Group-based Distributed Auction Algorithms for Multi-Robot Task Assignment

Xiaoshan Bai, Andres Fielbaum, Maximilian Kronmüller, Luzia Knoedler and Javier Alonso-Mora

***Abstract*—This paper studies the multi-robot task assignment problem in which a fleet of dispersed robots needs to efficiently transport a set of dynamically appearing packages from their initial locations to corresponding destinations within prescribed time-windows. Each robot can carry multiple packages simultaneously within its capacity. Given a sufficiently large robot fleet, the objective is to minimize the robots' total travel time to transport the packages within their respective time-window constraints. The problem is shown to be NP-hard, and we design two group-based distributed auction algorithms to solve this task assignment problem. Guided by the auction algorithms, robots first distributively calculate feasible package groups that they can serve, and then communicate to find an assignment of package groups. We quantify the potential of the algorithms with respect to the number of employed robots and the capacity of the robots by considering the robots' total travel time to transport all packages. Simulation results show that the designed algorithms are competitive compared with an exact centralized Integer Linear Program representation solved with the commercial solver Gurobi, and superior to popular greedy algorithms and a heuristic distributed task allocation method.**

***Note to Practitioners* — This work presents two group-based distributed auction algorithms for a sufficiently large fleet of robots to efficiently transport a set of dynamically appearing dispersed packages from their initial locations to corresponding destinations within prescribed time-windows. Each robot can carry multiple packages simultaneously within its capacity, and the objective is to minimize the robots' total travel time to transport all the packages within the prescribed time-windows. The paper's practical contributions are threefold: First, the multi-robot task assignment problem is formulated through a robot-group assignment strategy, which enables complex logistic scheduling for tasks grouped according to their distributions and time-windows. Second, we theoretically show that the multi-robot task assignment problem is an NP-hard problem, which implies the necessity for designing approximation task assignment algorithms. Third, the proposed group-based distributed auction algorithms are efficient and can be adapted for real scenarios.**

***Index Terms*—Multi-robot, task assignment, time-windows, NP-hard, distributed auction algorithm.**

This work has been supported in part by the Amazon Research Award "Predictive Multi-objective Fleet Routing and Assignment", in part by AIR-Lab, an AI for Retail Lab, funded by Ahold Delhaize, in part by the National Natural Science Foundation of China under Grant 62003217, and in part by the European Union's Horizon 2020 research and innovation programme under Grant 101017008.

Xiaoshan Bai is with the College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Cognitive Robotics, Delft University of Technology, Delft, 2628 CD, The Netherlands (e-mail: baixiaoshan@szu.edu.cn).

Andres Fielbaum, Maximilian Kronmüller, Luzia Knoedler and Javier Alonso are with the Department of Cognitive Robotics, Delft University of Technology, Delft 2628 CD, The Netherlands (e-mail: A.S.FielbaumSchnitzler@tudelft.nl; M.Kronmuller@tudelft.nl; L.Knoedler@tudelft.nl; j.alonsomora@tudelft.nl).

I. INTRODUCTION

To enable robots to effectively accomplish various tasks such as terrain mapping, environmental monitoring, disaster rescue and logistics scheduling, the multi-robot task assignment problem must be solved [1–4]. For the task assignment of multi-robot systems, the overall group task generally consists of a group of subtasks that individual robots can complete simultaneously [5, 6]. A typical task assignment scenario is to deploy several robots to visit a set of target locations [7, 8], which is a variant of the vehicle routing problem (VRP) [9]. For the VRP, a fleet of vehicles with limited capacity needs to deliver products from one or several depots to a group of dispersed customers [9]. It has been shown that the VRP and some multi-robot task assignment problems are NP-hard [10], requiring extremely long computation time to achieve the optimal solution with an increasing number of vehicles and customers. For solving NP-hard problems, heuristic algorithms are usually designed to achieve a sub-optimal assignment solution [11]. The VRP with pickup and delivery with time windows (VRPPDTW) is the extension of the VRP in which each request, having a specific pickup point and a delivery location, must be served by a vehicle within an associated time-window [12–15]. The VRP and VRPPDTW have given rise to a number of computation methods, including exact algorithms, such as branch-and-cut algorithms and dynamic programming approaches [13–15], and heuristic algorithms [16, 17] where genetic algorithms and simulated annealing algorithms are popular [12, 18].

This paper studies the task assignment problem for multiple robots to efficiently transport a set of dispersed packages from different initial locations to corresponding destinations in a warehouse within prescribed time-windows. Robots can carry multiple packages simultaneously not exceeding their maximal capacity. To transport each package, a robot needs to first move to the initial location of the package to pick it up, and then transport it to the associated destination. Given a sufficiently large robot fleet, the objective is to minimize the total operation cost for transporting all packages within their time-windows, which is quantified by the robots' total travel time.

The main contributions of this paper are as follows. First, this paper utilizes a novel robot-group assignment strategy to formulate the distributed multi-robot task assignment problem as a two-stage optimization problem, inspired by the centralized methods for ridesharing [19]. In the first stage, each robot individually calculates feasible package groups. Each feasible package group contains a set of packages that is feasible for the robot to serve while respecting the corresponding time-windows and the robot's capacity constraint. In the second stage, an integer programming model is used to formulate an auction-based distributed assignment of the feasible package groups to the robots. Second, we theoretically show that determining whether a feasible solution exists for the resulting optimization problem is NP-complete, and that if we assume its existence the associated optimization problem is NP-hard, which shows the necessity for designing approximation algorithms to solve the problem. Third, we propose two group-based distributed auction algorithms, which can achieve satisfying solutions to the task assignment problem compared with several baseline methods.

The rest of this paper is organized as follows. Section II presents the related work on the studied task assignment problem. In Section III, the formulation of the multi-robot package delivery task assignment problem is given. Section IV presents the group-based distributed auction algorithms and the associated analysis. We show the simulation results in Section V and conclude the paper in Section VI.

II. Related work

Task assignment problems for a team of robots to visit a set of targets, with each target assigned to exactly one robot and one robot can be assigned to multiple targets, can be solved by either centralized or distributed algorithms [10].

A. Centralized algorithms

Considering vehicles' limited loading capacity, a genetic algorithm was designed for task assignment of multiple unmanned aerial vehicles (UAVs) in [20]. Later on, a Dubins car model was integrated with a genetic algorithm in [21] for multi-UAV target assignment considering the UAVs' limited turning radius. Centralized algorithms can achieve near-optimal or optimal assignments; however, in dynamic environments they require persistently reliable communication between individual robots and the central server to obtain global information. The performance of centralized algorithms can deteriorate as communication gets worse [22]. In [19], a centralized on-demand ride-sharing algorithm was proposed to solve the ride-sharing problem, where multiple vehicles with limited capacity are assigned to pickup and deliver a set of passengers within a prescribed maximal travel delay. For a set of dynamically appearing requests, the algorithm first calculates a group of feasible trips that can be served by each vehicle, and then optimally solves an integer linear program to match vehicles to the trips.

To reduce computational time, Simonetto $et\ al.$ [23] leverage a one-to-one assignment strategy and an insertion heuristic, which inserts at most one new task/customer into each vehicle's current route at one optimization run. It reduces the ridesharing optimization problem to a linear assignment problem, and then solves the integer linear programming in a centralized fashion. It has been shown that the dynamic ridesharing algorithm in [23] has competing performance compared with the on-demand ride-sharing algorithm in [19]. Several multi-robot task assignment problems have been shown to be NP-hard [10], demanding high computation capacity for the central server. As a result, centralized methods may not be scalable with the growing numbers of targets and/or robots, and are not suitable for environments where a central server is not available.

B. Distributed algorithms

In parallel, distributed algorithms have been developed to enable each robot to plan its route [24–26]. In [27], a heuristic distributed algorithm was proposed. It changes the assignment of a target based on the cost saved by removing the target from the ordered robot's route containing the target, and the smallest marginal cost for inserting the target into another ordered robot's route. Several distributed algorithms were proposed in [8] to minimize the robots' total travel distance until every target location was occupied by one robot under limited communication and sensing ranges. The routing of multiple robots to serve spatially distributed requests at specified time instants was formulated as a pure assignment problem in [28]. However, the corresponding set of planar positions that require simultaneous service at each time instant is assumed to be initially known by every robot. In [29, 30], the pickup and delivery task assignment for multiple robots was studied. The objective of [29] is to minimize the average service time to execute each task, and the objective in [30] is to minimize the largest execution time of all the tasks. However, the robots' capacity to carry packages is one in [29, 30], which does not make use of robots with a higher capacity. In contrast, the robots in this paper can carry multiple packages simultaneously. This complicates the assignment problem due to a larger solution space resulting from the robots' higher capacities.

In recent years, auction-based methods are popular for solving task assignment problems due to their computational efficiency [11, 31–33]. For the static task assignment of homogeneous robots, the auction-based algorithm proposed in [31] guarantees that the robots' total travel time is at most twice of the optimal assuming the connectivity of the communication network, i.e., two arbitrary robots can exchange information with each other through direct or indirect communication. Under the assumption of connectivity of the communication network, an auction algorithm, namely the consensus-based bundle algorithm (CBBA), was designed for multi-agent task allocation [11]. A robot using CBBA bids for a task based on the cost incurred by inserting the task into the robot's target bundle. However, most of the discussed auction-based methods iteratively assign a single unassigned target to one robot until all the targets are assigned. This paper borrows the centralized vehicle-group assignment algorithmic idea from [19] to design two auction-based distributed task assignment algorithms, which iteratively assigns feasible package groups to the robots.

III. Problem formulation

Consider that multiple dispersed robots need to transport a group of dynamically appearing packages, located at dispersed workstations, to their destinations within prescribed time-windows. Each package has a release time that is the earliest time the robots can pick it up, and a latest time to be delivered to its destination. The release times and the positions of the newly appearing packages are not initially known. The information is collected during a fixed time span, after which new packages, as well as packages that have not been picked up yet, are assigned to the robots in the team. The new assignment may modify the schedule for packages already on the robots, but they remain on the robot that is carrying them. Robots have the same capacity in terms of the number of packages that each robot can carry simultaneously. Each robot moves with a constant unit speed for transporting packages and stops moving after finishing its tasks.

It is assumed that the number of robots is sufficiently large to transport all packages within their time-windows such that a feasible solution exists. The aisles of the warehouse are spacious compared with the size of the robots, where the robots have good sensing capabilities to avoid robot-robot or robot-environment collisions. In this work, we do not consider the robots' battery constraint with the assumption that the robots have a quick battery swapping or an efficient inductive charging for battery, where each workstation can be equipped with an inductive charging area and a robot is charged once it is in this area [34]. The objective is to minimize the robots' total travel time to transport all the packages within their time-windows.

A. *Formulation as an optimization problem*

We use $\mathcal{P} = \{1, \cdots, n\}$ to denote the set of indices of n randomly distributed packages that need to be transported from their initial locations to corresponding dispersed destinations. $\mathcal{R} = \{n+1, \cdots, n+m\}$ denotes the set of indices of $m > 1$ robots that are initially located at dispersed depots. We use o_k to represent the origin of robot $k \in \mathcal{R}$. Each package $i \in \mathcal{P}$ is associated with a given tuple (o_i, r_i, d_i, l_i), where o_i is the origin of i, r_i is the release time of i, d_i is the destination of i, and l_i is the latest time to transport i to d_i. The tuple information on the newly appearing packages is collected during each fixed time span. It is assumed that the robots can carry a maximum of C packages simultaneously. Let $n_k(t)$ be the number of packages carried by robot $k \in \mathcal{R}$ at time instant t, and $\mathcal{I} = \{o_1, ..., o_{n+m}, d_1, ..., d_n\}$. We use variable $a(j)$, initialized as $a(j) = 0$, to denote the time when a robot reaches location $j \in \mathcal{I}$. For any pair of $i, j \in \mathcal{I}$, the variable $t(i, j)$ is used to denote the shortest time needed for a robot to travel from i to j. Obviously $t(i, i) = 0$ for each $i \in \mathcal{I}$. For each package $i \in \mathcal{P}$, it is assumed that at least one robot is viable to transport i to its destination within the corresponding time-window.

Assume that at an arbitrary task assignment time instant, new packages, as well as packages that have not been picked up yet construct the package set $\mathcal{P}' \subseteq \mathcal{P}$. We use $\pi(k, \mathcal{P}_k)$ to denote the optimal route for robot k to transport all the packages in set $\mathcal{P}_k \subseteq \mathcal{P}'$ assigned to it with the minimum total travel time while respecting the robot's capacity constraint and the time-window to transport each package in $\mathcal{P}_k$. The optimal route $\pi(k, \mathcal{P}_k)$ can be achieved by an exhaustive search or approximated by heuristic methods, which will be discussed later. The minimum total travel time for robot k to transport all the packages in $\mathcal{P}_k$ is

$$c(k, \mathcal{P}_k) = \sum_{i=1}^{|\pi(k,\mathcal{P}_k)|-1} t(\pi_i(k, \mathcal{P}_k), \pi_{i+1}(k, \mathcal{P}_k)),$$

where $\pi_i(k, \mathcal{P}_k)$ is the index of the vertex located at the ith position of $\pi(k, \mathcal{P}_k)$. Then, the objective to minimize the robots' total travel time for transporting all the packages within their time-windows at the task assignment time instant can be expressed as follows.

Problem 1 Given a set of package transportation requests $\mathcal{P}'$ and a sufficiently large robot fleet at each task assignment time instant, find a package set $\mathcal{P}_k \subseteq \mathcal{P}'$ and the associated route $\pi(k, \mathcal{P}_k)$ for each robot $k \in \mathcal{R}$ to solve

$$\min \sum_{k \in \mathcal{R}} c(k, \mathcal{P}_k), \tag{1}$$

subject to

$$\begin{aligned} \mathcal{P}_k \cap \mathcal{P}_j &= \emptyset, \forall k, j \in \mathcal{R}, j \neq k; && (2)\\ \cup_{k\in\mathcal{R}} \mathcal{P}_k &= \mathcal{P}'; && (3)\\ n_k(t) &\leq C, \forall k \in \mathcal{R}, \forall t \in \mathbb{R}^+; && (4)\\ a(d_i) &\leq l_i, \forall i \in \mathcal{P}'; && (5)\\ a(o_i) + \max\{r_i - a(o_i), 0\} &\leq a(d_i) - t(o_i, d_i), \forall i \in \mathcal{P}'. && (6) \end{aligned}$$

Constraint (2) requires that each package is transported exactly by one robot; (3) ensures that all the packages will be transported; (4) ensures that the robots' capacity constraint is always satisfied; (5) ensures that the time for delivering each package to its destination is no later than the corresponding latest delivery time; (6) implies the earliest time that a package can be delivered to its destination.

Problem 1 imposes that all packages are transported within their specified time-windows, entailing that this formulation and the subsequent solution methods are intended for scenarios in which the number of available robots is sufficiently large. An alternative approach would be to include a rejection penalty for each package that is not served, which is well suited for centralized algorithms, and was employed by [19], where rejections occurred. If not requiring every package to be served, Problem 1 could be formulated as a submodular one with *matroid constraints*[1], for which sub-optimality bounds are known [35]. However, the objective function would be different as it would include the mentioned rejection penalties. For the reminder of this paper, we assume that a sufficiently large robot fleet is available. A relaxation of this assumption, potentially by reformulating the problem as discussed, is outside the scope of this paper.

Remark 1. *At each task assignment time instant, the studied multi-robot task assignment problem is in essence the NP-hard VRPPDTW, which generalizes the vehicle routing problem with time windows (VRPTW) [36]. Indeed, even finding a feasible*

[1]The main reason precluding our approach to fit into a matroid optimization scheme is that not assigning any task should be regarded as feasible, which is not the case when we require that every package is served. For details, see [35].

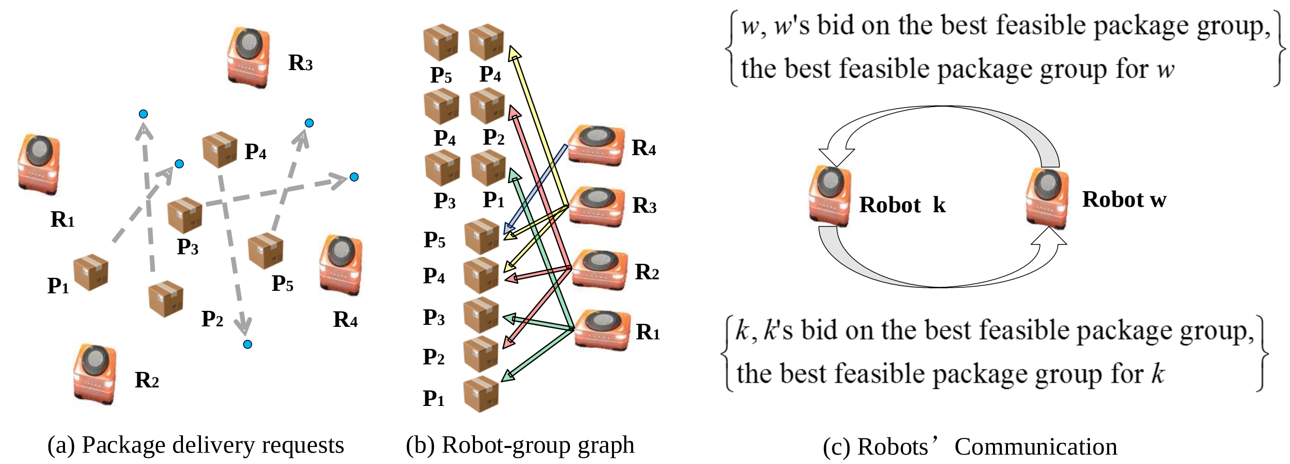

Fig. 1. Schematic overview of the group-based distributed auction algorithms: (a) Illustration of 5 packages (orange box, origin; blue circle, destination) and 4 robots (orange autonomous ground vehicle, origin); (b) RG-graph consists of all the feasible package groups that each robot can serve while respecting the prescribed time-windows and robots' capacities, in which a robot is connected to a package group if the group is feasible for the robot to serve; (c) The information exchange between communication-connected robots guided by ADA.

solution to the VRPTW with a fixed fleet size is NP-complete [37].

IV. AUCTION-BASED DISTRIBUTED ALGORITHMS

As Problem 1 is NP-hard, it is time-consuming to solve (1) optimally. In this section, we propose two group-based distributed auction algorithms to solve the package transport task assignment problem under the assumption of a sufficiently large robot fleet. The two algorithms consist of two main procedures: 1) each robot first distributively computes feasible package groups that it can serve considering the time-windows to transport the packages and the robot's capacity constraint; 2) robots decide which robot should win which package group based on their bids for transporting the packages contained in each feasible package group.

A. Generating feasible package groups

In this paper, we refer to a set of packages as a group, and a package group $\mathcal{G} \subseteq \mathcal{P}'$ is feasible for a robot $k \in \mathcal{R}$ to serve if the robot can transport all the packages in $\mathcal{G}$ to their corresponding destinations while satisfying the time-windows to transport the packages and the robot's capacity constraint. In Fig. 1 (b), the Robot-group graph (RG-graph) represents which package groups are feasible for which robots to serve considering the prescribed time-windows, the robots' capacities and current positions, which is inspired by the batch assignment graphs in [19]. In the RG-graph, a package $P_i \in \mathcal{P}'$ might be directly connected to a robot $R_k \in \mathcal{R}$ if robot R_k can transport package P_i to its destination while satisfying the time-window to transport the package. The edge is denoted by $e(P_i, R_k)$. Two packages P_i and P_j belong to the same package group if an empty robot, starting at its origin, could transport the two packages to their destinations while satisfying the corresponding time-windows.

For a group $\mathcal{G}$ that is feasible for robot k to serve, we use $\pi(k, \mathcal{G})$ to denote the optimal route for k to transport all the packages in $\mathcal{G}$, and $c(k, \mathcal{G})$ is the corresponding total travel time for k to transporting all the packages in $\mathcal{G}$ by following the route $\pi(k, \mathcal{G})$. $\mathcal{F}_k \subseteq S(\mathcal{P}')$ is used to contain all the package groups feasible for robot k to serve, where $S(\mathcal{P}')$ is the power set of the package set $\mathcal{P}'$. To determine the feasibility of a package group relative to a robot and the optimal route for the robot to serve a feasible package group, one needs to solve a VRP with time-windows for a single robot starting at a given initial position and transporting each package in the group from the package's origin to its destination within the corresponding time-window. We use function feasible$(k, \mathcal{G})$ to check the feasibility of package group $\mathcal{G}$ relative to robot k, and travel$(k, \mathcal{G})$ to achieve the optimal route $\pi(k, \mathcal{G})$ for k to serve a feasible group $\mathcal{G}$. Then, the minimum total travel time $c(k, \mathcal{G})$ for robot $k \in \mathcal{R}$ to transport all the packages in the feasible package group $\mathcal{G}$ is

$$c(k,\mathcal{G}) = \begin{cases} \sum_{i=1}^{|\pi(k,\mathcal{G})|-1} t(\pi_i(k,\mathcal{G}), \pi_{i+1}(k,\mathcal{G})), & \text{if } \mathcal{G} \subseteq \mathcal{F}_k, \\ \infty, & \text{otherwise,} \end{cases} \tag{7}$$

where $\pi_i(k, \mathcal{G})$ is the index of the ith ordered element on the route $\pi(k, \mathcal{G})$ and $|\pi(k, \mathcal{G})|$ is the number of elements contained in $\pi(k, \mathcal{G})$. For robots with low capacity and packages with tight time-windows, feasible$(k, \mathcal{G})$ and travel$(k, \mathcal{G})$ can be implemented via an exhaustive search. For robots with higher capacity, efficient heuristic methods such as the topological sorting based algorithms [38], marginal-cost based algorithms [39] and genetic algorithms [40] can be used to achieve near-optimal routes. It should be noticed that several package groups of varying sizes might contain a particular package, and a package group might be feasible for more than one robot to serve.

Algorithm 1 Each robot $k \in \mathcal{R}$ independently calculates the feasible package groups that it can serve at the task assignment time instant.

Input: Package set $\mathcal{P}'$, and (o_i, r_i, d_i, l_i) for each package $i \in \mathcal{P}'$.
Output: The set $\mathcal{F}_k$ contains all package groups feasible for k.

$\mathcal{F}_k^1 \leftarrow \emptyset$.
for $\forall i \in \mathcal{P}'$ **do**
  if feasible$(k, \{i\})$ **then**
    $\mathcal{F}_k^1 \leftarrow \mathcal{F}_k^1 \cup \{i\}$.
  end if
end for
$s \leftarrow 2$.
while $\mathcal{F}_k^{s-1} \neq \emptyset$ **do**
  $\mathcal{F}_k^s \leftarrow \emptyset$.
  for $\forall \mathcal{G} \subseteq \mathcal{F}_k^{s-1}$ **do**
    for $\forall i \in \mathcal{F}_k^1$ **do**
      if feasible$(k, \mathcal{G} \cup \{i\})$ and $i \notin \mathcal{G}$ **then**
        $\mathcal{F}_k^s \leftarrow \mathcal{F}_k^s \cup \{\mathcal{G} \cup \{i\}\}$.
      end if
    end for
  end for
  $s \leftarrow s + 1$.
end while
$\mathcal{F}_k \leftarrow \mathcal{F}_k^1 \cup \mathcal{F}_k^2 \cup \cdots \cup \mathcal{F}_k^s$.

The group feasibility has the useful property that all the subsets/subgroups of a feasible package group are feasible. In other words, we have $\mathcal{G}' \subseteq \mathcal{F}_k, \forall \mathcal{G}' \subset \mathcal{G}$, if a package group $\mathcal{G}$ satisfies $\mathcal{G} \subseteq \mathcal{F}_k$. This property enables us to design a procedure that iteratively generates the sets $\mathcal{F}_k^1, \mathcal{F}_k^2, \ldots$, containing feasible package groups of size $1, 2, \ldots$ for robot k. For robots with higher capacity and packages with wide time-windows, we calculate the feasible package groups $\mathcal{G} \subseteq \mathcal{F}_k^s$ for each robot k with increasing group size $s = |\mathcal{G}|$ as below. First, when the group size s is small, we use exhaustive search to achieve all the feasible package groups $\mathcal{G} \subseteq \mathcal{F}_k^s$ for each robot k as well as the associated optimal route $\pi(k, \mathcal{G})$ for k to transport all the packages in each $\mathcal{G}$. Then, to check whether $\mathcal{G} \cup \{i\}, i \notin \mathcal{G}$ the union of a feasible package group $\mathcal{G}$ and package i is feasible, we try all the potential positions for inserting i at the optimal route $\pi(k, \mathcal{G})$. If one feasible route is found for k to transport all the packages in $\mathcal{G} \cup \{i\}$ by using $\pi(k, \mathcal{G})$, the function feasible$(k, \mathcal{G} \cup \{i\})$ returns to 1, and the associated optimal route $\pi(k, \mathcal{G} \cup \{i\})$ is calculated by comparing the performance of the feasible route with the feasible route, resulting from the marginal-cost based algorithm [37], for k to transport all the packages in $\mathcal{G} \cup \{i\}$. The above procedures are repeated to compute other feasible package groups.

Each robot first uses Algorithm 1 to distributively calculate feasible package groups incrementally in group size that it can serve. If exhaustive search is used to compute all feasible package groups, Algorithm 1 is complete; otherwise it is not. The second procedure of the group-based distributed auction algorithms is to solve the following robot-group assignment optimization problem, which is a mapping of robots to feasible package groups.

Problem 2 (Robot-Group Assignment) Given a sufficiently large robot fleet $\mathcal{R}$, a set of feasible package groups $\mathcal{F}_k, \forall k \in \mathcal{R}$, and the minimum total travel time $c(k, \mathcal{G})$ for robot k to serve each group $\mathcal{G} \subseteq \mathcal{F}_k$ at each task assignment time instant, solve

$$\min \sum_{k \in \mathcal{R}, \mathcal{G} \subseteq \mathcal{F}_k} x_{k\mathcal{G}} \cdot c(k, \mathcal{G}), \tag{8}$$

subject to

$$\sum_{\mathcal{G} \subseteq \mathcal{F}_k} x_{k\mathcal{G}} \leq 1, \ \forall k \in \mathcal{R}; \tag{9}$$

$$\sum_{k \in \mathcal{R}, \mathcal{G} \subseteq \mathcal{F}_k} x_{k\mathcal{G}} \cdot \mathbf{1}_{\mathcal{G}(i)} = 1, \ \forall i \in \mathcal{P}'; \tag{10}$$

$$x_{k\mathcal{G}} \in \{0, 1\}, \ \forall k \in \mathcal{R}, \forall \mathcal{G} \subseteq \mathcal{F}_k, \tag{11}$$

where the operator $\mathbf{1}_{\mathcal{G}(i)} = 1$ if $i \in \mathcal{G}$, and otherwise $\mathbf{1}_{\mathcal{G}(i)} = 0$. Constraint (9) ensures that at most one package group is assigned to each robot; (10) enforces that every package is assigned to exactly one robot; (11) implies that $x_{k\mathcal{G}} = 1$ if package group $\mathcal{G}$ is assigned to robot k, and otherwise $x_{k\mathcal{G}} = 0$. Later on, we will show that an optimal solution of Problem 2 is optimal for Problem 1.

B. Assigning feasible package groups

Having reformulated the task assignment problem, we now construct two group-based distributed auction algorithms to solve Problem 2.

1) Auction-based distributed algorithm: After computing the set of feasible package groups that each robot can serve, robots, guided by the first auction-based distributed algorithm (ADA), will iteratively bid on feasible package groups to optimize Problem 2. As each robot might be able to serve multiple package groups, it is both challenging and important to design a proper bidding mechanism to assign package groups to robots to optimize (8). The package set $\mathcal{P}^u$, initialized as $\mathcal{P}'$, contains packages currently unassigned, and the robot set $\mathcal{R}^u$, initialized as $\mathcal{R}$, contains all robots currently without any assignments. According to (7) and (8), minimizing the robots' total travel time to transport all packages is equivalent to minimizing the robots' average travel time to transport each package. As a result, the bid for robot $k \in \mathcal{R}$ on package group $\mathcal{G} \subseteq \mathcal{P}^u$ can be set to

$$b(k, \mathcal{G}) = \begin{cases} c(k, \mathcal{G})/|\mathcal{G}|, & \text{if } \mathcal{G} \subseteq \mathcal{F}_k \text{ and } \mathcal{G} \neq \emptyset, \\ \infty, & \text{otherwise}, \end{cases} \tag{12}$$

where $|\mathcal{G}|$ is the number of packages contained in $\mathcal{G}$.

Then, each robot $k \in \mathcal{R}^u$ computes the package group $\mathcal{G}_k^\star \subseteq \mathcal{F}_k$ with the lowest average cost for k to serve, where

$$\mathcal{G}_k^\star = \underset{\mathcal{G} \subseteq \mathcal{P}^u, \ \mathcal{G} \subseteq \mathcal{F}_k}{\operatorname{argmin}} \ b(k, \mathcal{G}). \tag{13}$$

Afterwards, each robot $k \in \mathcal{R}^u$ simultaneously relays the information tuple $\{k, b(k, \mathcal{G}_k^\star), \mathcal{G}_k^\star\} \subseteq \mathbb{R}^{n+2}$ to its communication-connected neighbours, where n binary digits can be used to identify and store a group (subset) of the n packages in set $\mathcal{G}_k^\star$. It uses a matrix $\mathcal{X}_k \in \mathbb{R}^{m*(n+2)}$ with its wth row $\{w, b(w, \mathcal{G}_w^\star), \mathcal{G}_w^\star\}$ to store the information tuple relayed from neighbour robot $w \in \mathcal{R} \setminus \{k\}$. It is straightforward to check that after at most $d \leq m$ rounds of simultaneous communication, with d the diameter of the communication network, each robot k has updated all the information in $\mathcal{X}_k$. At this time, each robot guided by ADA will determine and agree that the unassigned robot $k^\star \in \mathcal{R}^u$ wins the corresponding package group $\mathcal{G}^\star_{k^\star} \subseteq \mathcal{F}_{k^\star}$, where

$$k^\star = \underset{k \in \mathcal{R}^u}{\operatorname{argmin}}\, b(k, \mathcal{G}^\star_k). \tag{14}$$

Meanwhile, $\mathcal{P}^u$ and $\mathcal{R}^u$ are updated to

$$\mathcal{P}^u = \mathcal{P}^u \setminus \mathcal{G}^\star_{k^\star}, \qquad \mathcal{R}^u = \mathcal{R}^u \setminus \{k^\star\}. \tag{15}$$

The target assignment procedure from (13) to (15) continues until $\mathcal{P}^u$ is empty.

If not all of the packages in $\mathcal{P}^u$ are assigned when the robot set $\mathcal{R}^u$ is empty, ADA fails to find a feasible solution of Problem 2. ADA will, however, return a solution where a large number of packages are assigned and can be transported. This is further discussed in Section IV-C.

2) Marginal-cost based auction algorithm: Inspired by the strategy of using regret for not making an assignment [41], we propose the marginal-cost based auction algorithm (MAA), which iteratively assigns feasible package groups to robots based on each robot's current best feasible package group and the second best feasible package group. In the same way as ADA, each robot $k \in \mathcal{R}^u$ guided by MAA first calculates the best package group $\mathcal{G}^{1^\star}_k \subseteq \mathcal{F}_k$ that satisfies

$$\mathcal{G}^{1^\star}_k = \underset{\mathcal{G}^1_k \subseteq \mathcal{F}_k}{\operatorname{argmin}}\, b(k, \mathcal{G}^1_k). \tag{16}$$

Then, if $\mathcal{F}_k \setminus \mathcal{G}^{1^\star}_k \neq \emptyset$, each robot $k \in \mathcal{R}$ calculates the second best feasible package group $\mathcal{G}^{2^\star}_k$ that satisfies

$$\mathcal{G}^{2^\star}_k = \underset{\mathcal{G}^2_k \subseteq \mathcal{F}_k \setminus \mathcal{G}^{1^\star}_k}{\operatorname{argmin}}\, b(k, \mathcal{G}^2_k), \tag{17}$$

and otherwise let $b(k, \mathcal{G}^{2^\star}_k) = 0$.

Afterwards, each robot k simultaneously relays the information tuple $\{k, b(k, \mathcal{G}^{1^\star}_k), b(k, \mathcal{G}^{2^\star}_k), \mathcal{G}^{1^\star}_k\} \in \mathbb{R}^{n+3}$ to its communication-connected neighbours, and uses a matrix $\mathcal{X}_k \in \mathbb{R}^{m*(n+3)}$ with its wth row $\{w, b(w, \mathcal{G}^{1^\star}_w), b(w, \mathcal{G}^{2^\star}_w), \mathcal{G}^{1^\star}_w\}$ to store the information tuple relayed from robot $w \in \mathcal{R} \setminus \{k\}$. After at most d rounds of simultaneous communication, each robot k has updated all the information in $\mathcal{X}_k$. At this time, each robot guided by MAA determines that $k^\star \in \mathcal{R}^u$ wins the corresponding package group $\mathcal{G}^{1^\star}_{k^\star} \subseteq \mathcal{F}_{k^\star}$, where

$$k^\star = \underset{k \in \mathcal{R}^u}{\operatorname{argmax}}\, |b(k, \mathcal{G}^{1^\star}_k) - b(k, \mathcal{G}^{2^\star}_k)|. \tag{18}$$

Then, $\mathcal{P}^u$ and $\mathcal{R}^u$ are updated to

$$\mathcal{P}^u = \mathcal{P}^u \setminus \mathcal{G}^{1^\star}_{k^\star}, \qquad \mathcal{R}^u = \mathcal{R}^u \setminus \{k^\star\}. \tag{19}$$

The target assignment procedure from (16) and (19) continues until $\mathcal{P}^u$ is empty. If not all packages in $\mathcal{P}^u$ are assigned when the robot set $\mathcal{R}^u$ is empty, MAA fails to find a feasible solution of Problem 2. MAA will, however, return a solution where a large number of packages are assigned and can be transported. This is further discussed in Section IV-C.

Remark 2. *When the warehouse is cluttered with shelves and other obstacles, the designed algorithms ADA and MAA can also be applied to compute an assignment of packages to robots by calculating the shortest feasible time and the associated path for a robot to travel between any two locations in the warehouse using Dijkstra's algorithm [26].*

C. Theoretical analysis

We first show the motivation for optimizing Problem 2 by investigating its relationship with Problem 1.

Lemma 1. *Assume that, for each robot k and each group $\mathcal{G}$, the optimal travel cost $c(k, \mathcal{G})$ is known. Then, an optimal solution of Problem 2 is also optimal for Problem 1.*

We omit the proof for Lemma 1 since it is straightforward based on (1), (7), and (8).

A feasible solution to Problem 2 requires that all packages are served, which might be impossible to do depending on the available number of robots and the rigid time-windows as shown by [37]. We now show that the feasibility of Problem 2 is a complex issue. To do so, we first introduce the well-known Boolean satisfiability problem abbreviated as SAT.

Definition 1. (SAT [42]) Given a set of m boolean logical variables $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ where corresponding to each variable u_j are a *true* literal u_j and a *false* literal $\neg u_j$, and a set of n' clauses $\{C_1, C_2, \ldots, C_{n'}\}$ where each clause consists of a set of boolean variables linked by logical *OR* operator such as $C_1 = u_1 \vee \neg u_2 \vee u_m$, SAT is to determine whether a solution exists for the formula $C_1 \wedge C_2 \wedge \cdots \wedge C_{n'} = 1$ by setting the boolean variables, where $\wedge$ is the logical operator *AND*.

SAT is an NP-complete problem [42]. We now analyze the computational complexity to determine whether Problem 2 has a feasible solution.

Theorem 1. *Determining whether Problem 2 has a feasible solution is NP-complete.*

Proof. To prove the theorem, it suffices to show that determining whether Problem 2 under a particular instance has a feasible solution is NP-complete. We prove it by reducing SAT to the particular instance of Problem 2 in which each robot has at most 2 maximal feasible package groups to choose to serve and relaxing (10) as $\sum_{k \in \mathcal{R}, \mathcal{G} \subseteq \mathcal{F}_k} x_{k\mathcal{G}} \cdot \mathbf{1}_{\mathcal{G}(i)} \geq 1,\ \forall i \in \mathcal{P}'$. A feasible package group $\mathcal{G}$ is called maximal for a robot, if no package group G' with $\mathcal{G} \subset \mathcal{G}'$ is feasible for the robot to serve.

We construct the particular instance of Problem 2 where the number of packages in $\mathcal{P}'$ to be assigned is n' and each robot has at most two maximal feasible package groups to choose to serve. Each package $i \in \mathcal{P}'$ is associated with a clause C_i, and each robot j is associated with a logical variable u_j. For each robot j associated with the variable u_j, the clauses corresponding to the packages in one of j's maximal package groups contain u_j, and the clauses corresponding to the packages in the other maximal group contain $\neg u_j$. Then, each clause C_i in SAT can be represented as which robots can be used to serve each package, such as $C_1 = u_1 \vee \neg u_2 \vee u_m$ implying that package 1 can be served by robots 1, 2 and m. The reduction from SAT to the particular instance of Problem 2 is in polynomial time.

After reducing SAT to the particular instance of Problem 2, it is straightforward to see that determining whether all the packages can be assigned in the particular instance of Problem 2 is equivalent to determining whether a solution exists for the formula of SAT. As SAT is NP-complete, determining whether the particular instance of Problem 2 has a feasible solution is NP-complete. Thus, the proof is complete. □

Theorem 1 states that not only Problem 2 might be unfeasible, but also that determining whether that is the case might not be doable in practice.

Remark 3. *Theorem 1 affects both of our algorithms ADA and MAA, as they both run in polynomial time with respect to the number of feasible package groups in Problem 2 and therefore it is impossible that they guarantee finding a feasible solution when there is one (unless $P = NP$). ADA and MAA might fail to find a feasible solution to Problem 2 even if a feasible solution exists. As a result, it is necessary to check whether the package set $\mathcal{P}^u$, containing all currently-unassigned packages, is empty when the robot set $\mathcal{R}^u$ is empty.*

Thus, the proposed auction-based algorithms are primarily designed for scenarios where feasibility is not expected to be an issue, namely when the number of robots is large enough. This is formalized in Assumption 1 below, which we hold for the rest of the paper (including all of our numerical simulations in Section V).

Assumption 1: *The robot number is large enough so that both algorithms ADA and MAA are able to find a feasible solution.*

Without Assumption 1, the main problem could be to maximize the number of transported packages as [43]. Adapting our framework for this problem is beyond the scope of this paper, but is a promising direction for future research.

When multiple feasible assignment solutions exist in the presence of Assumption 1, achieving the optimal solution to Problem 2 might be still non-trivial. To analyze it rigorously, we first introduce the weighted set-partitioning problem.

Definition 2. (Weighted set-partitioning problem [44]) Given a collection of sets $\mathcal{S} = \{S_1, S_2, \ldots, S_l\}$ where each set S_i has a nonnegative weight w_i and let $\mathcal{F} = \cup_{i=1}^{l} S_i$, the *weighted set-partitioning problem* is to find a subset of the family specified by $I \subseteq \{1, 2, \ldots, l\}$ such that $\mathcal{F} = \cup_{i\in I} S_i$ for which the total weight $\sum_{i\in I} w_i$ is minimized while any two different $i, j \in I$ satisfy $S_i \cap S_j = \emptyset$.

The weighted set-partitioning problem has been shown to be NP-hard [45]. We now analyze the computational complexity to optimize Problem 2.

Remark 4. *Optimally solving Problem 2 is NP-hard.*

We show the NP-hardness of Problem 2 under a particular instance in which all the robots have the same feasible package groups as well as the same optimal cost to transport all the packages in each feasible package group. In Problem 2, each package group $\mathcal{G}$, feasible for a robot k to serve while satisfying the corresponding time-windows, can be treated as a set in the weighted set-partitioning problem [44], and the corresponding total travel time $c(k, \mathcal{G})$ for robot k to serve all the packages in $\mathcal{G}$ is the associated weight for the set. The goal of Problem 2 is to choose a set of such un-overlapping feasible package groups that each package group is specified for one certain robot (each specified robot is assigned exactly with one feasible package group), and the union of the chosen package groups contains all the packages to be delivered while minimizing the total cost for the robots to serve the feasible package groups. Then, it is straightforward that the NP-hard weighted set-partitioning problem can be reduced to Problem 2 under the particular instance within polynomial computation time based on Definition 2, where an optimal solution to the latter leads to an optimal solution to the weighted set-partitioning problem. Then, optimally solving Problem 2 is NP-hard.

Remark 5. *Both of our algorithms can also be used when a feasible solution is not obtained. In such scenarios, they lead to an assignment in which some of the packages will be delivered (although not necessarily the maximum number).*

V. SIMULATIONS

We test the performance of the proposed algorithms by solving the package delivery task assignment problem under various instances in comparison with a popular Greedy algorithm (GA), the heuristic distributed task allocation (HDTA) method for multi-vehicle multi-task problems in [27], and a centralized exact cutting-edge algorithm, which is executed through the commercial Integer Linear Program solver Gurobi named as Gurobi in the following. Gurobi is able to achieve the optimal assignment solution for Problem 2 if all the packages are released initially. In other words, Gurobi can get the optimal assignment at each task assignment instant, which can be used as a benchmark to test the performance of the designed algorithms. For the GA, each robot always first moves to the initial location of a package that can be transported to its destination by the robot with the shortest travel time compared with the other robots, and then moves to the initial location of the next best unassigned package. All experiments are performed on an Intel Xeon W-2123 CPU 3.60 GHz with 16 GB RAM, and the algorithms are compiled by Matlab under Windows 10 Enterprise.

A. The algorithms' performance for solving problems with small sizes

Monte Carlo simulations are first carried out to test the performance of the algorithms for guiding a fleet of robots to transport a group of $n = 100$ packages to their prescribed destinations within time-windows. The 100 packages are dynamically released with the frequency of one new package appearing every 5 seconds. To reduce the impact of the randomness, 10 instances of the initial locations of 20 robots and the 100 packages, and the packages' destinations are randomly generated in a square warehouse with edge length 100m. The tuple information on the newly appearing packages is collected for each 15 seconds in our experiments, after which the new packages, as well as packages that have been picked up, are assigned to the robots by a task assignment.

TABLE I. The robots' average total travel time (s) under 10 instances for m robots guided by different task assignment algorithms to transport $n = 100$ dynamically released packages.

m	Capacity C	Robots' total travel time (s)				
		GA	HDTA	ADA	MAA	Gurobi
10	3	3469	2572	1837	1961	**1725**
	4	3469	2532	1710	1904	**1610**
	5	3469	2471	1700	1774	**1582**
12	3	3539	2622	1794	1970	**1744**
	4	3539	2548	1753	1908	**1612**
	5	3539	2507	1691	1750	**1542**
14	3	3573	2625	1794	1975	**1730**
	4	3573	2596	1756	1907	**1599**
	5	3573	2580	1693	1786	**1549**
16	3	3628	2684	1794	1973	**1694**
	4	3628	2621	1732	1915	**1564**
	5	3628	2628	1676	1761	**1505**
18	3	3651	2671	1730	1973	**1689**
	4	3651	2656	1717	1889	**1544**
	5	3651	2667	1750	1760	**1495**
20	3	3684	2691	1769	1983	**1634**
	4	3684	2672	1687	1892	**1533**
	5	3684	2675	1663	1775	**1475**
Ratio to Gurobi		2.249	1.635	1.081	1.175	1.000

TABLE II. The average computation time (s) at each task assignment time under 10 instances for m robots guided by the task assignment algorithms to transport $n = 100$ dynamically released packages.

m	Capacity C	Average computation time (s)				
		GA	HDTA	ADA	MAA	Gurobi
10	3	**0.0066**	0.5828	0.0618	0.0865	0.0879
	4	**0.0059**	0.6185	0.2230	0.5396	0.5411
	5	**0.0063**	0.5612	0.8524	2.1624	3.1988
12	3	**0.0070**	0.4857	0.0508	0.0893	0.0885
	4	**0.0068**	0.5337	0.2516	0.5393	0.4817
	5	**0.0066**	0.5728	0.6149	2.1715	2.3748
14	3	**0.0076**	0.4552	0.0484	0.0848	0.0819
	4	**0.0078**	0.5131	0.2239	0.5223	0.4244
	5	**0.0078**	0.4764	0.5837	2.4652	2.3097
16	3	**0.0102**	0.4517	0.0433	0.0819	0.0744
	4	**0.0093**	0.5075	0.1743	0.5051	0.3694
	5	**0.0091**	0.4843	0.6151	2.1052	1.9407
18	3	**0.0103**	0.4272	0.0372	0.0804	0.0677
	4	**0.0100**	0.4725	0.1513	0.5059	0.3702
	5	**0.0098**	0.4672	0.4566	2.0931	1.6409
20	3	**0.0113**	0.4114	0.0372	0.0803	0.0643
	4	**0.0105**	0.4425	0.1204	0.5413	0.3220
	5	**0.0106**	0.4601	0.3010	2.1472	1.3783

For each instance, we investigate the algorithms' performance when different numbers of the 20 robots with different capacities are employed, where $m \in \{10, 12, 14, 16, 18, 20\}$ and $C \in \{3, 4, 5\}$. The time-window is set as $(l_i - r_i) = 8$ minutes empirically for transporting each package $i \in \mathcal{P}$, which enables an arbitrary available robot to transport each package from its initial location to its destination within the time-window wherever the robot is in the square operation environment. The robots' average total travel time resulting from the four algorithms is shown in Table I, where the smallest travel time under each scenario is marked in bold. The corresponding average computation time at each task assignment time instant is listed in Table II. For the centralized GA and CO, the average computation time is the algorithms' average computation time at each task assignment time instant. For the distributed HDTA, ADA and MAA, the average computation time is the average computation time of individual robots at each task assignment time instant.

First, for each $m \in \{10, 12, 14, 16, 18, 20\}$ robots, Table I shows that *a higher capacity C in general decreases the robots' total travel time for HDTA, ADA, MAA and Gurobi*. This might be caused by the fact that the robots with a higher capacity C can carry more packages at one time. However, Table II shows that *increasing C generally leads to a larger computation time for HDTA, ADA, MAA and Gurobi*. This is because more feasible routes/package groups exist when the robots have a higher C, which leads to a larger computation time. For GA, its performance is independent of the robots' capacities for each $m \in \{10, 12, 14, 16, 18, 20\}$. It might be due to the mechanism of GA under which each robot starts moving to the initial location of the next package only after transporting the currently carried package to its destination, which does not have a good use of the robots' capacity.

Second, *for each $C \in \{3, 4, 5\}$, a bigger m generally has a larger positive influence on optimizing Problem 2 for ADA, MAA and Gurobi* as shown in Table I. The reason is that more efficient routes can be chosen to transport each package if more robots are initially dispersed in the warehouse. However, the contrary happens with HDTA. This might be because HDTA initially assigns each robot with all the packages that the robot can serve.

Third, Table I shows that *ADA and MAA have a better performance compared with the GA and HDTA, where ADA has the best performance while GA has the worst performance relative to the baseline assignment results of Gurobi*. The reason lies partly in the fact that the GA assigns each robot maximally one package at each optimization run. However, Table II shows that ADA, MAA and Gurobi in general have a larger computation time compared with the GA and HDTA when the robots have capacity $C = 5$. This is because a higher capacity leads to more feasible solutions, and achieving a better assignment solution to the NP-hard problem requires a longer computational time.

Finally, *for the robots with a fixed capacity guided by HDTA, ADA and Gurobi, their average computation time decreases when increasing the robots' number m*. The reason might be that a large number of robots leads to a small number of packages to be assigned at each optimization run as more packages are already transported by the robots. This results in less computation time for each robot to calculate its feasible package groups.

The box plots of the robots' total travel times resulting from the algorithms under different scenarios are shown in Fig. 2. First, the box plots of GA and HDTA are comparatively higher than those of ADA, MAA and Gurobi with ADA generally having the lowest box plots among the compared heuristic algorithms. This shows the better performance of ADA and MAA as illustrated by Table I. Second, *for a given number of robots with a larger C, the box plots of HDTA, ADA, MAA and Gurobi are generally lower in comparison with the GA, suggesting that HDTA, ADA, MAA and Gurobi have a good use of the robots' capacity.* Third, with the increasing robots'

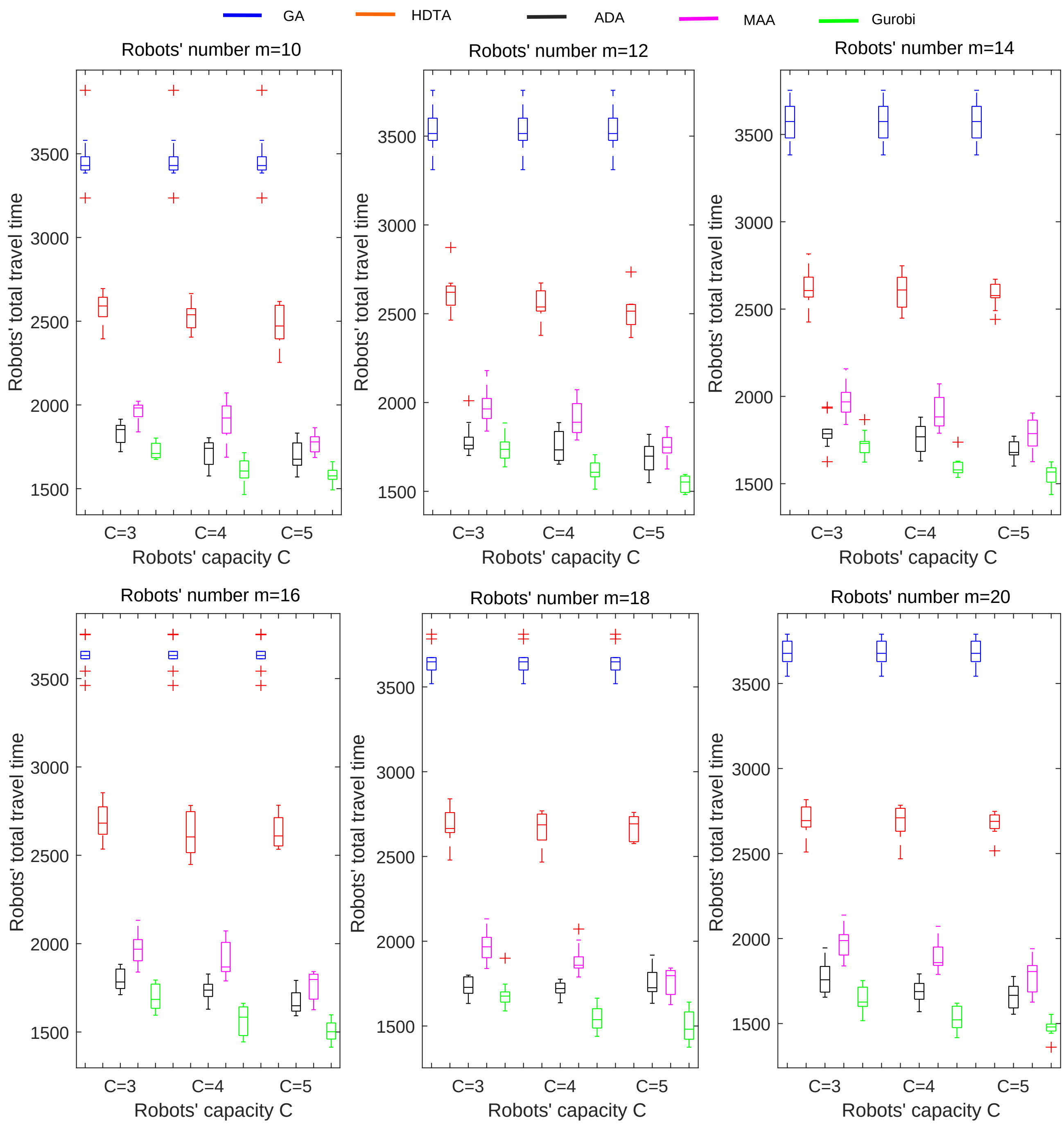

Fig. 2. Box plots of the robots' total travel times (s) under 10 instances to transport 100 dynamically released packages with different numbers of robots and different capacities, where the maximum time-delay for transporting each package is 8 minutes.

number m with each capacity, the box plots of ADA and MAA are generally lower and closer to Gurobi compared with the GA and HDTA, indicating that *ADA and MAA are more scalable than GA and HDTA*. For further comparison, we also carried out the two-tail Wilcoxon signed-rank test with the 5% significance level for each pair of the algorithms to guide different numbers of robots to transport the 100 packages. It is clear that *the robots' total travel times of the* 18 *scenarios differ significantly between the algorithms (the total travel time from small to large corresponds to Gurobi* $\rightarrow$ *ADA* $\rightarrow$ *MAA* $\rightarrow$ *HDTA* $\rightarrow$ *GA)*. This implies the algorithms have an increasingly better performance as GA – HDTA – MAA – ADA – Gurobi, which is consistent with those shown in Table I.

B. The algorithms' performance for solving problems with relatively large sizes

To test the scalability of the proposed algorithms, we increase the number of dynamically released packages to $n = 200$, and increase the frequency of dynamically released

TABLE III. The robots' average total travel time (s) under 10 instances for m robots guided by each algorithm to transport $n = 200$ dynamically released packages.

m	Capacity C	Robots' total travel time (s)				
		GA	HDTA	ADA	MAA	Gurobi
40	3	7466	4981	3234	3594	**3074**
50	3	7635	5158	3241	3621	**3049**
60	3	7781	5103	3238	3614	**2995**
70	3	7889	5164	3100	3592	**3002**
80	3	7970	5167	3257	3586	**2929**
Ratio to Gurobi		2.576	1.700	1.068	1.197	1.000

TABLE IV. The average computation time (s) at each task assignment time under 10 instances for m robots guided by each algorithm to transport 200 dynamically released packages.

m	Capacity C	Average computation time (s)				
		GA	HDTA	ADA	MAA	Gurobi
40	3	**0.0532**	2.7004	0.3571	0.6620	0.4980
50	3	**0.0658**	2.4470	0.3014	0.6547	0.4270
60	3	**0.0767**	2.1068	0.2584	0.6437	0.3873
70	3	**0.0884**	2.1287	0.2173	0.6247	0.3475
80	3	**0.1003**	1.9898	0.1889	0.6342	0.3080

packages to two newly appearing packages every 5 seconds. To reduce the impact of the randomness, 10 instances of the initial locations of 80 robots and 200 packages, and the packages' destinations are randomly generated in a square warehouse with edge length 100m. The tuple information on the newly appearing packages to be transported is collected for each 15 seconds, after which the new packages, as well as packages that have been picked up, are assigned to the robots by a task assignment. For each instance, the influence of the number of the 80 robots that can be used in the assignment is investigated, where $m \in \{40, 50, 60, 70, 80\}$. At this point, the robots' capacity is 3 and the time-window for transporting each package is set as 8 minutes.

The robots' average total travel time resulting from the algorithms is shown in Table III, and the corresponding average computation time of each task assignment time instant is listed in Table IV. First, Table III shows that *the robots' total travel time resulting from ADA, MAA and Gurobi generally decreases when increasing the robots' number m* while it is not the case for the GA and HDTA. This is consistent with those shown in Table I. For GA, the reason might be that each robot guided by the algorithm always first serves the package with the smallest travel time, which does not make good use of the robots' capacity. For HDTA, it might be because the algorithm initially assigns each robot with all the packages that the robot can serve. Second, Table IV shows that *the average computation times resulting from HDTA, ADA, MAA and Gurobi in general decrease when increasing m* while those resulting from GA increase, which shows the scalability of HDTA, ADA, MAA and Gurobi. The reasons maybe that a larger number of robots not only leads to less iterations for assigning the packages at individual task assignment time instants, but also leads to a small number of un-served packages to be assigned at each task assignment time instant as more packages are already transported by the robots.

The box plots of the robots' total travel time for transporting the 200 dynamically released packages under different scenarios are shown in Fig. 3. First, the box plots of GA and HDTA shown in Fig. 3 are comparatively higher than those of ADA, MAA and Gurobi with ADA generally having the lowest box plots among the compared heuristic algorithms. This shows the better performance of ADA and MAA as illustrated in Table III and Fig. 2. Second, with the increasing robots' number m, the box plots of HDTA, ADA and MAA are generally lower compared with GA, *indicating that HDTA, ADA and MAA are more scalable than GA*. To further evaluate the algorithms' performance for guiding different numbers of robots to transport the 200 packages, the Wilcoxon signed-rank test is carried out in a two-tail test with the 5% significance level for each pair of the algorithms. *The test clearly shows that the algorithms have an increasingly better performance as GA − HDTA − MAA − ADA − Gurobi, which is consistent with those shown in Table III. As a conclusion, ADA and MAA are competitive compared with the standard solver Gurobi, and perform better than GA and HDTA at the cost of a relatively larger computation time, where ADA has the better performance compared with MAA considering both the robots' total travel time and average computation time.*

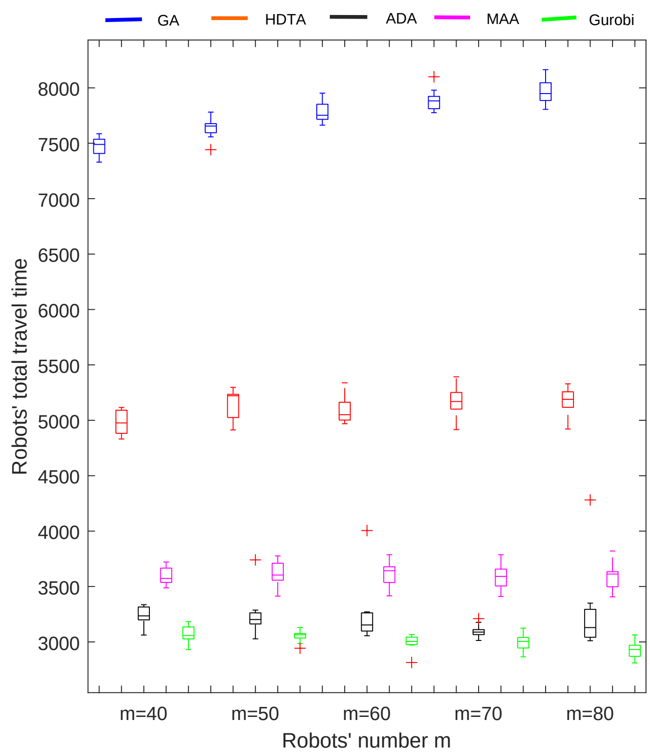

Fig. 3. Box plots of the robots' total travel times (s) under 10 instances to transport 200 dynamically released packages with different numbers of robots, where the maximum time-delay for transporting each package is 8 minutes and $C = 3$.

VI. CONCLUSION

In this paper, we have studied the multi-robot task assignment problem in which multiple dispersed robots need to efficiently transport a set of packages to their destinations within time-windows. The robot-group assignment strategy is first used to formulate the multi-robot task assignment problem as an optimization problem, which is shown to be an NP-hard problem. Two group-based distributed auction algorithms are designed which enable the robots to distributively calculate feasible package groups that they can serve, and then assign package groups to robots. The proposed task assignment algorithms are shown to outperform the greedy algorithm and the heuristic distributed task allocation method in simulation with a sufficiently large robot fleet. Future work may concern a multi-objective task assignment problem such as transporting the maximum number packages while minimizing the robots' total travel cost when some of the packages cannot be transported in time. Additional research directions are to consider the case where robots exchange packages already picked up, and the adaption of our formulation to utilize the techniques for submodular problems with matroid constraints.

References

[1] S. Moon, E. Oh, and D. H. Shim, "An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 303–313, 2013.

[2] G. Binetti, D. Naso, and B. Turchiano, "Decentralized task allocation for surveillance systems with critical tasks," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1653–1664, 2013.

[3] M. Turpin, K. Mohta, N. Michael, and V. Kumar, "Goal assignment and trajectory planning for large teams of interchangeable robots," *Autonomous Robots*, vol. 37, no. 4, pp. 401–415, 2014.

[4] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[5] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[6] T. S. Dahl, M. Matarić, and G. S. Sukhatme, "Multi-robot task allocation through vacancy chain scheduling," *Robotics and Autonomous Systems*, vol. 57, no. 6, pp. 674–687, 2009.

[7] S. L. Smith and F. Bullo, "Monotonic target assignment for robotic networks," *IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2042–2057, 2009.

[8] J. Yu, S.-J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 327–341, 2015.

[9] P. Toth and D. Vigo, *Vehicle routing: problems, methods, and applications*. Siam, 2014, vol. 18.

[10] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[11] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[12] R. Bent and P. Van Hentenryck, "A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows," *Computers & Operations Research*, vol. 33, no. 4, pp. 875–893, 2006.

[13] S. Ropke and J.-F. Cordeau, "Branch and cut and price for the pickup and delivery problem with time windows," *Transportation Science*, vol. 43, no. 3, pp. 267–286, 2009.

[14] A. Bettinelli, A. Ceselli, and G. Righini, "A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 5, pp. 723–740, 2011.

[15] M. Mahmoudi and X. Zhou, "Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations," *Transportation Research Part B: Methodological*, vol. 89, pp. 19–42, 2016.

[16] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.

[17] J. W. Escobar, R. Linfati, P. Toth, and M. G. Baldoquin, "A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem," *Journal of Heuristics*, vol. 20, no. 5, pp. 483–509, 2014.

[18] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.

[19] 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.

[20] T. Shima, S. J. Rasmussen, A. G. Sparks, and K. M. Passino, "Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research*, vol. 33, no. 11, pp. 3252–3269, 2006.

[21] E. Edison and T. Shima, "Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research*, vol. 38, no. 1, pp. 340–356, 2011.

[22] M. Otte, M. J. Kuhlman, and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," *Autonomous Robots*, vol. 44, no. 3, pp. 547–584, 2020.

[23] A. Simonetto, J. Monteil, and C. Gambella, "Real-time city-scale ridesharing via linear assignment problems," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 208–232, 2019.

[24] X. Han, S. Mandal, K. R. Pattipati, D. L. Kleinman, and M. Mishra, "An optimization-based distributed planning algorithm: a blackboard-based collaborative framework," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 673–686, 2014.

[25] S.-Y. Wang and L. Wang, "An estimation of distribution

algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 1, pp. 139–149, 2016.

[26] X. Bai, W. Yan, M. Cao, and D. Xue, "Distributed multi-vehicle task assignment in a time-invariant drift field with obstacles," *IET Control Theory & Applications*, vol. 13, no. 17, pp. 2886–2893, 2019.

[27] W. Zhao, Q. Meng, and P. W. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 902–915, 2016.

[28] S. Chopra and M. Egerstedt, "Spatio-temporal multi-robot routing," *Automatica*, vol. 60, pp. 173–181, 2015.

[29] H. Ma, J. Li, T. Kumar, and S. Koenig, "Lifelong multi-agent path finding for online pickup and delivery tasks," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 837–845.

[30] M. Liu, H. Ma, J. Li, and S. Koenig, "Task and path planning for multi-agent pickup and delivery," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1152–1160.

[31] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2004, pp. 698–705.

[32] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 98–104, 2007.

[33] W. Wang, Y. Jiang, and W. Wu, "Multiagent-based resource allocation for energy minimization in cloud computing systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 2, pp. 205–220, 2017.

[34] B. Zou, X. Xu, R. De Koster *et al.*, "Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system," *European Journal of Operational Research*, vol. 267, no. 2, pp. 733–753, 2018.

[35] R. K. Williams, A. Gasparri, and G. Ulivi, "Decentralized matroid optimization for topology constraints in multi-robot allocation problems," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 293–300.

[36] P. Toth and D. Vigo, *The vehicle routing problem*. Society for Industrial and Applied Mathematics, 2002.

[37] M. W. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations research*, vol. 4, no. 1, pp. 285–305, 1985.

[38] X. Bai, M. Cao, W. Yan, S. S. Ge, and X. Zhang, "Efficient heuristic algorithms for single-vehicle task planning with precedence constraints," *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 6274–6283, 2020.

[39] X. Bai, M. Cao, W. Yan, and S. S. Ge, "Efficient routing for precedence-constrained package delivery for heterogeneous vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 248–260, 2019.

[40] X. Bai, W. Yan, S. S. Ge, and M. Cao, "An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field," *Information Sciences*, vol. 453, pp. 227–238, 2018.

[41] F. A. Tillman and T. M. Cain, "An upperbound algorithm for the single and multiple terminal delivery problem," *Management Science*, vol. 18, no. 11, pp. 664–682, 1972.

[42] S. A. Cook and D. G. Mitchell, "Finding hard instances of the satisfiability problem: A survey," *Satisfiability Problem: Theory and Applications*, vol. 35, pp. 1–17, 1997.

[43] X. Bai, W. Yan, and S. S. Ge, "Efficient task assignment for multiple vehicles with partially unreachable target locations," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3730–3742, 2020.

[44] E. Balas and M. W. Padberg, "Set partitioning: A survey," *SIAM review*, vol. 18, no. 4, pp. 710–760, 1976.

[45] M. R. Garey and D. S. Johnson, *Computers and intractability A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

Xiaoshan Bai received his Ph.D. in Systems Engineering from the University of Groningen, the Netherlands, in 2018. In January to July 2015, he was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. In November 2018 to July 2019, he was a Lecturer with the Faculty of Science and Engineering, University of Groningen. In August 2019 to July 2020, he was a Postdoctoral Fellow with the Department of Cognitive Robotics, Delft University of Technology, the Netherlands. Since September 2020, he has been a Assistant Professor with the College of Mechatronics and Control Engineering, Shenzhen University, China. His main research interests include multi-vehicle/robot task assignment, path planning, logistic scheduling, and heuristic algorithms.

Andres Fielbaum received his Ph.D. in Systems Engineering at Universidad de Chile in 2019. Since November 2019, he is a Postdoctoral Fellow at the Department of Cognitive Robotics, Delft University of Technology, the Netherlands. Between 2015 and 2019 he was an adjunct lecturer at Universidad Federico Santa María, Chile. During 2019, he was a Research Fellow at Universidad de O'Higgins, Chile. His main research interests include public transport, on-demand mobility and logistics, and combinatorial optimization. Dr. Fielbaum received the first prize among young researchers worldwide to attend the World Conference on Transport Research 2019.

Maximilian Kronmueller is a Ph.D. student under the supervision of Dr. Javier Alonso-Mora and Dr. Robert Babuska. He is affiliated with the department of Cognitive Robotics at the University of Technology Delft. Previously, he obtained his Master of Science in Physics (2018) at the Technical University of Munich. His research interests lie in the following areas: online multi-task assignment, dynamic vehicle routing problems, learning-based methods to support routing decisions and dynamic optimization.

Luzia Knoedler is a Ph.D. student under the supervision of Dr. Javier Alonso-Mora and Dr. Robert Babuska. She is affiliated with the Department of Cognitive Robotics at the Delft University of Technology, The Netherlands. Before joining TU Delft, Luzia Knoedler received a master's degree in Control Engineering from the University of Stuttgart, Germany, and a bachelor's degree in Medical Engineering from the University of Stuttgart and the University of Tuebingen, Germany. Her main research interests include social and interactive navigation in human-centered environments and multi-robot systems.

Javier Alonso-Mora (Senior Member, IEEE) received the Ph.D. degree in robotics from ETH Zurich, Zürich, Switzerland, in 2014.,He is currently an Associate Professor with the Delft University of Technology, Delft, The Netherlands, where he leads the Autonomous Multi-Robots Laboratory, and a Principal Investigator with the Amsterdam Institute for Advanced Metropolitan Solutions, Amsterdam, The Netherlands. During the Ph.D., he worked in a partnership with Disney Research Zurich, ETH Zurich. He was a Postdoctoral Associate with the Massachusetts Institute of Technology, Cambridge, MA, USA. His main research interests include motion planning and control of autonomous mobile robots and multirobot systems interacting in dynamic environments and on-demand transportation.,Dr. Alonso-Mora was the recipient of multiple prizes and grants, including the ICRA Best Paper Award on Multi-Robot Systems in 2019 and a Talent Scheme Veni Award from the Netherlands Organisation for Scientific Research in 2017. His main research interests include mobile robots, path planning, collision avoidance, predictive control, and multi-robot systems.