JOURNAL OF BUSINESS LOGISTICS
STRATEGIC SUPPLY CHAIN RESEARCH

Journal of Business Logistics, 2020, 1–24
doi: 10.1111/jbl.12237

Do You See What I See? A Simulation Analysis of Order Bundling within a Transparent User Network in Geographic Space

Tomas Ambra[1], An Caris[2], and Cathy Macharis[1]

[1]*Vrije Universiteit Brussel*
[2]*Hasselt University*

The Physical Internet (PI) concept presents a radical change with the aim to revert the unsustainable practices that are used for transporting goods. It identifies dedicated freight flows and transforms them into transparent open logistics networks which can be accessed by other users, such as shippers and carriers. In this paper, we test the universal network openness in which the users can tap into the PI network and place orders that will be assigned to the nearest available transport service and consequently delivered to the order sender. The objective of our paper is to investigate the impact of inserting extra service points into existing dedicated freight flows of a service-driven company. We simulate different transparency levels and routings to new pickup locations and evaluate the impact in terms of altered lead times, covered distances, and fill rates. The novel aspects presented herein are (1) deliveries based on decentralized location detection of the nearest order sender, (2) dynamically changing speed parameters of agents within specific geographic clusters based on their geo-locations in order to account for congestion levels, (3) more realistic routing strategies that consider the urban layout, and (4) transparent querying of nearest agents in space and time that meet specific conditions such as current ongoing processes, available capacity, and position. Finally, we identify the impact from a general/holistic perspective that emerges once extra orders are assigned to the service-driven company's fleet.

Keywords: agent-based modeling; Geographic Information Systems; computational modeling; Physical Internet; SYMBIT; last mile

INTRODUCTION

Freight transport and logistics of today are environmentally and socially unsustainable due to higher and more frequent demand. The ambitious climate goals and higher capacity as well as infrastructure utilization cannot be met while relying on business-as-usual methods of freight distribution. The road freight is expected to increase even further by around 40% by 2030 and 80% by 2050 (EC, 2016). Such current and projected market developments call for a radical change. The Physical Internet (PI) presents such a radical change which has the ability to reshape how we think about moving goods from origins to destinations. This need for change is addressed in the work of Montreuil (2011) who terms the current situation as the global logistics sustainability challenge. The Physical Internet (PI) concept is to identify dedicated freight flows and transform them into transparent open logistics networks which can be accessed by other users, such as shippers and carriers. In this regard, the longer dedicated freight movements should be decentralized and bundled locally based on available local assets and their parameters. Departing from the universal network openness, the users can tap into the PI network and place orders which will be assigned to the nearest available transport service and consequently delivered to the order sender. Since PI is inspired by the metaphor of the digital Internet, the goods should be routed in the most efficient way through existing links that have available capacities, just like message route via an open interconnected network in the digital Internet. Our paper focuses on how this can be done and what the implications would be when doing so. Using other existing network requires a certain level of transparency so that such a network can be identified in the first place. In this respect, the Internet of Things (IoT) connects objects via the Internet and mobile networks. These networks can be accessed by different geo-platforms which integrate GPS devices and sensor information in a server which then processes the spatial and temporal attributes of objects (vans, parcels, distribution centers) and detect other objects in surrounding areas.

The objective of our paper is to investigate the impact of inserting extra service points into existing dedicated freight flows of a service-driven company. The impact is measured in terms of lead times, covered distances, and fill rates. The assets of the service-driven company (vans) seek delivery solutions locally and deliver newly inserted orders en route. To achieve such a spatial and temporal awareness of the assets' and orders' surroundings, we combine agent-based modeling (ABM) and Geographic Information Systems (GIS) to account for decentralized parallel processes of agents in geographic space. In this regard, we use object-oriented programming where agents are depicted as objects that can roam a virtual environment. Such an environment is represented by a digital map that is today used by many users via GPS devices for finding the best route and points of interest. The model described in this paper can simulate information exchange among agents that can be queried as "things" in the IoT notion. Geo-servers that enable geo-fence and IoT integration are rather expensive, which is why our modeling approach can mimic similar transparency systems, and assess various what-if scenarios in a risk-free environment, to gain more insight before empirical tests or industrial implementations. The bundling of orders is assessed from a carrier perspective in the Brussels Central Region, from the distribution center to the final end consumer(s).

The paper contributes to the existing body of literature, described in the next section, by applying simulation modeling which depicts a more detailed time-based component (continuous flow of time) in geo-referenced space. The novel aspects

Corresponding author:
Tomas Ambra, Vrije Universiteit Brussel, Pleinlaan 2 (PL.5.4.36), 1050 Brussels, Belgium; E-mail: tomas.ambra@vub.be

presented herein are (1) en route deliveries based on decentralized location detection of the nearest order sender, (2) dynamically changing speed parameters within specific geographic clusters based on geo-locations of agents and the time of the day which determines congestion levels, (3) more realistic routing strategies that consider the urban layout, (4) transparent querying of nearest agents in space and time that meet specific conditions such as current ongoing process, available capacity, and position, and (5) identification of impact on lead times and fill rates. We simulate an integrated logistics system where orders of two different shippers/retailers are delivered by the same vehicles owned by a service-driven company. The focus is not on the end-to-end supply chain, multitier inventory sourcing, etc., but rather on the carrier transportation performance within the last-mile delivery in order to assess how such a PI business application could influence established/dedicated delivery processes. Such an application also addresses the concerns of Sternberg and Norrman (2017) who point out that all the PI-related studies do not cover return trips, and this paper is to address such a deficiency. The above contributions also relate to the call of Bell and Griffis (2010) for further research on traffic patterns, variable capacity, and en route changes.

We break the black box paradigm—where routes are predefined in advance and individual decentralized logic of entities cannot be probed during model run-time—by creating an open and transparent assignment of orders with a higher level of detail. This assignment is done based on geo-locations of moving assets, their ongoing working conditions, and spatial and temporal attributes. The study considers spatial characteristics of the built environment that govern asset (agent) movements, altered distances caused by extra service points, a temporal dimension such as the time of the day which changes agent speed parameters in a dynamic manner, and distances to existing service points, but also distances to newly inserted service points. The simulations test different logistics strategies and network design options with regard to the following:

- Order assignment to dedicated vans while considering priority and en route deliveries. $\mathbf{H_1}$: Insertion of extra website orders into existing van deliveries has a significant effect on core/priority customer orders in terms of lead times, and vans' load factors and distances. $\mathbf{H_2}$: The delivery logic (priority or en route) has a significant effect on core customer orders in terms of lead times, and vans' load factors and distances.
- A new central location that serves as a PI hub. This is to evaluate potential decrease in lead times, higher service levels, and lower environmental impact. $\mathbf{H_3}$: Location has a significant effect on core customer orders in terms of lead times, and vans' load factors and distances.
- Flexible order assignment to the nearest vans that are within a certain radius and have spare capacity. $\mathbf{H_4}$: Transparent order allocation performs better than dedicated order allocation in terms of lead times, load factors, and distances.

The motivation behind the setting is to simulate the network transparency by assuming that vans carry sensors and share information within their environment via IoT and GPS devices. Our experiments mimic geo-spatial servers that are capable of imposing conditional filtering rules and geo-fences for asset detection and message/notification exchange among entities such as moving and stationary assets. The variations caused by the three factors are statistically analyzed by factorial ANOVA.

RELATED WORK AND POSITIONING

The relevance of online order placements has been gaining more attention due to purchasing habits of customers and their requirements/preferences. The last-mile delivery presents a crucial dimension when fulfilling order deliveries in a timely manner (Esper et al. 2003; Boyer et al. 2009; Rao et al. 2011). The online purchasing context is studied by Esper et al. (2003) who point out the importance of carriers to fulfill orders within the last mile. Their findings indicate higher costumer delivery expectations when disclosing carrier information, especially well-known and established carriers such as FedEx and UPS.

The link between inventory liquidity and fulfillment guarantees related to service-driven companies is assessed by Rabinovich (2004) who focuses on the physical distribution part. In this study, the quality depends on fulfilling the right order in a timely manner to its right destination. The author stresses that Internet retailers must rely on the transportation providers (carriers and 3PLs) to fulfill promised order delivery times. As a matter of fact, the involvement of third-party logistics providers as orchestrators in supply chain management is gaining more importance (Zacharia et al. 2011).

In terms of lead times and cost, Nguyen et al. (2019) show that customers tend to adjust their preferences and accept longer lead times of online order deliveries if they come at a lower fee. Most recently, Muir et al. (2019) devise a simulation model to understand the challenge of reverse/return logistics in retail environments. Their work calls for an alignment of return policy decisions with reverse logistics design structures. Hence, last-mile routings and reroutings can build on such return policy structures once there is a certain awareness or exposure of assets with regard to where they are and what they can do during their return trips. According to Griffis et al. (2012), faster returns management can contribute to higher future online purchases as well. The Physical Internet (PI) concept could increase the efficiency of order fulfillments and improve asset utilization based on interconnectivity and information exchange.

The first formal definition of the PI (sometimes referred to as π) was introduced by Montreuil et al. (2013) who describe it as an open global logistics system founded on the physical, digital, and operational interconnectivity through encapsulation, interfaces, and protocols. Following this notion, several authors have published work within the PI context. Lin et al. (2014) devise a model for selecting standard modular containers (boxes) for a set of products. Sarraj et al. (2014) numerically demonstrate the potential of merging container flows by interconnecting logistics networks and protocols. An explicit research on π-containers has been carried out by Landschützer et al. (2015) who describe a first engineering process for developing modular and multifunctional load units within the fast-moving consumer goods industry. Pan and Ballot (2015) demonstrate the benefits of knowing asset positions via a framework to optimize the repositioning open container tracing based on radio frequency identifiers (RFID). They provide an exploratory simulation study of inventory control models in PI. Qiu et al. (2015) propose a new business model based on and IoT-enabled infrastructure. Darvish et al. (2016) link the vehicle routing problem with lot-sizing problem in order to address a more holistic production-routing problem.

One of the first pricing models in the PI context is investigated by Qiao et al. (2019) to facilitate carriers' decision making with regard to price propositions in a dynamic bidding environment for less-than-truckloads. As far as the inner π-hub operations are concerned, Kong et al. (2016) transform the auction business into a new paradigm in combination with the PI. Walha et al. (2016) study the railroad π-hub allocation problem where the π-hub is distinguished from a classical road–rail terminal by having modular and standard π-containers. Yao (2017) applies the shared and open PI logistics network in the context of optimizing one-stop delivery scheduling in online shopping. Venkatadri et al. (2016) assess the PI from a shipment consolidation perspective by analyzing traditional distribution and consolidated distribution within a European city network. Zhang et al. (2016) create a new product service system based on a smart box and propose real-time optimization via a cloud computing platform. Zhong et al. (2016) introduce manufacturing executive system that makes use of RFID for real-time data collection. Fazili et al. (2017) quantify the benefits and performance of PI compared to a conventional logistics system. Tran-Dang et al. (2017) propose a solution that has the ability to facilitate container encapsulation by detecting errors and providing updates. Sallez et al. (2016) focus on the (pro)activeness and information exchange among containers where different groping strategies within a rail terminal are tested. Yang et al. (2017b) study the impact of disruptions on hubs and factory plants and assess inventory model resilience within a PI environment of interconnected logistics services. Yang et al. (2017a) introduce a PI-based inventory optimization control model. The authors propose a vendor-managed inventory strategy where facilities and transport means are shared based on user demands. As the purpose of this paper is not to review all the Physical Internet publications in detail, we refer the reader to the work of Ambra et al. (2019) and Sternberg and Norrman (2017) who provide a more explicit overview of the current body of literature.

Nevertheless, little research has been done in terms of order deliveries in cities as most of the PI literature covers inner-hub operations and large national road networks. The first PI city application is addressed by Crainic and Montreuil (2016) who introduce a hyperconnected city logistics idea with its fundamental concepts. More recently, Ben Mohamed et al. (2017) study the urban transportation problem in a PI-enabled setting by using different types of vehicles. Chen et al. (2017) make use of the extra loading capacity of taxis to collect returned goods in a city. These existing city applications make use of directed graphs and analytical approaches. With regard to general urban/city logistics applications which are not PI-related, the review by Lagorio et al. (2016) also indicates that current applications use traditional analytical approaches which decompose system components into separate parts and are consequently assessed individually. This type of reductionism does not allow to capture effects that the separated parts have in reality (Daellenbach et al. 2012; Lagorio et al. 2016). For instance, newly incoming orders from a different order group would lead to altered lead times and distances generated by vehicle deviations. Such deviations are difficult to capture as linear programming models cannot generate such values in abstract modeling space during model executions.

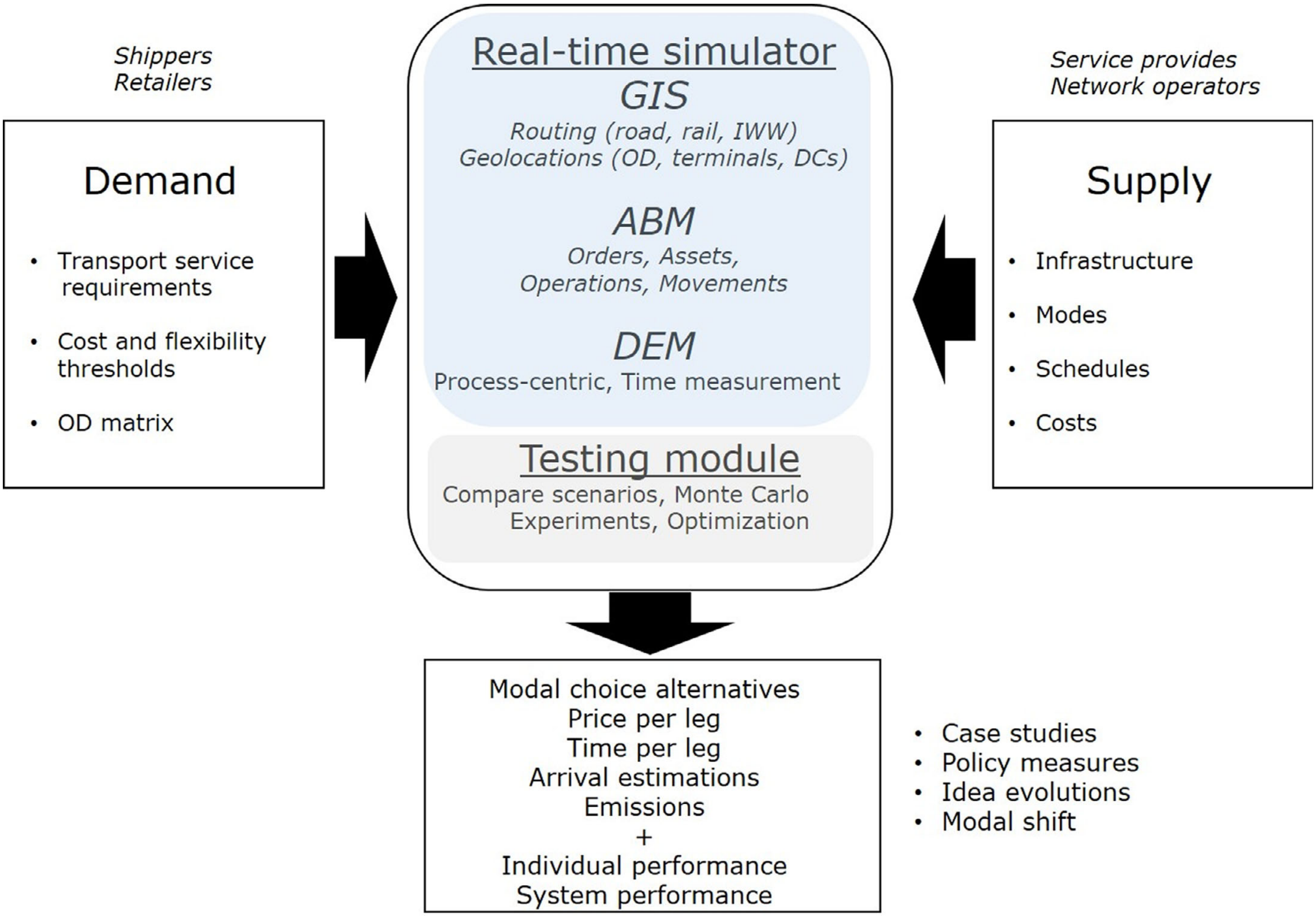
Multimethod approaches are thus necessary to introduce more details and accurate estimations of flows. The early work of Campbell et al. (2001) presents a hybrid approach combining the shortest path distance with analytical distance approximation to assist with snow disposal in Canada. Nevertheless, their work did not rely on digital road network databases due to the computational intensity of the network. Geographic Information Systems (GIS) can be very useful for depicting realistic distances and routes. In our paper, the detailed routing via road vector files (shapefiles) is possible due to the computational advancements in recent years. In terms of new approaches, Bell and Griffis (2010) compare the tradition heuristic Clarke–Wright savings algorithm with a more recent metaheuristic ant colony optimization. They provide evidence that classic approaches do not perform well when exposed to variable spatial locations that reflect real-world conditions. Our approach, depicted in more detail in the following section, accounts for dispersed spatial delivery locations that change due to customer order placements. We further include congestion levels that are missing in actual practice, as acknowledged by Donati et al. (2008) who study a time-dependent vehicle routing problem.

METHODOLOGY

SYMBIT model is a computational model, which computes freight movements based on scheduling, decentralized behavior rules, and information flows of entities/agents. The goal of the computational approach is to facilitate shippers' and also carriers' decision making by computing and estimating what-if scenarios in a risk-free environment while considering their own lead time and cost constraints.

The main modeling canvas is a digital map that comprises road, rail, and iww shapefiles. The GIS environment presented herein provides our agents with real-world locations based on the WGS84 geographic coordinate system, having Greenwich (0, 0) as its prime meridian. The reason behind choosing the WGS84 coordinate system is its broad application, used as a reference system by GPS and Google Maps as well as by Microsoft in its Bing Maps. This digital infrastructure is part of the transport "Supply" (Figure 1, right) that also includes existing services and schedules that induce agent movements. We have chosen AnyLogic software as it offers ready-to-use agent and GIS components. SYMBIT uses agents as its core elements which flow through its subparts. The parts consist of moving and stationary agents. The former are objects that flow between stationary agents such as distribution centers (DCs) and retailers/end consumers. Compared to abstract mathematical modeling environments, agents have the ability to roam geographic space and record data. The advantages of using GIS for these processes lie in more accurate and realistic routing strategies, higher level of detail, detection of events in space and time during execution, and efficient response actions that are facilitated by location intelligence. In other words, the moving agents collect distance data that they have covered or are about to cover. The data

Figure 1: General overview of SYMBIT's composition together with its inputs and outputs.



collection may be done dynamically without switching manually to Google Maps or other providers to acquire distances. Each agent also possesses a speed parameter that governs the agent's speed throughout its movement. Although there exist other approaches that utilized GIS to depict the geographic nature of transport problems (Campbell et al. 2001), our approach builds on the advances in computation that allow for decentralized distance calculations during model executions (not before or after). More specifically, technological developments have induced a move from ESRI's ArcMacro, ArcView, and AML to industry-standard programming and scripting languages such as Java, C++, Visual Basic, JScript, and Python which have the ability to incorporate GIS software libraries such as OpenStreetMap, Landsat, and GeoTools (Crooks and Castle 2012). GIS serve as a medium for communicating results and assessing patterns which we will generate by simulation runs. In general, GIS present a modeling canvas full of geocoded information and location intelligence which facilitate the movement of agents and contribute to more informed decisions.

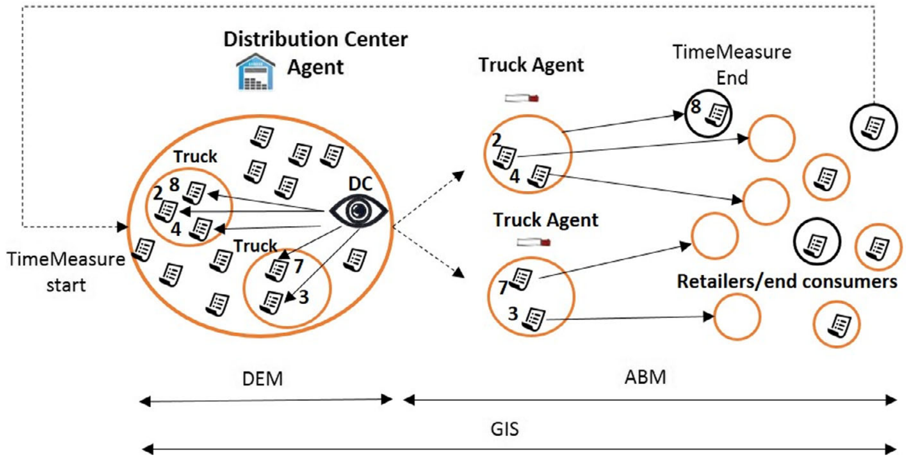
Model formulation

Figure 2 represents a conceptual overview of an order flow starting at a distribution center agent and ending at a destination node which can be a retailer or an end consumer. In-house DC processes (stationary agents) comprise discrete-event modeling, whereas movement between stationary agents is carried out by moving agents (vans, cars, etc.). GIS provide location and routing attributes to moving and stationary agents. SYMBIT continuously monitors the state of agents from the point when the order agent is sent to the DC (Figure 2, left). It then logs the type of transport means the order is carried by, covered individual distances, dwelling time, and elapsed delivery time which stops the monitoring process of a specific order.

It is assumed that each agent possesses sensors that serve as input connectors via which events $e \in E$ can be read. We make use of state charts, developed by the AnyLogic company, which are the most advanced constructs capable of describing time- and event-driven behaviors. The triggering events E induce functions which represent code snippets that are embedded in agents' states $\omega \in \Omega$. These function snippets contain decentralized algorithmic logic (depicted in the following sections) and initiate transfers between different ω_i and ω_j states.

Stationary agents ($\mathbb{S}$) have a fixed location in space and time. The GIS environment (Figure 3) is populated with four groups of stationary agents labeled as Customer1 ($G = \{0, 1, \ldots, 243\}$), Customer2 ($B = \{0, 1, \ldots, 214\}$), wholesalers ($W = \{0, 1, \ldots, 18\}$), and distribution centers ($DC = \{0, 1\}$).

Figure 2: Illustration of SYMBIT's stationary (DCs, retailers) and moving (vans, cars) agents.

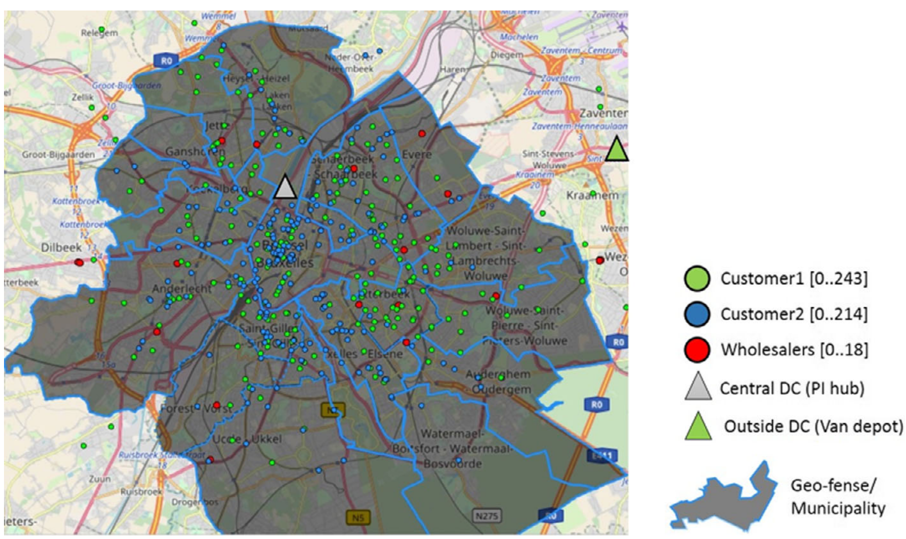


Locations of G (Figure 3, green) represent core customers of the logistics service provider (LSP). B locations (blue) represent store-owners and are taken from the Atrium database—the Chamber of Commerce of the Brussels-Capital Region. Locations of W are selected wholesalers (red) where each $b \in B$ tends to replenish its store. The last two locations are $dc_p \in DC$ agent, which corresponds to a PI hub (gray triangle), and $dc_o \in DC$, which is a van depot of the LSP (green triangle) that governs a fleet of 20 assets. For all locations, a "Google Maps Geocode API" is used to acquire latitude and longitude. These coordinates determine geo-locations where $\mathbb{S}$ entities are created.

Moving agents ($\mathbb{M}$) roam the geo-referenced environment between $\mathbb{S}$ agents. The model distinguishes three types of $\mathbb{M}$ agents: the fleet of assets $V = \{0, 1, \ldots, 20\}$ operated by dc_o; a fleet of cars $C = \{0, 1, \ldots, 214\}$ operated by $b \in B$ assuming that each b owns a car; and lastly, two sets of orders Og and Ob where every $og \in Og$ is generated by group G and $ob \in Ob$ by group B. The main focus of our work is on the V group that delivers Og orders to G locations as the purpose of this paper is to assess the vehicle performance after inserting extra orders (Ob). Each individual moving agent $v \in V$ will take decentralized decisions based on their surrounding context. The decisions will be affected by the following sets and parameters that will be used in our flowcharts in Section 4 and pseudocodes in the appendix:

$Og_{collection} = \{0, 1, \ldots, n\}$ = set of onboard orders $og \in Og$
$Ob_{collection} = \{0, 1, \ldots, m\}$ = set of onboard orders $ob \in Ob$
p_s = position determined by longitude and latitude
n_g = nearest g location
n_b = nearest b location
dn_g = distance to nearest g location
dn_b = distance to nearest b location
A_v = Boolean parameter indicating availability
S_{ch} = Boolean parameter related to schedule
v_c = vehicle maximum capacity
s_p = speed parameter in km/hr

Figure 3: Study area depicting the geographic region of Brussels and its municipalities.



The parameters and variables listed below compose the main KPIs generated by the model:

- vd = total distance covered by each $v \in V$
- T_{vd} = total distances covered by fleet V
- v_{lf} = vehicle load factor
- Ld = lead time of an order (og and ob)

Equation 1 summarizes all the distance that will be generated by every van agent. As the distances are not predefined in advance, and the nearest locations are changeable as they depend on the geo-local context of each v agent, equation 1 will summarize the covered distances upon the van's arrival at dc_o. The n_g and n_b are taken from the $Og_{collection}$ and $Ob_{collection}$ from which the van agent v queries orders and compares distances to their delivery locations from a decentralized perspective after each delivery. The return-distance is depicted by $dist||dc_o - 1 \rightarrow dc_o||$ which means that the distance to the dc_o is calculated from the last known location of the van before returning to dc_o (this can be dn_{gn} or dn_{bm}, depending on the imposed delivery policy). Equation 2 then consequently calculates the overall distances generated by the LSP's fleet (V).

$$vd = \sum_{i=1}^{n}(dn_{gi} + \ldots + dn_{gn}) + \sum_{j=1}^{m}(dn_{bj} + \ldots + dn_{bm}) + diszt||dc_o - 1 \rightarrow dc_o|| \quad (1)$$

$$T_{vd} = \sum_{i=1}^{20} vd_i \quad (2)$$

Contrary to equation 1 which is van-specific, equation 3 is order-specific; it accumulates order lead time of og (Ld_{og}) determined by the van's travel time and its working conditions (loading/unloading). The same applies to external orders where og is replaced by ob in equation 3. As the deliveries will depend on the policy (priority or en route) introduced in Section 4, x in dn_x can be any proceeding order location that has higher priority or is closer than $og \in Og_{collection}$. The real-time simulator records the delivery time upon van's arrival at the order's geo-location and subtracts the order placement time represented by an event e that occurred at time t (e_t). The dn_g represents the location which is embedded in each og.

$$Ld_{og} = \sum_{i=1}^{og}(\frac{dn_{xi} + \ldots + dn_g}{s_p}) - e_t \quad (3)$$

The final parameter determining the Ld is s_p that governs how fast a van moves through the GIS space. In this regard, Section 3.2 elaborates on how the speed is dynamically adjusted via geo-fences, and Section 3.3 sheds more light on the load factor calculation (v_{lf}).

Geo-fencing and speed adjustment

When V agents move through the environment from/to $\mathbb{S}$, it is unrealistic to use a constant average speed parameter. Therefore, we deploy geo-fences that probe the s_p parameters of vans. In order to determine what speed corresponds to a municipality, we sampled historical data from commercial map providers. The s_p parameter that governs the movement of $\mathbb{M}$ is deduced from distances and elapsed times by applying basic physics

$$s_p = \frac{d}{t} \times 3.6 \quad (4)$$

, where d is distance in meters and t is elapsed time in second. Factor 3.6 is used as a conversion to acquire km/hr. We measure d within every cluster from the cluster's four edges to the centroid. The t is obtained from the commercial map provider who offers elapsed time given traffic congestion during different times of the day. Alternatively, speed profiles can be used for governing $\mathbb{M}$ agents' s_p parameters by the properties of the route polylines. However, these polylines contain maximum allowed speeds but no realistic daily speeds. The proposed geographic clusters are characterized by a list of latitude–longitude pairs. Since $\mathbb{M}$ agents may roam this geographic space, the s_p parameter can by dynamically probed while moving; once an agent enters a certain geo-fence, a matching algorithm is deployed to compare x- and y-coordinates of the agent's location with the cluster's latitude–longitude pairs to identify which speed each agent has to adapt. Another parallel algorithm monitors the real-time simulator in order to switch speed values according to the time of the day (Table 1). This ensures that morning, noon, and afternoon deliveries reflect realistic congestion levels. The routing and service points are not predefined in advance since O_b orders can be delivered by a different agent based on its current position and ongoing process in space and time. For this reason, cluster speeds may be more advantageous compared to predefined individual speed profiles per entity. The van agent's speed logic could be also linked to an API that fetches real-time data; however, this approach goes beyond the scope of our paper.

Object detection and load factor calculation

Given the decentralized nature of agents, load factor calculations may be carried out bottom-up without averaging, hence losing, individual specificities that are generated at the local level. In other words, equation-based or analytical models do not generate movements themselves, but use observables that "are moved" by the high-level system. Van agents store their distances they covered but also the objects they carried and unloaded. Given such an object-oriented approach, the vehicle load factor v_{lf} is calculated as

$$v_{lf} = \sum_{i=1}^{n}\left(\frac{D_{lfi}}{vd} \times L_{fi}\right) + \ldots + \left(\frac{D_{lfn}}{vd} \times L_{fn}\right) \quad (5)$$

, where L_{fi} is load factor of the first delivery, D_{lfi} is distance traveled with L_{fi}, and vd is the total distance. To elaborate further on this, example A in Figure 4 depicts the vehicle load factor as $v_{lf} = (0.5 \times 80\%) + (0.5 \times 0\%) = 40\%$. However, the number of stops may vary as individual vans receive a different amount of generated orders. In this regard, example B provides a more accurate representation that is relevant to our case study.

However, the load factor does not always have a decreasing trend in function of distance and unloading, but can be increased

Table 1: Speed parameters (km/hr) per each geographic cluster during a given time of the day

Cluster	8–10 hr	10–12 hr	12–14 hr	14–16 hr	16–18 hr	18–20 hr
Brussels	10.4	11.4	12.9	12	14	12.8
Schaerbeek	16.5	16.5	16.5	18.5	14.6	16.5
Etterbeek	12.6	15.7	15.7	15.7	12.6	12.6
Ixelles	13.3	18.4	18.4	18.4	16	17.1
Saint-Gilles	13	13	13	13	12	13
Anderlecht	16.8	13.5	13.5	12.4	12.4	16.2
Molenbeek-St-Jean	15.7	18.7	18.7	16.6	13	17.6
Koekelberg	14.4	13.1	13.1	14.6	12	13.1
Berchem-St-Agathe	18	18	16.6	15.4	14.4	18
Ganshoren	13.2	16.5	16.5	14.6	13.2	16.5
Jette	20	17.1	18.4	17.1	16	17.1
Evere	18.4	18.4	17.3	19.7	16.2	18.4
Woluwe-St-Pierre	27.6	30.6	30.6	30.6	30.6	25.1
Auderghem	14.5	17	20.4	18.5	12.6	20.4
Watermael-Boitsfort	22.3	24	24	24	18.4	24
Uccle	16.2	40.5	40.5	21.6	19.1	27
Forest	18.9	22	22	18.5	13.2	20.3
Woluwe-St-Lambert	18.5	22	26.4	22	26.4	22
St-Josse-ten-Noode	10	12	13	11.1	7.8	13
Ringroad	31.4	61	61	54.9	36.6	54.9

in case the v departs to the centrally located dc_o to collect Ob orders; after the last destination (D3), the load factor will increase once the van collects an order from the central DC until the extra customer destination (not shown). The overall v_{lf} per trip is calculated upon the van's return to the DC once we know the total distance. In our case application, one order represents 20% of the vans capacity which makes the total van capacity (v_c) limited to 5 orders.

EXPERIMENTAL DESIGN AND RESULTS

To demonstrate the abilities of SYMBIT, the model is applied to a case which considers existing vans with spare capacity that are physically present in the city (Figure 5, number 4). These vans transport "air" most of the time, which is why orders from B locations are considered to utilize the spare van capacity and prevent B group from going to the nearest wholesaler $w \in W$ to replenish (number 3). The $c \in B$ car trip (number 2) from its origin location $b \in B$ (number 1) can be avoided by placing an order (o_b) via a retail platform; the order will be consequently delivered by the van (v) of a service-driven company (dc_o). This prevents the customer from set B from using his/her car to travel to the nearest wholesaler to replenish. The initial location of cars C is evenly distributed among B locations, assuming that each store-owner has one vehicle at his/her disposal.

Van agents V are the core elements governed by the experimental design depicted in Figure 5. The study uses order generation based on Kin et al. (2018); one single customer in G group generates demand up to three times per day, whereas stores B replenish 3–6 times per week by going to the nearest wholesaler. We converted this rate to daily generation between 0 and 2 times as our simulations terminate after one day. The model run-time is confined to one day given the computational power required for agent speed control and geo-fencing. The general experimental design consists of four simulations (Figure 6) that are textually described in the following subsections. More detailed pseudocodes are provided in the appendix.

The simulation descriptions and results are streamlined in a form of "simulation description→ results, simulation description→ results, etc." The simulation output produced by our computational model is assessed by a statistical analysis. This is carried out by means of ANOVA as running multiple t-tests would increase the type 1 error rate (alpha error)—higher probability that we will reject the null hypothesis when it is in fact true. We thus used a factor significance test ANOVA to assess the interaction effect of the factors on the dependent variables. The following subsections describe these effects per dependent variable. Significance level of .05 is used which relates to confidence intervals of 95%. Furthermore, Bonferroni confidence interval adjustment is deployed to compare main effects of the mean values. To ensure comparability, each simulation is executed by using a fixed random seed for numbers taken from the uniform distribution functions which mainly concern the demand generation function; this is to account for reproducible and comparable simulations. Hence, parametric variations are taken into account and are consequently assessed statistically. In this regard, our model does not include parameter values with stochastically fluctuation values that would require multiple model runs and Monte Carlo experiments.

The simulation logic and results are presented in the following order: The status quo is reproduced in Section 4.1. Simulation 1 description and results will be conveyed when inserting extra service points and carrying out deliveries en route or priority

Figure 4: Conceptual overview of load factor calculations for a vehicle round trip. O and D depict origins and destination.

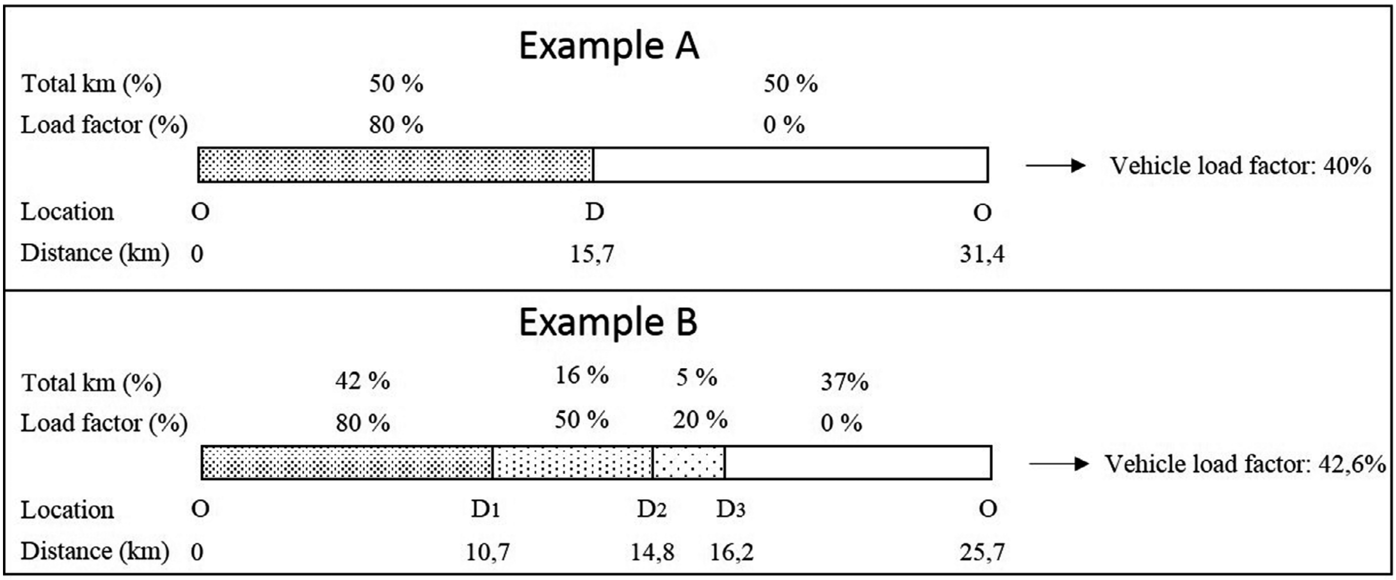
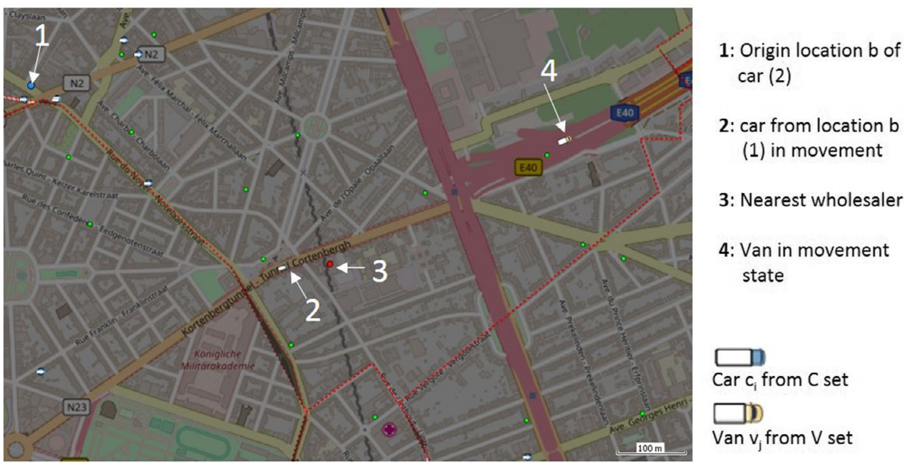


Figure 5: An illustration of overlapping flows that are to be eliminated.



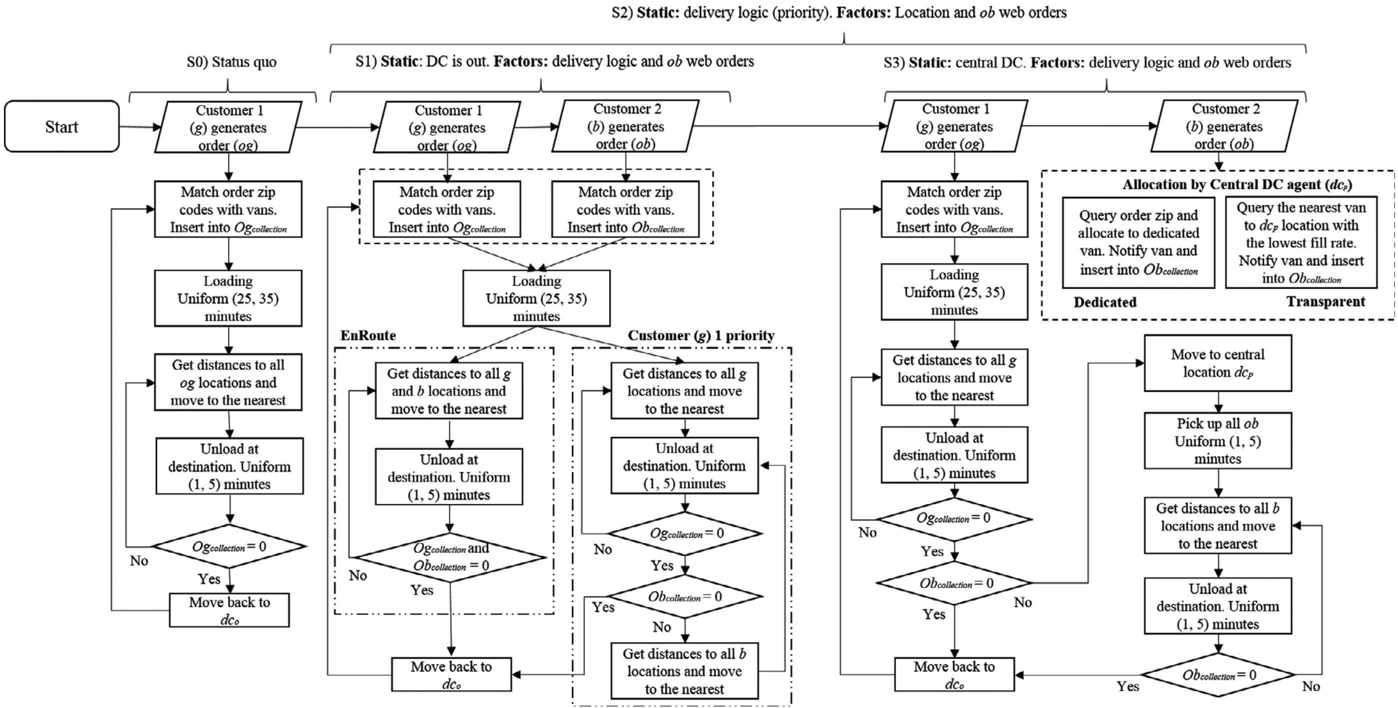
(Section 4.2). Thirdly, the extra *Ob* order insertion and results with different DC locations will be presented in Section 4.3. Lastly, Simulation 3 logic and results are described where the focus is on transparent allocation and its impact (Section 4.4).

Status Quo (S0)

This subsection reproduces the base scenario that will be later used for comparing the implications of different parameter variations. The business-as-usual case (status quo) depicts *B* agents replenishing at the nearest wholesaler *W* by using their own assets *C*. In parallel, van agents (*V*) deliver to their customer locations (*G*) who generate orders (*Og*) with embedded geo-coordinates of their origin. This demand is defined by a uniform distribution function with bounds between 0 and 3. The *V* agents are initiated by a service schedule S_{ch} that starts at 8am, and *C* agents initiate according to schedule WG_{sch} which indicates opening hours of *W* from 08:30 until 20:00.

The van agents are initiated by the service schedule (S_{ch}) that starts at 9am, and they consequently follow processes displayed under status quo depicted in Figure 6. Once a matching algorithm assigns orders to corresponding vans based on zip codes, the van agent calculates all distances to the order destinations and departs to the nearest one by route (the road layout is considered and not a straight line). After unloading, it recalculates all order distances from its new geo-location and moves to the next nearest order location. If the van's order list queue ($Og_{collection}$) is empty, the van agent returns back to the dc_o. The calibration and validation of van agent movements were conducted in comparison with operational log files provided by the service-driven company. The log files contained detailed time stamps regarding the vehicle stops, ignition on and off events, closing and opening of vehicle doors, and delivery durations between stops. Such time events enabled us to calibrate and validate the agents to closely mimic realistic delivery conditions.

Figure 6: Schematic overview of the business-as-usual case (status quo) and 3 experimental simulations and their composition.



Pseudocodes in Appendix A.1 provide more detail with regard to the simulation logic. Having established these dedicated flows, 3 factors will be introduced in order to assess how the current structure would be impacted and whether the impact is of significance. The factors, evaluated in the following subsections, are location (outside, central), external website orders (10%, 25%, 50%, 75%, 100%), and delivery logic (priority, en route, dedicated, transparent). The impact is measured in terms of distances, lead times, and fill rates.

Simulation 1

Simulation 1 (S1) varies the delivery logic factor and *ob* Web orders that are placed by Customer2 group from *B* locations. The order (*ob*) placement and replenishment processes of *B* agents are executed at the same time. Once the online order is received at dc_o, the matching algorithm inserts the *ob* agent into $v \in V$ that corresponds with the ZIP code. Algorithm 3 thus provides input for Algorithm 4. We refer more interested readers to Appendix A.2 that contains the two algorithms as well as more information on how the *ob* variations are computed. As far as the delivery logic is concerned, S1 applies priority-based and en route deliveries (Figure 6). The former, priority-based, means that individual *v* agents (vans) deliver to their core customers first ($og \in Og_{collection}$). As soon as the $Og_{collection}$ is empty, which is an array list of *og* orders that the van has onboard, the extra online orders ($ob \in Ob_{collection}$) are delivered to their corresponding *B* locations. The van agents consequently return back to dc_o. In the latter setting, en route, a decentralized algorithm queries all the order locations (*b* and *g*) in geo-referenced space, and the van agent recalculates and compares distances to each one of them from its individual location which changes after every order delivery. In both cases, priority and en route, the van agents have loaded the extra *ob* Web orders at the dc_o. Based on the simulation composition, certain assumptions are made to set the scene in terms of expectations. These expectations can be accepted or rejected by our simulations. Thus, the following hypotheses are posed:

> ***H1*** **:** *Insertion of Ob* website orders into existing van *V* deliveries has a significant effect on core customer *Og* orders in terms of lead times, and vans' load factors and distances.

> ***H2*** **:** *The delivery logic (priority or en route) has a significant effect on Customer1 Og* orders in terms of lead times, and vans' load factors and distances.

S1 Results and Discussion: Impact of extra service points and delivery logic

Let us recall that the impact is studied from the LSP's perspective and the results indicate performances of *V* agents (distances, load factors) and *Og* orders (lead times). Simulation results for lead time (Figure 7, left) show a significant increase in Customer1 order deliveries when inserting extra service points of *B* locations (p = .000). In order to establish the stage at which the variance becomes significant, pairwise comparison is used that indicates 75% of orders as a threshold (p = .008) when the extra service points start to have an impact on the core *Og* deliveries. This means *Og* orders are not substantially affected if the service-driven company receives 50, 25, or 10% of website orders from *B* locations. In terms of delivery logic, Figure 7 (left) shows slightly better performance when *Og* orders are delivered as priority, and the delay increases with en route as vans attend

Figure 7: Average lead time of *Og* orders in hours after inserting *Ob* website orders of *B* locations in combination with delivery logic (left) and van *V* load factors (right). The horizontal axis represents the percentage of inserted orders.

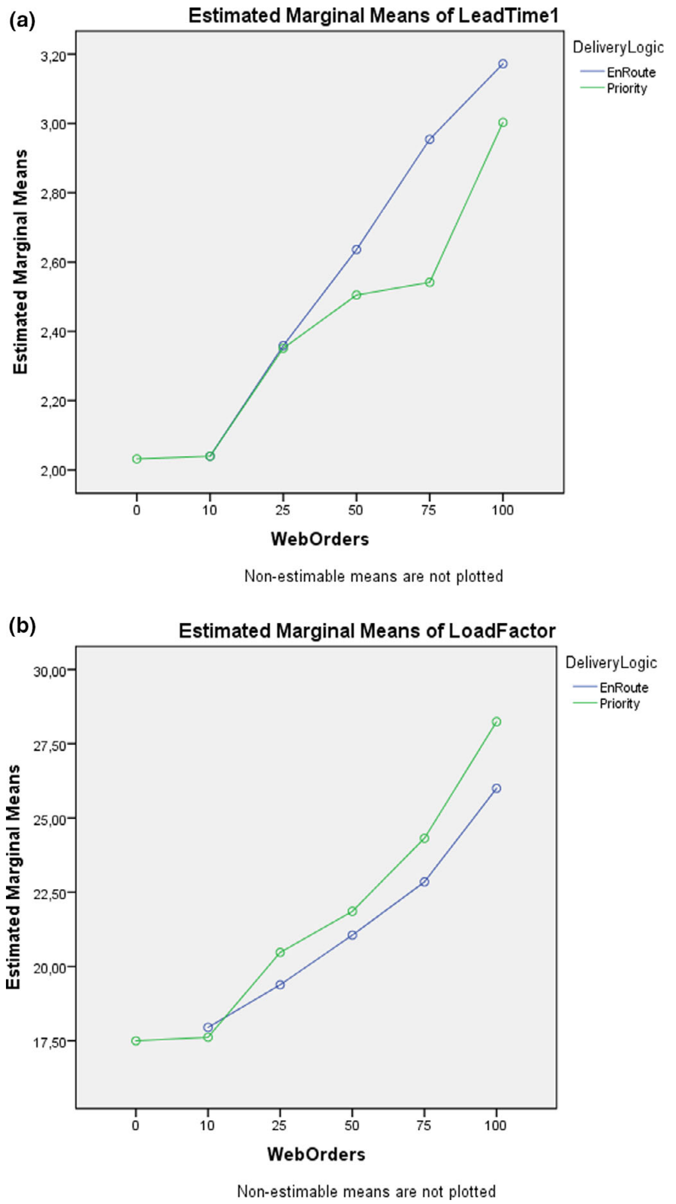
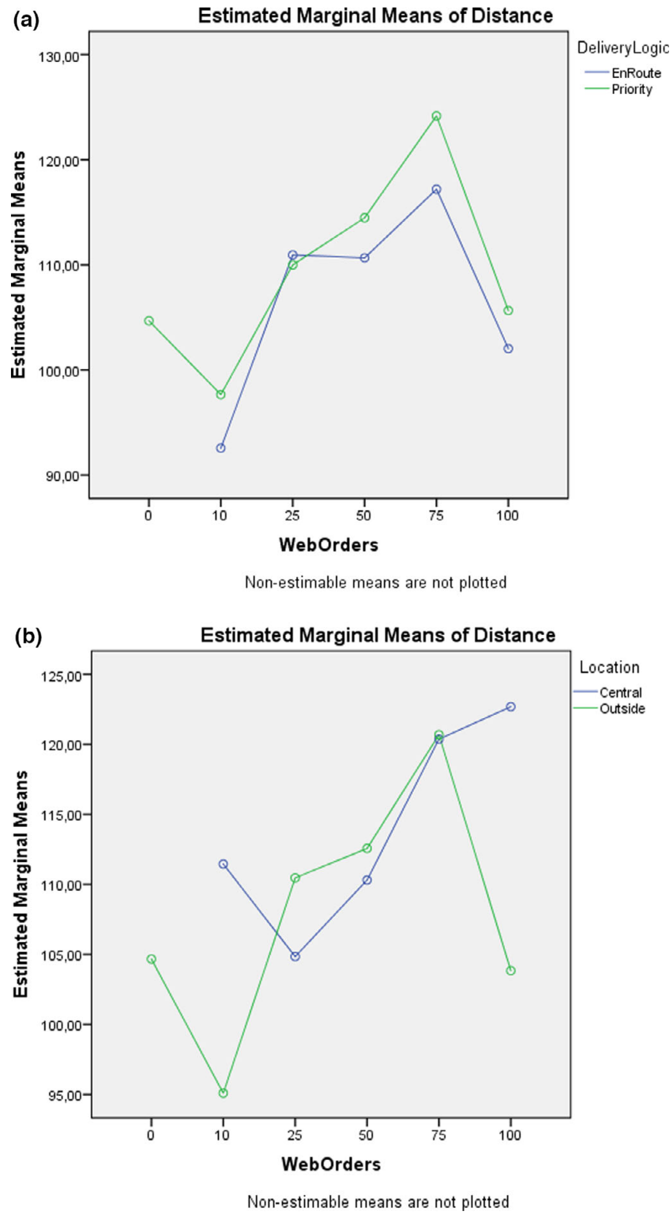


Figure 8: Average distances of vans in kilometers after inserting *Ob* website orders in combination with delivery logic (Simulation 1) and location (Simulation 2).



to *B* locations of *Ob* orders first, if they are closer. This slightly increases the delay of core orders *Og* (blue line) since vans deviate from their original route. However, the variance between en route and priority is not significant (p = .177).

Figure 7 (right) depicts van load factor variations. Website orders (p = .216) and delivery logic (p = .688) have statistically no significant effect on the van load factors when following priority or en route delivery logic (left). Furthermore, both order groups are delivered from the outside DC location (dc_o), as a result of which vans still cover a lot of empty kilometers on their return trips.

En route deliveries are visually more efficient in terms of kilometers (Figure 8, left) but the statistical analysis yields insignificance (p = .743). The rapid decline in distance from 75% to 100% can be explained by the spatial attributes of the urban layout and the routing algorithm used by our vans. The vans follow the fastest route, and a smaller amount of orders located on the western side of Brussels will result in vans taking the ring road to reach the order's delivery geo-locations. However, with more orders entering the scene, some vans are "dragged" closer to the center of Brussels from the clusters' peripheries. This is an interesting emerging phenomenon that causes the vans to ignore the ring road and take the inner roads instead, when returning back to dc_o. To address our posed hypotheses, website orders of *B* locations do have a significant effect on core *Og* lead times starting from 75%, but no significant effect is observed regarding load factors and distances. The delivery logic does not significantly affect lead times, load factors, nor covered distances.

Simulation 2 (S2)

Unlike the previous simulation (S1), the *ob* Web orders are not loaded at the dc_o but must be collected at the central dc_p location. The central location is perceived as a new PI hub which is located in the port of Brussels. S2 thus measures the performance of the van fleet (*V*) when deviating to the dc_p. As shown in Figure 6, once all core customer orders ($og \in Og_{collection}$) are delivered and there exist *ob* orders to be collected by the van that services the geo-cluster/municipality, the van does not return back to dc_o, but drives to dc_p. It then collects the *ob* orders, delivers them to their *b* locations by choosing the nearest, and returns back to dc_o. Appendix A.3 sheds more light on the procedure and S2 output. Our third hypothesis concerns the *DC* location (**H$_3$**):

> **H** : *Location has a significant effect on core Og* orders in terms of lead times, and vans' load factors and distances.

S2 Results and Discussion: Impact of deviating to central location

As far as the location factor is concerned, distance variations are also not significant (p = .472). Although there is a visual difference once the vans deviate to the central dc_p to collect *Ob* website orders (Figure 8, right), the variations are negligible. These variations in van distances do not follow a clear pattern. En route deliveries are visually more efficient in terms of kilometers, but the statistical analysis yields insignificance (p = .743). Website orders also do not affect traveled distances (p = .792). This is a promising results showing that extra orders do not necessarily generate substantially more driven kilometers if vans collect *Ob* Web orders from the central dc_p location (PI hub).

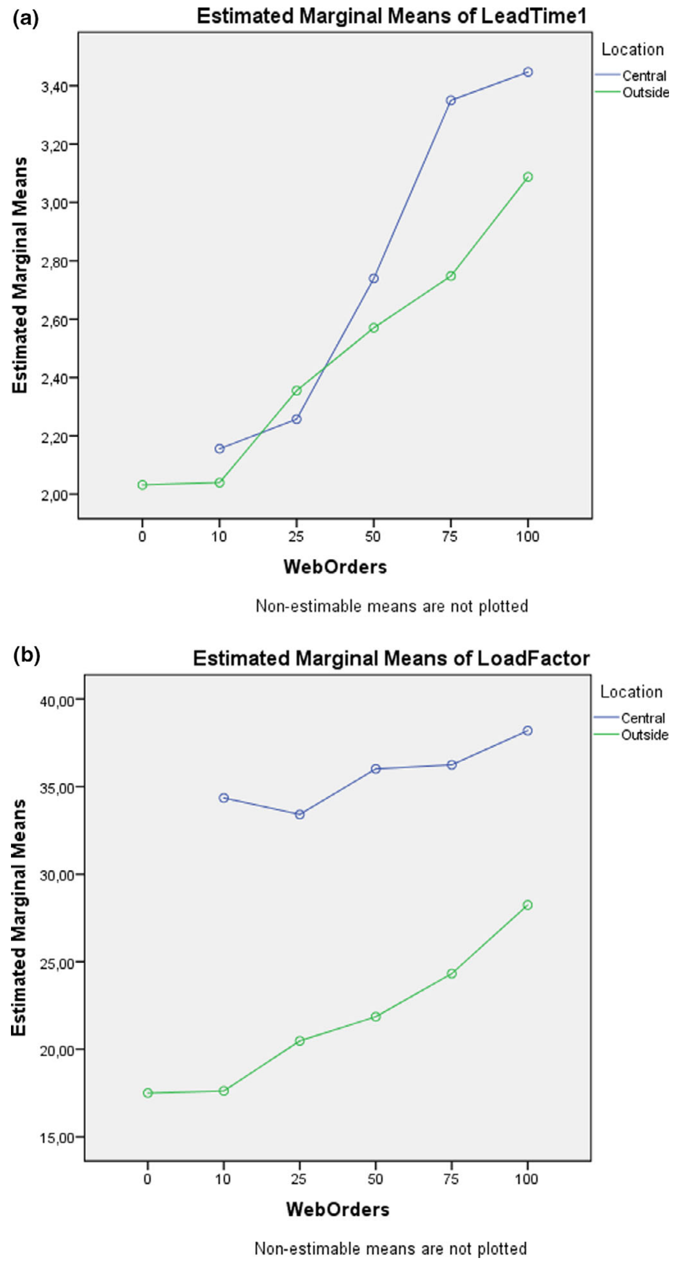
Regarding the delivery times (Figure 9, left), lead times increase once vans do not load *Ob* orders at their depot (outside dc_o location) but need to travel to the central dc_p location to collect the orders. The deviation leads to a significant increase in core *Og* order lead times (p = .012) as vans spend more time in the city and return back later to their depot to collect new *Og* orders.

Unlike in S1 where the load factor did not change at all, in S2 the load factor increases significantly (p = .000) when vans decrease empty kilometers by going to the central dc_p (Figure 9, right) and consequently deliver *Ob* orders. In other words, the vans cover less kilometers transporting "air." In relation to our hypothesis for S2, the location does have a significant effect on lead times and it substantially increases load factors of vans without significant variations of covered distances. It can be inferred that the LSP's assets may cope sufficiently with an influx of *Ob* orders until 75% without severely affecting *Og* lead times. Load factors increase significantly more when collecting extra orders from the central location.

Simulation 3 (S3)

The last simulation (S3) omits dc_o and considers only the central dc_p location. It simulates 2 different order allocation and notification schemes as indicated in Figure 6: Firstly, the central dc_p (PI hub) notifies dedicated vans that serve the specific clusters/municipalities where the *Ob* come from. Secondly, the *Ob* allocation is carried out in a transparent manner when the central dc_p has the ability to spatially detect the nearest moving agent, but also the number of orders/objects the van has onboard. It then requests the nearest van agent with the lowest number of orders onboard. This case works with an assumption that vans carry sensors and share information within their environment. In this regard, "Do you see what I see?" refers to the level of transparency the 2 allocation schemes in S3 have, in which the first allocation logic (dedicated) is not aware of nearest vans and their parameters and the second allocation logic (transparent) is. The pseudocode is provided in Appendix A.4. The last simulation is to test our fourth hypothesis (**H$_4$**):

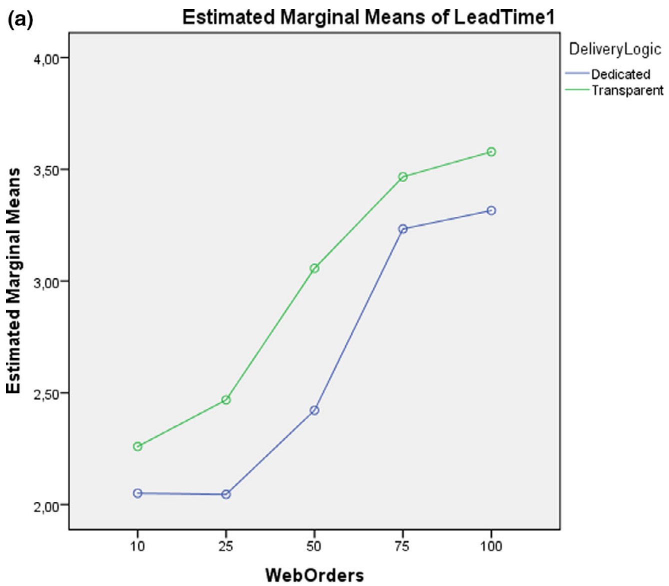
Figure 9: Average load factors of vans in % after inserting website orders of Customer2 in combination with delivery logic (left) and DC location (right).

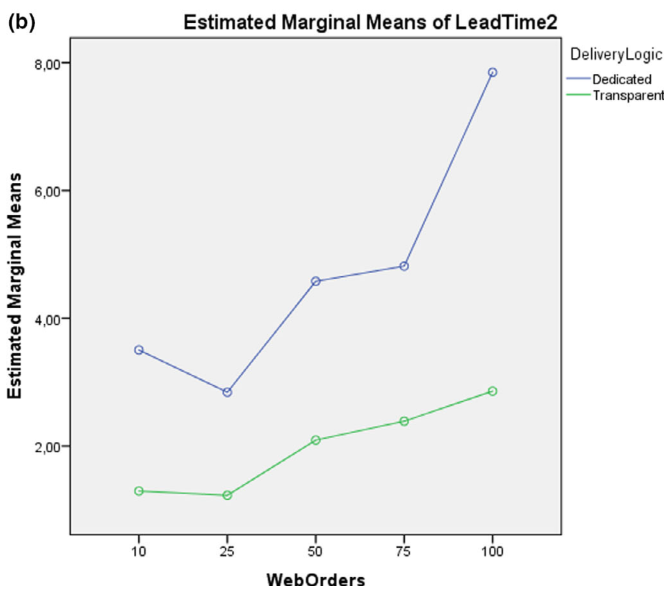


H : *Transparent allocation/delivery logic performs better than dedicated allocation/delivery logic in terms of lead times, load factors, and distances.*

S3 Results and Discussion: Impact of spatial detection and transparent allocation of orders from a central (PI) location
This section concerns dedicated and transparent order assignments/deliveries from the central location with a goal to evaluate which dependent variables benefit from transparency and transparent order allocation to vans with exposed parameters such as fill rates and geo-locations. Also the lead times of *Ob* are displayed, to demonstrate the impact of transparent allocation on both order groups. The previous sections excluded *Ob* lead times as the focus was on the *Og* and LSP's fleet performance, whereas in this subsection, we intend to demonstrate how transparent allocation impacts both groups within the PI context. In other words, the *Ob* lead times serve as an argument when comparing both lead times in order to justify why the dc_p could or should be located more centrally. In the previous section, we established that the amount of Web orders has a significant effect on *Og* lead times starting from 75%. Figure 10 (left) illustrates that transparent order allocation (green) to the nearest van with the lowest fill rate causes a significant (p = .020) increase in *Og* order lead times. On the other hand, transparent order allocation has stronger significance (p = .000) by decreasing *Ob* order lead times (right). Based on the visual inspection of the figure below, transparent deliveries are less severe for core *Og* orders (left) than dedicated deliveries for website *Ob* orders (right). In this regard, transparent allocation can benefit mainly the group of *Ob* orders with a slightly lower delivery performance for the *Og* orders. This development could attract more users to join the PI open network due to better service levels, and additional delays of core *Og* orders could be offset by additional revenues generated by new customers. Extra fleet or additional service-driven companies could also mitigate the burden of extra order influx.

Figure 10: Average lead times of *Og* (left) and *Ob* (right) orders in hours after inserting website orders of *B* locations in combination with delivery logic.

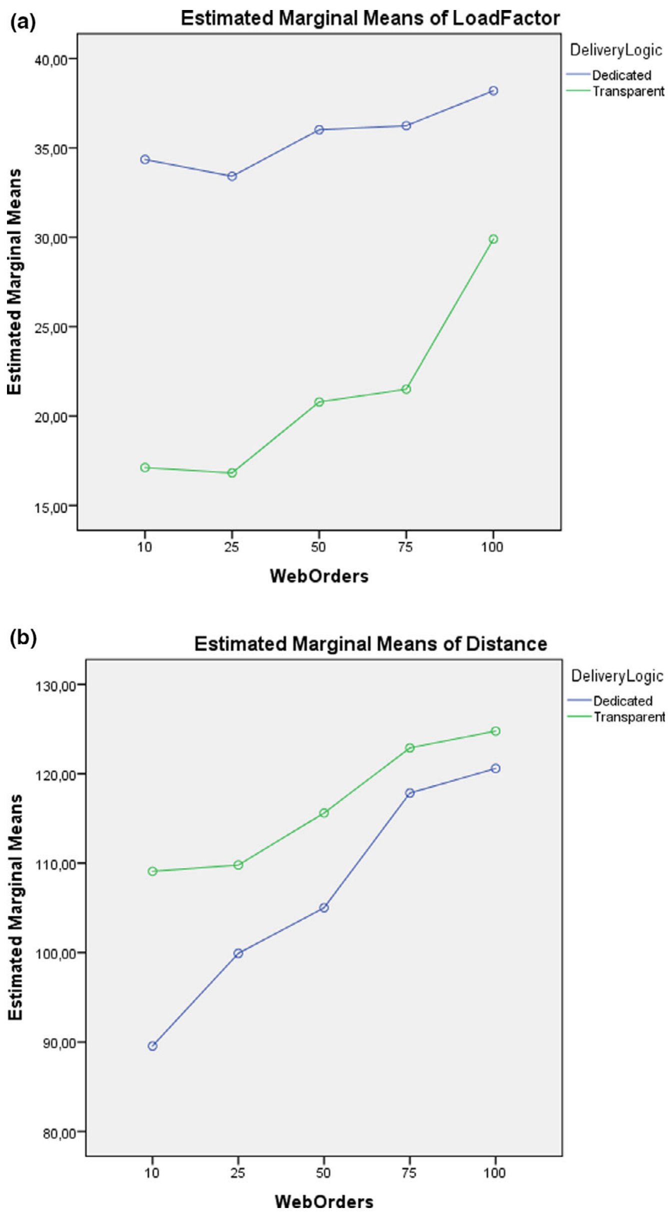




In terms of load factors (Figure 11, left), the amount of Web orders does not cause wide variations (p = .429), as also described in the previous section. However, the delivery logic does affect the load factor (p = .000) where transparent allocation, perhaps surprisingly, decreases the load factor compared to dedicated allocation. This can be explained by the already low amount of orders in the vans that were passing by the central DC. In other words, orders that would be normally allocated to dedicated vans, hence increasing overall fill rates of rather full vans of specific clusters, are now allocated to nearly empty vans that serve not so dense clusters. This setting thus contributes to more kilometers traveled with lower fill rates. A testament to this fact is provided by the same figure (right) which indicates that vans start generating more kilometers as the central location agent assigns vans to *Ob* orders which may be located in two different clusters, subsequently increasing vehicle kilometers.

As far as $\mathbf{H_4}$ is concerned, transparent allocation has a significant effect on lead times as well as on vans' load factors. Distances are not substantially affected by the transparent delivery logic. The transparent allocation does perform better for *Ob* Web orders but does not perform better for the core *Og* group and van load factors. The transparent allocation can be perceived as a good selling proposition in terms of *Ob* lead times at the expense of lower fill rates and slightly more kilometers. The fill rate variations are significant, but the extra distances are not. From an environmental point of view, a rather stable amount of kilometers and lower fill rates may emit less pollutants compared to more loaded vans. In fact, the transparent delivery logic is faster for *Ob* group of orders, does not yield significantly more kilometers, and could emit less pollutants. From a general perspective, the vans are still present in the city anyway, but they eliminate car ($c \in B$) vehicle kilometers (vkm) to the nearest wholesaler (Figure 12). It can be observed that despite the vkm increase in vans caused by extra service points (blue), the car vkm (orange) of *B* locations overcompensates for this increase, which, in fact, results in a general decrease in total vkm from a general perspective (gray). Hence, such a development may lead

Figure 11: Average load factors of vans in % (left) and average distances of vans (right) in kilometers after inserting website *Ob* orders of *B* locations (left).



to a reduction in the amount of vehicles in cities and less external effects such as congestion, noise, and emissions.

Having a reliable and fast service offer will convince new users to place an order within a PI-like network instead of using their own transport means for individual replenishments. An alternative to this approach is to follow the dedicated allocation which generates higher fill rates, but also significantly higher lead times for newly incoming *Ob* orders that could consequently cause a reverse effect and decrease the probability of customers from the *B* locations to place an order online. However, this would be at the expanse of slightly lower service levels for the core *Og* orders; in this regard, the core customers (*G* locations) would have to relax their strict time windows and accept slight delays caused by deviations to *B* locations.

CONCLUDING REMARKS

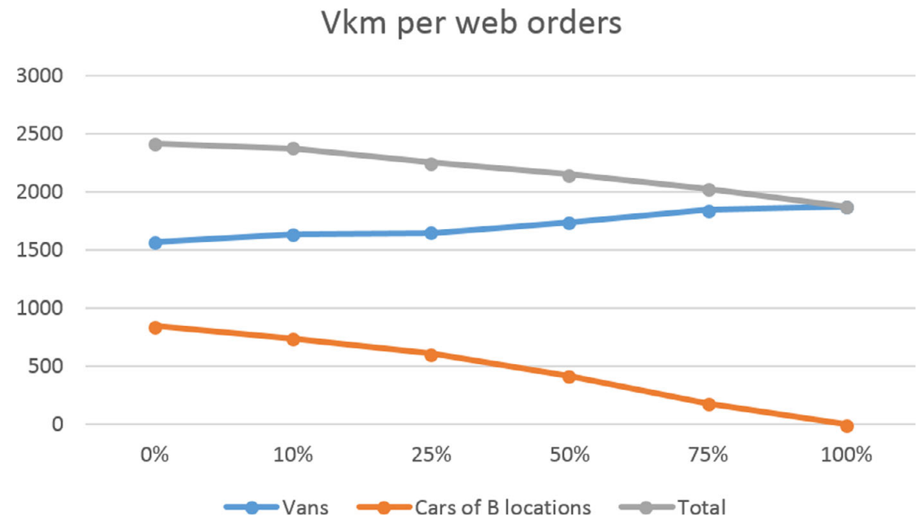
Research implications

Our work presents decentralized and autonomous allocation in geographic space that allows for testing various delivery logics in a more realistic manner within a risk-free environment. The contributions fill the gap on the inclusion of congestion levels, variable vehicle capacity, and en route changes that are echoed in Bell and Griffis (2010). Compared to mathematical/analytical approaches, an object-oriented agent-based model can capture more details and make moving entities aware of their surroundings. Such a level of agent decentralization linked to spatial and temporal awareness of surrounding entities as well as information can offer higher accuracy and more precise forecasts of emerging, and not-yet-well-understood, phenomena. In other words, a simulation study can help identify risk and find more robust solutions before pilot implementations.

In terms of geo-fencing and speed adjustment, data fetching tools could be connected to provide constant/continuous speed monitoring when deployed for other days or seasons of the year. Our geo-fences do not necessarily need to have a speed governing purpose only, but may be transformed into a notification source that notifies costumers or DCs about van's location and their estimated times of arrival. As IoT promises better visibility of operations and improved control over assets by remotely diagnosing problems and inducing diversion of an in-transit shipment, it is also noted by Goldsby et al. (2019) that emerging technologies are depicted mostly by the "promise" or "potential" terms which can prove to be hard to capture. The study presented herein offers quantification of the "promised" transparency levels and the implications they have on asset performances. As for the Physical Internet applications, our study could be linked to the existing body of literature, which focuses on inner PI-hub operations, by allowing hubs to proactively adjust their local solutions once they become informed about asset arrivals. Given the fact the simulations yielded positive results when introducing the PI hub closer to the consumers, it may serve as a starting point to link urban flows (perceiving city operations as the local "intranet") to interregional flows (perceiving these longer distance flows as the physical "Internet"). The PI-hub location is centered within the port of Brussels which can serve as a confluence of interregional shipments being carried by inland waterways and urban shipments to account for synchromodal door-to-door solutions. In fact, such a link could reduce road congestion in Flanders (around Brussels) by decreasing truck movements and inducing more barge movements.

Future research could focus on extending our work by accounting for CO_2 and *PM* emissions. More complexity could be also included in the current solution where vans do not collect orders from the central location after priority-based orders are delivered, but carry out this collection when the vans pass nearby the hub. This type of en route collection setting would require the experimental design to shift from "soft" allocation, where the matching algorithm notifies the van with the lowest fill rate to collect a new order after delivering all orders onboard, to "hard" allocation, where the algorithm is less benevolent and takes the first nearest van passing nearby the PI hub regardless of the van's ongoing process or the amount of core customer orders

Figure 12: A general perspective depicting total vehicle kilometers(vertical axis) of cars and vans within a city as a reaction to increasing percentage of inserted *Ob* online orders (horizontal axis) of customers from *B* locations.



onboard. Furthermore, the vans recalculate the distances to the next order from their decentralized location when unloaded, and depart to the nearest one by route; future work can address time windows of these orders (release dates) and include delivery duration priorities rather than focusing on shortest distances to the order locations.

Managerial implications

From the core customer group (*G*) perspective and its LSP, the service-driven company's fleet is impacted once 75% of new customers (*B*) place an order. In other words, the number of LSP assets becomes critical. It does not matter whether the orders are delivered in a priority-based or en route fashion. This means the core business of the service-driven company does not have to be necessarily affected as the core customers may be delivered as first, provided the extra orders do not exceed 75%. When the extra *Ob* online orders are collected from the LSP's depot, the load factors do not increase as return trips are still empty. In this regard, the best way to increase load factors is to collect *Ob* orders from a central location which is also beneficial to the *B* group in terms of lead times. From the perspective of the *B* group, transparent allocation generates faster lead times.

According to Rabinovich (2004), the Internet retailers can create new businesses by promising a fulfillment performance that reflects the actual performance. We captured implications newly incoming orders can have on the transport service provider. The transport provider in this case is the power house that accounts for actual performances, which is also why detailed simulations are relevant to assess the implications of new logic to depict, or come close to, the actual performance when promising certain service levels online. Querying multiple carries in an automated manner can provide higher service levels and more robust solutions in case of disruptions or unforeseen events when delivering orders. Once extra order insertion exceeds the 75% threshold, additional service-driven companies, taxi services, or crowdsourcing solutions could ameliorate delivery times and provide more reliable and faster service. Alternatively, longer lead times can be justified by lower delivery fees as a result of goods bundling and higher fill rates, as the study of Nguyen et al. (2019) shows positive acceptance levels of consumers when trading shorter lead times for lower delivery fees. The central PI-hub location would be beneficial to these potential new service providers as it is located closer to the customer demand.

In terms of application transferability to other geographic regions, higher delivery-related expectations are assigned to carriers that have higher consumer awareness such as FedEx, Airborne, and others in the United States (Esper et al. 2003). Such carrier exposures online may undermine the potential in order-delivery efficiency if nearest assets are ignored even though they could provide a better service. Omitting smaller and less known carriers at the expense of established brands can result in missed opportunities. The PI is thus a black box-like delivery solution where the carriers do not necessarily have to be exposed to the customer when ordering online. If costumers ease their expectations, the carriers will gain more time for consolidation and bundling. In relation to Griffis et al. (2012), Sternberg and Norrman (2017), and Muir et al. (2019) who depict the importance of returns logistics, the Physical Internet context demonstrated in our work can be linked to the identification of vans/agents roaming in the city, and query their working conditions and current states for the purpose of return flows. In this regard, insertion of return requests could be dynamically added to the routing of assets that are located nearby and have a similar delivery location as the product to be returned. The Physical Internet network could thus offer alternatives to consumers to return their items when purchasing online.

Our work illustrates that private car journeys can be eliminated by existing service flows of a service-driven company with reasonable delays incurred in their priority flows. From a general perspective/holistic point of view (Figure 12), not only could the negative impact of freight logistics and private mobility be reduced to zero, but it could also become negative, when compared to status quo. However, this research avenue still needs more consideration in terms of used vehicles and engine types combined with our load factors and driven kilometers. In reality, service-driven companies can use the decentralized local detection via GIS platforms that provide real-time updates of asset locations and their parameters. Such platforms allow to impose geo-fences to receive notifications and messages once moving assets enter a certain catchment of a DC or any other location. Such allocations can be tested with the resource pool of a single service-driven company, or with combined assets of more companies with similar flows in the area.

In conclusion, given the manifestation of new technologies and increasing customer requirements, facilitating decisions of logistics service providers and retailers can become rather challenging as vehicle routing problems become more and more complex. Current routing software does not have the ability to detect nearest assets and order senders, which makes new delivery practices rather untested. Hence, deployment of untested services can result in higher risk and potential money losses if not pretested in risk-free environments. This is where simulations can contribute to create a quantifiable basis for introduction of new technologies and delivery approaches, prior to investments and consortium commitments. To make the agent-based simulations in GIS more actionable, current transport management systems (TMS) can be connected to simulation modules in which the proposed logic in this paper could be realized. In such a case, the van agents can serve as digital twins that take over the geo-locations and working conditions of physical assets (physical twins) via sensors and IoT devices for the purpose of assessing future developments and potential routing and bundling opportunities. After evaluating multiple scenarios in parallel, the simulation module can return information or process specification back to the TMS to be executed in the physical system (real world). The virtual environment can then adapt and shift along with the physical environment. This actionable element that connects the virtual simulation environment to the real physical system will be studied more explicitly by the authors in their future endeavors together with several companies. Connecting research and models with companies will be crucial in order to translate the Physical Internet into daily use. Besides academia, the PI will also need more industrial and governmental leadership. The industrial and governmental involvement is gaining traction as more industrial players and governmental representatives (such as the European Commission) participated and presented palpable steps and support at the latest IPIC 2019 conference held in Westminster, London, as compared to earlier events that addressed more of the theoretical aspects of the concept.

APPENDIX A

When $\mathbb{M}$ agents move between $\mathbb{S}$ agents in GIS space, they enter a movement state $\omega_m \in \Omega$. Therefore, $\omega_m(.)$ means that agents are moving toward a destination between the brackets (.). The following expression $\|.\|$ is the distance between two points in geo-referenced space. This space consists of vector files, also known as shapefiles in GIS software. The distances $\mathbb{M}$ agents cover are recorded after reaching an $\mathbb{S}$ agent. Parameters such as WG_{sch} and S_{ch} are Boolean parameters indicating opening hours: *true* if open, and *false* when closed.

Simulation 0

Algorithms 1 and 2 run in parallel, but do not affect each other. Simulation A.1 sets the scene by reproducing the baseline setting where orders are not combined or inserted into vans. Algorithm 1 simulates replenishment of Customer2 group (B) and Algorithm 2 the deliveries to Customer1 group (G) by vans (V).

Algorithm 1: Status quo (S0)

input : B, C, S_{ch}, WG_{sch}
output: $dist_c, departed_c$ (figure 12 with last simulation S3)

// simulating cars replenishing at the nearest wholesaler
1 **for** $\forall\ b_j \in B$ **do**
2   Generate replenishment event E uniform (0, 2)
3   **if** E *initiates* b_j *and* $WS_{sch} = true$ **then**
4   **for** $\forall\ c_i \in B$ *initiated by* $e \in E$ **do**
5   ω_m(the nearest $w \in W$)
6       $departed_c \leftarrow$ add 1 // record presence of c_i in the city
7   **while** c_i *in* ω_m *state* **do**
8   get current $p_s(lat, lon)$ and update s_p according to table 1
9       $dist_c \leftarrow (\|b_j \rightarrow w_i\|)$ // record covered distance of c_i
10       $\omega_m(\ b_j)$ // return to home location
11   **while** c_i *in* ω_m *state* **do**
12   same as line 8
13       $dist_c \leftarrow (\|w_i \rightarrow b_j\|)$ // add the return trip to covered distance of c_i

Algorithm 2: Status quo (S0)

input : $G, dc_o, V, S_{ch}, WG_{sch}$
output: vd, v_{lf}, Ld_{og}

`// simulating van deliveries to their customer in parallel with Algorithm 1`

1 **for** $\forall\ g_i \in G$ **do**
2   Generate o_g uniform (0, 3) and send to dc_o
3   **if** S_{ch} *is true* **then**
        `// at distribution center (`dc_o`)`
4   Match ZIP code of og_i with $v_j \in V$ which services the geo-cluster/ municipality
5   **for** $\forall\ v_j \in V$ **do**
6   Add og_i to $Og_{collection}$ of v_j
7   **if** $Og_{collection} < v_c$ **then**
8   **if** $Og_{collection} > 0$ **then**
9                     $n \leftarrow$ number of orders in $Og_{collection}$
10   **for** $i \leftarrow 1$ **to** n **do**
11                         $n_g \leftarrow$ get the nearest $g_i \in Og_{collection}$
12                         $\omega_m(n_{gi})$ `//` v_j `moves to its nearest customer location`
13   **while** v_j *in* ω_m *state* **do**
14   get current $p_s(lat, lon)$ and update s_p according to table 1
15                         $dn_g \leftarrow (||g_i - 1 \rightarrow g_i||)$, record L_{fi} and D_{lfi}
16                         $Ld_{ogi} \leftarrow$ Equation 3
17                         $n \leftarrow n - 1$; remove delivered o_{gi} agent from $OG_{collection}$
18   **if** *n=0* **then**
                        `// once all orders delivered, move back to DC`
19                         $\omega_m(dc_o)$
20   **while** v_j *in* ω_m *state* **do**
21   same as line 14
22                         $dn_g \leftarrow (||g_n \rightarrow dc_o||)$, record L_{fn} and D_{lfn}
23                         $vd \leftarrow$ Equation 1
24                         $v_{lf} \leftarrow$ Equation 5
25   collect pending orders or wait for new ones (jump to line 3)
26   **else**
27   v_j stays at dc_o
28   **else**
29   Orders above v_c stay at dc_o for next delivery

Simulation 1

Simulation 1 executes Algorithms 3 and 4 in parallel. The distribution center that presents a static parameter is located outside of Brussels (dc_o), whereas the delivery logic (en route and priority) and *ob* Web borders (10%, 25%, 50%, 75%, 100%) are varied. Every store has a local Boolean parameter named *OrderOnline* directly linked to a parameter called *OnlinePreference*. This parameter contains values of type double between 0 and 1. A RandomTrue Java function is used to generate random numbers between these bounds where 0 is the lower and 1 is the upper bound. For instance, values between 0 and 0.5 indicate *false* and between 0.5 and 1 *true*. The bounds and the random numbers generated within the bounds are depicted by a uniform distribution function. If the function generates 0.786, the condition is more likely true and a $b \in B$ is keener on ordering via the website, which will be delivered by the service-driven company. If the function generates 0.221, the condition is more likely false and the $b \in B$ will depart to the wholesaler. For instance, based on a sample of ten decision events, a website preference of 0.221 would result in approximately three website orders and seven wholesaler trips. The randomness is further elaborated on in Appendix B.

Algorithm 3: Simulation 1 (S1)

input : $B, C, W, S_{ch}, WG_{sch}$
output: $dist_c, departed_c$

1 **for** $\forall\ b_j \in B$ **do**
2   Generate replenishment event E_r uniform (0, 2)
3   **if** E_r *initiates* b_j *and* WS_{sch} *is true* **then**
        `// generate random numbers for` $OnlinePrefereme$`. Bounds are set by the modeller`
4   **if** $Orderonline = true$ **then**
5   send *ob* to dc_o (jump to Algorithm 4)
6   **else**
7   reproduce movement from Algorithm 1 (lines 5 - 13)

Algorithm 4: Simulation 1 (S1)

input : $ob_{Algorithm3} \in Ob, G, dc_o, V, S_{ch}, WG_{sch}$
output: $vd, v_{lf}, Ld_{og}, Ld_{ob}$ (figure 7 and figure 8 left)

1 generate orders from core customers ($o_g \in G$) from Algorithm 2
2 Match ZIP code of og_i and ob_i from algorithm 3 with $v_j \in V$ which services the geo-cluster/municipality `// at distribution center`
3 **for** $\forall\ v_j \in V$ **do**
4 Add og_i to $Og_{collection}$ and ob_i to $Ob_{collection}$ of v_j
5 **if** $Og_{collection} + Ob_{collection} < v_c$ **then**
6 **if** $Og_{collection} > 0$ *and/or* $Ob_{collection} > 0$ *and* $S_{ch} = true$ **then**
`// ------------------Customer 1` $g \in G$ `has priority-------------------`
7 $R \leftarrow$ number of og orders in $Og_{collection}$
8 $L \leftarrow$ number of ob orders in $Ob_{collection}$
9 **for** $i \leftarrow 1$ **to** R **do**
10 $n_g \leftarrow$ get the nearest $g_i \in Og_{collection}$
11 $\omega_m(n_g)$ `//` v_j `moves to its nearest` g_i `location`
12 **while** v_j *in* ω_m *state* **do**
13 get current $p_s(lat, lon)$ and update s_p according to table 1
14 $dn_g \leftarrow (||g_i - 1 \rightarrow g_i||)$, record L_{fi} and D_{lfi}
15 $Ld_{ogi} \leftarrow$ Equation 3
16 $R \leftarrow R - 1$; remove og agent from $Og_{collection}$
17 **if** $R = 0$ *and* $L > 0$ **then**
18 **for** $k \leftarrow 1$ **to** L **do**
19 $n_b \leftarrow$ get the nearest $b_k \in Ob_{collection}$
20 $\omega_m(n_b)$ `//` v_j `moves to its nearest` b_k `location`
21 **while** v_j *in* ω_m *state* **do**
22 same as line 13
23 $dn_b \leftarrow (||b_k - 1 \rightarrow b_k||)$, record L_{fk} and D_{lfk}
24 $Ld_{obk} \leftarrow$ Equation 3
25 $L \leftarrow L - 1$; remove ob agent from $Ob_{collection}$
26 **else**
`// once all orders delivered, move back to DC`
27 $\omega_m(dc_o)$
28 **while** v_j *in* ω_m *state* **do**
29 same as line 13
30 $dn_g \leftarrow (||n_{gr}$ or $n_{bl} \rightarrow dc_o||)$, record L_{fn} and D_{lfn}
31 $vd \leftarrow$ Equation 1
32 $v_{lf} \leftarrow$ Equation 5
33 collect pending orders or wait for new ones (jump to line 3)
34 **else**
35 v_j waits for orders
36 **else**
37 Orders above v_c stay at dc_o for next delivery
`// --------------------------En Route delivery---------------------------`
38 remove condition from line 17. Once n_g from $Og_{collection}$ and n_b from $Ob_{collection}$ are known, v_j compares n_g and n_b against each other and departs to the nearest location regardless of the order type.
39 record dn, L_f, D_{lf}, Ld_{og} and Ld_{ob} values after each delivery
40 repeat until both $Og_{collection}$ and $Ob_{collection}$ are empty
41 jump to line 26 until line 33

Simulation 2

Simulation 2 executes Algorithms 3 and 5 in parallel. In this case, the *ob* Web order variations and location variations are taken into account. The delivery logic is static, applying priority deliveries to Customer1 orders from *Ob*.

Algorithm 5: Simulation 2 (S2)

input : $ob_{Algorithm3} \in Ob, G, dc_o, V, S_{ch}, WG_{sch}$
output: $vd, v_{lf}, Ld_{og}, Ld_{ob}$ (figure 8 right, figure 9)

1 copy lines 1 to 12 from algorithm 4
2 exclude ob_i insertion into $Ob_{collection}$ in line 4 `// the orders will be loaded at the central` dc_p `location upon` v_j `arrival`
3 **if** $R = 0$ *and* $L > 0$ **then**
  `// the corresponding` v_j `(based on ZIP code) receives a message that there is/are` Ob `orders waiting for pick-up.` L `becomes a temporary array list where` Ob `are stored`
4   $\omega_m(dc_p)$ `//` v_j `moves to the central` dc_p `location to collect` Ob `orders`
5   **while** v_j *in* ω_m *state* **do**
6   get current $p_s(lat, lon)$ and update s_p according to table 1
7   $dn_b \leftarrow (||g_i \rightarrow dc_p||)$ `// get distance from the last visited` $g_i \in G$ `location to the` dc_p `central location`
8   record L_{fk} and D_{lfk}
  `// At` dc_p `location`
9   Add ob_i from L waiting at dc_p to $Ob_{collection}$ of v_j
10   $N \leftarrow$ number of ob orders in $Ob_{collection}$
11   **for** $k \leftarrow 1$ **to** N **do**
12     $n_b \leftarrow$ get the nearest $b_k \in Ob_{collection}$
13     $\omega_m(n_b)$ `//` v_j `moves to its nearest` b_k `location`
14   **while** v_j *in* ω_m *state* **do**
15   same as line 6
16     $dn_b \leftarrow (||b_k - 1 \rightarrow b_k||)$, record L_{fk} and D_{lfk}
17     $Ld_{obk} \leftarrow$ Equation 3
18     $N \leftarrow N - 1$; remove ob agent from $Ob_{collection}$
19   **if** $N = 0$ **then**
20   finish movement by following the same logic from algorithm 4 (lines 26 - 33)
21 **else**
22   finish movement by following the same logic from algorithm 4 (lines 26 - 33)

Simulation 3

Simulation 2 executes Algorithms 3 and 6 in parallel. In this case, the *ob* Web order variations and the delivery logic variations are taken into account. The location is set to dc_p in Brussels from which both *Og* and *Ob* are picked up.

Algorithm 6: Simulation 3 (S3)

input : $ob_{Algorithm3} \in Ob, G, dc_o, V, S_{ch}, WG_{sch}$
output: $vd, v_{lf}, Ld_{og}, Ld_{ob}$ (figures 10, 11 and 12)

// -------------------Dedicated allocation by central dc_p -----------------------
1 copy lines 1 - 37 from algorithm 4; // Algorithm 4 is reproduced where dc_o is replaced by dc_p; all the proceses start and end at the central location
// ------------------Transparent allocation by central dc_p ----------------------
2 copy lines 1 - 9 from algorithm 2
3 **if** $dc_pRequest = false$ **then**
4   **for** $i \leftarrow 1$ **to** R **do**
5     $n_g \leftarrow$ get the nearest $g_i \in Og_{collection}$
6     $\omega_m(n_g)$ // v_j moves to its nearest g_i location
7     $A_v \leftarrow true$ // v_j availability parameter becomes $true$ indicating vehicle presence in the city while roaming the GIS environment
8   **while** v_j *in* ω_m *state* **do**
9   get current $p_s(lat, lon)$ and update s_p according to table 1
10     $dn_g \leftarrow (||g_i - 1 \rightarrow g_i||)$, record L_{fi} and D_{lfi}
11     $Ld_{ogi} \leftarrow$ Equation 3
12     $R \leftarrow R - 1$; remove og agent from $Og_{collection}$
13   **if** $R = 0$ **then**
    // once all orders delivered, move back to DC
14     $\omega_m(dc_p)$
15   **while** v_j *in* ω_m *state* **do**
16   same as line 9
17     $dn_g \leftarrow (||g_n \rightarrow dc_o||)$, record L_{fn} and D_{lfn}
18     $vd \leftarrow$ Equation 1
19     $v_{lf} \leftarrow$ Equation 5
20   collect pending orders or wait for new ones (jump to line 2)
21 **else**
  // request v_j regardless of the cluster it serves
22   Finish last delivery(ies)from $Og_{collection}$
23   $\omega_m(dc_p)$ // go to to dc_p to collect Ob orders. The real-time simulator continuously monitors the $dc_pRequest$ boolean parameter
24   **while** v_j *in* ω_m *state* **do**
25   same as line 9
26   $dn_b \leftarrow (||g_i \rightarrow dc_p||)$, record L_{fi} and D_{lfi}
27   copy lines 9 - 20 from algorithm 5
// At dc_p location
28 **if** $S_{ch} = true$ *and* $L > 0$ **then**
  // L is a temporary storage array for Ob orders as introduced in algorithm 5
29   **for** $\forall\ v_j \in V$ *where* $A_v = true$ **do**
    // this forloop runs in parallel with the loop in line 4
30   get distances to each v_j from dc_p
31   get the size of $Og_{collection}$ parameters of each v_j
32     $v_{jr} \leftarrow$ the v_j with lowest $Og_{collection}$ number AND nearest to dc_p
33   send message to (v_{jr}) and change its local $dc_pRequest$ parameter to $true$ // jump to line 22

APPENDIX B

As explained in A.2, the *OnlinePreference* parameter contains values of type double between 0 and 1. The bounds and the random numbers generated within the bounds are depicted by a uniform distribution function. The results presented in main body of the paper are based on reproducible simulation runs with fixed random seeds, to preserve comparability of the results. In this appendix, the randomness of the *OnlinePreference* parameter is captured by executing 10 replication of each simulation with unique random seeds. This means that, for instance, 50% order placements were evaluated by a uniform distribution function that "flipped a coin between 0.0 and 1.0." For 75%, for instance, the function would flip a coin between 0.5 and 1.0. Hence, the order placement coming from the B group will not be precisely 50 or 75%, but may vary as illustrated in the replicated outputs below. The variations do not fluctuate wildly and show that the selected order stream, depicted in the results section, is representative. The fluctuations are not so wild due to the fact that order placements which started coming from other clusters were then accommodated by the corresponding van that services the cluster/municipality.

Simulation 1 replications

Figure B1: Plotted averages of core orders' lead times after 10 replications of Simulation 1. The left side represents priority deliveries and the right en route.

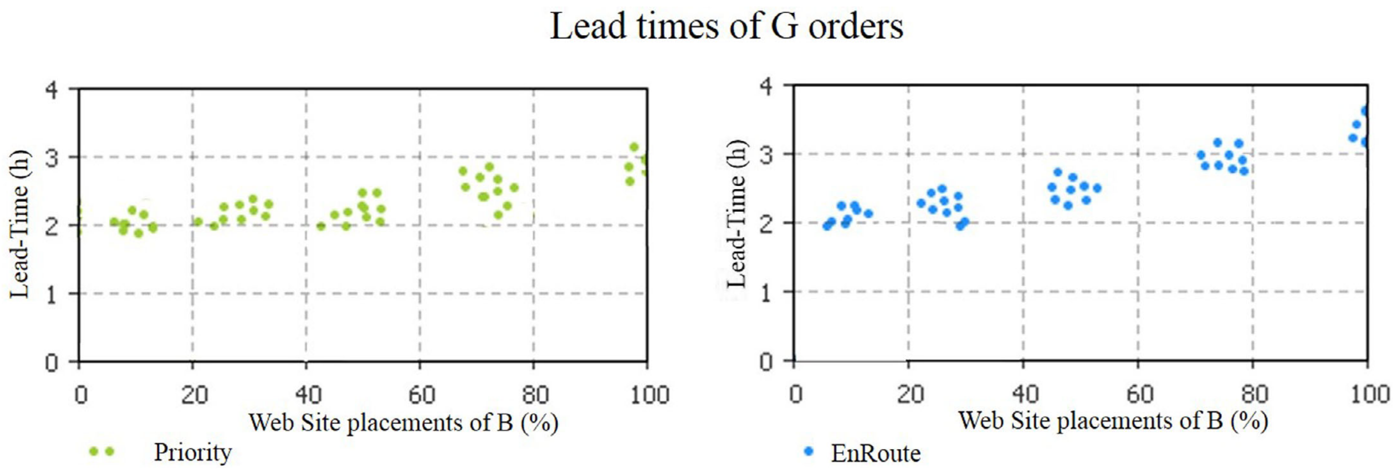
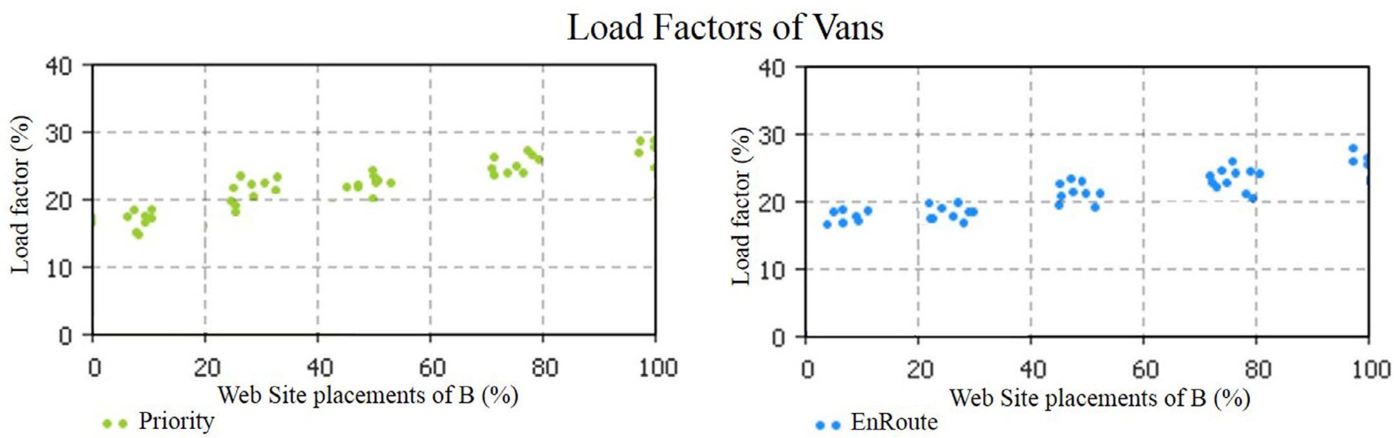


Figure B2: Plotted averages of vans' load factors after 10 replications of Simulation 1. The left side represents priority deliveries and the right en route.



Simulation 2 replications

Figure B3: Plotted averages of core orders' lead times after 10 replications of Simulation 2. The left side represents van deliveries from the original outside DC location, and the right side depicts deliveries from the central location.

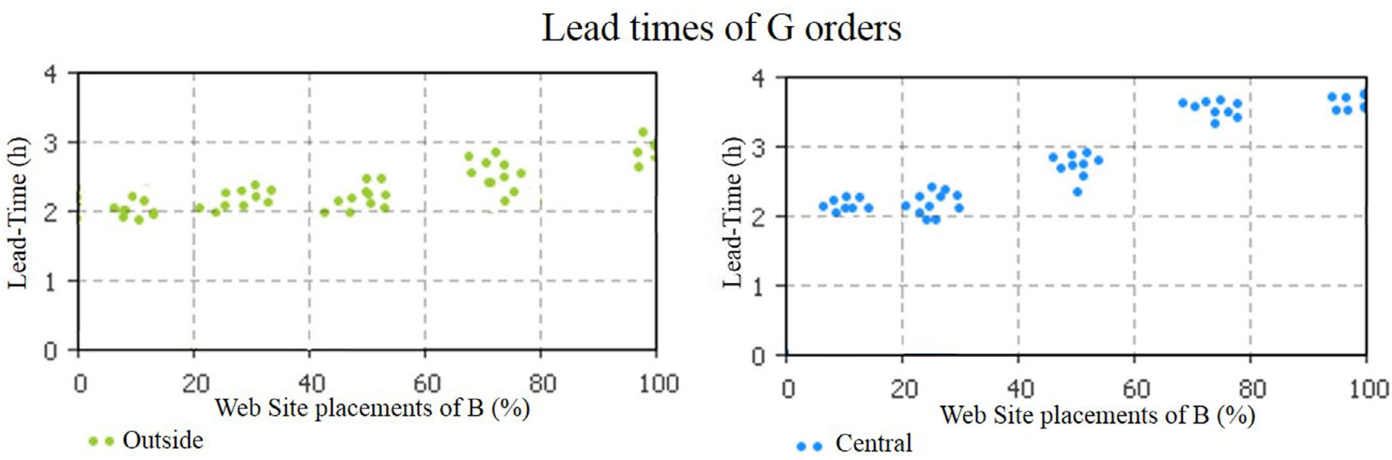
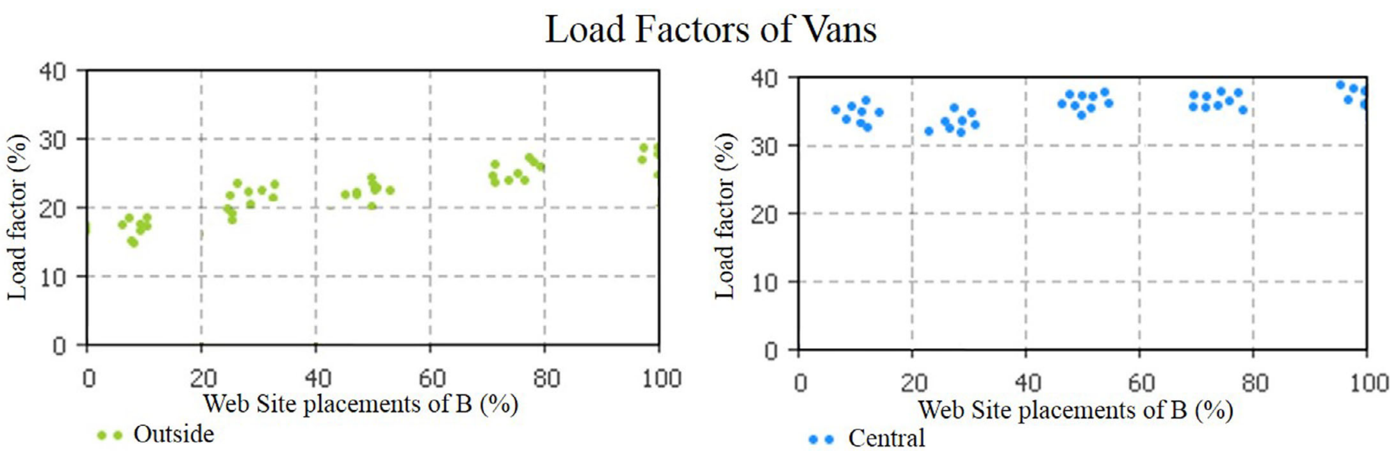


Figure B4: Plotted averages of vans' load factors after 10 replications of Simulation 2. The left side represents van deliveries from the original outside DC location, and the right side depicts deliveries from the central location.



Simulation 3 replications

Figure B5: Plotted averages of core orders' lead times after 10 replications of Simulation 3. The left side represents transparent order allocation of new B orders and the impact they have on the core G orders. The right side depicts dedicated allocation of B orders.

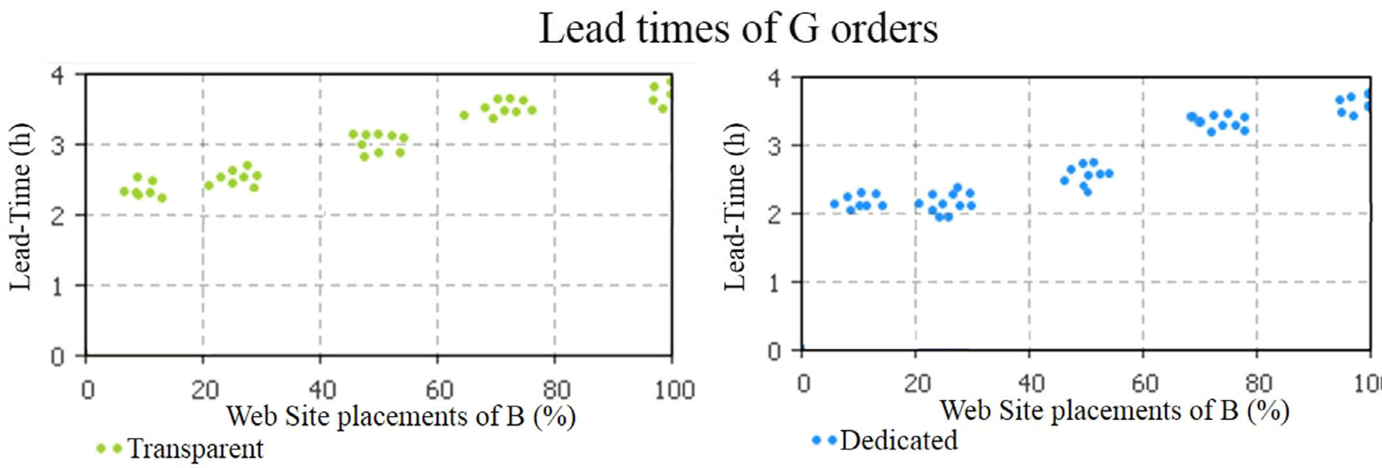


Figure B6: Plotted averages of extra B orders' lead times after 10 replications of Simulation 3. The left side represents the impact of transparent order allocation on B lead times, and the right side shows the impact of dedicated allocation.

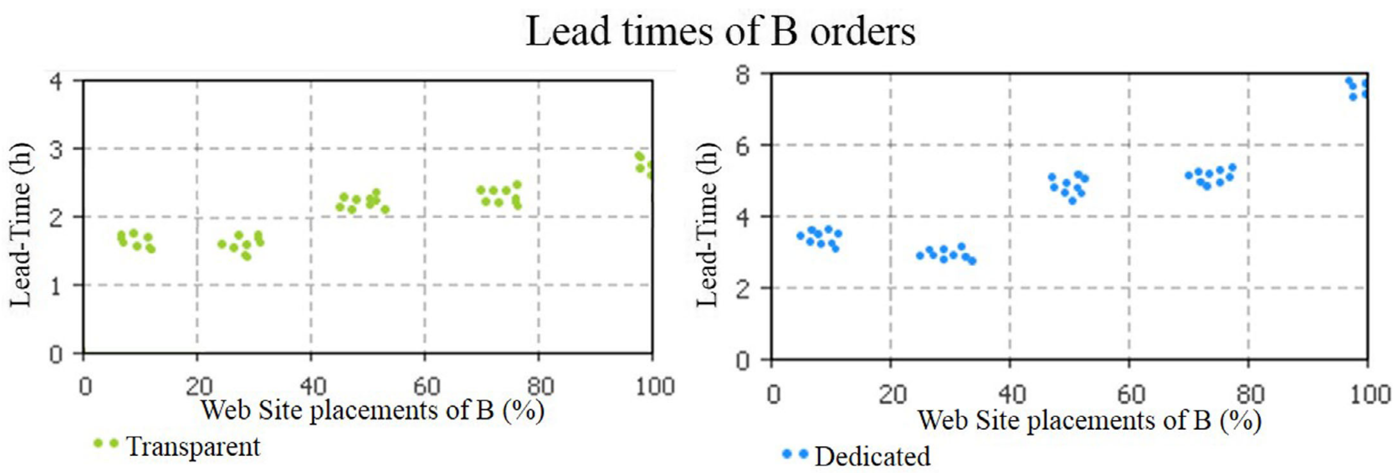
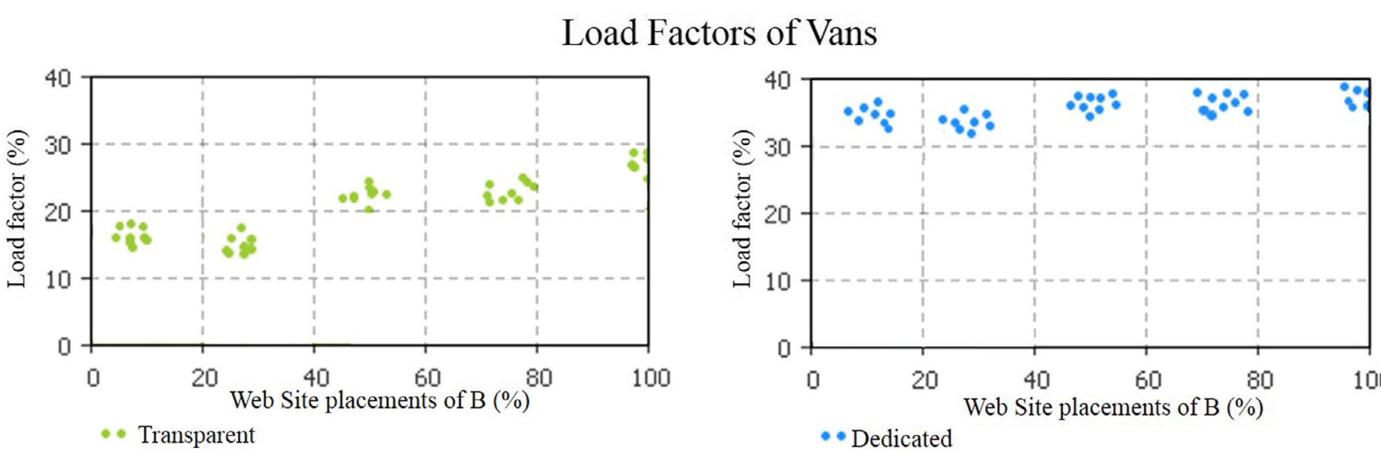


Figure B7: Plotted averages of vans' load factors after 10 replications of Simulation 3. The left side represents transparent order allocation and the right dedicated allocation to vans.



REFERENCES

Ambra, T., Caris, A., and Macharis, C. 2019. "Towards Freight Transport System Unification: Reviewing and Combining the Advancements in the Physical Internet and Synchromodal Transport Research." *International Journal of Production Research* 57(6):1606–23.

Bell, J.E., and Griffis, S.E. 2010. "Swarm Intelligence: Application of the Ant Colony Optimization Algorithm to Logistics-Oriented Vehicle Routing Problems." *Journal of Business Logistics* 31(2):157–75.

Ben Mohamed, I., Klibi, W., Labarthe, O., Deschamps, J.-C., and Babai, M.Z. 2017. "Modelling and Solution Approaches for the Interconnected City Logistics." *International Journal of Production Research* 55(9):2664–84.

Boyer, K.K., Prud'homme, A.M., and Chung, W. 2009. "The Last Mile Challenge: Evaluating the Effects of Customer Density and Delivery Window Patterns." *Journal of Business Logistics* 30(1):185–201.

Campbell, J.F., Labelle, A., and Langevin, A. 2001. "A Hybrid Travel Distance Approximation for a GIS-Based Decision Support System." *Journal of Business Logistics* 22(2):165–81.

Chen, C., Pan, S., Wang, Z., and Zhong, R.Y. 2017. "Using Taxis to Collect Citywide e-Commerce Reverse Flows: A Crowdsourcing Solution." *International Journal of Production Research* 55(7):1833–44.

Crainic, T.G., and Montreuil, B. 2016. "Physical Internet Enabled Hyperconnected City Logistics." *Transportation Research Procedia* 12:383–98.

Crooks, A.T., and Castle, C.J. 2012 "The Integration of Agent-Based Modelling and Geographical Information for Geospatial Simulation." In *Agent-Based Models of Geographical Systems*, edited by Alison J. Heppenstall, Andrew T. Crooks, Linda M. See, and Michael Batty, 219–51. Dordrecht: Springer.

Daellenbach, H., McNickle, D., and Dye, S. 2012. "Management Science: Decision-Making through Systems Thinking. Macmillan International." *Higher Education* 2:10–19..

Darvish, M., Larrain, H., and Coelho, L.C. 2016. "A Dynamic Multi-plant Lot-Sizing and Distribution Problem." *International Journal of Production Research* 54(22):6707–17.

Donati, A.V., Montemanni, R., Casagrande, N., Rizzoli, A.E., and Gambardella, L.M. 2008. "Time Dependent Vehicle Routing Problem with a Multi Ant Colony System." *European Journal of Operational Research* 185(3):1174–91.

EC. 2016. REFIT Ex-Post Evaluation of Combined Transport Directive 92/106/eec. page 59. Accessed 25 April 2018.

Esper, T.L., Jensen, T.D., Turnipseed, F.L., and Burton, S. 2003. "The Last Mile: An Examination of Effects of Online Retail Delivery Strategies on Consumers." *Journal of Business Logistics* 24(2):177–203.

Fazili, M., Venkatadri, U., Cyrus, P., and Tajbakhsh, M. 2017. "Physical Internet, Conventional and Hybrid Logistic Systems: A Routing Optimisation-Based Comparison Using the Eastern Canada Road Network Case Study." *International Journal of Production Research* 55(9):2703–30.

Goldsby, T.J., Zinn, W., Closs, D.J., Daugherty, P.J., Stock, J.R., Fawcett, S.E., and Waller, M. 2019. "Reflections on 40 Years of the Journal of Business Logistics: From the Editors." *Journal of Business Logistics* 40(1):4–29.

Griffis, S.E., Rao, S., Goldsby, T.J., and Niranjan, T.T. 2012. "The Customer Consequences of Returns in Online Retailing: An Empirical Analysis." *Journal of Operations Management* 30(4):282–94.

Kin, B., Ambra, T., Verlinde, S., and Macharis, C. 2018. "Tackling Fragmented Last Mile Deliveries to Nanostores by Utilizing Spare Transportation Capacity—A Simulation Study." *Sustainability* 10(3):653.

Kong, X.T., Chen, J., Luo, H., and Huang, G.Q. 2016. "Scheduling at an Auction Logistics Centre with Physical Internet." *International Journal of Production Research* 54 (9):2670–90.

Lagorio, A., Pinto, R., and Golini, R. 2016. "Research in Urban Logistics: A Systematic Literature Review." *International Journal of Physical Distribution & Logistics Management* 46 (10):908–31.

Landschützer, C., Ehrentraut, F., and Jodin, D. 2015. "Containers for the Physical Internet: Requirements and Engineering Design Related to FMCG Logistics." *Logistics Research* 8(1):8.

Lin, Y.-H., Meller, R.D., Ellis, K.P., Thomas, L.M., and Lombardi, B.J. 2014. "A Decomposition-Based Approach for the Selection of Standardized Modular Containers." *International Journal of Production Research* 52(15):4660–72.

Montreuil, B. 2011. "Toward a Physical Internet: Meeting the Global Logistics Sustainability Grand Challenge." *Logistics Research* 3(2–3):71–87.

Montreuil, B., Meller, R.D., and Ballot, E. 2013. "Physical Internet Foundations." In *Service Orientation in Holonic and Multi-agent Manufacturing and Robotics*, edited by Theodor Borangiu, Andre Thomas, and Damien Trentesaux, 151–66. Heidelberg, Berlin: Springer.

Muir, W.A., Gris, S.E., and Whipple, J.M. 2019. "A Simulation Model of Multi-echelon Retail Inventory with Cross-channel Product Returns." *Journal of Business Logistics* 40(4):322–38..

Nguyen, D.H., de Leeuw, S., Dullaert, W., and Foubert, B.P. 2019. "What Is the Right Delivery Option for You? Consumer Preferences for Delivery Attributes in Online Retailing." *Journal of Business Logistics* 40(4):299–21.

Pan, S., and Ballot, E. 2015. "Open Tracing Container Repositioning Simulation Optimization: A Case Study of FMCG Supply Chain." In *Service Orientation in Holonic and Multi-agent Manufacturing*, edited by Theodor Borangiu, Damien Trentesaux, André Thomas, Paulo Leitão, and José Barata Oliveira, 281–91. New York: Springer.

Qiao, B., Pan, S., and Ballot, E. 2019. "Dynamic Pricing Model for Less-Than-Truckload Carriers in the Physical Internet." *Journal of Intelligent Manufacturing* 30(7):2631–43.

Qiu, X., Luo, H., Xu, G., Zhong, R., and Huang, G.Q. 2015. "Physical Assets and Service Sharing for IoT-Enabled Supply Hub in Industrial Park (SHIP)." *International Journal of Production Economics* 159:4–15.

Rabinovich, E. 2004. "Internet Retailing Intermediation: A Multilevel Analysis of Inventory Liquidity and Fulfillment Guarantees." *Journal of Business Logistics* 25(2):139–69.

Rao, S., Goldsby, T.J., Gris, S.E., and Iyengar, D. 2011. "Electronic Logistics Service Quality (e-LSQ): Its Impact on the Customer's Purchase Satisfaction and Retention." *Journal of Business Logistics* 32(2):167–79.

Sallez, Y., Pan, S., Montreuil, B., Berger, T., and Ballot, E. 2016. "On the Activeness of Intelligent Physical Internet Containers." *Computers in Industry* 81:96–104.

Sarraj, R., Ballot, E., Pan, S., Hakimi, D., and Montreuil, B. 2014. "Interconnected Logistic Networks and Protocols: Simulation-Based Efficiency Assessment." *International Journal of Production Research* 52(11):3185–208.

Sternberg, H., and Norrman, A. 2017. "The Physical Internet–Review, Analysis and Future Research Agenda." *International Journal of Physical Distribution & Logistics Management* 47 (8):736–62.

Tran-Dang, H., Krommenacker, N., and Charpentier, P. 2017. "Containers Monitoring through the Physical Internet: A Spatial 3D Model Based on Wireless Sensor Networks." *International Journal of Production Research* 55(9):2650–63.

Venkatadri, U., Krishna, K.S., and Ülkü, M.A. 2016. "On Physical Internet Logistics: Modeling the Impact of Consolidation on Transportation and Inventory Costs." *IEEE Transactions on Automation Science and Engineering* 13 (4):1517–27.

Walha, F., Bekrar, A., Chaabane, S., and Loukil, T.M. 2016. "A Rail-Road PI-Hub Allocation Problem: Active and Reactive Approaches." *Computers in Industry* 81:138–51.

Yang, Y., Pan, S., and Ballot, E. 2017a. "Innovative Vendor-Managed Inventory Strategy Exploiting Interconnected Logistics Services in the Physical Internet." *International Journal of Production Research* 55(9):2685–702.

Yang, Y., Pan, S., and Ballot, E. 2017b. "Mitigating Supply Chain Disruptions through Interconnected Logistics Services in the Physical Internet." *International Journal of Production Research* 55(14):3970–83.

Yao, J. 2017. "Optimisation of One-Stop Delivery Scheduling in Online Shopping Based on the Physical Internet." *International Journal of Production Research* 55(2):358–76.

Zacharia, Z.G., Sanders, N.R., and Nix, N.W. 2011. "The Emerging Role of the Third-Party Logistics Provider (3PL) as an Orchestrator." *Journal of Business Logistics* 32(1):40–54.

Zhang, Y., Liu, S., Liu, Y., and Li, R. 2016. "Smart Box-Enabled Product–Service System for Cloud Logistics." *International Journal of Production Research* 54(22):6693–706.

Zhong, R.Y., Gong, H., Xu, C., and Lu, S. 2016. "Physical Internet-Enabled Manufacturing Execution System for Intelligent Workshop Production." *International Journal of Signal Processing, Image Processing and Pattern Recognition* 9(6):121–32.

SHORT BIOGRAPHIES

Tomas Ambra, PhD, is Senior Researcher and Team Leader at the Vrije Universiteit Brussel, MOBI research center, and Visiting Professor at Hasselt University, Belgium. His research interests lie in simulating real-time dynamics of freight transport processes in geo-referenced environments with a focus on synchromodal/intermodal transport and the Physical Internet. He currently coordinates an interuniversity DISpATch project (Digital TwIn for SynchromodAl Transport).

An Caris, PhD, is Associate Professor of Operations Management and Logistics at Hasselt University, Belgium. Her research interests lie in modeling intermodal transport, warehouse optimization, data-driven logistics, and healthcare logistics. From a technical viewpoint, she is also involved in the statistical analysis of algorithms for efficient parameter setting of metaheuristic algorithms.

Cathy Macharis, PhD, is Professor at the Vrije Universiteit Brussel, Belgium. She teaches courses in supply chain management and sustainable mobility and logistics. She is specialized in the assessment of policy measures and innovative concepts in the field of sustainable logistics and urban mobility. She is Head of the MOBI (Mobility, Logistics and Automotive Technology) research center.