevant information is not always available in advance (i.e., stochastic/online version) (see, e.g., (Dayarian and Savelsbergh, 2020) and (Archetti et al., 2021)) or the case in which electric vehicles are used (see, e.g., (Macrina and Guerriero, 2018) and (Macrina et al., 2020c)).

As regards the use of blockchain technology, the scientific contributions in the context of the sharing economy and crowdsourcing mainly concern ride-sharing.

A decentralized peer-to-peer approach to the ride-sharing process is an efficient way to get around problems with an intermediary or 3^{rd} party (see, e.g., (Sánchez et al., 2016)).

Consequentially, some researchers used blockchain technology to solve these problems. For example, (Vazquez and Landa-Silva, 2021) proposed a way to make a ride-sharing system decentralized, by using a smart contract to execute a cautionary deposit by the passenger and the driver, and another one to make the payment at the end of the ride.

In other scientific contributions, the importance of investigating the multiple features that technology can bring to ride-sharing to make the system sustainable and safe at the same time is discussed (see, e.g., (Baza et al., 2021), (Kudva et al., 2020) and (Renu and Banik, 2021)).

3 THE VRPODTW DESCRIPTION

In this section we report the mathematical programming model of the VRPODTW, where only ODs are considered. This formulation has been derived from the model proposed by (Macrina et al., 2017). Let C be the set of customers and s the shipper depot. Let K be the set of available ODs and V the set of their destination nodes v_k. We define the node set as $N := C \cup V \cup \{s\}$. Each node pair $(i, j) \in N \times N$ has a positive cost c_{ij} and a travel time t_{ij}, which satisfies the triangle inequality. Each customer has a request d_i, while each OD k has a maximum transport capacity Q_k, and finally each $i \in C \cup K \cup V$ is assigned a time window $[e_i, l_i]$.

Let r_{ij}^k be a binary variable equal to 1 if and only if OD k traverses the arc (i, j). Let w_i^k be the available capacities of k, after delivering to $i \in C$ and f_i^k be the arrival time of k at customer i. The problem that aims at minimizing the total cost can be mathematically formulated as follows:

$$\min \sum_{k\in K}\sum_{i\in C\cup\{s\}}\sum_{j\in C}\rho_k c_{ij} r_{ij}^k - \sum_{k\in K}\sum_{j\in C}\rho_k c_{sv_k} r_{sj}^k \quad (1)$$

$s.t.$

$$\sum_{j\in C\cup\{v_k\}} r_{ij}^k - \sum_{j\in C\cup\{s\}} r_{ji}^k = 0 \qquad \forall i \in C, k \in K \quad (2)$$

$$\sum_{j\in C\cup\{v_k\}} r_{sj}^k - \sum_{j\in C\cup\{s\}} r_{jv_k}^k = 0 \qquad \forall k \in K \quad (3)$$

$$\sum_{k\in K}\sum_{j\in C\cup\{v_k\}} r_{sj}^k \le |K| \quad (4)$$

$$\sum_{j\in C} r_{sj}^k \le 1 \qquad \forall k \in K \quad (5)$$

$$w_j^k \ge w_i^k + d_i r_{ij}^k - Q_k(1 - r_{ij}^k) \quad (6)$$

$$\forall i \in C \cup \{s\}, j \in C \cup \{v_k\}, k \in K$$

$$w_s^k \le Q_k \qquad \forall k \in K \quad (7)$$

$$f_i^k + t_{ij} r_{ij}^k - M(1 - r_{ij}^k) \le f_j^k \quad (8)$$

$$\forall i \in C, k \in K, j \in C \cup \{v_k\}$$

$$f_i^k \ge e_{v_k} + t_{si} \qquad \forall i \in C, k \in K \quad (9)$$

$$f_{v_k}^k \le l_{v_k} \qquad \forall k \in K \quad (10)$$

$$e_i \le f_i^k \le l_i \qquad \forall i \in C \quad (11)$$

$$\sum_{k\in K}\sum_{j\in C\cup\{v_k\}} r_{ij}^k = 1 \qquad \forall i \in C \quad (12)$$

$$r_{ij}^k \in \{0, 1\} \qquad \forall i, j \in N, k \in K \quad (13)$$

$$w_i^k \in [0, Q_k] \qquad \forall i \in C \cup \{s, v_k\}, k \in K \quad (14)$$

$$f_i^k \ge 0 \qquad \forall i \in C \cup \{s, v_k\}, k \in K. \quad (15)$$

The objective function minimizes the total cost, which is defined by considering a differentiated delivery routing cost for the ODs. In particular, each driver k has a different routing parameter ρ_k that modifies the value of the cost. The parameter keeps track of the reliability of k and it is calculated as a convex combination of the three scores (indicated as sc_i) introduced in the Sect.1. The coefficients (we_i) are chosen, following company policies, by the shipper user, and the value of ρ_k is defined as follows:

$$\rho_k = \sum_{i=1}^{3} \frac{we_i}{sc_{i,k}}. \quad (16)$$

In this way, partial priority is given to the ODs with the highest scores. For example, between two drivers with the same destination, the OD with the best weighted score will be chosen. On the other hand, there could be scenarios in which a driver with low scores is chosen but whose deviation for deliveries is very low.

As regarding the constraints, equations (2) and (3) represent flow conservation of the ODs. Constraints

(4) establish a maximum limit of available ODs, while (5) ensure that each OD leaves their origin node at most once. Equations (6) and (7) are capacity constraints. Finally, constraints (8)–(11) manage the time windows of all nodes. In particular, constraints (8) compute the arrival time at node j, while constraints (9)–(11) ensure that each customer is visited within their own time windows and each OD makes deliveries within their own time windows. Constraints (12) allow each customer to be served once and only once. Equations (13)–(15) define the domains of the variables.

4 BLOCKCHAIN INTERACTION

Blockchain technologies are widely used in problems where certification and trust are the basic components. In the proposed VRPODTW system, this technology is useful to handle customers, drivers and delivery data. In particular, the blockchain manages the entire shipments life cycle and updates the scores of ODs, but due to the complexity of the proposed exact model, we do not use it to implement the VRPODTW algorithm. In fact, execute a CPU-bound process into the blockchain is not very efficient. Blockchain computations suffer from high cost and high latency, with respect to classical centralized data processing. Therefore, a semi-decentralized approach is used: the solution of the exact model and, consequently, the assignments of deliveries to ODs are performed in a centralized way, while a decentralized approach is used for the data updating and the deliver confirmations.

Given the considerations reported above, some operations will be on-chain and other off-chain. Consequently, to be reliable and certified, the entire shipments life cycle is handled in a decentralized way and this guarantees the following properties:

- Transparency of DBs;
- Transparency of ODs scores;
- Trust of ODs;
- Certification of shipment receiving.

Figure 1 offers a high-level overview of how blockchain interacts with the VRPODTW system. In particular, it is possible to see the main positive flow and main interactions between customers, shipper, platform, algorithm, ODs and blockchain. The steps are the following and they are of two kinds: on-chain and off-chain.

(1). After purchasing online, each customer creates a delivery request. Since no data is stored in the chain, then this is considered an off-chain step.

(2). Shipper creates a Delivery Box (DB), that is a set of the information necessary to perform the deliveries. In particular, pickup and destination location, the package details, the time windows within which to make the deliveries and the compensation are specified. This is an off-chain step because here data are only submitted to VRPOD Platform that confirms the creation of DB and sends the request in step (3) to VRPOD Algorithm.

(3). The VRPOD platform takes the DB from step (2) and manages the information for the next steps and creates the inputs for the algorithm. Again, this step is off-chain.

(4). The VRPOD algorithm finds the optimal solution by minimizing costs, also taking into account the scores of all the available drivers and then notifies the selected ODs about the shipment assigned to them. This step is off-chain and mainly consists of a notification to each chosen OD, before the on-chain steps (5) and (6).

(5). It is the first on-chain step. Here ODs confirm to take care of shipments. This is a preliminary important step; in fact it is where blockchain comes into the play. When an OD confirms the shipment, then the procedure is started to write on the blockchain the necessary information to keep track of this event.

(6). This is the other important on-chain step. In fact, here the procedure guarantees the correct receiving of packages by the customers. After this step, in addition to keeping track of the confirmation event on the blockchain, the process automatically updates the scores of the OD.

(7). This is the last step and it is off-chain. Here the procedure notifies the VRPOD algorithm with the new update scores.

The cost of performing each transaction in the blockchain is measured in terms of gas fees and as shown in the figure, (5) and (6) are the only steps in which gas fees are paid. These two steps represent the decentralized phase and guarantee the trust and reliability of ODs as well as certify that customers have received shipments. All computational heavy calculations are performed by the algorithm.

In the Alg.1 there are two smart contracts to manage the system just described. The first one, *Shipment* contract is used to register all shipments. The information that the class considers are arrival time, departure time, shipment information and other useful data, including *nonce*. The latter is simply a random number, that, in combination with private key of

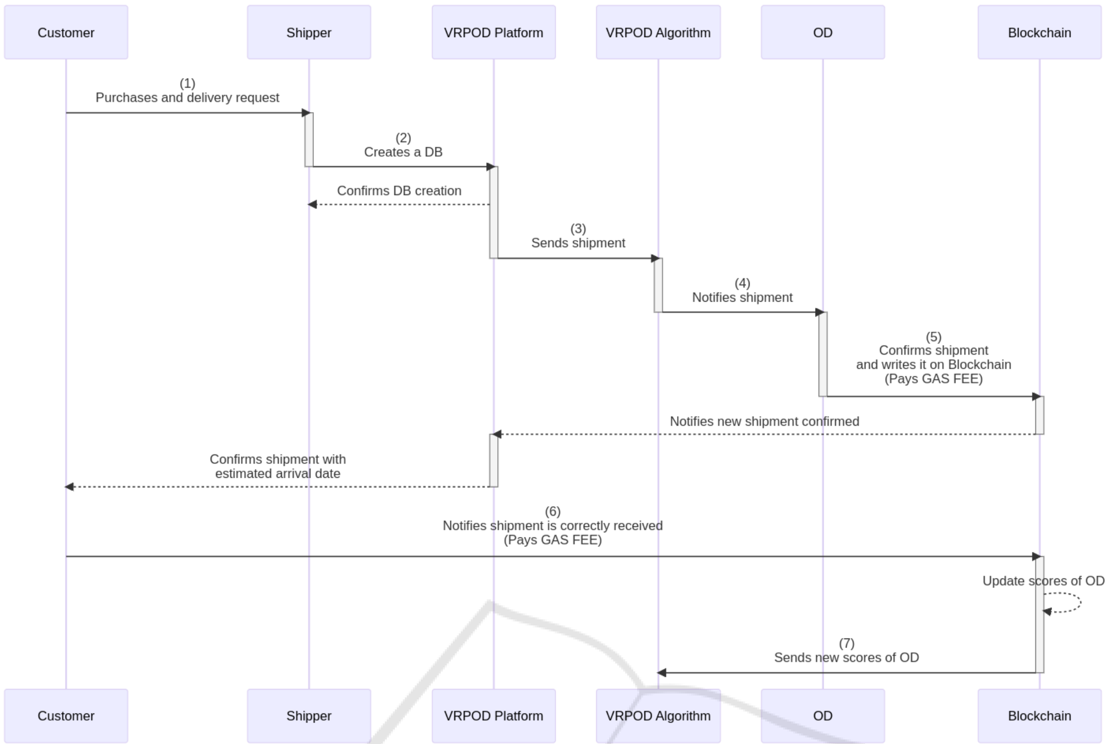

Figure 1: Blockchain and VRPODTW system integration - Main flow.

signer (customer), can confirm the correct arrival of the shipment to customer. When a shipment is confirmed using *confirmShip* function, then the *shipmentOD* (*shipmentDriver* variable) is set and a random nonce is created. The *setIsReceived* function confirms that a shipment is effectively received by the customer, by taking the signature of random nonce of shipment. Nonce can be public and visible to everybody, but only the owner of private key can sign this number. This guarantees that the only person who can confirm the receipt of shipment is the customer who owns the private key used to generate the *custAddress* address variable stored into the shipment. Then, after this function is called, the scores of the OD will be updated. *Driver* contract instead, is used to represent a driver with the corresponding scores. Finally, when a shipment is received, then the scores of the OD are updated with *updateScores()*.

5 COMPUTATIONAL STUDY

In this section, we summarize the results of our computational experiments. The exact model described was implemented in Java and solved to optimality using CPLEX 12.10. The computational tests were conducted using a 2.6 GHz Intel Core i7-3720QM processor and 8 GB 1600 MHz DDR3 of RAM running macOs Catalina 10.15.7. As regarding the blockchain we use three different EVM-compatible blockchains: Ethereum, Polygon and VeChain. We interact with blockchain using ethers.js library and smart contracts are done with solidity version 0.8.14. To evaluate the sustainability of the entire system and cost of the technologies we compared the results obtained by the proposed approaches for several small-size instances.

5.1 Generation of Instances

In order to evaluate the performance of the proposed integrated VRPODTW system in terms of cost sustainability, computational experiments have been carried out by considering three types of test problems. More specifically, we generated clustered-type, random-type and mixed-type instances, starting from the test problems used in (Macrina et al., 2020a) and we have grouped them into three sets, based on the number of customers, that is C5, C10, C15, with 5, 10 and 15 customers, respectively. Each set contains 12 problems. For each instance, we kept the same features and information of customers and shipper depot as those described in the work of (Macrina et al., 2020a). Instead, as regarding the drivers, we eliminated the company drivers and kept the main features of the ODs.

Algorithm 1: Smart contracts.

```
contract Shipment:
    struct ShipmentInfo:
            uint256 height;
            uint256 width;
            uint256 depth;
    private ShipmentInfo shipmentInfo;
    private Driver shipmentDriver;
    private address custAddress;
    private date departureTime;
    private date arrivalTime;
    private boolean isReceived = false;
    private uint256 nonce;
    constructor(custAddress):
        this.custAddress = custAddress;
    getNonce() external view;
    confirmShip(driverAddress) external:
        nonce = randomNumber();
        this.driverAddress = driverAddress;
        departureTime = currentTimestamp;
    setIsReceived(signedNonce, shipment)
      external:
        if verifySignature is true then
            driverAddress.updateScore();
contract Driver:
    private address driverAddress;
    private uint256 scores = (sc1, sc2, sc3);
    updateScores() public;
```

Since some drivers have been deleted from the original version of the instances, to adapt them to the specific scenario considered in this paper, then in order to guarantee feasibility, we modified in the instances of the set C15 the maximum capacities of ODs, by randomly choosing values in the interval $[15,80]$ and we added in all instances some more ODs. More specifically, for each instance we generated randomly the location of three ODs in the rectangle $R := \{(x,y) \in \mathbb{R}^2 \mid x \in [x_m, x_M],\ y \in [y_m, y_M]\}$, where x_m, x_M, y_m, y_M are the minimum and maximum of the abscissas and ordinates of the coordinates of all nodes of the original instance. Therefore, there are 6 ODs in the C5 and C10 sets, while 8 in the C15 set. Finally, for each OD we generated, in a random way, the three scores, by choosing the values in the interval $[1,30]$.

As regarding the shipper user parameters, described in (16) in the Sect. 3, we set the first coefficient we_1 equal to 0.6 and both the other two we_2, we_3 to 0.2, following a preliminary test.

5.2 Numerical Results

In Table 1 we report, for each set of test problems, the average costs of the system, using Ethereum, Polygon and VeChain. In particular, for each instance we measured the blockchain costs related to the main two functions showed in the steps (5) and (6) (see Section 4). Under the columns *GL* and *GP* there are the average values of the gas limit and gas price, respectively. In particular, gas limit is the maximum amount of gas estimated to execute a single blockchain function for a customer or a driver, while gas price is the cost of a single gas limit unit. This last metric alone does not represent an accurate measure, because it is influenced by the status of network, the volatility and other external factors. However, it is useful to evaluate the final price. Under the columns $Cost_B$, the average cost in dollars for using a given technology, that depends on the gas limit and gas price.

Table 2 gives the numerical results obtained by solving the mathematical model described in Section 3. In particular, under the *Cost* and *Time* columns the average optimal values and the average running time obtained with the exact model are given. Under the last column (*#EV*) there are the average number of main events to be tracked on the blockchain, that is customer confirmations or updating of the OD scores. These events depend on the number of the ODs selected and on the number of customers to be served by each OD.

It is important to note that the computational times shown in Table 2 are measured in seconds. In addition, we assume that, in all solutions, each OD makes all deliveries on time, thus all scores are incremented by one.

From the Table 2, it is evident that when the number of customers increases, the number of ODs needed to make deliveries also increases. Thus, the higher the number of customers, the higher the routing cost and the number of events to be recorded on the blockchain. On the other hand, the average cost per OD remains almost unchanged. As far as the execution time is concerned, the computational results underline that in all the three sets of test problems, the exact model spends on average a relatively short amount of time to find optimal solutions. However, the observed trend for the computational time underlines that a heuristic procedure should be implemented to solve large-scale instances.

Even though, the computational experiments have been carried out on small-size instances, the results reported in Table 1 clearly underline that, in these cases, the choice of the blockchain affects the performance of the developed system in terms of both

Table 1: Blockchains results.

Class	VeChain			Polygon			Ethereum		
	GL (10^5)	$GP(10^{-4})$	$Cost_B$	GL (10^5)	$GP(10^{-7})$	$Cost_B$	GL (10^5)	$GP(10^{-7})$	$Cost_B$
C5	8.18	1.36	0.008$	6.50	3.60	0.011$	6.50	1.32	8.23$
C10	14.72	2.53	0.015$	11.57	6.60	0.019$	11.57	2.43	14.68$
C15	20.25	3.57	0.021$	15.76	9.23	0.026$	15.76	3.41	20.02$

Table 2: Routing results.

Class	*Cost*	$Time(s)$	*#OD*	*#EV*
C5	206.88	0.20	2.33	12.00
C10	357.98	9.91	4.00	22.00
C15	470.68	33.16	5.17	30.50

sustainability and efficiency. From Table 1, it is evident the superiority of VeChain and Polygon in terms of cost compared to Ethereum blockchain.

Indeed, with VeChain and Polygon the average cost for the largest set is about 0.023$, while for the Ethereum blockchain the cost is about 20$. Since the values of the gas limit *GL* is estimated as a function of the number of operations performed with the smart contract, as expected, the higher the number of events performed *#EV*, the higher the value of *GL* on all the three blockchains. In particular, for each blockchain, the average *GL*, referring to the class with 15 customers, increases approximately 2.5 times compared to the value of the class with 5 customers, exactly the same increase in the number of events tracked on the blockchain.

Looking at the values of *GL* of Polygon and Ethereum in more details, we observed the same values for all the classes. However, a relevant difference is detected in the $Cost_B$ values caused by a substantial price difference as indicated in the *GP* column.

To make a preliminary assessment on the possibility of obtaining a scalable business system, it is important to consider the number of events to be tracked on the blockchain, that is *#EV*.

From the results of Tables 1 and 2, we can notice that, although in the C15 class the average cost of Polygon and VeChain increases a little more than double compared to the C5 class, the cost of a single event remains lower than 0.001$, and respecting this trend, then as the size of the instances increases, the use of the two blockchains would continue to be convenient. Consequently, the sustainability of the OD system in terms of costs seems to be really possible with at least two of the blockchains used.

6 CONCLUSIONS

In this paper, we propose a way to solve multiple issues related to VRPOD, including the great pressure of deliveries in the last-mile and the need to achieve transparency, semi-decentralization and reliability of the non-professional driver system. We introduced a VRPODTW system consisting only of occasional drivers supported by blockchain technology, in which a shipper decides to delegate some deliveries to the OD network for logistical needs or at the request of customers. The algorithm, that selects the most suitable ODs, is guided by three scores that keep track of the punctuality of picking, delivery and the number of deliveries completed. To evaluate the cost sustainability of this system, a computational study has been conducted considering three blockchains on three instance classes with 5, 10 and 15 customers. The results showed that the economic sustainability of the presented system is possible, and also that the choice of the correct blockchain, in terms of scalability and performance can greatly influence the costs of a single shipment. Furthermore, the collected computational results underlined the needs to implement a heuristic to make the interaction with the blockchain more efficient for large-scale instances. For future work, we intend to develop a fully managed VRPODTW system on blockchain, increase the size of the experiments and develop a metaheuristic to adequately manage the scalability of the system.

ACKNOWLEDGEMENTS

This work is supported by the Italian Ministry of University Education and Research (MIUR), project: "Innovative approaches for distribution logistics" - Code DOT1305451 - CUP H28D20000020006.

REFERENCES

Alnaggar, A., Gzara, F., and Bookbinder, J. (2021). Crowd-sourced delivery: A review of platforms and academic literature. *Omega*, 98:102–139.

Archetti, C., Guerriero, F., and Macrina, G. (2021). The online vehicle routing problem with occasional drivers. *Computers & Operations Research*, 127:105144.

Archetti, C., Savelsbergh, M., and Speranza, M. (2016). The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472–480.

Baza, M., Lasla, N., Mahmoud, M., Srivastava, G., and Abdallah, M. (2021). B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain. *IEEE Transactions on Network Science and Engineering*, 8(2):1214–1229.

Calabrò, R., Zanardo, E., and D'Atri, G. (2021). Cratichain: An eco-sustainable blockchain based on a light consensus.

Dayarian, I. and Savelsbergh, M. (2020). Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29(9):2153–2174.

Di Puglia Pugliese, L., Guerriero, F., Macrina, G., and Scalzo, E. (2021). *Crowd-Shipping and Occasional Depots in the Last Mile Delivery*, pages 213–225. Springer International Publishing, Cham.

Festa, P., Guerriero, F., Resende, M., and Scalzo, E. A brkga with implicit path-relinking for the vehicle routing problem with occasional drivers and time windows. In *Lecture Notes in Computer Science – MIC 2022: 14th Metaheuristics International Conference, forthcoming.*

Kudva, S., Norderhaug, R., Badsha, S., Sengupta, S., and Kayes, A. (2020). Pebers: Practical ethereum blockchain based efficient ride hailing service. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pages 422–428.

Macrina, G., Di Puglia Pugliese, L., and Guerriero, F. (2020a). A variable neighborhood search for the vehicle routing problem with occasional drivers and time windows. In *Proceedings of the 9th International Conference on Operations Research and Enterprise Systems*, volume 1, pages 270–77.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., and Laganà, D. (2017). The vehicle routing problem with occasional drivers and time windows. In *Optimization and Decision Science: Methodologies and Applications*, volume 217, pages 577–587, Cham. Springer.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., and Laporte, G. (2020b). Crowd-shipping with time windows and transshipment nodes. *Computers & Operations Research*, 113:104806.

Macrina, G. and Guerriero, F. (2018). *The Green Vehicle Routing Problem with Occasional Drivers*, volume 1, pages 357–366. Springer International Publishing, Cham.

Macrina, G., Pugliese, L. D. P., and Guerriero, F. (2020c). Crowd-shipping: a new efficient and eco-friendly delivery strategy. *Procedia Manufacturing*, 42:483–487. International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019).

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260.

Renu, S. and Banik, B. (2021). Implementation of a secure ridesharing dapp using smart contracts on ethereum blockchain. *International Journal of Safety and Security Engineering*, 11(2):167–173.

Sampaio, A., Savelsbergh, M., Veelenturf, L., and Van Woensel, T. (2020). Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. *Networks*, 76(2):232–255.

Sánchez, D., Martínez, S., and Domingo-Ferrer, J. (2016). Co-utile p2p ridesharing via decentralization and reputation management. *Transportation Research Part C: Emerging Technologies*, 73:147–166.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35:254–265.

Vazquez, E. and Landa-Silva, D. (2021). Towards blockchain-based ride-sharing systems. In *ICORES*, pages 446–452.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS