An Auction-Based Multi-Agent System for the Pickup and Delivery Problem with Autonomous Vehicles and Alternative Locations

Los, Johan; Schulte, Frederik; Spaan, Matthijs T.J.; Negenborn, Rudy R.

DOI
10.1007/978-3-031-05359-7_20

Publication date
2022

Document Version
Final published version

Published in
Dynamics in Logistics

Citation (APA)
Los, J., Schulte, F., Spaan, M. T. J., & Negenborn, R. R. (2022). An Auction-Based Multi-Agent System for the Pickup and Delivery Problem with Autonomous Vehicles and Alternative Locations. In M. Freitag, A. Kinra, H. Kotzab, & N. Megow (Eds.), *Dynamics in Logistics : Proceedings of the 8th International Conference LDIC 2022, Bremen, Germany* (pp. 244-260). (Lecture Notes in Logistics). Springer. https://doi.org/10.1007/978-3-031-05359-7_20

An Auction-Based Multi-Agent System for the Pickup and Delivery Problem with Autonomous Vehicles and Alternative Locations

Johan Los[1], Frederik Schulte[1(✉)], Matthijs T. J. Spaan[2], and Rudy R. Negenborn[1]

[1] Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
{F.Schulte,R.R.Negenborn}@tudelft.nl

[2] Department of Software Technology, Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands
M.T.J.Spaan@tudelft.nl

Abstract. The trends of autonomous transportation and mobility on demand in line with large numbers of requests increasingly call for decentralized vehicle routing optimization. Multi-agent systems (MASs) allow to model fully autonomous decentralized decision making, but are rarely considered in current decision support approaches. We propose a multi-agent approach in which autonomous vehicles are modeled as independent decision makers that locally interact with auctioneers for transportation orders. The developed MAS finds solutions for a realistic routing problem in which multiple pickup and delivery alternatives are possible per order. Although information sharing is significantly restricted, the MAS results in better solutions than a centralized Adaptive Large Neighborhood Search with full information sharing on large problem instances where computation time is limited.

Keywords: Autonomous vehicle routing · Pickup and delivery problem · Alternative locations · Preferences · Multi-agent system · Auctions

1 Introduction

The overarching trends towards automation and service orientation in transportation (Speranza 2018) go along with a rising need for decentralized decision support of individuals and autonomous vehicles. Autonomous transportation services may no longer depend on (human-controlled) centralized routing, but may autonomously optimize routes on the level of a single vehicle and may therefore even act as independent vehicular entrepreneurs. As Mobility as a Service (MaaS) solutions, such services rely on a digital platform (mobile app or web page) through which the end-users can access all the necessary resources for

M. Freitag et al. (Eds.): LDIC 2022, LNLO, pp. 244–260, 2022.
https://doi.org/10.1007/978-3-031-05359-7_20

their trips (Jittrapirom et al. 2017). The big players in the automobile industry anticipate this development: while Toyota sees itself in the transition from a car manufacturer to a mobility service provider (Buckland and Sano 2018), Volkswagen even envisions a mobility platform on which vehicles would act as autonomous entrepreneurs (Munford 2018).

When vehicles act as independent intelligent agents, their coordination and cooperation becomes increasingly significant (Qu et al. 2008), particularly for cooperative routing and traffic management (Zhou et al. 2017). While many cooperative transportation models assume a centralized planning approach with full information sharing (Guajardo and Rönnqvist 2015), there are other applications where competition seriously limits information sharing (Feng et al. 2017). While mobility platforms enable horizontal collaboration with multiple advantages, they also can easily turn into a problem of "coopetition", describing a situation in which logistics service providers are competitors in one market and cooperate in another market (Cruijssen et al. 2007). This raises a need for decentralized control: fully centralized planning requires a full exchange of information, which is not in the partners' interest when they are competitors in other markets (Cleophas et al. 2019). Moreover, assuming that such platforms may become dominant design in future mobility (Atasoy et al. 2020), there might easily be 100.000 vehicles or more and a respective number of requests—posing a tremendous computational challenge for centralized approaches based on NP-hard problems. Embedding agent-based routing models in multi-agent systems (MASs) is one way to explicitly model decentralized optimization with limited information sharing (Los et al. 2020b). This approach differs from combinatorial auctions, as reviewed by Gansterer and Hartl (2018), among others in the fact that each request is evaluated by agents locally and no centrally defined bundles are auctioned. Thus, agents are given a larger degree of freedom.

In this work, we develop a multi-agent approach for solving a decentralized Generalized Pickup and Delivery Problem with Preferences (GPDPP) (Los et al. 2018), where customers or operators can specify multiple alternative time-location combinations for pickup or delivery. Autonomous vehicle agents solve their individual decentralized subproblems based on the requests they receive. Moreover, order agents are responsible for the individual transport orders, that is, they try to find an assignment of the order to a vehicle, which can be understood as an intelligent (algorithmic) contract. We compare the decentralized MAS approach with a centralized single-agent system (SAS) approach assuming full information availability (see Fig. 1). Despite the limited information sharing, the MAS solutions outperform the solutions of the single-agent approach under certain conditions. The results demonstrate that the MAS is particularly useful when computation time is limited and the problem size is large.

We introduce the problem in Sect. 2. In Sect. 3, we describe the developed MAS in detail. Then, in Sect. 4, we introduce the centralized SAS to compare the different approaches in Sect. 5. Finally, in Sect. 6, we summarize our findings and give proposals for future research.

2 Decentralized Generalized Pickup and Delivery Problem with Preferences

In this section, we describe the GPDPP this paper deals with. First, we give a general motivational introduction. Next, we describe the problem from the local vehicles' and orders' perspective, and finally, we consider the problem from the global perspective. An extensive formal description is given by Los et al. (2018).

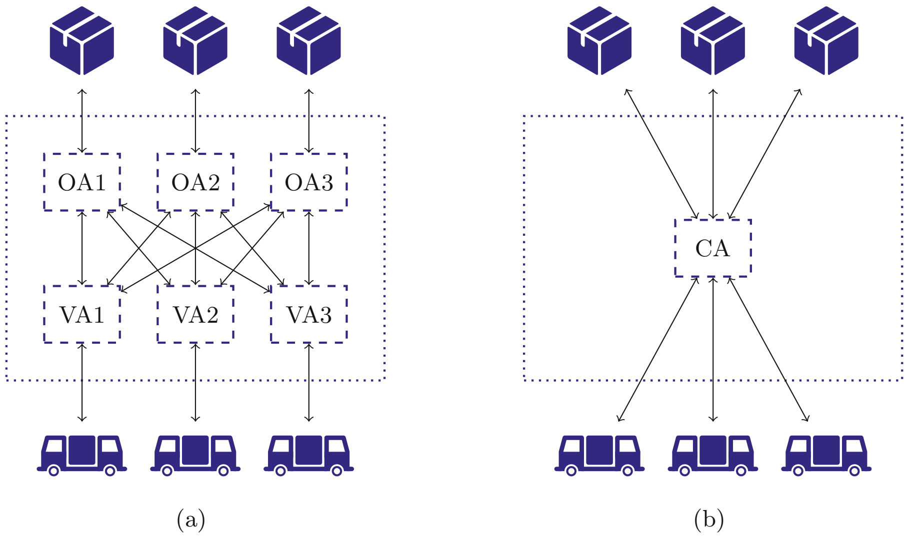

Fig. 1. Different computational approaches. In (a), a decentralized multi-agent approach is shown, where all orders and vehicles are represented by order agents and vehicle agents, respectively, that interact locally with each other, without communicating with a central manager. In (b), a centralized single-agent approach is shown, for which all vehicle and order information is sent to a central manager, and there is no direct local communication. **OA**: order agent; **VA**: vehicle agent; **CA**: central agent. The dotted box represents the computational system, outside is the physical system. Information flow is represented by arrows.

2.1 Problem Motivation

Our problem differs in two aspects from classical Pickup and Delivery Problems (PDPs) (Cordeau et al. 2008) or Dial-a-Ride Problems (Molenbruch et al. 2017). First, we consider the problem to be inherently decentralized, that is, vehicles are independent, can attach to or detach from the system at any moment, and might not be willing or able to share all information. Transport requests also might continuously appear, disappear, or change, and require immediate actions. Computing a central solution might be no longer possible in such situations.

Second, we consider a realistic problem with alternative locations and preferences. Instead of a single pickup location and a single delivery location per order, as is the case in the classical PDP, we allow for multiple alternative pickup and/or delivery locations per order, with their individual time windows. Furthermore, different preference values can be assigned to each location. One of the possible pickup locations and one of the possible delivery locations is chosen and attended by a vehicle (Los et al. 2018). A motivation for this generalization of the PDP is the high absence rate of customers in regular home delivery processes. Next to the (still preferred) option of home delivery between 10:00 and 12:00, for example, a (less appreciated but still acceptable) delivery at the locker box station two streets away might be allowed, with the advantage of a larger time window. Also, different pickup or delivery time windows can be assigned to the same physical location. In this scenario with multiple alternatives, a higher delivery success rate can be achieved, and the transport operator has more flexibility in designing efficient routes.

2.2 Local Problem Definitions

A problem instance consists of a set C of transport orders and a set V of vehicles. In the next sections, we consider the problem from the perspective of the individual orders and vehicles.

Order Problem. Each order $c \in C$ has a load quantity Q_c, a set of possible pickup alternatives P_c, and a set of possible delivery alternatives D_c. A pickup or delivery alternative i is defined by a tuple $\langle n_i, e_i, l_i, d_i, p_i \rangle$, where n_i is the location where the order can be picked up or delivered, the earliest service start time e_i and the latest service start time l_i determine the time window in which the service can start, the service duration d_i determines the time that is needed for loading or unloading at n_i, and the preference value $p_i \in (0, 1]$ describes the relative satisfaction for the alternative. We assume that each order has at least one pickup alternative i and one delivery alternative j with $p_i = p_j = 1$, meaning that there are no other alternatives preferred over these.

An order $c \in C$ needs to be served by a vehicle $k \in V$, that is, it needs to be picked up by k as described by one pickup alternative $i \in P_c$ and delivered by k as described by one delivery alternative $j \in D_c$, while $M_k^c + \beta((1-p_i)+(1-p_j))$ should be minimized. Here, M_k^c are the marginal routing costs for vehicle k to include order c into its route, and β is a positive parameter representing the weight of dissatisfaction relative to travel cost.

Vehicle Problem. Each vehicle $k \in V$ has a capacity B_k, a start location α_k and an end location ω_k. For each pair of nodes $\langle i, j \rangle$, we denote the travel time and travel costs from location n_i to location n_j by t_{ij} and c_{ij}, respectively.

A feasible route for a vehicle $k \in V$ is a sequence of locations that meets the following requirements:

- the vehicle starts its route at α_k and stops at ω_k;
- all time constraints are respected, that is, if the vehicle serves an alternative i, it arrives at n_i between e_i and l_i, it leaves n_i not before d_i after arrival time, and traveling from n_i to n_j takes a time of at least t_{ij};

- all precedence constraints are respected, that is, if the vehicle serves a pickup alternative i and a delivery alternative j belonging to the same order, the arrival at n_i takes place before the arrival at n_j;
- all capacity constraints are respected, that is, the load of the vehicle may never exceed its capacity B_k throughout its route, but is increased or decreased with the order load quantity at a pickup or delivery location, respectively.

A vehicle $k \in V$ can transport orders if it keeps a feasible route, and the vehicle agent should minimize the travel costs. Thus, the vehicle agent tries to find combinations of a pickup alternative and a delivery alternative of the new order that can most efficiently be incorporated into its current route, such that the previously agreed on alternatives from already included orders still can be served. Hence, the vehicle agent locally solves multiple instances of a standard single-vehicle PDP (see, e.g., Parragh et al. 2008).

2.3 Local and Global Perspective

From a local perspective, order agents need transportation by a vehicle that realizes a preferred pickup and delivery, but is not too bad in terms of vehicle route costs. Vehicle agents must obtain routes with minimal travel costs.

These local objectives contribute to the objective from the global perspective: to find a minimal cost solution. A global solution consists of a set of feasible routes (one for each vehicle $k \in V$), such that each order $c \in C$ is served by one vehicle. Global cost is defined as the sum of travel costs and dissatisfaction costs, where the sum of c_{ij} values for i and j such that the edge from n_i to n_j is part of a vehicle's route constitutes the travel costs, and all non-preferred alternatives i that are served by a vehicle contribute a term $\beta(1 - p_i)$ to the dissatisfaction costs.

3 Multi-Agent System

We propose a MAS approach to solve the GPDPP in a decentralized manner, in accordance with the assumption that vehicles and orders can independently attach to a platform. We introduce two types of agents that represent the main stakeholders of the problem: order agents, each responsible for getting one of the orders transported, and vehicle agents, each representing one vehicle. For finding an assignment of orders to vehicles, the order agents and vehicle agents communicate with each other in a multi-agent auction (Wooldridge 2009). Order agents act as auctioneers that offer a transportation task. Vehicle agents act as bidders. Hence, there is no central auctioneer, and no central authority that is aware of the global routing plan, but information is exchanged locally (see Fig. 2). Although the solution quality might be suboptimal from a global perspective, this approach resembles a transportation demand and supply market that allows for a fast response to dynamic events. Our auction mechanism is based on the systems described by Máhr (2011), Gath (2016), and Los et al. (2020a; 2022; 2020b).

3.1 General Approach

Order agents try to make a contract with a vehicle agent for a pickup and delivery with high preference values, but are cooperative in the sense that they take vehicle routing costs into account and accept lower preference values if this decreases the routing costs enough. Vehicle agents are responsible for making contracts with order agents, but have the local goal of minimizing the sum of travel costs while keeping a feasible route.

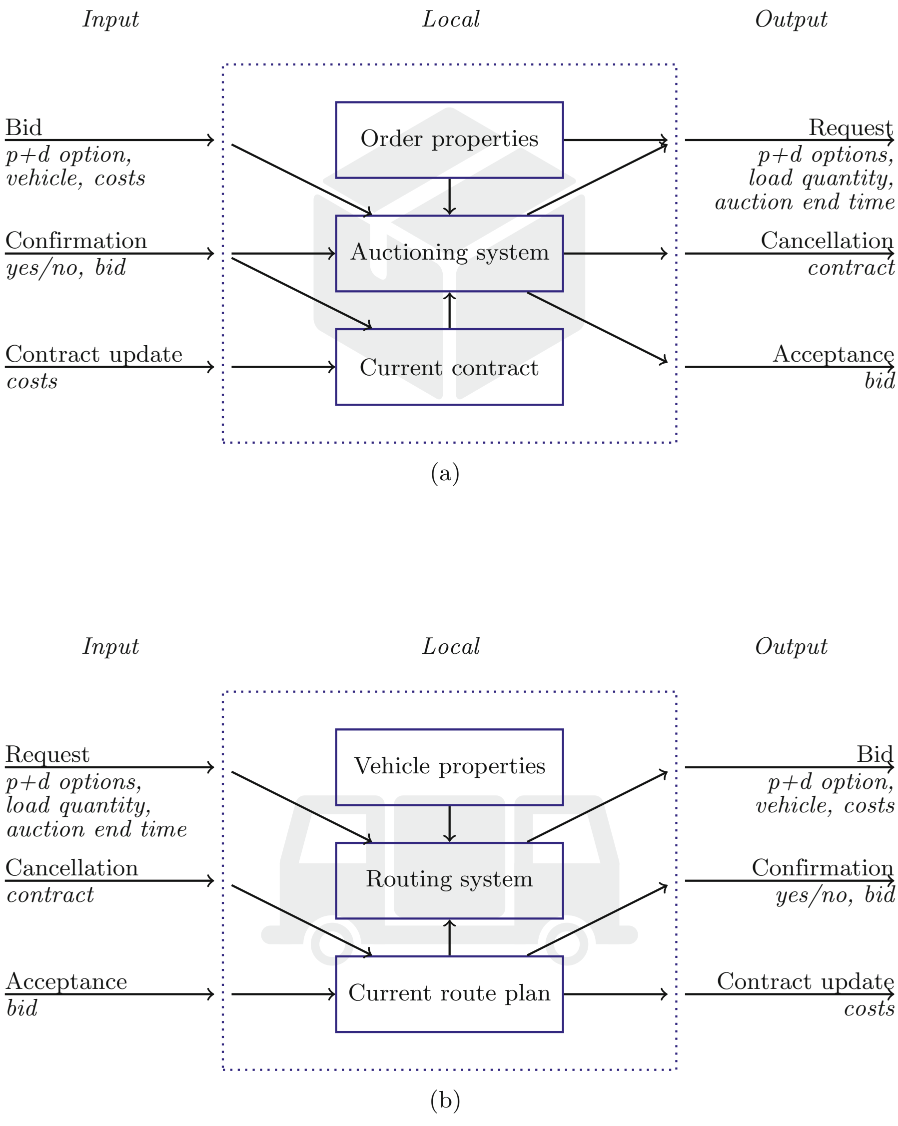

Fig. 2. Information input and output, as well as local information flows, for an order agent (a) and a vehicle agent (b).

When they enter the system, order agents send a request for transportation to a well-selected set of vehicle agents. These compute the routing costs for inserting the order into their current route (by solving multiple single-vehicle subproblems, each with a different combination of an alternative pickup and delivery location of the new order, together with the orders already included in their route) and propose different bids. Order agents evaluate the bids from different vehicles and choose the one that is best, based on the routing costs of the bid and their own preferences for the alternative locations. If no changes have occurred meanwhile in the route of the chosen vehicle agent, the order will be inserted into its route. Orders auction themselves again after some time to check if there are better options due to the high dynamics within vehicle routes.

In line with Los et al. (2020b), order agents interact only with a well-chosen subset of the vehicles (instead of with all vehicles) to limit the communicational and computational load. Although we might lack some good bids from other vehicles, we expect a better result due to a gain in time. In contrast to other MAS approaches, we introduce two properties that are specific for the GPDPP. First, preference costs are locally considered by the order agent. Second, it is possible for vehicle agents to send multiple bids to an order agent, resulting from different combinations of pickup and delivery alternatives.

An example of the general auction procedure is shown in Figs. 3 and 4, and detailed agent algorithm descriptions are given in the next sections.

3.2 Order Agent

An order agent keeps track of the contract of the order, consisting of a transporting vehicle, one pickup and one delivery alternative that are agreed on, as well as the costs for transportation. Initially, there is no contract; the order agent organizes auctions for obtaining and improving a contract.

An order agent starts its first auction in the system immediately after its release time. First, it selects a set of vehicle agents to send a request for transportation. As in other approaches, the set of all vehicles in the system can be used, but this can result in an overload of messages and subsequent vehicle computations, although not all of them have a high potential of being useful. For example, consider an order that needs to be picked up and delivered in the northern part of a city, and a vehicle that has only pickups and deliveries in the southern part of the city in its current plan. A match is not likely in this case. Different selection heuristics are possible, based on, e.g., the current vehicle locations, the planned routes, and the occupancy rate of the vehicles. In this paper, we select the vehicles based on the spatiotemporal distance of the different pickup and delivery alternatives of the order to the planned routes of the vehicles. The order agent opens the auction by sending all its possible pickup and delivery locations, the corresponding time windows and service durations, the load quantity, and the time at which the auction will end to all selected vehicles.

If an order agent receives a bid (consisting of a pickup alternative i, a delivery alternative j, and the marginal travel costs) from a vehicle agent, it adds its dissatisfaction costs $\beta(2 - p_i - p_j)$ (see Sect. 2) for the specific alternatives to

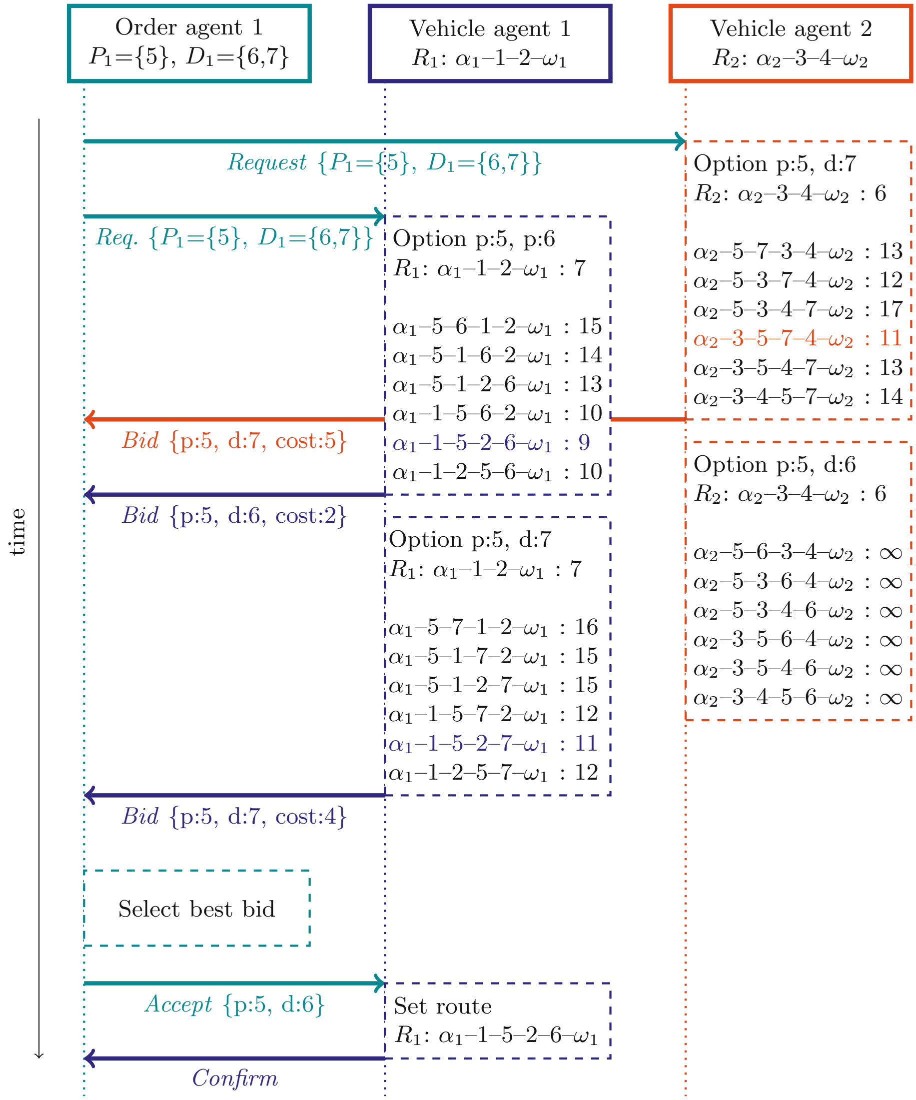

Fig. 3. Schematic overview of an auction round in the MAS, with an order agent having one pickup option and two delivery alternatives, and two vehicles with current routes α_1–1–2–ω_1 and α_2–3–4–ω_2. The order agent sends a request with its pickup option 5 and delivery alternatives 6 and 7 to both vehicle agents. The vehicle agents each consider the two options, one with delivery alternative 6 and one with delivery alternative 7. They insert the new locations into their current routes and compare the costs (as defined by the graph of Fig. 4) of the different new routes to the cost of their current routes. A bid with the least increase in costs is sent back. Note that insertion of delivery alternative 6 is not feasible for vehicle agent 2; hence, only one bid is sent back. The order agent selects the best bid (consisting of delivery alternative 6 with a cost of 2), and notifies vehicle agent 1 of its acceptance. (The example abstracts from time windows and preferences.)

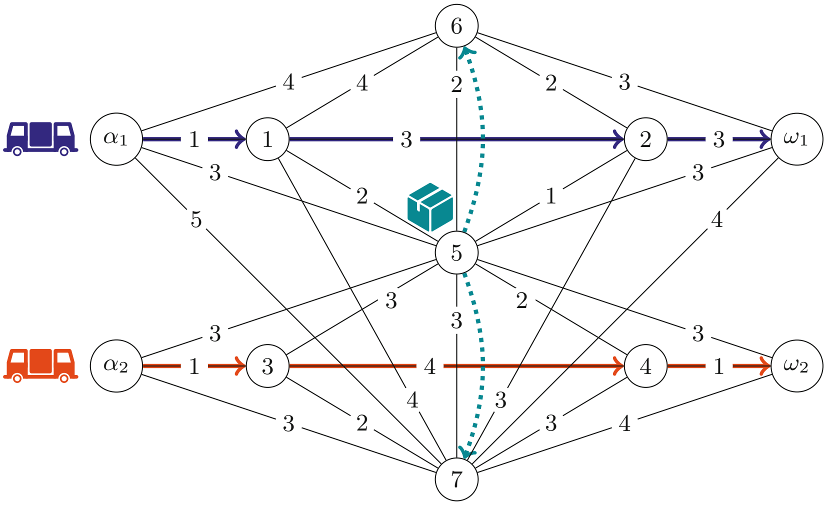

Fig. 4. Graph with initial vehicle routes and an order with two possible delivery locations corresponding to the auction process overview of Fig. 3. Edges represent the travel costs between locations.

the travel costs to obtain the total costs for the bid. Subsequently, it stores the bid in a sorted list with increasing bid costs. When the auction time has ended, the order agent selects the first bid of its list, compares the costs of that bid to the costs of its current contract, if possible, and acts appropriately:

- If the costs of the selected bid are lower than that of the current contract, or there is no current contract, the order agents asks the vehicle agent that proposed the bid to insert the order into its route. If a positive response follows, the order agent updates its current contract, cleans up its bid list and schedules to start a new auction after some time. Furthermore, a message is sent to the vehicle agent of the previous contract (if applicable) to inform this agent that the order can be removed from its route. In case of a negative response of the vehicle agent, the bid has become outdated. In this case, the order agent possibly includes a new bid of the vehicle agent into its bid list, selects the next bid of its bid list and repeats the procedure.
- If the costs of the selected bid are not lower than the costs of the current contract, the current contract is still the best option. The agent cleans up its bid list and schedules to start a new auction after some time.
- If there is no bid selected (i.e., the bid list was empty) and there is no current contract, the order agent immediately starts a new auction. If vehicle routes have been changed in the meantime, probably it will obtain some bid from the new auction. This is urgent since there is no contract yet.

3.3 Vehicle Agent

A vehicle agent keeps track of the planned route of the vehicle, along with earliest and latest possible times for each location, and the used vehicle capacity at each trajectory. Initially, the route only consists of the vehicle's start and end locations.

When a vehicle agent receives a request from an order agent, it checks whether the auction has not yet ended. If there is still time, it computes the marginal travel costs for inserting each combination of alternatives into its current route, that is, it solves the single-vehicle PDP multiple times: once for each possible combination of a pickup and a delivery alternative of the new order. If an insertion is possible, a bid consisting of the marginal travel costs (the costs of the new route minus the costs of the current route), the pickup alternative and the delivery alternative is sent to the order agent. Hence, a vehicle agent can return multiple bids based on one request.

For quick vehicle computations, we use a fast greedy insertion heuristic instead of solving the local vehicle problem in an exact way. The current sequence of the route will be kept, and feasibility (of time windows and capacities) will be checked for insertion of the new pickup and delivery at all possible positions (see Fig. 5).

If a vehicle agent receives the acceptance of a bid from an order agent, it checks whether including the corresponding pickup and delivery alternatives into its route is still possible for the same (or less) costs. If this can be done, the vehicle agent updates its route accordingly and confirms this to the order agent. Otherwise, it sends a negative response to the order agent, together with a new bid for the same pickup and delivery alternatives, if possible. The rationale is that the vehicle still might have a better offer than other vehicles, although the costs might be higher than in the initial bid.

Each time a vehicle agent changes its route plans (after insertion or removal of an order), it informs all order agents that are affected by the changes about their new routing costs: for all order agents that have a pickup or delivery directly before or after an inserted or deleted location in the route, the vehicle agent computes what it would gain by removing the pickup and delivery of that order. These actual routing costs will be sent to the corresponding order agents; they update the costs of their contracts, which is useful when they compare bids to their contract in a new auction.

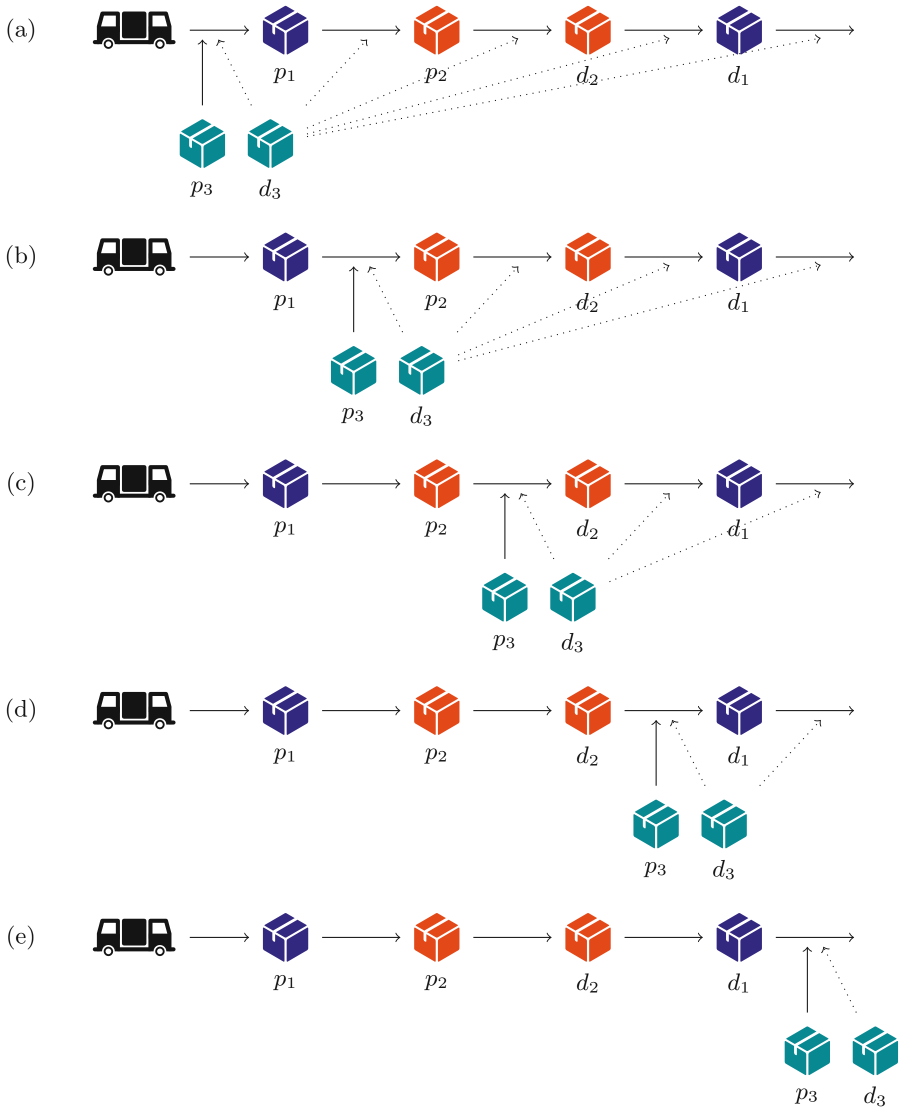

Fig. 5. Different insertion possibilities for a new pickup (p_3) and delivery (d_3) into a vehicle's route consisting of two orders. The greedy heuristic keeps the sequence of the current route ($\langle p_1, p_2, d_2, d_1 \rangle$). The number of possible routes to check is quadratic in the number of pickups and deliveries.

4 Single-Agent System

To measure the performance of our MAS, we compare with a solution that is computed in a centralized way. For such a SAS, it is assumed that all order and vehicle information is available at one central place. We use an Adaptive Large Neighborhood Search (ALNS) algorithm (Ropke and Pisinger 2006), adapted to the situation with multiple locations and preferences, as SAS. First, an initial solution is computed by a greedy heuristic. Then, ALNS iteratively looks for improvements of the current solution by changing parts of it. In each iteration, some orders are removed from the current solution, and reinserted again into the remaining routing plan. Different heuristics for removal and reinsertion can be used; they are selected based on their performance in previous iterations. For details, see Algorithm 1 and Ropke and Pisinger (2006). An indication of the quality of ALNS applied to the GPDPP is given by Los et al. (2018).

Algorithm 1: Adaptive Large Neighborhood Search

```
Input: feasible solution x
Initialize best solution x^b ← x
Initialize weights w
while stop criterion not met do
    Select destroy and repair heuristics d and r based on w
    x^t ← r(d(x))
    if accept(x, x^t) then
        x ← x^t
    end
    if costs(x^t) < costs(x^b) then
        x^b ← x^t
    end
    Update w
end
return x^b
```

5 Computational Study

To gain insight into the performance of the MAS, we compare the MAS results for different auction sizes with the SAS results on instances of moderate size.

5.1 Problem Instances

We generated a problem set with instances of 500 to 2000 orders. All orders have a random load between 0 and 100, 2 pickup alternatives, and 4 delivery alternatives on a 100×100 area. Travel times and travel costs correspond to euclidean distances between the locations. Time windows are randomly generated with length at least 30, whereas the time horizon was set to 960. For each instance, $\beta = 20$. The number of vehicles equals 20% of the number of orders, and vehicles have capacity 250 or 500.

5.2 MAS vs. SAS

To compare the performance of the MAS and the SAS and the influence of the number of vehicles in an auction, we consider the following methods:

- **Baseline**: The centralized reference baseline first computes an initial solution with a greedy heuristic and thereafter gets the full computation time to improve this solution by applying ALNS. Hence, it can be seen as a SAS where an initial solution is already given.
- **MAS-25%, MAS-50%, MAS-100%**: The three MAS methods differ in vehicle interaction percentage: order agents send in each auction a request to the most promising 25%, 50% or 100% of the available vehicles, respectively, as described in Sect. 3.2.
- **SAS**: The SAS, as described in Sect. 4, starts from scratch and uses the available computation time to both build an initial greedy solution and improve this solution by applying ALNS.

We implemented the different methods in Go and ran them 5 times on 5 problem instances of 500, 1000, and 2000 orders. Since solutions need to be provided quickly in highly dynamic real-world cases, we limited the computation time of our experiments to 5 and 10 min on an i5-4590 CPU at 3.30 GHz with 8 GB of RAM.

Table 1 shows the normalized costs relative to the baseline result, along with averages per group. Some experiments did not result in feasible solutions since some orders were not assigned to a vehicle at all, due to limited time. In general, it can be seen that for smaller computation times, lower vehicle interaction percentages, and larger problem instances, the MAS solutions get closer to the baseline solutions. Although the SAS produces better results than the MAS for the smaller instances, there is an opposite result for the larger instances: for 1000 order instances and a time limitation of 10 min, the MAS-25% method outperforms the SAS, and for the 5 min case, both the MAS-25% and the MAS-50% methods outperform the SAS. Furthermore, for the 2000 order instances, the MAS-25% method is still able to find a solution in 10 min while the SAS is not. In addition, the MAS-25% method is even highly competitive with the baseline for 1000 orders in 5 min and for 2000 orders in 10 min. Note that there are two problem instances (no. 5 of the 1000 order series and no. 2 of the 2000 order series) for which the mean MAS-25% result even outperforms the baseline result.

Table 1. Normalized costs for the different methods relative to the baseline solution (mean ± standard deviation of 5 runs per instance).

O	T	I	Baseline	MAS-25%	MAS-50%	MAS-100%	SAS
500	5	1	1.00 ± 0.04	1.14 ± 0.02	1.15 ± 0.02	1.18 ± 0.02	0.97 ± 0.01
		2	1.00 ± 0.03	1.08 ± 0.01	1.13 ± 0.00	1.12 ± 0.03	0.95 ± 0.02
		3	1.00 ± 0.06	1.15 ± 0.01	1.15 ± 0.03	1.22 ± 0.04	0.98 ± 0.02
		4	1.00 ± 0.05	1.18 ± 0.01	1.20 ± 0.03	1.25 ± 0.03	1.06 ± 0.04
		5	1.00 ± 0.03	1.20 ± 0.02	1.20 ± 0.02	1.23 ± 0.01	1.09 ± 0.06
		Avg.	1.00 ± 0.04	1.15 ± 0.01	1.17 ± 0.02	1.20 ± 0.03	1.01 ± 0.03
	10	1	1.00 ± 0.03	1.19 ± 0.02	1.21 ± 0.02	1.23 ± 0.04	0.96 ± 0.03
		2	1.00 ± 0.06	1.17 ± 0.00	1.19 ± 0.02	1.20 ± 0.02	0.95 ± 0.01
		3	1.00 ± 0.03	1.25 ± 0.02	1.24 ± 0.02	1.30 ± 0.02	1.02 ± 0.02
		4	1.00 ± 0.02	1.26 ± 0.02	1.26 ± 0.02	1.29 ± 0.03	1.04 ± 0.04
		5	1.00 ± 0.05	1.22 ± 0.02	1.25 ± 0.02	1.26 ± 0.02	1.02 ± 0.02
		Avg.	1.00 ± 0.04	1.22 ± 0.02	1.23 ± 0.02	1.25 ± 0.02	1.00 ± 0.02
1000	5	1	1.00 ± 0.03	1.00 ± 0.03	1.07 ± 0.04	⊗	⊗
		2	1.00 ± 0.03	1.03 ± 0.02	1.12 ± 0.02	⊗	1.14 ± 0.01
		3	1.00 ± 0.02	1.05 ± 0.03	1.17 ± 0.01	⊗	⊗
		4	1.00 ± 0.02	1.06 ± 0.04	1.17 ± 0.02	⊗	⊗
		5	1.00 ± 0.03	0.98 ± 0.02	1.06 ± 0.02	⊗	⊗
		Avg.	1.00 ± 0.03	1.02 ± 0.03	1.12 ± 0.02	⊗	1.14 ± 0.01
	10	1	1.00 ± 0.02	1.09 ± 0.04	1.14 ± 0.02	1.27 ± 0.01	1.10 ± 0.05
		2	1.00 ± 0.02	1.14 ± 0.04	1.16 ± 0.02	1.31 ± 0.02	1.11 ± 0.01
		3	1.00 ± 0.04	1.10 ± 0.02	1.15 ± 0.02	1.28 ± 0.02	1.16 ± 0.04
		4	1.00 ± 0.02	1.07 ± 0.03	1.14 ± 0.02	1.28 ± 0.02	1.11 ± 0.04
		5	1.00 ± 0.03	1.07 ± 0.03	1.13 ± 0.01	1.24 ± 0.01	1.05 ± 0.03
		Avg.	1.00 ± 0.02	1.09 ± 0.03	1.14 ± 0.02	1.28 ± 0.02	1.10 ± 0.03
2000	5	1	1.00 ± 0.03	⊗	⊗	⊗	⊗
		2	1.00 ± 0.04	⊗	⊗	⊗	⊗
		3	1.00 ± 0.02	⊗	⊗	⊗	⊗
		4	1.00 ± 0.03	⊗	⊗	⊗	⊗
		5	1.00 ± 0.01	⊗	⊗	⊗	⊗
		Avg.	1.00 ± 0.03	⊗	⊗	⊗	⊗
	10	1	1.00 ± 0.01	1.02 ± 0.14	⊗	⊗	⊗
		2	1.00 ± 0.03	0.95 ± 0.04	⊗	⊗	⊗
		3	1.00 ± 0.04	1.04 ± 0.13	⊗	⊗	⊗
		4	1.00 ± 0.02	1.10 ± 0.16	⊗	⊗	⊗
		5	1.00 ± 0.02	1.08 ± 0.10	⊗	⊗	⊗
		Avg.	1.00 ± 0.02	1.04 ± 0.11	⊗	⊗	⊗

O: Number of orders per instance; **T**: Computational time in minutes; **I**: Instance number; ⊗: No solution found within the given time.

5.3 Discussion

From our study, we obtain several insights for the use of MASs to model autonomous transportation:

1. Adopting a multi-agent approach for routing problems models real-world requirements by limiting information sharing, but has still good results compared with a centralized approach. Hence, the method can compete with current approaches, but shows advantages when information sharing is limited, e.g., due to a lack of trust or fierce competition. Moreover, the approach is robust in the sense that it obtains solutions when information is missing. This is a very relevant property in a multitude of future autonomous transportation systems in which cooperation is essential, but full information sharing cannot not be taken for granted.
2. Decentralized approaches result in good quality solutions for large problems when computation time is limited. Limiting the number of vehicles that take part in an auction can improve the results. Especially for the scenarios with 1000 orders and 5 min of computation time as well as for 2000 orders and 10 min of computation time, the MAS outperforms a centralized solution. For some problem instances, we demonstrate that the results of the decentralized approach even outperform a baseline centralized solution. Along with the aforementioned properties, this makes the developed MAS a suitable approach for large-scale and dynamic routing problems.

6 Conclusions and Future Research

Several studies have addressed cooperative vehicle routing problems and decentralized transportation problems. However, up to now, cooperation is commonly modeled under the sometimes biased assumption of complete information sharing. In this work, we modeled autonomous vehicles acting as independent decision makers that solve a realistic pickup and delivery problem where multiple alternative locations per order are possible within a MAS with auctions.

We find that modeling limited information sharing does not significantly worsen solutions. Moreover, the performance of the distributed MAS approach improves for large-scale, time-limited instances in comparison to a centralized approach. In this way, our results also contribute to modeling and solving large-scale dynamic cooperative vehicle routing problems appropriately. The described MAS approach allows to model the behaviour of autonomous vehicles as independent physical decision makers, which is considered an important feature of future autonomous vehicle routing. Modeling limited information sharing, this work also considers scenarios of limited trust or coopetition. Both, autonomous distributed decision making and limited information sharing, are considered important characteristics of autonomous transportation systems that will gain more importance in the next stages of vehicular automation.

Nonetheless, the numerical experiments of this work are still limited and future work could especially focus on the interaction of solution methods applied by vehicle and order agents. In this respect, the quality vs. time performance of different strategies for the local vehicle problems could be investigated. Furthermore, the approach could be extended by, e.g., allowing transfers of orders to other vehicles when they are already en route, or considering different policies regarding switching to a different pickup or delivery alternative when a contract already has been made.

Acknowledgements. This research is supported by the project "Dynamic Fleet Management (P14-18 – project 3)" (project 14894) of the Netherlands Organization for Scientific Research (NWO), domain Applied and Engineering Sciences (TTW).

References

Atasoy, B., Schulte, F., Steenkamp, A.: Platform-based collaborative routing using dynamic prices as incentives. Transp. Res. Rec. **2674**, 670–679 (2020)

Buckland, K., Sano, N.: Millennials, Toyota wants you to buy its internet-ready Corolla. Bloomberg (2018). https://www.bloomberg.com/news/articles/2018-06-26/millennials-toyota-wants-you-to-buy-its-internet-ready-corolla

Cleophas, C., Cottrill, C., Ehmke, J.F., Tierney, K.: Collaborative urban transportation: recent advances in theory and practice. Eur. J. Oper. Res. **273**, 801–816 (2019)

Cordeau, J.-F., Laporte, G., Ropke, S.: Recent models and algorithms for one-to-one pickup and delivery problems. In: Golden, B., Raghavan, S., Wasil, E. (eds.) The Vehicle Routing Problem: Latest Advances and New Challenges, pp. 327–357. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-77778-8_15

Cruijssen, F., Dullaert, W., Fleuren, H.: Horizontal cooperation in transport and logistics: a literature review. Transp. J. **46**, 22–39 (2007)

Feng, F., Pang, Y., Lodewijks, G., Li, W.: Collaborative framework of an intelligent agent system for efficient logistics transport planning. Comput. Ind. Eng. **112**, 551–567 (2017)

Gansterer, M., Hartl, R.F.: Collaborative vehicle routing: a survey. Eur. J. Oper. Res. **268**, 1–12 (2018)

Gath, M.: Optimizing Transport Logistics Processes with Multiagent Planning and Control. Springer Fachmedien Wiesbaden, Wiesbaden (2016). https://doi.org/10.1007/978-3-658-14003-8

Guajardo, M., Rönnqvist, M.: Operations research models for coalition structure in collaborative logistics. Eur. J. Oper. Res. **240**, 147–159 (2015)

Jittrapirom, P., Caiati, V., Feneri, A.-M., Ebrahimigharehbaghi, S., Alonso-González, M.J., Narayan, J.: Mobility as a service: a critical review of definitions, assessments of schemes, and key challenges. Urban Plan. **2**, 13–25 (2017)

Los, J., Schulte, F., Gansterer, M., Hartl, R.F., Spaan, M.T.J., Negenborn, R.R.: Decentralized combinatorial auctions for dynamic and large-scale collaborative vehicle routing. In: Proceedings of the 11th International Conference on Computational Logistics, ICCL 2020, Enschede, The Netherlands, pp. 215–230 (2020a)

Los, J., Schulte, F., Gansterer, M., Hartl, R.F., Spaan, M.T.J., Negenborn, R.R.: Large-scale collaborative vehicle routing. Ann. Oper. Res. (2022). https://doi.org/10.1007/s10479-021-04504-3

Los, J., Schulte, F., Spaan, M.T.J., Negenborn, R.R.: The value of information sharing for platform-based collaborative vehicle routing. Transp. Res. Part E Logist. Transp. Rev. **141**, 102011 (2020b)

Los, J., Spaan, M.T.J., Negenborn, R.R.: Fleet management for pickup and delivery problems with multiple locations and preferences. In: Proceedings of the 6th International Conference on Dynamics in Logistics, LDIC 2018, Bremen, Germany, pp. 86–94 (2018)

Máhr, T.: Vehicle routing under uncertainty. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands (2011)

Molenbruch, Y., Braekers, K., Caris, A.: Typology and literature review for dial-a-ride problems. Ann. Oper. Res. **259**, 295–325 (2017)

Munford, M.: Volkswagen's chief digital officer joins blockchain non-profit Iota Foundation. Forbes (2018). https://www.forbes.com/sites/montymunford/2018/01/23/volkswagens-chief-digital-officer-joins-cryptocurrency-iota-foundations-supervisory-board

Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems: part II: transportation between pickup and delivery locations. J. für Betriebswirtschaft **58**, 81–117 (2008)

Qu, Z., Wang, J., Hull, R.A.: Cooperative control of dynamical systems with application to autonomous vehicles. IEEE Trans. Autom. Control **53**, 894–911 (2008)

Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. **40**, 455–472 (2006)

Speranza, M.G.: Trends in transportation and logistics. Eur. J. Oper. Res. **264**, 830–836 (2018)

Wooldridge, M.: An Introduction to Multi-agent Systems. Wiley, Chichester (2009)

Zhou, M., Qu, X., Jin, S.: On the impact of cooperative autonomous vehicles in improving freeway merging: a modified intelligent driver model-based approach. IEEE Trans. Intell. Transp. Syst. **18**, 1422–1428 (2017)