

Routing of Heterogeneous Fleets for Flash Deliveries via Vehicle Group Assignment

Kronmüller, Maximilian ; Fielbaum, Andres; Alonso-Mora, Javier

DOI

[10.1109/ITSC55140.2022.9922062](https://doi.org/10.1109/ITSC55140.2022.9922062)

Publication date

2022

Document Version

Final published version

Published in

Proceedings of the IEEE 25th International Conference on Intelligent Transportation Systems (ITSC 2022)

Citation (APA)

Kronmüller, M., Fielbaum, A., & Alonso-Mora, J. (2022). Routing of Heterogeneous Fleets for Flash Deliveries via Vehicle Group Assignment. In *Proceedings of the IEEE 25th International Conference on Intelligent Transportation Systems (ITSC 2022)* (pp. 2286-2291). IEEE.
<https://doi.org/10.1109/ITSC55140.2022.9922062>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Routing of Heterogeneous Fleets for Flash Deliveries via Vehicle Group Assignment

Maximilian Kronmueller*, Andres Fielbaum* and Javier Alonso-Mora*

Abstract—This paper presents a novel approach to route heterogeneous fleets for flash delivery operations. Flash deliveries offer to serve customers' wishes in minutes. We investigate a scenario that allows to pick up orders at multiple depots with a heterogeneous vehicle fleet leveraging different modes of transportation. We propose the Heterogeneous Vehicle Group Assignment (HVGA) method, which, given a problem state, identifies potential pick-up locations, calculates potential trips for all modes of transportation and last chooses from the set of potential trips. Experiments to analyze the proposed method are executed using a fleet featuring two modes of transportation, trucks and drones. We compare to a state-of-the-art method. Results show that HVGA is able to serve more orders while requiring less total traveled distance. Further, the effects of the fleet size and fleet composition between drones and trucks are examined by simulating three hours of a flash delivery operation in the city center of Amsterdam.

I. INTRODUCTION

Using heterogeneous fleets for last-mile delivery operations allows leveraging the strengths of different modes of transportation. For example, a drone can maneuver independently of roads and traffic and thus deliver quickly in hard-to-reach areas and a truck can load many parcels simultaneously and deliver multiple orders in one neighborhood. In addition, grocery deliveries within minutes have established themselves within many cities. In the Netherlands alone, consumers spent around 40 million euros per month on flash deliveries at the end of 2021, a trend that is continuously rising [1]. Combining these two aspects poses a highly relevant and interesting question: How do we route heterogeneous fleets for on-demand last-mile deliveries?

This paper proposes a novel optimization-based approach to route heterogeneous fleets for on-demand last-mile deliveries, considering multiple depots and short delivery times. The proposed method is able to combine many types and forms of transportation while staying scalable. We specifically investigate the use case of supporting ground-based vehicles with drones.

The main virtues are threefold. First, we generalize a well-known method for ride-sharing, called Vehicle Group Assignment, to be able to handle heterogeneous fleets, including various modes of transportation. Second, this work combines dynamic multi-depot vehicle routing with heterogeneous fleets. Third, to the best of our knowledge, this work is first in investigating the use of drones for flash delivery operations.

*Delft University of Technology, Mekelweg 5, 2628 CD Delft, The Netherlands m.kronmueller@tudelft.nl

II. RELATED WORK

The problem we face can be classified as a dynamic and heterogeneous vehicle routing problem (pick-up and delivery problem). Thus it is related to the two broad fields of Dynamic Vehicle Routing Problems (DVRP) and Vehicle Routing Problems with heterogeneous fleets. For an overview of dynamic routing problems, see [2]–[5], and for an overview of routing for heterogeneous fleets see [6]. Heterogeneous routing problems are further divided by the type of vehicles they are considering and by the way these can interact with each other. For a more detailed analysis, we exclusively focus on dynamic problems, as approaches deployed between static and dynamic vehicle routing problems differ strongly.

Proposed methods to tackle the dynamic vehicle routing problem with a heterogeneous fleet are mainly based on heuristics. Large Neighbourhood Search based strategies were applied by [7] considering a fleet of trucks with different capabilities and by [8] tackling the technician routing problem (technicians differ in skills and repair parts carried). [9] used a Tabu Search to improve routes while dynamically integrating new incoming orders. They consider different vehicle speeds and capacities. A problem of dynamically refuelling aeroplanes by trucks with different speeds, capacities and fit to planes was studied by [10] using a genetic ant colony algorithm.

More specifically, the problem studied in this paper is a specific case of a Same-Day Delivery (SDD) problem [11]–[16]. Works that combine an SDD problem and a heterogeneous fleet are sparse. We identified three works, analyzed in the following.

[13] proposes a method routing a heterogeneous fleet of trucks and drones for a Same-Day Delivery (SDD) operation, splitting it into two. For each new order, it is decided if it is served by a drone or a truck or is ignored. Subsequently, each mode of transportation is routed independently. To determine if a drone or a truck should be used, they apply policy function approximation based on geographical districting. Orders which have a travel time longer than a given threshold are preferably served by drones. [15] builds upon [13] by adapting the way orders are assigned to a mode of transportation. They leverage Q-learning to do so and improve on previous results. Improvements are most substantial for small fleets but vanish fully if more resources are provided. [17] investigates a SDD problem, in which drones are used to resupply trucks, which exclusively deliver to customers. They examine two approaches for resupplying: first, only empty trucks can receive a resupply and second, resupplies are

possible anytime. They optimize on finding the best resupply locations, serving as many orders as possible.

In contrast, this work investigates a specific variation of SDD, a flash delivery operation featuring multiple depots. HVGA further differs from the works proposed for heterogeneous SDD by not splitting into assigning orders to vehicles and routing but does so jointly.

The proposed approach in this paper is building on a ride-sharing method called Vehicle Group Assignment (VGA) [18], which shows great scalability. A retail context variation considering multiple depots was investigated by [19]. This work extends VGA to heterogeneous fleets.

As we test our method considering scenarios in which drones and trucks are utilized, we want to point interested readers to an overview on routing problems featuring the usage of drones (mostly static), see [20].

III. PROBLEM DESCRIPTION

This section presents the problem statement covering the whole operation and introduces the used notation (Section III-A). As our problem is dynamic, we model it as a Markov Decision Process (MDP), explicitly capturing the problem's dynamics (Section III-B).

A. Notation and Problem Statement

We consider a heterogeneous fleet of vehicles \mathcal{V} consisting of multiple sub-fleets, each having a different mean of transport, noted by the subscript m . Each vehicle, independent of its mode, is denoted by $v \in \mathcal{V}$. For the sake of simplicity, we describe the method assuming a truck fleet \mathcal{L} of H identical trucks and a drone fleet \mathcal{D} of G identical drones. The generalization towards more than two sub-fleets is straightforward. Vehicles move on a weighted and directed graph $G_m = (N, A_m)$, thereby the arcs and arcs' weights are mode-dependent. Apart from the network they travel on, the fleets differ in their maximum capacity of each vehicle C_m and their traveling speed s_m . A set of depots De , locations where orders can be picked-up, is placed within the graph at specific nodes $\{de_1, \dots, de_{|De|}\}^1$. Each customer is defined as an order o . Customers can request the delivery of goods to a location of their desire. We denote the goal location by g_o , which is assumed to be a node. The time a customer orders is called the request time t_o . All orders are summarized within the demand \mathcal{O} . Each order is unique and has a size of one. We additionally introduce time-dependent demand subsets: First, the set of loaded orders \mathcal{O}_t^{load} consisting of all orders currently loaded to a truck or drone. Further, \mathcal{O}_t^{deli} denotes the set of delivered orders and \mathcal{O}_t^{igno} the set of ignored orders (i.e., orders that were not delivered within the given constraints). Last, the set of known open orders \mathcal{O}_t^{open} consisting of all orders currently known and unloaded nor delivered nor ignored.

The operation starts at $t = 0$ and ends at $t = T$, with t being the current time. During a short time span of duration

δ_T before T no new orders are accepted. Further, δ_{pick} and δ_{drop} are fixed time spans needed to load or to deliver an order.

General Problem Statement: Consider a heterogeneous fleet \mathcal{V} consisting of two sub-fleets, a truck fleet \mathcal{L} of H identical trucks and a drone fleet \mathcal{D} of G identical drones. Each mode of transportation operates on a weighted and directed graph $G_m = (N, A_m)$. Multiple depots $de \in De$ are placed within the graph. Each truck and drone is in a known initial state (starting locations and no prior load). Customers continuously place orders, summarized as the demand \mathcal{O} , where each order is specified by a request time $t_o \in [0, T - \delta_T]$ and a goal location g_o . The operation starts at $t = 0$ and ends at $t = T$. Find a set of routes for each truck and drone to pick up, thus including the choice of a depot, and to deliver the orders such that a given cost function \mathcal{J} is minimized subject to a set of given constraints.

We consider two constraints: First, each vehicle's maximum capacity C_m may not be exceeded. Second, each order has a maximum delivery time $t_{o,max,m}$. We calculate the maximum delivery time as the sum of the request time t_o , the optimal delivery time $t_{o,opt,m}$ and a fixed time interval, called maximum delay θ_{max} , i.e., $t_{o,max,m} = t_o + t_{o,opt,m} + \theta_{max}$. Thereby, the optimal delivery time $t_{o,opt,m}$ is calculated as the time a vehicle needs to travel from the closest depot to the goal location of the order plus the time to pick-up δ_{pick} and to deliver δ_{drop} the order. Note that the optimal delivery time is dependent on the used mode of transportation. An order might not be delivered within the given constraints and is thus considered ignored.

B. Markov Decision Process

An MDP captures the dynamics of a problem by modelling subsequent states \mathcal{S}_t connected by a decision taken and a transition between them.

Decision Points: Decisions are taken at specific points in time, summarized in the set² ψ . Individual decisions and according states are enumerated by k . We make decisions in fixed time steps of Δt , i.e., $t_{k+1} = t_k + \Delta t$.

Problem State: The problem state \mathcal{S}_k at time t_k is fully characterized by the time itself t_k , the state of both sub-fleets and the set of open orders \mathcal{O}_t^{open} . To describe the fleet's states each individual vehicle v is described by its location $l_{v,t}$, its set of loaded orders $\mathcal{O}_{v,t}^{load}$ and the plan it follows currently $\pi_{v,t}$. For example, the truck fleet's state is described as $\mathcal{L}_t = ((l_{i,t}, \mathcal{O}_{i,t}^{load}, \pi_{i,t}) \forall i \in \mathcal{L})$. This results in the overall state definition as

$$\mathcal{S}_t = (t, \mathcal{L}_t, \mathcal{D}_t, \mathcal{O}_t^{open}).$$

Note that a plan $\pi_{v,t}$ consists of an ordered sequence of actions (pick-up and drop-off of orders or rebalancing) with associated locations. Between locations, the vehicle follows the shortest path. As soon as an action is executed, it is

¹Also shared between trucks and drones.

² ψ can be determined during operation or beforehand.

removed from the plan. Further, a plan can be updated at a later point in time.

Decision: The decision at decision point k is to update the plan $\pi_{v,t}$ of each vehicle v , which it will follow until the next decision point t_{k+1} . Note that a vehicle's plan can change if a subsequent decision updates it, including newly obtained information.

Transition: The transition can be split into a deterministic part and an unknown part. In the deterministic part, we update the truck and drone fleet's status. Each vehicle follows its plan $\pi_{v,t}$ determined within the taken decision. Doing so, a vehicle's location and its loaded orders change (orders can get delivered and new orders can be loaded). As time propagates, between subsequent states, customers may place new orders (which is the unknown aspect). As a result, new orders are added to the set of open orders \mathcal{O}_{k+1}^{open} . If an order can not be delivered within its constraints anymore, we consider it ignored. The order is removed from the set of open orders³ \mathcal{O}_{k+1}^{open} .

Objective: The goal of the posed problem is to minimize a combination of costs, considering the total driven distance, service quality measured as the delay θ , and a penalty term α for orders that are not delivered (Equation 1).

$$\mathcal{J}_T = \left[(1 - \beta) \cdot \sum_{o \in \mathcal{O}_T^{deli}} \theta_o + \beta \cdot \sum_{v \in \mathcal{V}} \eta_{v,T} + \sum_{o \in \mathcal{O}_T^{igno}} \alpha \right] \quad (1)$$

Thereby, the delay θ_o is defined per order as the difference between the actual delivery time and the optimal delivery time. The total driven distance of a vehicle $v \in \mathcal{V}$ at time t is denoted by $\eta_{v,t}$. α is a constant predefined penalty for ignoring an order. Note that if α is set to a large constant, the objective function puts the highest priority onto serving as many orders as possible. This penalty can be interpreted as the cost to hire a third party to deliver the respective order, such that no customer is neglected. β is a tuneable weighting parameter between operator cost and service quality. Note that we can not directly map a decision taken at a specific state S_t to Equation 1. Each vehicle will follow its plan $\pi_{v,t}$ until the next decision state, then each plan may be altered due to new information. Thus, a current plan can not be directly evaluated by Equation 1.

IV. METHOD

In this section, we describe the proposed method, called Heterogeneous Vehicle Group Assignment (HVGA). Following the MDP (Section III-B), each time a decision point is hit, a decision given the current state S_k , determining a set of routes $\pi_{v,t}$ for every vehicle $v \in \mathcal{V}$, needs to be taken. To do so, we follow a sequence of steps explained in the following. First, we determine multiple potential pick-up locations for each order (Section IV-A). Second, we calculate sets of potential trips each truck and drone could take (Section IV-B). Last, we pick from those sets of potential

³Note that when vehicles load orders following their plan, they also get removed from the set of open orders.

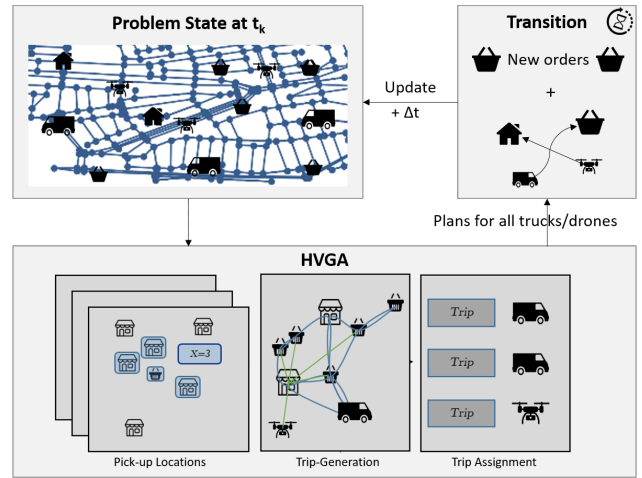


Fig. 1: Given a problem state, a decision is taken on how to update the plans of all trucks and drones. Heterogeneous Vehicle Group Assignment (HVGA) creates plans for all vehicles by first identifying potential pick-up locations for each order. Subsequently, a large set of potential trips is calculated for each vehicle. Which of these trips are executed is determined in the last step called trip assignment. During a transition phase, all trucks and drones follow their determined plan and new orders are received until a new decision state is met.

trips, which each vehicle should carry out by solving an Integer Linear Problem (ILP) (Section IV-C).⁴ An illustrative overview is depicted in Figure 1. The proposed method builds on top of previous work for flash deliveries for homogeneous fleets, introduced in [19].⁵

A. Selecting Potential Pick-up Locations

As orders only specify their delivery location g_o , the decision of where to pick up the corresponding goods is raised. To acknowledge this, we introduce a concept called candidate. A candidate combines an order with a potential pick-up location. We define it as follows: A candidate c is a tuple containing an order $o_c \in \mathcal{O}$ and an associated pick-up location $p_c \in De$. Thus a candidate is described as $c = (o_c, p_c)$. Each order can have multiple candidates associated with it. Further, we introduce a heuristic that reduces the number of candidates by only considering a subset of depots. We consider the x depots closest in delivery time by truck to the orders goal location. x is a predetermined tuneable parameter.

B. Finding Potential Trips

A trip, denoted as Γ , is defined as a set of candidates $\{c_1, \dots, c_j\}$, a vehicle and a plan, that serves all candidates

⁴Note that the proposed method does not take any future information into account and thus operates myopic. Doing so could improve obtained results [21], [22], but is regarded as future work.

⁵[19] also provides a more detailed analysis of their proposed method. Interested readers are referred to their work for further details.

of the trip, $\Gamma = (\{c_1, \dots, c_j\}, v, \pi_{v,t})$. The goal of the trip generation step is to find all potential trips for each truck and drone, summarized as the set Z^{all} . This set is formed by combining the sets of all potential trips Z_v for each vehicle, i.e. $Z^{all} = \bigcup_{v \in \mathcal{V}} Z_v$. We calculate the sets Z_v separately for each truck and drone and thus also separately for each mode of transportation. Each trip is further tagged with the cost to execute it.

The general workflow for a truck and drone are identical but differ in the used network G_m , speed s_m and vehicle's capacity C_m . If a trip Γ is feasible for a specific vehicle v , a plan $\pi_{v,t}$ can be found, picking up and delivering all candidates of this trip without violating any constraint. This includes orders that are already on board of the considered vehicle. To generate the complete set of all feasible trips Z_v , the method builds onto the idea that a trip can only be feasible if all its sub-trips are feasible as well. As a result, we start searching for trips of size one. Subsequently, we combine obtained trips to form larger trips, successively increasing in size. The cost of a trip Γ is given by γ_Γ and is derived from Equation 1:

$$\gamma_\Gamma = (1 - \beta) \cdot \sum_{o \in \Gamma} \theta_o(\Gamma) + \beta \cdot travel(\Gamma) \cdot s_m \quad (2)$$

$travel(\Gamma)$ determines the required distance traveled to serve the according trip Γ . Note that the delay θ_o of an order, as well as the time needed to complete a trip, depends on the trip and the used mode of transportation. The way to navigate between different trip stops is determined by the used graph G_m , and the time needed to traverse a given arc is shaped by the speed of the vehicle s_m . To determine the cost and plan $\pi_{v,t}$ of a trip Γ , we perform an exhaustive search on all possible sequences and continue solely with the cheapest option.

C. Assigning Trips

Given the set Z^{all} of all potential trips each truck and drone can take, this step decides which of them are executed. Thereby we want to coordinate the individual decisions to maximize performance and minimize cost. This problem is formalized and solved as an Integer Linear Problem (ILP), see Equations 3-6.

$$\operatorname{argmin}_\chi \sum_{\Gamma \in Z^{all}} (\gamma_\Gamma - \gamma_{loaded,v}) \epsilon_\Gamma + \sum_{o \in \mathcal{O}_t^{open}} \alpha \chi_o \quad (3)$$

$$\sum_{\Gamma \in Z_v} \epsilon_\Gamma \leq 1 \quad \forall v \in \mathcal{V} \quad (4)$$

$$\sum_{\Gamma \in Z^{all} | o_c \in \Gamma} \epsilon_\Gamma + \chi_o = 1 \quad \forall o \in \mathcal{O} \quad (5)$$

$$\epsilon_\Gamma \in \{0, 1\} \quad \text{and} \quad \chi_o \in \{0, 1\} \quad (6)$$

Equation 3 presents the cost function. Note it differs slightly from Equation 2 as we only account for changes in vehicles plans by subtracting the costs needed to serve the already loaded orders of the according vehicle $\gamma_{loaded,v}$.

We introduce the constraints that each truck and drone is maximally used once (Equation 4). Also, each order is maximally served once or ignored for now (Equation 5). Equation 6 introduces the binary variable ϵ_Γ , which takes the value of one if a trip Γ is executed; and the binary variable χ_o taking the value one if an order o is not served by the chosen trips.⁶

As a result, each truck and drone are either assigned a new trip, which they execute until the next decision is taken, or they follow their plan as previously determined. If a vehicle becomes idle (i.e., it has no orders to serve after the trips have been assigned), we assign it a special plan, we call rebalancing. The vehicle is routed to its closest depot, such that is in a promising position for future decisions.

Remark: Note that HVGA has a large potential to unify different types of vehicles and modes of transportation due to assigning a cost to each trip and using it to make decisions. Trips and their associated costs can thereby be calculated using entirely different approaches. If approaches are similar, this ensures more straightforward comparability. For example, one can only adapt the graph vehicles operate on, from a road network to a water canal system or the speed can be adjusted to the capabilities of an individual vehicle.

V. RESULTS

First, we compare HVGA to a state-of-the-art method to evaluate its performance in Section V-A. Section V-B investigates the effect of the size and composition of a heterogeneous fleet of trucks and drones.

All experiments presented in this section use a time window of 3 hours and 10 minutes, whereby no new orders are placed within the last 10 minutes. During this time, 1828 orders are placed randomly within the operation area.⁷ As an operation area, we simulate the city centre of Amsterdam, with trucks driving along the street network with a speed of 10 meters per second, and assuming that drones can fly directly to their goal location with a speed of 15 meters per second. Trucks can load up to six orders simultaneously, whereas drones have a maximum capacity of one. Each order is allowed a maximum delay of 8 minutes. To load an order 15 seconds are needed and 30 seconds to deliver it. HVGA was performed every 100 seconds, i.e., $\Delta t = 100$. The penalty for ignoring an order was chosen as a high constant $\alpha = 10000$ [sec] and the cost weighting parameter as $\beta = 0.3333$.

As a first impression, a snapshot of an area of a particular state is depicted in Figure 2. The plan for one truck and one drone are highlighted.

A. Comparison to Other Approaches

Next to HVGA, we implemented a second approach resembling the work by Ulmer et. al. [13]. This method first decides which mode of transportation is going to be used for which order, followed by a separate routing step. To

⁶To solve this problem, standard software, like Mosek or Gurobi, can be used. We used Mosek [23].

⁷We keep the used demand distribution and number constant to allow for better comparison between all analyzed scenarios.

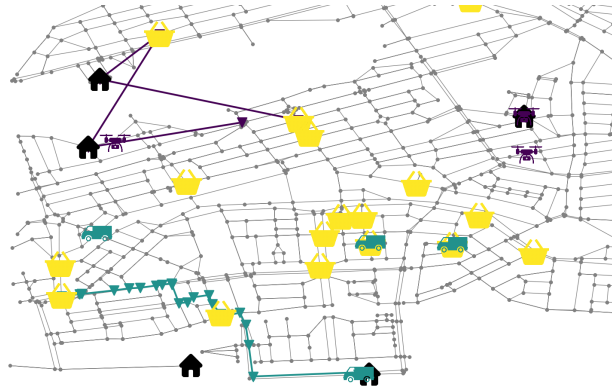


Fig. 2: A cutout of one particular state of the simulation is illustrated. The road network graph, on which trucks move is shown in grey. The plan of one drone and one truck are highlighted in purple and green. Goal locations of known and non-delivered orders are shown in yellow.

assign the mode of transportation, they use a policy function approximation based on geographical districting, preferably serving orders with long travel times by drones. To allow for a fair comparison, we adapted the approach [13] to use the routing method as introduced in this paper. Details on this adapted approach can be found in the Appendix (Section VII). In the following, we call this approach Split&Route. Here we investigate a scenario in which a single depot placed in the centre of the graph is used, as Split&Route is not specifically designed to work with multiple depots. Obtained results are depicted in Figure 3. HVGA improves on Split&Route by increasing the service rate by about 17%. Despite serving more orders, the total driven distance reduces by 108 km. On the other hand, the average delay increases by 24 seconds. Figure 4 illustrates the difference in

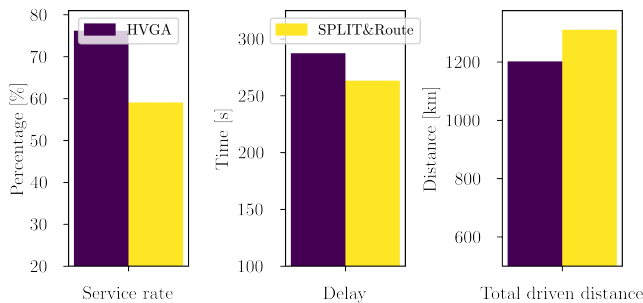


Fig. 3: Comparison of HVGA and Split&Route

the orders that get served in regard to their distance between the depot and drop-off location, based on the street network. A histogram of orders against travel distance is shown, where orders served by trucks are shown purple and orders served by drones in yellow (HVGA: Figure 4a, Split&Route: Figure 4b). HVGA primarily utilizes drone to serve short-distance orders, standing in direct contrast to Split&Route. This results in a greater amount of orders being served by drones (HVGA: 341, Split&Route: 138). The results suggest that the assumption to serve long-distance orders using drones does

not hold for flash delivery operations. Not doing so allows drones to serve a larger share of the requests.

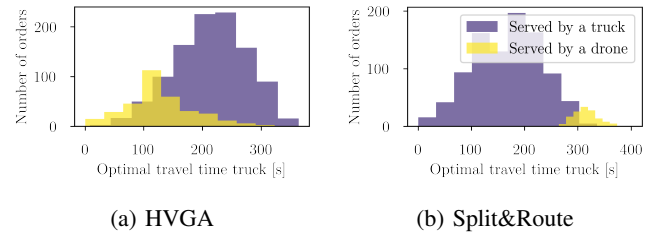


Fig. 4: Illustration highlighting the difference in served orders regarding their distance between depot and drop-off location between HVGA, left, and Split&Route, right.

B. Fleet Composition

As this work is the first (to the best of our knowledge) in investigating the effect of deploying drones within a flash delivery operation, we analyzed the fleet composition in more detail. We increase the number of pick-up locations in the service area to 20, to leverage that our method can handle multiple depots; three are considered per order ($x = 3$). Figure 5 shows the change in service rate, delay and total travelled distance for three fleets of total size 15, 20 and 25 and their composition in various ratios of trucks to drones. Independent of the total fleet size, we see an increase in service rate the more drones are used. This comes with an increase in total travelled distance. We see one exception for a fleet size of 25, when the service rate comes close to 100%. Then resources become available, which the approach can use to stronger optimize on traveled distance and delay. Generally, delay decreases the higher the percentage of drones used.

For flash delivery operations drones hold great potential. Short times between ordering and delivery and short distances between drop-off and pick-up location (strengthened by considering multiple depots) fit well with drones' benefits. High capacities are less important if comparing flash deliveries to traditional next-day operations, where times between leaving and returning to a depot can cover many hours.

VI. CONCLUSIONS

This paper presented an optimization-based approach to route a heterogeneous fleet of vehicles for an on-demand last-mile delivery operation. Orders are served within minutes after ordering, posing a special variant of a Same-Day Delivery problem. To analyze the results, a fleet of trucks and drones was studied and compared to a method inspired by [13]. HVGA was able to serve more orders while driving fewer kilometres in total. Further, the size and composition of various fleets have been studied. A larger amount of drones increases obtained service rates at the cost of increased travelled distance.

Future work involves a more detailed representation of used vehicle types, enabling a more accurate study about their use for flash deliveries. Further, the scope of experimental

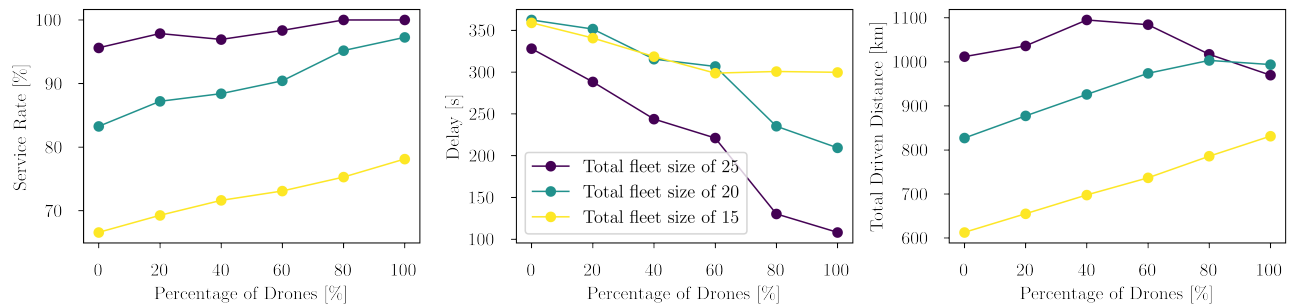


Fig. 5: Service rate, delay and total travelled distance are depicted for various compositions of the total fleet. Further the effect of the total fleet size is investigated by setting it to 15, 20 and 25.

analysis can be broadened such that the most critical drivers for flash delivery operations can be identified and studied.

VII. APPENDIX: DETAILS ON SPLIT&ROUTE

Split&Route selects a set of orders to be potentially served by drones and routes the drones exclusively for this set of orders. We attempt to serve as many orders as possible by drones. To do so, we take an iterative approach. The individual steps are outlined in the following.

1. We select y orders to be potentially served by drones, further called the drone order set. The y longest orders, based on their traveling distance on the road network, are selected. Note that y can be larger than the number of drones, as it is possible to assign multiple orders to one drone and deliver them subsequently, even though the maximum capacity is set to one. y is a predefined tuneable parameter.
2. We calculate a solution using HVGA (potential pick-up locations, trip generation and trip assignment) for all drones and the drone order set exclusively.
3. We check if all orders of the drone order set are assigned to a drone's plan and none is ignored. If yes, we update the drone plans $\pi_{v,t}$ and route all trucks using HVGA for the set of remaining open orders. If not, we reduce the drone order set by excluding the order with the shortest optimal travel distance and repeat starting from step 2.

ACKNOWLEDGMENT

This research was supported by Ahold Delhaize. All content represents the opinion of the author(s), which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

- [1] Kantar, "www.kantar.com/nl/kantar-nieuws/onderzoek-markt-flitsbezorging."
- [2] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *European Journal of Operational Research*, vol. 202, no. 1, pp. 8–15, 2010.
- [3] V. Pillac, M. Gendreau, C. Gu  ret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.
- [4] H. N. Psaraftis, M. Wen, and C. A. Kontovas, "Dynamic vehicle routing problems: Three decades and counting," *Networks*, vol. 67, no. 1, pp. 3–31, 2016.
- [5] N. Soeffker, M. W. Ulmer, and D. C. Mattfeld, "Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review," *European Journal of Operational Research*, vol. 298, no. 3, pp. 801–820, 2022.
- [6]   ađrı Ko  , T. Bekta  , O. Jabali, and G. Laporte, "Thirty years of heterogeneous vehicle routing," *European Journal of Operational Research*, vol. 249, no. 1, pp. 1–21, 2016.
- [7] A. Goel and V. Gruhn, "Solving a dynamic real-life vehicle routing problem," *Operations Research Proceedings*, vol. 2005, pp. 367–372, 01 2005.
- [8] V. Pillac, C. Gu  ret, and A. Medaglia, "A fast reoptimization approach for the dynamic technician routing and scheduling problem," *Operations Research/Computer Science Interfaces Series*, pp. 347–367, 01 2018.
- [9] F. Ferrucci and S. Bock, "Real-time control of express pickup and delivery processes in a dynamic environment," *Transportation Research Part B: Methodological*, vol. 63, pp. 1–14, 2014.
- [10] M. Schyns, "An ant colony system for responsive dynamic vehicle routing," *European Journal of Operational Research*, vol. 245, no. 3, pp. 704–718.
- [11] N. Azi, M. Gendreau, and J.-Y. Potvin, "A dynamic vehicle routing problem with multiple delivery routes," *Annals of Operations Research*, vol. 199, no. 1, pp. 103–112, 2012.
- [12] S. A. Voccia, A. M. Campbell, and B. W. Thomas, "The same-day delivery problem for online purchases," *Transportation Science*, vol. 53, no. 1, pp. 167–184, 2017.
- [13] M. W. Ulmer and B. W. Thomas, "Same-day delivery with heterogeneous fleets of drones and vehicles," *Networks*, vol. 72, no. 4, pp. 475–505, 2018.
- [14] M. A. Klapp, A. L. Erera, and A. Toriello, "Request acceptance in same-day delivery," *Transportation Research Part E: Logistics and Transportation Review*, vol. 143, p. 102083, 2020.
- [15] X. Chen, M. W. Ulmer, and B. W. Thomas, "Deep Q-learning for same-day delivery with vehicles and drones," *European Journal of Operational Research*, vol. 298, no. 3, pp. 939–952, 2022.
- [16] M. W. Ulmer, "Dynamic pricing and routing for same-day delivery," *Transportation Science*, vol. 54, no. 4, pp. 1016–1033.
- [17] I. Dayarian, M. Savelsbergh, and J.-P. Clarke, "Same-day delivery with drone resupply," *Transportation Science*, vol. 54, no. 1, pp. 229–249, 2020.
- [18] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.
- [19] M. Kronmueller, A. Fielbaum, and J. Alonso-Mora, "On-demand grocery delivery from multiple local stores with autonomous robots," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pp. 29–37, 2021.
- [20] S. H. Chung, B. Sah, and J. Lee, "Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions," *Computers and Operations Research*, vol. 123, p. 105004, 2020.
- [21] A. Fielbaum, M. Kronmueller, and J. Alonso-Mora, "Anticipatory routing methods for an on-demand ridepooling mobility system," *Transportation*, 09 2021.
- [22] W. B. Powell, "A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers," *Transportation Science*, vol. 30, no. 3, pp. 195–219, 1996.
- [23] M. ApS, *MOSEK for c++*. Version 7.1.