

Received 12 March 2024, accepted 9 May 2024, date of publication 23 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404823

## RESEARCH ARTICLE

# Distributed Allocation and Scheduling of Tasks With Cross-Schedule Dependencies for Heterogeneous Multi-Robot Teams

BARBARA ARBANAS FERREIRA<sup>1</sup>, (Member, IEEE), TAMARA PETROVIĆ<sup>1</sup>, (Member, IEEE),  
MATKO ORSAG<sup>1</sup>, (Member, IEEE), J. RAMIRO MARTÍNEZ-DE DIOS<sup>2</sup>, (Member, IEEE),  
AND STJEPAN BOGDAN<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia

<sup>2</sup>GRVC Robotics Laboratory, University of Seville, 41092 Seville, Spain

Corresponding author: Barbara Arbanas Ferreira (barbara.arbanas@fer.hr)

This work was supported in part by European Union's Horizon Europe Research Program Widening Participation and Spreading Excellence through Project Strengthening Research and Innovation Excellence in Autonomous Aerial Systems (AeroSTREAM) under Grant 101071270, and in part by Croatian Science Foundation under SPECULARIA Project UIP-2017-05-4042.

**ABSTRACT** To enable safe and efficient use of multi-robot systems in everyday life, a robust and fast method for coordinating their actions must be developed. In this paper, we present a distributed task allocation and scheduling algorithm for missions where the tasks of different robots are tightly coupled with temporal and precedence constraints. The approach is based on representing the problem as a variant of the vehicle routing problem, and the solution is found using a distributed metaheuristic algorithm based on evolutionary computation (CBM-pop). Such an approach allows a fast and near-optimal allocation and can therefore be used for online applications. Simulation results show that the approach has better computational speed and scalability without loss of optimality compared to the state-of-the-art distributed methods. An application of the planning procedure to a practical use case of a greenhouse maintained by a multi-robot system is given.

**INDEX TERMS** Multi-robot systems, multi-robot coordination, task allocation, task scheduling, vehicle routing problem, distributed optimization.

## I. INTRODUCTION

Research on cooperative multi-robot systems (MRS) has received considerable attention in recent years due to its compelling advantages over single-robot applications. The main challenge is to create a robust and intelligent control system that enables seamless communication and task coordination between robots so that they can work efficiently as a team. Therefore, control architecture design, communication, and mission planning are the main problems discussed and solved in the literature. In this paper, we discuss the problems of task allocation (the question of who does what?) and task scheduling (the question of how to arrange the tasks in time?) of multi-robot systems, which are often summarized under the common term mission (task) planning.

The associate editor coordinating the review of this manuscript and approving it for publication was Emanuele Crisostomi<sup>1</sup>.

In this paper, we propose a distributed solution for task planning problem in heterogeneous multi-robot teams for problems within the class XD[ST-SR-TA] defined in the taxonomy in [1]. These task types require execution by a single robot (SR, single-robot tasks), and robots are allowed to execute only one task at a time (ST, single-task robots). The task allocation and scheduling procedure considers both current and future assignments (TA, time-extended assignment). In terms of complexity, these tasks involve cross-schedule dependencies (XD), where various constraints relate tasks from plans of different robots. The cross-schedule dependencies we consider are precedence constraints and transitional dependencies. The heterogeneity within the system arises from various factors, including the different hardware configurations and physical capabilities of each robot, which include sensor diversity, perceptual capabilities and different actuation mechanisms. The intended roles

and behaviors, such as specialization in certain tasks, also contribute significantly to this diversity.

In our approach, we reinterpret the task planning problem as a variant of the Vehicle Routing Problem (VRP). We combine the task planning problem with a variant of the well-studied Multi-Depot VRP model (MDVRP), in particular, the MDVRP with a heterogeneous fleet [2], hereafter referred to as HF-MDVRP. By doing so, we define a generic model of task planning problems that can be applied to different domains of multi-robot and multi-agent systems. Another advantage of the proposed modeling is that it exposes the task planning problems to a wide range of optimization techniques already available in the VRP literature, thus advancing the state of the art in task planning.

To solve the task planning problem modeled as HF-MDVRP, we propose a distributed metaheuristic algorithm based on the Coalition-Based Metaheuristic (CBM) paradigm [3] (CBM-pop<sup>1</sup>). We chose a distributed approach because it offers more reliability, flexibility, adaptability and robustness. This approach is particularly suitable for dynamic field applications with resource-constrained robots and can provide near-optimal solutions to difficult optimization problems in a reasonable amount of time. To empirically validate the proposed algorithm, we created a generic set of multi-robot task planning problems (a new benchmarking dataset, [4]) with precedence constraints and transitional dependencies and compared our solution with the following techniques:

- 1) Gurobi Optimizer [5]. Gurobi is a centralized solution that uses exact mathematical methods to solve Mixed-Integer Linear Programming (MILP) problems. Gurobi provides an optimality gap for each solution and can therefore serve as a measure of the optimality of the proposed solution.
- 2) State-of-the-art distributed auction-based approach with a single central agent acting as an auctioneer [6]. We slightly adapt the method to suit our problem class.

Using an extensive series of simulation runs of these solutions, we have shown that our method is state-of-the-art in terms of optimality. The advantage of our approach is **better computational speed and scalability**, which is essential for online operation of multi-robot systems. Another important advantage is that our solution produces results in a distributed manner. The auction algorithm, on the other hand, assumes a central auctioneer agent that processes bids from other agents and makes task assignments. In distributed systems, this may be regarded as a tactically vulnerable point. Although our proposed method is suitable for dynamic setups with on-the-fly mission rescheduling, dynamic applications are beyond the scope of this paper and are deferred for future work.

To illustrate the effectiveness of the method, we apply it to a case of multi-robot collaboration in a robotized greenhouse, as presented in the SPECULARIA project [7]. This contribution is part of a more comprehensive work

presented in the dissertation by Arbanas Ferreira [8]. A more detailed examination of the results presented here can be found in the full dissertation.

The paper is organized as follows. In the next section, we outline approaches to MDVRP problems and justify the solution we chose to model our problem. In Section III, we formally define the task planning problem and present our original contributions. Next, we define the unified model, which combines task planning and HF-MDVRP. In Section V, we present CBM-pop to solve the defined unified model. Section VI contains a detailed evaluation of the results and a discussion. In the same section, we show how the proposed approach can be applied to the SPECULARIA use case. Finally, we draw conclusions and give an outlook on future work in Section VII.

## II. RELATED WORK

To improve readability, this section begins with a description of the current state of mission planning. We then focus on solutions specifically targeted at VRP-based models.

### A. MISSION PLANNING

Some of the best-known distributed solutions to the mission planning problem are auction and market-based approaches [6], [9], [10]. These generally fall into the domain of task allocation (and omit scheduling of tasks), where robots use bidding mechanisms for a set of simple tasks to allocate tasks among themselves. However, the partial ordering between tasks and tight coupling, which are the basis for our cooperative missions, are rarely considered. These constraints concern a class of problems [1] where an agent's effective utility for a task depends on other tasks assigned to that agent (In-Schedule Dependencies, ID) or additionally on the schedules of other agents (Cross-Schedule Dependencies, XD). More recently, the authors in [6] and [11] have addressed the problem of precedence constraints in iterative auctions. These algorithms can be run offline or online, and they solve the same class of problems as those considered in this work. However, all these solutions still require a central auctioneer node that is in constant communication with all other nodes, which can be considered as a weak point of the system.

To alleviate this issue, solutions that use consensus algorithms for bid resolution have been suggested. A pioneering work in this area is the Consensus Based Bundle Algorithm (CBBA) [10]. Authors in [12] extended the basic algorithm to incorporate temporal constraints. However, the objective function remains rather simple, as it only accounts for a number of completed tasks without considering the usual metrics such as cost or duration of the mission.

On the other side of the spectrum, various optimization-based methods attempt to solve the task planning problem. They range from exact offline solutions [13] to heuristic approaches such as evolutionary computation and other AI optimization methods [14], [15]. In the former, a slightly broader class of problems is considered than in this

<sup>1</sup>[https://github.com/barbara0811/cbm\\_pop\\_mdvrp\\_optimization](https://github.com/barbara0811/cbm_pop_mdvrp_optimization)

work, where robots may perform more than one task simultaneously. The problem is modeled as an instance of a MILP problem and solved using offline solvers or well-known optimization methods. Although optimal, the method involves a high computational burden and lacks reactivity in dynamic environments.

In our previous work, we proposed the decentralized framework for multi-robot coordination [16], [17], which defines the design of particular decision-making modules and their architecture, as well as coordination mechanisms that ensure coordinated behavior. In these examples, fairly simple algorithms are used for task allocation and scheduling. In [16], tasks are assigned to robots using a greedy function that allots each task to the robot with the best score for that task and ignores the effects on other task assignments. In recent work [17], we extended the assignment function to a simple market-based task allocation scheme that aims to minimize the total mission duration.

### B. TASK PLANNING AS VRP

The problems considered here (XD[ST-SR-TA]) fit well into the MDVRP model, particularly the HF-MDVRP variant. MDVRP is a VRP variation that deals with problems where customers are served from multiple depots with a given fleet of vehicles. It is a classic example of an NP-hard [18] combinatorial optimization problem. Even for relatively small problem sizes, it is challenging to solve MDVRP optimally. Given the rapid combinatorial explosion, it quickly becomes impossible to obtain optimal solutions for this type of problem.

Similar modeling was proposed in [19], where task planning refers to the Dial-a-Ride Problem (DARP), a variant of VRP with pickup and delivery. To solve the problem, the authors use a centralized bounded optimal branch-and-price algorithm. Although the idea is similar, the set-partitioning formulation [19] is mainly suitable for solutions employing exact methods and cannot be directly used to solve non-trivial VRP instances due to the large number of possible routes. The method suffers from a high computational cost, scalability issues, and the inability to work online in distributed systems. Therefore, in this paper, we have defined a more manageable and general representation that can be easily used in various heuristic approaches.

Many heuristic methods have been proposed for MDVRP problems, and distributed approaches have been particularly suitable for multi-robot systems. More recently, Distributed Artificial Intelligence (DAI) has extended to multi-agent systems that deal with complex combinatorial problems. These multi-agent concepts can be readily applied to various metaheuristics, in particular population-based, hybrid and distributed metaheuristics. The distributed approach provides higher computational performance by executing tasks in parallel and improves robustness and efficiency through agent collaboration. Our algorithm is inspired by the Coalition-Based Metaheuristic [3], which applies established DAI principles to solve VRP problems.

Other approaches include distributed heuristic optimization algorithms such as Multi-Point Stochastic Insertion Cost Gradient Descent (MuPSICGD) [20], a distributed learning-based evolutionary algorithm [21], and the artificial bee colony algorithm [22]. In [23], the authors use a distributed game theoretic model to distribute products to customers without a central authority. Although each of these approaches would be suitable for solving our HF-MDVRP problem, we found that they are either comparatively slower or have worse optimality on the MDVRP benchmark dataset. The full comparison was not included in the paper for brevity, but can be found at [24].

### ORIGINAL CONTRIBUTIONS OF THE PAPER

The **original contributions** presented and discussed in this paper are:

- 1) *The unified task planning model* based on VRP paradigm. It combines problems of task planning with a variant of the well-studied Multi-Depot VRP (MDVRP) model, which generalizes the problem and makes it domain-agnostic. In particular, we consider MDVRP with heterogeneous fleet, HF-MDVRP.
- 2) *Distributed metaheuristic* algorithm for the defined problem based on the Coalition-Based Metaheuristic (CBM) paradigm (CBM-pop).
- 3) *An open benchmark dataset repository* of task planning problems of class XD[ST-SR-TA]. Our proposed benchmark dataset includes problems with  $2^n$ ,  $2 \leq n \leq 10$ ,  $n \in \mathbb{N}$  tasks and 2–8 robots, providing a large-scale test of the efficiency and scalability of task planning algorithms.

### III. TASK PLANNING PROBLEM

In this paper we consider two mathematical formulations. First, we formally define the task planning problem in this section, followed by the mathematical representation of the *unified model* that represents task planning as a VRP variant. The solution to the task planning problem proposed in this paper is based on the unified mathematical model.

We consider a problem where a team of heterogeneous robots  $R = \{1, \dots, m\}$  is available to perform a collection of simple single-agent tasks (actions)  $A = \{1, \dots, n\}$ . One or more robots can perform each  $a \in A$ , and we specify the set of actions that robot  $i$  can perform as  $A_i$ . Redundancy is possible, and in general  $A_i \cap A_j \neq \emptyset$ ,  $i \neq j$ , and  $i, j \in R$ .

In addition, we define precedence constraints on the set of actions. If the action  $a \in A$  must be completed before the action  $b \in A$  starts, we can specify a constraint between the two as  $prec(a, b)$ . This constraint forces  $a^f < b^s$ , where  $a^f$  and  $b^s$  indicate the times when the action  $a$  finishes and  $b$  starts.

In addition to strict precedence constraints, the task planning problem also involves transitional dependencies between actions. The problem involves the transition time between two actions (setup time), which includes all the operations required to go from the execution of one action

to the start of processing the next action. This term represents an additional temporal dependency on the order of actions, since direct predecessors and successors of actions strongly influence the total duration of the mission and the associated costs (setup costs).

Another naturally occurring constraint inherent in the physical system itself is the capacity constraint, which is expressed in the limited battery resources of each robot. Each action requires a certain amount of energy to be performed, which depends on the physical properties of the robot and the current state of the system (i.e., robot position, battery status, and current payload). For a robot  $i \in R$  with the capacity  $Q_i$ , we define the capacity constraint as  $\sum_{a \in S_i} q_i(a) \leq Q_i$ , where  $S_i (S_i \subseteq A_i)$  is the set of actions assigned to the robot  $i$  for execution, and  $q_i(a)$  is the energy requirement of the action  $a$  to be completed by robot  $i$ .

A solution to the described problem is a set of time-related actions (schedule) for all robots that do not violate the specified constraints. Formally, the schedule  $s_i$  for each robot  $i \in R$  is defined as  $s_i = \{(a, a^s, a^f) \forall a \in S_i\}$ , where  $S_i$  is the set of actions assigned to the robot  $i$ , and  $a^s(a^f)$  are the start (finish) times of the action  $a$ .

In the evaluation procedure, each action  $a \in A$  is assigned a couple  $(d_a(i), c_a(i))$ ,  $\forall i \in R$ , where  $d_a(i)$  stands for the duration of the action  $a$  when performed by robot  $i$  and  $c_a(i)$  for the cost of the action. The robot estimates the duration and cost of a future task based on the current state of the system. The planning procedure aims to find a solution that meets all constraints and maximizes the global reward of the system. The objective function can be chosen at will, depending on the system requirements.

All defined sets and variables are later summarized in Table 1, in direct comparison with HF-MDVRP and the unified model of task planning as VRP.

#### IV. UNIFIED TASK PLANNING MODEL FORMULATION

The problem modeling in this section is based on the VRP paradigm, which is included in Appendix A of this paper for the convenience of readers. In this section, we describe the elements of HF-MDVRP through the lens of task planning and equate all the building blocks of the VRP-based model with a concept of task planning. Then we define the full unified mathematical model in terms of a MILP.

##### A. TASK PLANNING AS HF-MDVRP

In relating the task planning problem to the HF-MDVRP model, we associate the fundamental VRP concepts directly with the task planning paradigm. As shown in Figure 1, the idea of a depot in VRP problems is directly related to the initial position of the robot, and the vehicle in VRP represents a robot itself. Next, the concept of customer and customer demand is applied to the actions in task planning and the energy demand of each action, respectively. Consequently, the routes (sequences of customer nodes to be visited) as solutions to VRP problems represent the sequence of actions in the final robot schedules in the task planning model. Based

on the metaphor thus specified, we describe the mathematical model of task planning problems defined as a VRP problem variant.

In the modeling of task planning as a VRP variant, we use the term customer for the simple single-robot tasks, actions. We designate the set  $N = \{1, \dots, n\} = A$ , where  $A$  is the set of actions, and  $N$  is the set of customers in the original VRP modeling. This definition includes only actionable elements from the robot's task structure, and all tasks must be decomposed down to the action level where vehicle routing is optimized.

In the task planning paradigm, we directly equate the concept of a vehicle with robots,  $K = \{1, \dots, m\} = R$ , where  $K$  is the set of vehicles, and  $R$  the corresponding robot set. Typically, the VRP paradigm requires vehicles to start at the depot, serve assigned customers along the route, and return to the depot. In our modeling, we equate the term depot with the initial location of the robot. Unlike the vehicles in typical VRP problems, the robot is not required to return to the starting point. Such a variant of VRP is referred to in the literature as open VRP [25].

The proposed model distinguishes between two different cost variants. On the one hand, the cost of transitioning between two tasks, setup cost, is directly related to the travel cost of HF-MDVRP. This cost can include any movement between locations and the possible setup cost between two tasks and is defined as  $c_{ijr}, i, j \in \{0\} \cup A, i \neq j, r \in R$ . Closely related to the setup cost is the setup time,  $t_{ijr}, i, j \in \{0\} \cup A, i \neq j, r \in R$ , which defines the duration of the robot's setup between two tasks. Next, the energy requirement of the specific action  $i \in A$  is defined as  $q_{ir}, r \in R$  and characterized as the customer demand in HF-MDVRP.

According to the specified modeling, the robot schedules are constructed based on the routes in the HF-MDVRP solution. The parallel is apparent since both constructs represent temporally ordered sequences of tasks. In the open VRP paradigm, a route is a sequence  $r^* = (i_0, i_1, \dots, i_s)$  with  $i_0 = 0$ , where 0 denotes a depot node. In our modeling, we take the previously defined concept of a schedule and map the order of tasks in a schedule to the arranged sequence of tasks  $s_r^* = (i_0, i_1, \dots, i_{|S_r|})$ ,  $i_j \in S_r, j \in \{1, \dots, |S_r|\}$ ,  $r \in R$ , where  $i_0$  is a zero-cost task associated with the robot's starting location, and  $S_r$  is a set of scheduled tasks. Temporal elements of the schedule are the task start time and the task duration, defined respectively as:  $\omega_{ir} := a_i^s, i \in S_r, r \in R$  and  $\sigma_{ir} := d_i = a_i^f - a_i^s, i \in S_r, r \in R$ .

A schedule for a robot  $r \in R$  is considered feasible if the capacity constraint  $q_r(S_r) := \sum_{i \in S_r} q_{ir} \leq Q_r$  holds, no task is scheduled more than once,  $i_j \neq i_k$  for all  $1 \leq j < k \leq |A|$ , and precedence and transitional constraints are respected. When it is necessary to perform repetitive tasks, each occurrence of the task should be treated as a separate entity.

All previously defined relations are summarized in the Table 1 and additionally illustrated in Figure 1.



TABLE 1. Task planning elements as HF-MDVRP.

Task Planning	HF-MDVRP	Unified Model Formulation
actions, $A$	customers, $N$	$A$ , set of actions
robots, $R$	vehicles, $K$	$R$ , set of robots
schedule, $s$	route, $r^*$	$s_r^* = (i_0(r), i_1, \dots, i_{ S_r }), i \in A, r \in R$
start of the action, $a_i^s$	start of the service at node	$\omega_{ir}, i \in A, r \in R$
action duration, $d_i$	service duration	$\sigma_{ir}, i \in A, r \in R$
end of the action, $a_i^f$	end of the service at node	$\omega_{ir} + \sigma_{ir}, i \in A, r \in R$
initial position of the robot	depot	$i_0(r), r \in R$ , initial zero-cost action
action setup cost	travel cost	$c_{ijr}, i, j \in \{0\} \cup A, r \in R$
setup time	travel time	$t_{ijr}, i, j \in \{0\} \cup A, r \in R$
energy requirement of the action	customer demand	$q_{ir}, i \in A, r \in R$
robot energy capacity	vehicle capacity	$Q_r, r \in R$

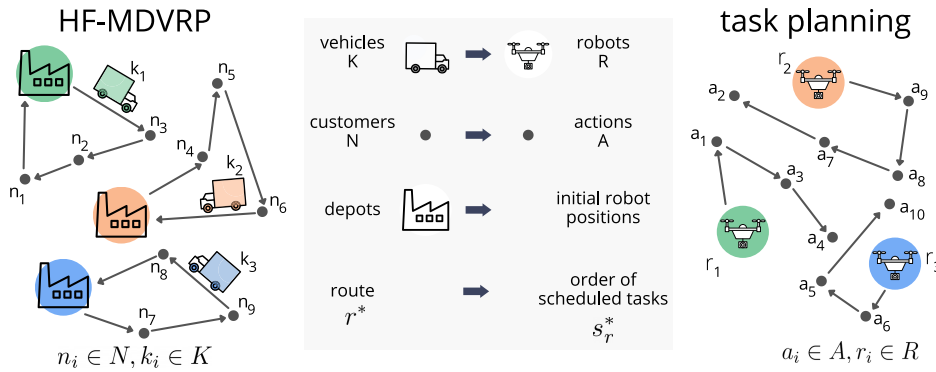


FIGURE 1. An illustration of the relationship of the HF-MDVRP model to the task planning paradigm. The diagram shows the direct relation between concepts in VRP and task planning problems, where depots represent initial robot positions, vehicles stand for robots, customers refer to actions, and vehicle routes represent final robot schedules.

**B. UNIFIED MODEL FORMULATION**

The mathematical representation of the unified task planning model as MILP is based on the graph structure. Let  $V$  be the set of vertices (or nodes) consisting of two distinct sets of nodes, the action nodes  $V_a = A = \{1, \dots, n\}$  and the start position nodes  $V_s = \{1, \dots, l\}$ , where  $n$  and  $l$  represent the number of available tasks and robot start positions, respectively. It holds that  $V = V_a \cup V_s$  and  $V_a \cap V_s = \emptyset$ . The underlying graph  $G = (V, E)$  is complete and directed with edge set  $E = \{e = (i, j) : i, j \in V, i \neq j, i \text{ and } j \text{ not both in } V_s\}$ .

We model heterogeneity motivated by a mixed fleet variant of MDVRP, in which the set of customers available to the vehicle corresponds in the task planning formulation to a subset of actions  $A_r$  that the robot  $r \in R$  can perform. To model unavailable tasks  $j \in A \setminus A_r$ , we set  $c_{ijr}$  to a sufficiently large number  $M$  for all  $(i, j) \in E$  related to the unreachable task  $j$ . Further aspects of heterogeneity of multi-robot systems are achieved by replacing the general coefficients of HF-MDVRP with robot-specific ones, e.g., the capacity  $Q$  by  $Q_r$ , the energy requirement  $q_i$  by  $q_{ir}$  for all  $i \in V_a$ , and the cost  $c_{ij}$  by  $c_{ijr}$  for all  $(i, j) \in E$ .

For the given precedence constraints  $prec(a_i, a_j), a_i, a_j \in A$  and assuming that tasks  $a_i(a_j)$  correspond to nodes  $i(j) \in V_a$ ,

we represent the precedence constraint in VRP notation by adding the pair of tasks to the set of constrained tasks  $\Pi = \{(i, j), i, j \in V_a\}$ . The constraint on the tasks is then

$$\omega_i + \sigma_i \leq \omega_j, (i, j) \in \Pi, \tag{1}$$

where  $\omega_i$  and  $\omega_j$  mark the beginning of the tasks  $i$  and  $j$  respectively, and  $\sigma_i$  denotes the duration of task  $i$ . We also distinguish particular  $\omega_{ir}$  as the start time of task  $i$  when it is executed by robot  $r \in R$ , and the corresponding task duration  $\sigma_{ir}$ . The expressions for these cases are defined in the full model representation.

TABLE 2. Defined sets in the unified model.

Set	Definition
$R$	set of robots
$V$	set of vertices (nodes)
$V_s$	set of robot initial locations
$V_a = A$	set of single-robot actions
$E = V \times V$	set of edges
$R_i$	set of robots starting from location $i, i \in V_s$
$\Pi$	set precedence constraints

For convenience, all sets, variables, and constants of the model are summarized in Tables 2 and 3.

**TABLE 3. Defined variables and constants in the unified model.**

Variable	Definition	Domain
$x_{ijr}$	if robot $r$ traverses edge $(i, j) \in E$	$\{0, 1\}$
$\omega_{ir}$	start time of action $i \in V_a$ performed by $r \in R$	$\mathbb{R}$
Constant		
$\sigma_{ir}$	duration of action $i \in V_a$ when performed by robot $r \in R$	
$t_{ijr}$	setup time between actions $i$ and $j$ , $\forall (i, j) \in E$ assigned to $r \in R$	
$c_{ijr}$	setup cost between actions $i$ and $j$ , $\forall (i, j) \in E$ assigned to $r \in R$	$\mathbb{R}$
$q_{ir}$	energy demand of action $i \in V_a$ for robot $r \in R$	
$Q_r$	energy capacity of robot $r \in R$	

**Definition 1 (Unified Task Planning Model Formulation):**

Based on the relations defined in this section, we define the unified task planning model built according to the HF-MDVRP MILP formulation. The binary decision variable  $x_{ijr}$  is defined to indicate whether the robot  $r \in R$  traverses an edge  $(i, j) \in E$  in a given solution. Then, the model is given as:

$$\min \begin{cases} \delta = \max(\sum_{r \in R} \sum_{i' \in V} x_{i'ir}(\omega_{ir} + \sigma_{ir})) \\ \quad - \min(\sum_{r \in R} \sum_{j' \in V} x_{j'jr} \omega_{jr}) \\ \gamma = \sum_{r \in R} \sum_{(i,j) \in E} x_{ijr}(c_{ijr} + q_{jr}) \end{cases} \quad (2.1)$$

$$\text{Subject to } \sum_{r \in R} \sum_{i \in V, i \neq j} x_{ijr} \leq 1, \quad \forall j \in V_a, \quad (2.2)$$

$$\sum_{j \in V_a} x_{ijr} \leq 1, \quad \forall i \in V_s, r \in R, \quad (2.3)$$

$$\sum_{(i,j) \in E} x_{ijr}(\omega_{ir} + \sigma_{ir} + t_{ijr} - \omega_{jr}) \leq 0, \quad \forall r \in R \quad (2.4)$$

$$\sum_{r \in R} \sum_{i' \in V} x_{i'ir}(\omega_{ir} + \sigma_{ir}) \leq \sum_{r \in R} \sum_{j' \in V} x_{j'jr} \omega_{jr}, \quad \forall (i, j) \in \Pi, \quad (2.5)$$

$$\sum_{i \in V} \sum_{j \in V \setminus \{0\}, j \neq i} q_{jr} x_{ijr} \leq Q_r, \quad \forall r \in R, \quad (2.6)$$

$$\sum_{r \in R} \sum_{i \in S} \sum_{j \in S, j \neq i} x_{ijr} \leq |S| - 1, \quad \forall S \subseteq V_a, \quad (2.7)$$

$$x_{ijr} \in \{0, 1\}, \quad \forall r \in R, (i, j) \in E, \quad (2.8)$$

$$\omega_{ir} \in \mathbb{R}, \quad \forall i \in V, r \in R. \quad (2.9)$$

The objectives of the optimization problem are represented by Equations (2.1). The goal is to find a solution that minimizes the makespan  $\delta$  (the difference between the latest action finish time in the whole mission and the earliest action start) and the cost  $\gamma$ . Actions  $i'$  and  $j'$  are possible direct predecessors of actions  $i$  and  $j$ , respectively, in the schedules of the same robot. The sum  $\sum_{i' \in V} x_{i'ir}$  is equal to 1 for exactly one action  $i'$  that precedes  $i$  in the schedule of robot  $r$ . Constraints (2.2) require that each task is executed at most

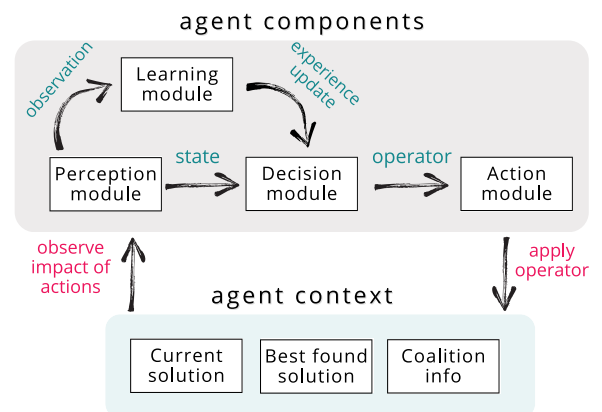
once. Equations (2.3) state that each robot may execute at most one schedule (only one edge starting at robot's initial position can be incorporated in the solution). Note that flow constraints represented in previously defined versions of VRP in Equations (7.4) are omitted here. This is because we do not restrict this model to provide closed-loop solutions, as we consider the open VRP formulation. Next, constraints (2.4) guarantee schedule feasibility with respect to considerations in the schedule of each robot. The  $t_{ijr}$  represents the setup time between actions  $i$  and  $j$  for robot  $r$ . The constraints defined by (2.5) enforce precedence constraints. Equations (2.6) ensure that the capacity constraints are met by the robots, while (2.7) eliminate all possible sub-tours in the solution. Finally, variable domains are provided in (2.8) and (2.9).

**V. SOLUTION APPROACH**

In this section, we present our algorithm inspired by the CBM paradigm to solve the problem described in Section IV. We provide a brief overview of the motivating algorithm that forms the basis for our approach. We then go into the details of our implementation, highlighting the novel elements.

**A. THE COALITION-BASED METAHEURISTIC (CBM)**

In CBM [3], multiple agents organized in a coalition simultaneously explore the solution space, cooperate, and self-adapt to solve the given problem collectively. The novelty introduced in this algorithm was the use of basic DAI principles, reinforcement, and mimetic learning, which not only allows agents to learn from their experiences and adapt their future behaviors accordingly, but also shares knowledge with other agents in the coalition. In addition to the learned behaviors, the agents also share the best solutions found, so that at the end of each iteration of the algorithm, the best global solution to the problem is obtained.



**FIGURE 2. CBM agent structure.**

The visual representation of the CBM agents is shown in Figure 2. During the search process, each agent maintains three solutions, similarly to particle swarm optimization [26]: a current solution, the best solution found by the agent, and the best solution found by the entire coalition. An agent uses

several operators that are applied to the current solution. The operators can be intensifiers or diversifiers. Intensifier operators concern improvement processes such as local search, and diversifier operators correspond to generation, mutation, or crossover procedures.

The choice of operators to apply is not completely stochastic as in Genetic Algorithms (GA). Instead, it is determined by a decision process that uses perceived state and past experience to select the most appropriate operators and coordinate intensification and diversification procedures. The selection of operators is based on heuristic rules. The search behavior of an agent is adapted during the optimization process through an individual reinforcement learning mechanism and mimetic learning. These mechanisms modify the rules of the decision process based on the experience results of previous explorations. Although all agents in the coalition use the same set of operators, the learning mechanisms may ultimately lead to different strategies.

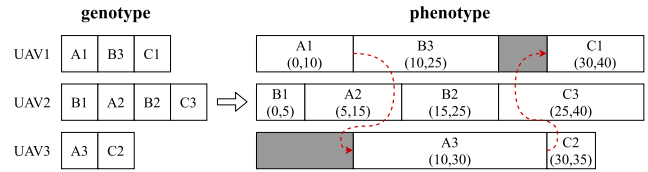
Agents cooperate in two ways. First, an agent can inform the rest of the coalition about the newly found best coalition solution. Second, agents share their internal decision rules to enable mimetic behavior. This fosters search behavior in which desirable solutions are often found.

In [3], the authors proposed the CBM solution for a case of VRP. In our case, we consider a HF-MDVRP and thus need to formulate a suitable set of operators. Moreover, we modified the basic CBM algorithm to keep more than one current solution so that a population of solutions is preserved, similar to GA methods. In the rest of the paper, we refer to the proposed algorithm as CBM-pop. Details on the implementation of the algorithm follow in this section.

**B. DISTRIBUTED METAHEURISTIC FOR HF-MDVRP**

The first feature to consider in the design of the algorithm is the representation of the solution. Since this metaheuristic is based on a set of operators commonly used in genetic algorithms, we use the same encoding of the solutions in terms of the chromosome. Inspired by an evolutionary process, each chromosome contains genetic material that defines a solution (genotype). In the case of HF-MDVRP, this refers to the assignment of actions to different robots and their order within a sequence of tasks in the schedule. An indirect coding, based on permutations of actions, similar to the one in [27], was used. Each chromosome is associated with a phenotype that evaluates the genetic material and, in our case, generates schedules for task sequences based on the temporal properties of the tasks.

An example of a chromosome and its genotype and phenotype is shown in Figure 3. On the left is shown the genetic material of a solution containing specified task groupings of robots (robot1, robot2, robot3) and ordering. The genotype representation is maintained respecting intra-schedule precedence constraints. On the right is an example of a phenotype generated from the specified genotype. The schedule is formed by introducing time



**FIGURE 3. Solution representation – chromosome genotype and phenotype. The tasks presented here are arbitrarily named generic tasks ((A, B, C) × (1, 2, 3)). The idle times introduced in the schedules are a consequence of precedence constraints  $prec(A1, A3)$  and  $prec(A3, C1)$ , since task A3 cannot start before task A1 finishes, and task C1 cannot start before the end of A3.**

elements into the ordered tasks (task durations, task setup times). If necessary, minimal idle times are inserted to ensure consistency with the defined precedence constraints. The phenotype represents the so-called semi-active schedule, where no left shift is possible in the Gantt graph. For any given sequence of robot operations, there is only one semi-active schedule [28]. One advantage of storing solutions in this way is faster exploration of the solution space, since all operators perform on a simpler genotype representation of the solution. The evaluation procedure renders the phenotype and evaluates the solutions found.

The next point to consider is the evaluation of the solution. The usual approach to solution evaluation is to use a fitness function that maps each chromosome in a population to a value of the utility function in  $\mathbb{R}$ . This usually works best when the search is limited to a single optimization objective. For multi-objective optimization problems, it is best to employ a ranking procedure because these objectives often interact in complex, nonlinear ways. In this work, we use a Pareto ranking procedure [29] that assigns ranks to all solutions based on the non-dominance property (i.e., a solution with a lower rank is clearly superior to solutions with a higher rank concerning all objectives). Therefore, solutions are stratified into multiple ranks based on their ability to meet the optimization objectives.

An important property of the Pareto front is that it allows us to determine the trade-offs of each decision, namely the reduction in the performance of the other objectives if one is improved along the frontier. Therefore, we can easily determine the priority of the different criteria for the solutions that are on the Pareto front. An alternative approach of using weighted sum function was discarded since the problem of determining weights is often non-trivial and may not produce good results, but also it is impossible to obtain points on non-convex parts of the Pareto-optimal set in the criterion space.

To evaluate solutions in a population  $P$ , we apply the double-rank strategy, which takes into account both the density information and the distribution of the solution in the rank. In the first step, an individual  $i \in P$  is assigned a dummy rank value  $R'(i)$  representing the number of solutions that dominate it in the current population  $P$ :

$$R'(i) = |\{j, j \in P, i \prec j\}|, \quad \forall i \in P, \quad (3)$$

where the symbol  $\prec$  corresponds to the Pareto dominance relation, i.e.,  $i \prec j$  if the solution  $j$  performs better than  $i$  given all optimization criteria. The final rank of solution  $R(i)$  is then defined as the sum of its own dummy rank value and that of its dominators:

$$R(i) = R'(i) + \sum_{j \in P, i \prec j} R'(j), \quad \forall i \in P. \quad (4)$$

The second part of the fitness function is the density function, which determines how similar the solution is to the other individuals in the population. Here we use a fairly simple solution where the density of an individual is inversely proportional to the distance to the nearest solution in the population and is calculated as follows:

$$\text{dens}(i) = \frac{1}{\min(d(i, j), \forall j \in P) + 2}, \quad \forall i \in P, \quad (5)$$

where  $d(i, j)$  represents the Euclidean distance between two individuals in the criteria space. Finally, the fitness of the solution is obtained as:

$$\text{fitness}(i) = \frac{1}{R(i) + \text{dens}(i) + 1}, \quad \forall i \in P. \quad (6)$$

In this setup, the rank of the solution has a much greater weight in the fitness function, since it is a natural number denoting a subset of solutions from the population, and the rank is a positive number smaller than  $1/2$ . The role of the density function is to discriminate between solutions of the same rank and to favour the more diverse solutions, as they are deemed more likely to explore new regions of the solution space.

Next, we briefly discuss the operators that form the core of the algorithm. We distinguish between generation, diversification (crossover and mutation operators), and intensification operators (local search algorithms). In our application, the generation operator is not used as a diversifier because it is applied during initial population creation. In the proposed solution, we use a single generation operator, a greedy insertion method that randomly takes an unassigned task and inserts it into existing routes at minimal cost, taking into account capacity constraints. Other operators are listed and described in Table 4. Diversification operators are first introduced in [30], and we implemented them for our specific problem. In the crossover procedure, we distinguish two cases of Best-Cost Route Crossover (BCRC), depending on the choice of parent chromosomes. One of the parents is always the current solution of the agent and the second parent is either the best solution found within the whole coalition or selected from the population. Similarly, we adapted the local search algorithms developed in [3] for a VRP problem class. In all these operators, the logic remains the same, but the additional ordering constraints arising from the precedence relation have been taken into account, which is a non-trivial task. For each chromosome, we maintain a directed graph with precedence constraints derived from the original constraints and the order of precedence-constrained nodes within the route. For each operator, we consider

the precedence graph and allow insertion of nodes only in places that do not break the consistency of this graph (cyclic dependencies and breaking of precedence dependencies are not allowed).

TABLE 4. Genetic operators used in our proposed solution.

Diversifiers [30]		
Best-Cost Route Crossover (BCRC)	Route	Crossover
intra depot reversal		Mutation
intra depot swapping		
inter depot swapping		
single action rerouting		
Intensifiers [3]		
two swap		
one move		

The behavior of a single CBM-pop agent is described in Algorithm 1. Before starting the algorithm, the agents exchange their specific problem parameters – task durations  $\sigma_{ir}$ , setup times  $t_{ijr}$ , setup costs  $c_{ijr}$ , energy demands  $q_{ir}$ , and energy capacities  $Q_r$ . During the runtime of the algorithm, the best solutions found and the weight matrices are exchanged among the agents, as noted in Algorithm 1. The procedure itself consists of Diversification-Intensification cycles (D-I cycles), where a diversification operator is first applied to the solution to guide the search out of the local optimum. After this perturbation, a series of local search procedures are applied to the solution to arrive at a new (local) optimum. The process is repeated until a termination criterion is reached. Further details on the definition of states, experience updates and learning mechanisms can originally be found in [3].

As mentioned, another novelty introduced here is that an agent stores a population of solutions instead of just one solution. The idea is to diversify the search further and allow for broader exploration. The role of the population is a dual one. First, after each  $n\_cycles$  cycles without improvement over the best solution found, a new starting solution is randomly selected from the population. Second, solutions from the population participate as a second parent in the crossover operator, thereby introducing novelty from the genetic pool.

## VI. EVALUATION AND DISCUSSION

The assessment involves large-scale testing on randomly generated task planning problems with precedence constraints and transitional dependencies [4]. Additionally, we apply the evaluation in a practical setting, specifically routing mobile



**Algorithm 1** Population-Based CBM Algorithm (CBM-Pop)

---

```

input :  $pop\_size$  – number of solutions in
          population
input :  $\eta$  – reinforcement learning factors
input :  $\rho$  – mimetism rate
input :  $n\_cycles$  – number of cycles before
          changing exploration origin
input :  $\epsilon$  – minimal solution improvement
variable:  $c_{best\ ag}$  – best solution found by the agent
output :  $c_{best\ coalition}$  – best found solution
/* initialization */
 $P \leftarrow generate\_population(pop\_size)$ 
evaluate\_population( $P$ )
 $c_{current} \leftarrow select\_solution(P)$ 
 $W \leftarrow init\_weight\_matrix()$ 
 $H \leftarrow init\_experience\_memory()$ 
while stopping criterion is not reached do
  /* calculate current state */
   $s \leftarrow perceive\_state(H)$ 
  if no change in best solution  $> \epsilon$  for  $n\_cycles$ 
    cycles then
      evaluate\_population( $P$ )
       $c_{current} \leftarrow select\_solution(P)$ 
    end
     $o \leftarrow choose\_operator(W, s)$ 
     $c_{new} \leftarrow apply\_op(o, c_{current}, P, [c_{best\ coalition}])$ 
    /* update experience history */
     $gain \leftarrow f(c_{current}) - f(c_{new})$ 
    update\_experience( $H, s, o, gain$ ) /* update
      solutions */
    if  $c_{best\ coalition}$  improved then
      broadcast\_solution( $c_{new}$ )
    end
    /* learning mechanisms */
    if end of D-I cycle then
      if  $c_{best\ coalition}$  improved in the cycle then
         $W \leftarrow individual\_learning(W, H, \eta)$ 
      else if  $c_{best\ ag}$  improved in the cycle then
         $W \leftarrow individual\_learning(W, H, \eta)$ 
        broadcast\_weight\_matrix( $W$ )
      end
    end
    if weight matrix received from a neighbor then
       $W \leftarrow mimetism\_learning(W, W_{received}, \rho)$ 
    end
  end

```

---

robots in a greenhouse scenario. Our focus lies on evaluating optimality, computational speed, and scalability.

The proposed solution is implemented in the Robot Operating System (ROS) [31] environment to enable vehicle-in-the-loop development of planning algorithms. This facilitates the rapid transition from the simulated environment to a real-world experiment. ROS also provides the underlying

communication infrastructure through its message and service protocols. All simulations were run on Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz x 8, 32 GB RAM running Ubuntu 18.04 LTS operating system.

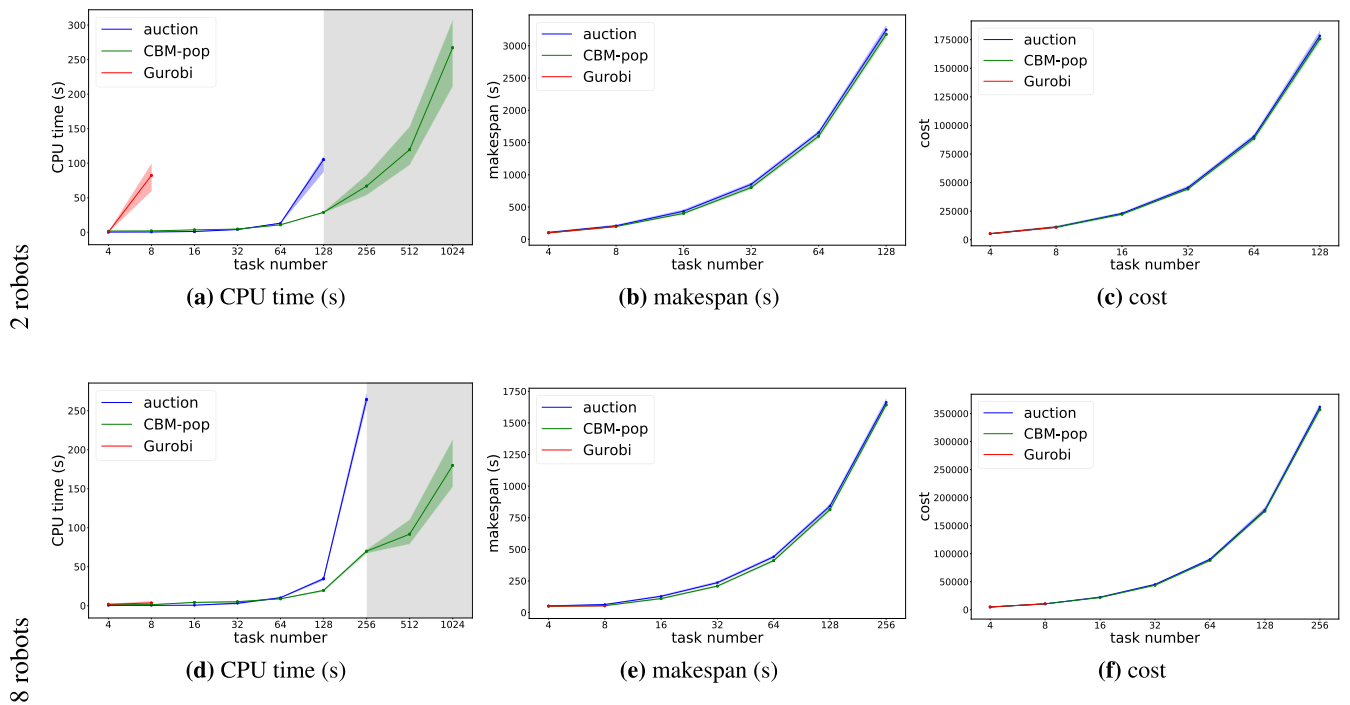
One of the important factors affecting the algorithm performance is the selection and tuning of the parameters. In this paper, we define the parameters empirically, following some common practices in evolutionary computing. First, the population size defines the diversity of the algorithm's solution pool. While a large diversity is good, larger populations lead to slower algorithm convergence. This parameter should be set considering the known properties of the problem to be solved. In our simulations, we set the  $pop\_size$  to 50. Next, the reinforcement learning factors  $\eta$  define the learning rate applied to the weight matrix after finding the best agent (coalition) solution. We set this parameter to [0, 5, 1], which determines a 50% larger weight increase for the case where the best coalition solution has improved compared to the only locally best solution. The mimetism rate  $\rho$ , which determines the rate of an agent's mimetic behavior, is set to 0.3, where an agent mimics another agent's weight matrix by 30% and retains 70% of its own learned behavior. Finally, the termination parameters  $n\_cycles$  and  $\epsilon$  are set specifically for each of the application scenarios. A typical value for  $n\_cycles$  would be no less than 10000 to obtain high-quality solutions, while  $\epsilon$  represents the smallest allowable improvement in the solution and we used the value of 0.01 to run the algorithm as long as there is progress.

In Appendix B, we conduct a comparative analysis between the proposed population-based CBM and the original single-solution version of the CBM algorithm.

#### A. COMPARATIVE ANALYSIS ON TASKS WITH CROSS-SCHEDULE PRECEDENCE CONSTRAINTS

Despite the considerable complexity of benchmark problems available in the VRP literature, they do not include some important elements of the problem that our approach solves, namely cross-schedule dependencies. Therefore, we have developed a separate set of benchmark examples to evaluate problems with precedence constraints. The problems are solved for a team of two and eight robots. The problem instances include 4, 8, 16, 32, 64, 128, 256, 512, and 1024 tasks, and we generated 50 randomized examples for each setting. In each example, 20% of the tasks are precedence-constrained. Task durations and costs are generated using assumed robot characteristics (speed, energy requirements). Setup times and costs are calculated using the above-mentioned agent properties and the Euclidean distance between tasks associated with random positions in 3D space. The full benchmark set and results can be found at [4].

In analyzing the results, we compare the proposed distributed metaheuristic CBM-pop with an exact MILP-based solution provided by the Gurobi solver [5] and a state-of-the-art distributed auction-based algorithm presented in [6]. We opted for the Gurobi optimizer because of its precise



**FIGURE 4.** Comparison of the performance of the proposed distributed metaheuristic algorithm CBM-pop (green), the Gurobi optimal solver (red), and the state-of-the-art auction-based distributed algorithm [6] (blue). The top row shows the results for 2 robots, and the performance for 8 robots is shown in the bottom row. The plot displays a graph for the mean of each of the observed values and the distribution highlighted by a transparent ray around the graph line. A time limit of 20min was introduced in the simulations, and only the results obtained in this runtime are presented. CPU time represents the total computational effort (time needed) to compute the solution, while makespan and cost refer to the duration and cost of the computed schedule, respectively. Figures (a) and (d) clearly show that our algorithm scales better than current methods, as it can solve more complex problems in the given time.

mathematical approach. However, it is limited to simpler problems in our dataset as it struggles with higher dimensionality (more tasks). From a small pool of distributed approaches, we chose the auction-based method [6] because it directly applies to our problem, covers all relevant constraints, and can be easily implemented based on the provided paper. The performance results of the algorithms for the defined benchmark set are shown in Figure 4.

Two rows of the figure represent the performance of the algorithms for teams of 2 and 8 robots, respectively. In the simulations, we introduced a computation time limit of 20 minutes. The optimal solver is able to obtain solutions for up to 8 tasks, the auction-based algorithm, for up to 128 tasks for the case of 2 robots, and 256 tasks for 8 robots, while the proposed algorithm can handle all examples in the benchmark. For the case of 2 robots and 256 tasks, and 8 robots and 512 tasks, the auction algorithm takes about 25 minutes to produce solutions.

The first property to be observed is the algorithm runtime and scalability of the above approaches. From the first column of the grid in Figure 4, it is clear how fast the combinatorial explosion manifests in the auction-based algorithm. It is even clearer for the optimal solver. The Gurobi solver succeeds on problems with up to 8 tasks for both sets of benchmarks. The auction method is able to solve problems

with at most 256 tasks in the given time. An exponential increase in computation time can be observed. On the other hand, CBM-pop copes very well with an increase in the number of robots and tasks. Also, a larger scatter in the CPU time in CBM-pop is observed for larger task examples. This is due to the stochastic nature of the protocol and the quality-based stopping criterion, which terminates the computation if no improvement in the solution has been achieved for a certain number of steps.

In Table 5, we can observe the average improvement of the CPU time of the proposed algorithm compared to the auction-based method. We provide information for more extensive problems where the qualities of the proposed method are highlighted. For simpler examples, the auction-based algorithm renders solutions faster (about 2-3 times). CBM-pop computes solutions in about 1.5s for the simplest cases (5s for more complex problems), and the auction manages to solve the problem in approximately 0.5s and 2s, respectively. This is the consequence of the constructive nature of auction-based algorithm, which generates only one solution, instead of performing a search of the solution space. For smaller examples, naturally, this renders less runtime.

We also investigate the performance of the algorithms in terms of optimality. As explained above, we model the optimization function in terms of Pareto optimality

**TABLE 5.** Summary of average improvements of the proposed algorithm (CBM-pop) compared to the state-of-the-art auction-based algorithm [6] on a basis of 50 randomized examples for each problem setting. For simpler examples, the auction-based algorithm obtains solutions faster. CBM-pop finds solutions in 1.5s for the simplest cases (5s for more complex problems), and the auction solves the problem in 0.5s and 2s, respectively.

	task number	average improvement (%)		
		makespan	cost	CPU time
2 robots	4	4.36	0	-
	8	5.25	0.51	-
	16	7.45	3.12	-
	32	5.03	2.36	-
	64	3.39	1.86	15.89
	128	2.39	1.57	71.60
8 robots	4	4.57	0	-
	8	14.39	0	-
	16	15.04	2.57	-
	32	11.48	2.70	-
	64	7.00	1.93	12.76
	128	3.43	1.27	42.35
	256	1.32	1.1	73.11

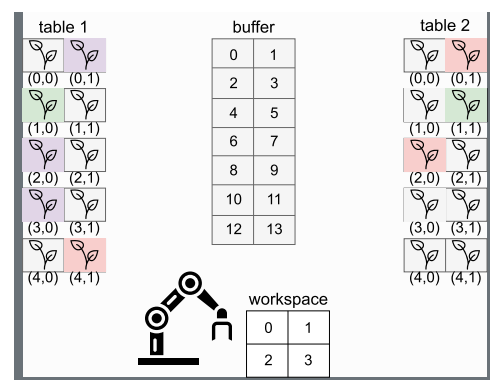
(as defined in Eq. 6) for two criteria, namely the makespan of the schedule and the total cost (Eq. 2.1). For the limited number of examples with computed optimal solutions, both the auction and our method follow the optimal solutions very closely (with a deviation of up to 0.5% from the optimum). In Figure 4, the last two columns represent the duration and the cost of the solution found, but the results are quite similar, so no significant difference can be seen in the graph. Therefore, we refer the reader to Table 5, which summarizes the performance of our algorithm. In all examples presented here, the CBM-pop algorithm outperforms the auction for both given criteria. For the case of 2 robots, the improvements range in makespan from 2% to 7% and in cost up to 3.12%. For 8 robots, the improvements range in makespan from 2-16% and in cost up to 2.6%.

By running these simulations, we have shown that our approach can keep up with the current state of the art in task planning in terms of optimality, while generating solutions in significantly less time, which is essential for all real-world applications.

### B. APPLICATION TO THE USE-CASE IN AGRICULTURAL ENVIRONMENT

The SPECULARIA project [7] aims to advance robotics in agriculture beyond conventional approaches, which usually focus on large machines for specific crops. Unlike traditional methods where a mobile robot with a manipulator is programmed to adapt to the farm environment [32], SPECULARIA seeks to embrace innovation in this rapidly growing industry. In SPECULARIA, the farm is built around a stationary robotic manipulator. This structures the manipulator's workspace and allows the robots to perform sophisticated manipulation tasks such as pruning or pollination. In turn, they rely on mobile robots to work as in a warehouse, moving plants grown in containers to ensure that each plant receives optimal care under ideal growing conditions. Ideally, the multi-robot system can plan and execute sequences to control the growth and hygiene of each plant from seed to harvest.

The mission given to our robotic team is to perform daily maintenance tasks in a robotized greenhouse. In this paper, we consider a team that consists of two types of robots – three unmanned ground vehicles (UGVs) equipped with a mechanism to transport plant containers to and from the workspace and a single robotic manipulator that performs actions on plants. Each UGV can pick up and transport one plant at a time and place them on specific tray holders in the greenhouse. We assume that missions to the system are issued in time frames greater than the time required to complete a single mission, i.e., the team has enough time to complete one mission before the next one arrives.



**FIGURE 5.** SPECULARIA use case greenhouse layout.

The layout of a greenhouse in a mission we analyze in this paper is shown in Figure 5. The greenhouse consists of two tables with plants placed along the walls of the greenhouse. Each table is organized into five rows and two columns. The tables are enumerated, as are the specific positions within the table. The convention for addressing the tray holders within the table is (row, column) and the indices begin with 0. The full address of each plant is defined by the triplet (table, row, column). Plants within the table can only be accessed by row, starting with the positions along the aisle. Thus, plants located by the wall of the structure can only be accessed by removing previous plants in the same row of the table. For example,

in Figure 5, the plant at position (1, 0) in table 1 can only be accessed after removing the plant at position (1, 1) from the table. A similar precedence relationship applies to table 2, except that plants are accessed from left to right.

In addition to the two tables, there is a buffer table structure in the middle of the greenhouse. The structure of a buffer is very similar to the structure of the tables, but the plants can be reached from either side, so there is no precedence relationship between the plant access tasks. The buffer is used to store plants that need to be put aside before the required plants are transported to the processing station (work station of the robot manipulator). Plants that are finished with the maintenance task are also put back into the buffer.

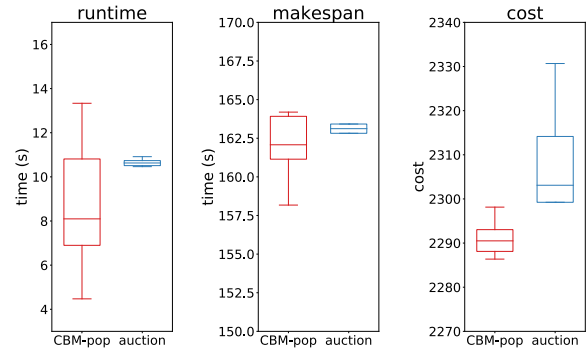
Finally, at the bottom of the greenhouse structure is a workspace table with four plant tray holders. The idea of the four positions is to allow for batch processing of the plants, which is especially advantageous for simpler tasks such as watering or spraying the plants.

Inputs to the planning procedure include the greenhouse layout, the groups of plants to be tended that day, and the procedures to be performed. In the given example from Figure 5, the specific groups are  $A = \{(1, 0, 1), (1, 2, 0), (1, 3, 0)\}$ , marked in purple in the figure,  $B = \{(1, 1, 0), (2, 1, 1)\}$  marked in green, and  $C = \{(1, 4, 1), (2, 0, 1), (2, 2, 0)\}$  marked in red. This means that to execute operation A, all three defined plants must be present in the workspace table. The same is true for the other two tasks.

In problem modeling, we distinguish between two actions that the UGV can perform on plants, namely transporting plants to the buffer and moving plants to the workspace. Based on these two actions and the defined inputs, we generate a set of actions to be planned for. For plants that are not scheduled for care on a given day and that interfere with the plants to be processed, the action of moving them to the buffer is generated. For example, we can identify the task to\_buffer(1,1,1). For plants that are scheduled for care, we define two precedence-constrained actions, moving them to the workspace and placing them in the buffer when processing is complete. For example, we define the tasks to\_workspace(1,1,0) and to\_buffer(1,1,0), with a precedence constraint in between. Additionally, there is a precedence for the tasks of accessing two adjacent plants, in this example the tasks to\_buffer(1,1,1) and to\_workspace(1,1,0). To estimate the duration and cost of each action, a path is calculated on a static map of the greenhouse and the length of the path is used to calculate the duration based on the speed of each vehicle.

On the side of the manipulator that tends the plants, we define three different actions. This was necessary to ensure the desired system behavior while keeping the problem within the scope of the defined modeling. For the maintenance task A example, the defined tasks are A\_ready, A\_perform, and A\_setup, all of which take precedence in the defined order. Task A\_ready signals that the workspace is empty and ready to receive the next batch of plants. This task precedes all tasks to\_workspace for the given procedure. After all plants are placed on the workspace, the task to

perform the procedure (A\_perform) begins. This relationship is also modeled by precedence constraints. Next, after the procedure is completed, the tasks of transporting the plants from the workspace to the buffer are activated. After all plants are removed from the workspace, the task A\_setup is executed. This task represents the tool change of a robot arm. When it is finished, new plants can be brought into the work area and the whole process starts again.



**FIGURE 6.** Comparison of the performance of the proposed CBM-pop (red) with the auction-based algorithm (blue). We observe the algorithm CPU time, solution makespan, and total cost for 10 runs of each algorithm.

The proposed system was implemented using the ROS Melodic distribution and the Gazebo simulator for environment modeling. Several simulations were conducted for the specified use case. As for the benchmark problem set, we compared the performance of the proposed algorithm with the auction-based state-of-the-art solution for task planning problems [6]. We ran 10 simulations for each algorithm, and the results are consonant with the conclusions drawn earlier. In Figure 6 we can observe CPU time, makespan, and cost distribution for the obtained solutions. Due to the stochastic nature of the CBM-pop algorithm and a quality-based stopping criterion, we can observe a large scatter in the runtime for each simulation run. There is also some standard deviation in the objective graph for each criterion, which is a feature of the Pareto-based objective function (for equally good solutions there can be a large dispersion in criteria values) and the properties of both algorithms (auction's constructive approach, while CBM-pop performs a stochastic search). For this example, the average reduction in CPU time of the proposed algorithm compared to the auction protocol is 13.23%. Regarding makespan and cost, a slight average improvement of 0.59% and 0.76%, respectively, is observed.

For this relatively small problem, the qualities of the obtained solutions are very similar for both algorithms. However, we have previously shown that our algorithm adapts better to the problem complexity. Another important advantage is that our solution produces results in a distributed manner. The auction algorithm, on the other hand, assumes a central auctioneer agent that processes bids from other agents and makes task assignments. In distributed systems, this may be regarded as a tactically vulnerable point.



**Glossary of acronyms.**

Acronym	Description
MRS	multi-robot systems
VRP	Vehicle Routing Problem
MDVRP	Multi-Depot VRP
HF-MDVRP	MDVRP with a heterogeneous fleet
CBM	Coalition-Based Metaheuristic
CBM-pop	population-based CBM
MILP	Mixed- Integer Linear Programming
DAI	Distributed Artificial Intelligence

The final generated schedule for the SPECULARIA use case is displayed in Appendix C for convenience. Tasks are color-coded according to the convention defined in Figure 5. Tasks of moving plants that do not require maintenance into the buffer are marked in gray. Precedence relations are indicated by arrows. Setup times are not displayed in this illustration to preserve a more compact representation of the schedule. Based on the generated schedule, we can conclude that the proposed method successfully handles the problem of planning tasks for this type of operations. The animated visualization of the execution of the obtained schedule is available at [24].

**VII. CONCLUSION AND FUTURE WORK**

In summary, in this paper we have developed a robust and fast task planning method for heterogeneous multi-robot systems. The planning method addresses problems with cross-schedule dependencies, in particular precedence constraints. We synthesized a general model that relates task planning (allocation and scheduling) to the well-studied VRP. This exposes task planning problems to various optimization techniques available in vehicle routing, which could lead to many compelling solutions for task planning in the future.

In this work, we have found a solution to the problem in a distributed manner by applying a metaheuristic approach based on evolutionary computation with knowledge sharing and mimetism. We have extensively tested the performance of the proposed algorithm. We established a benchmark dataset repository for planning for tasks of class XD[ST-SR-TA] and tested the proposed algorithm against existing task planning methods. Simulation results show that the approach has better computational speed and scalability without loss of optimality compared to state-of-the-art distributed methods. We have also provided a novel application of the planning procedure to a real-world use case of a greenhouse maintained by a multi-robot system.

As future work, we are interested in testing the proposed approach in a more dynamic setting and introducing protocols for handling disturbances in the system, including asynchronous and stochastic arrival of new tasks. Given the distributed nature of the proposed algorithm, we plan to investigate the robustness with respect to delays or information loss in the communication channel. Another interesting point to explore is the long-term operation of the system, where robot recharging would need to be considered and included in the problem model. The application of the

entire system in a real-life scenario is another promising topic for future development.

**APPENDIX A  
VEHICLE ROUTING PROBLEM**

VRP is widely used in transportation, distribution, and logistics because of its many practical applications. In essence, VRP is a problem in which vehicles with limited payloads must pick up or deliver items at various locations. The items have a certain quantity, such as weight or volume, and the vehicles have a maximum capacity that they can carry. The problem is to pick up or deliver the items at the lowest cost without exceeding the vehicle capacity.

The basic VRP [33] regards a set of nodes  $N = \{1, \dots, n\}$  representing  $n$  customers at different locations and a central depot (warehouse), which is usually denoted by 0. Customers are served from one depot by a homogeneous and limited fleet of vehicles. A vehicle serving a customer subset  $S \subseteq N$  starts at the depot, travels once to each customer in  $S$ , and finally returns to the depot. Each pair of locations  $(i, j)$ , where  $i, j \in N \cup \{0\}$ , and  $i \neq j$ , is associated with a travel cost  $c_{ij}$  that is symmetric,  $c_{ij} = c_{ji}$ .

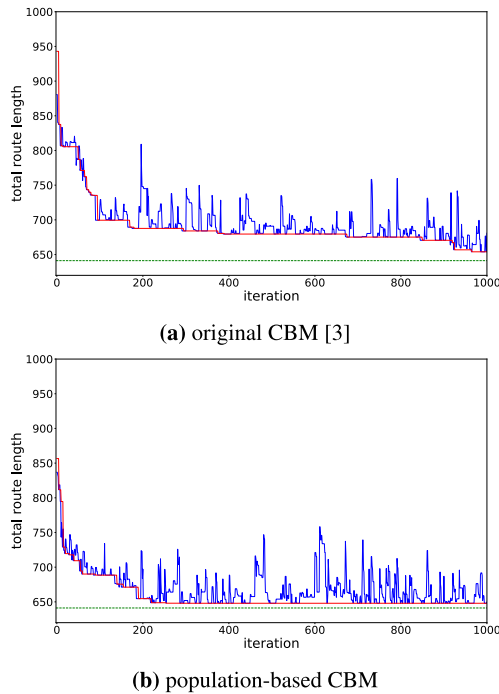
In VRP, each customer is assigned a demand  $q_i$ ,  $i \in N$  that corresponds to the quantity (e.g., weight or volume) of goods to be delivered from the depot to the customer. There is a set of vehicles,  $K = \{1, \dots, m\}$ , with capacity  $Q > 0$ , operating at identical cost. In the case of a heterogeneous fleet, the capacity  $Q$  is specifically defined for each vehicle (or type of vehicle).

A route is a sequence  $r^* = (i_0, i_1, \dots, i_s, i_{s+1})$  with  $i_0 = i_{s+1} = 0$ , and  $S = \{i_1, \dots, i_s\} \subseteq N$  is the set of visited customers. The route  $r^*$  has cost  $c(r^*) = \sum_{p=0}^s c_{i_p i_{p+1}}$ . A route is considered feasible if the capacity constraint  $q(S) := \sum_{i \in S} q_i \leq Q$  holds and no customer is visited more than once,  $i_j \neq i_k$  for all  $1 \leq j < k \leq s$ . In this case, the set  $S \subseteq N$  is considered a feasible cluster.

A solution of a VRP consists of  $m = |K|$  feasible routes, one for each vehicle  $k \in K$ .  $|K|$  represents the cardinality of the set  $K$ . Therefore, the routes  $r_1^*, r_2^*, \dots, r_m^*$  corresponding to the specific clusters  $S_1, S_2, \dots, S_m$  represent a feasible solution of VRP if all routes are feasible and the clusters form a partition of  $N$ .

The given model can be represented by an undirected or directed graph. Let  $V = \{0\} \cup N$  be the set of vertices (or nodes). In the symmetric case, i.e., if the cost of moving between  $i$  and  $j$  does not depend on the direction, the underlying graph  $G = (V, E)$  is complete and undirected with edge set  $E = \{e = (i, j) = (j, i) : i, j \in V, i \neq j\}$  and edge cost  $c_{ij}$  for  $(i, j) \in E$ . Otherwise, if at least one pair of vertices  $i, j \in V$  has asymmetric cost  $c_{ij} \neq c_{ji}$ , then the underlying graph is a complete digraph. We are concerned with the former.

*Definition 2 (VRP Model Formulation):* One of the most common mathematical representations of the VRP model is the MILP formulation [33]. The binary decision variable  $x_{ijk}$  is defined to indicate whether the vehicle  $k$ ,  $k \in K$  traverses



**FIGURE 7.** Illustration of typical trajectories of the current (blue) and the best found solution (red) for the two variants of the CBM algorithm. We analyze the convergence speed of single-solution CBM and the proposed CBM-pop. The best known solution is shown in the graph with a green line.

an edge  $(i, j) \in E$  in a given solution. Therefore, the integer linear programming model for VRP can be considered as written:

$$(VRP) \min \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x_{ijk} \quad (7.1)$$

$$\text{Subject to } \sum_{k \in K} \sum_{i \in V, i \neq j} x_{ijk} = 1, \quad \forall j \in V \setminus \{0\}, \quad (7.2)$$

$$\sum_{j \in V \setminus \{0\}} x_{0jk} = 1, \quad \forall k \in K, \quad (7.3)$$

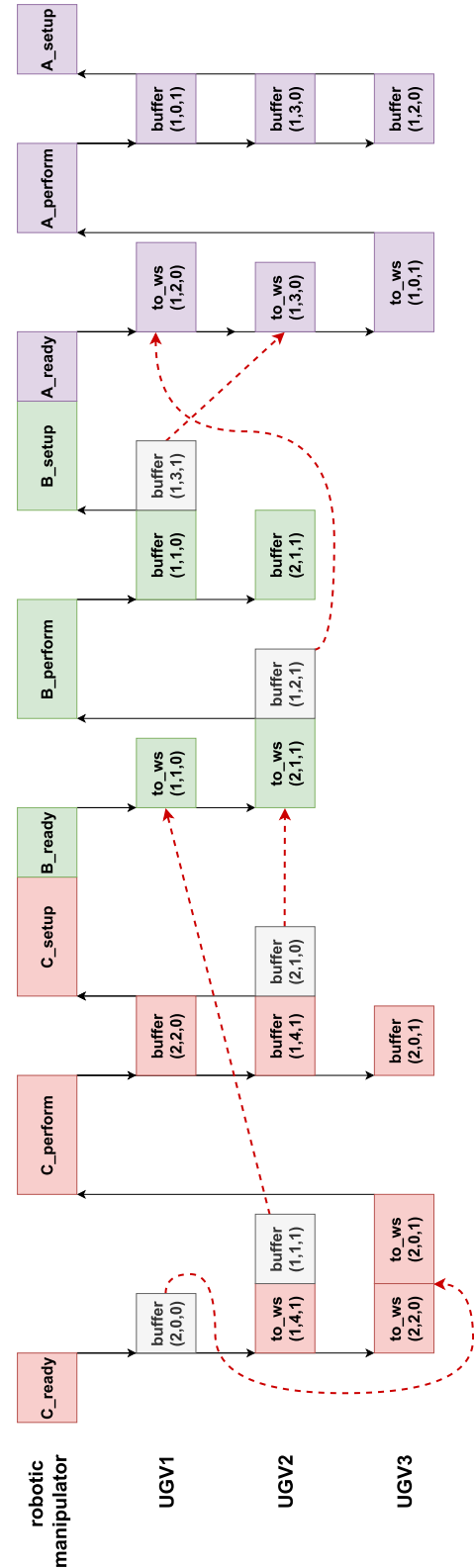
$$\sum_{i \in V, i \neq j} x_{ijk} = \sum_{i \in V} x_{jik}, \quad \forall j \in V, k \in K, \quad (7.4)$$

$$\sum_{i \in V} \sum_{j \in V \setminus \{0\}, j \neq i} q_j x_{ijk} \leq Q, \quad \forall k \in K, \quad (7.5)$$

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in S, j \neq i} x_{ijk} \leq |S| - 1, \quad \forall S \subseteq N, \quad (7.6)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall k \in K, (i, j) \in E. \quad (7.7)$$

The objective function (7.1) minimizes the total travel cost. The constraints (7.2) are the degree constraints that ensure that exactly one vehicle visits each customer. The constraints (7.3) and (7.4) guarantee that each vehicle leaves the depot only once, and that the number of vehicles arriving



**FIGURE 8.** Final schedule for the SPECULARIA use case, with tasks color-coded as per Figure 5. Tasks involving the movement of plants that do not require maintenance into the buffer are marked in gray. Precedence relations are indicated by arrows.

at each customer and returning to the depot is equal to the number of vehicles departing from that node. Capacity

constraints are expressed in (7.5), and ensure that the sum of the demands of the customers visited on a route is less than or equal to the capacity of the vehicle providing the service. The sub-tour elimination constraints (7.6) ensure that the solution does not contain cycles disconnected from the depot. The constraints (7.7) specify the domains of the variables. This model is known as a three-index vehicle flow formulation.

## APPENDIX B PERFORMANCE OF THE POPULATION-BASED CBM

First, we evaluate the performance of the proposed CBM-pop compared to the single-solution variant of CBM [3]. In Figure 7 is the visual representation of the current (blue) and best solution (red) trajectories for the two variants of the CBM algorithm on a benchmark problem for MDVRP. The two plots illustrate the contrast of the two approaches in the early search of the solution space. The CBM-pop manages to quickly jump through various solution configurations and explore different local optima, leading to faster convergence towards the optimal region. In Figure 7b, we can observe a rapid convergence (within the first 200 iterations of the algorithm) of the found solutions towards the green line, which is the best known solution for the given problem. In contrast, the solution trajectories in Figure 7a show a noticeably slower progress towards better solutions. This difference is the direct result of the genetic diversity of the obtained solution pool of CBM-pop and is consistent with the intended algorithm design.

## APPENDIX C SPECULARIA USE CASE SCHEDULE

See Figure 8.

## REFERENCES

- [1] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1495–1512, Oct. 2013.
- [2] S. Salhi, A. Imran, and N. A. Wassan, "The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation," *Comput. Oper. Res.*, vol. 52, pp. 315–325, Dec. 2014.
- [3] D. Meignan, A. Koukam, and J.-C. Créput, "Coalition-based meta-heuristic: A self-adaptive metaheuristic using reinforcement learning and mimetism," *J. Heuristics*, vol. 16, no. 6, pp. 859–879, Dec. 2010.
- [4] (2021). *Task Planning Benchmark Repository*. Accessed: Feb. 19, 2021. [Online]. Available: <https://sites.google.com/view/taskplanningrepository/home>
- [5] Gurobi Optimization LLC. (2019). *Gurobi Optimizer Reference Manual*. [Online]. Available: <http://www.gurobi.com>
- [6] E. Nunes, M. McIntire, and M. Gini, "Decentralized multi-robot allocation of tasks with temporal and precedence constraints," *Adv. Robot.*, vol. 31, no. 22, pp. 1193–1207, Nov. 2017.
- [7] M. Car, B. A. Ferreira, J. Vuletic, and M. Orsag, "Structured ecological cultivation with autonomous robots in agriculture: Toward a fully autonomous robotic indoor farming system," *IEEE Robot. Autom. Mag.*, vol. 30, no. 4, pp. 77–87, Dec. 2023.
- [8] B. A. P. Ferreira, "Decentralized mission planning for heterogeneous robotic teams based on hierarchical task representation," Ph.D. dissertation, Faculty Elect. Eng. Comput., Univ. Zagreb, Zagreb, Croatia, 2022.
- [9] J. Capitan, M. T. J. Spaan, L. Merino, and A. Ollero, "Decentralized multi-robot cooperation with auctioned POMDPs," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3323–3328.
- [10] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, Aug. 2009.
- [11] H. Mitiche, D. Boughaci, and M. Gini, "Iterated local search for time-extended multi-robot task allocation with spatio-temporal and capacity constraints," *J. Intell. Syst.*, vol. 28, no. 2, pp. 347–360, Apr. 2019.
- [12] J. Godoy and M. Gini, "Task allocation for spatially and temporally distributed tasks," in *Intelligent Autonomous Systems 12*. Berlin, Germany: Springer, 2013, pp. 603–612.
- [13] Z. G. Saribatur, E. Erdem, and V. Patoglu, "Cognitive factories with multiple teams of heterogeneous robots: Hybrid reasoning for optimal feasible global plans," in *Proc. IEEE/RJS Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 2923–2930.
- [14] P. Schillinger, M. Bürger, and D. V. Dimarogonas, "Auctioning over probabilistic options for temporal logic-based multi-robot cooperation under uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7330–7337.
- [15] S. Omidshafiei, A. Agha-Mohammadi, C. Amato, S. Liu, J. P. How, and J. Vian, "Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 231–258, Feb. 2017.
- [16] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan, "Decentralized planning and control for UAV-UGV cooperative teams," *Auto. Robots*, vol. 42, no. 8, pp. 1601–1618, Feb. 2018.
- [17] M. Krizmancic, B. Arbanas, T. Petrovic, F. Petric, and S. Bogdan, "Cooperative aerial-ground multi-robot system for automated construction tasks," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 798–805, Apr. 2020.
- [18] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: Freeman, 1990.
- [19] G. A. Korsah, "Exploring bounded optimal coordination for heterogeneous teams with cross-schedule dependencies," Ph.D. thesis, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Jan. 2011.
- [20] A. Soeanu, S. Ray, M. Debbabi, J. Berger, A. Boukhtouta, and A. Ghanmi, "A decentralized heuristic for multi-depot split-delivery vehicle routing problem," in *Proc. IEEE Int. Conf. Autom. Logistics (ICAL)*, Aug. 2011, pp. 70–75.
- [21] A. Soeanu, S. Ray, M. Debbabi, J. Berger, and A. Boukhtouta, "A learning based evolutionary algorithm for distributed multi-depot VRP," in *Proc. KES*, 2012, pp. 49–58.
- [22] S. Z. Zhang and C. K. M. Lee, "An improved artificial bee colony algorithm for the capacitated vehicle routing problem," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 2124–2128.
- [23] M. Saleh, A. Soeanu, S. Ray, M. Debbabi, J. Berger, and A. Boukhtouta, "Mechanism design for decentralized vehicle routing problem," in *Proc. 27th Annu. ACM Symp. Appl. Comput.* New York, NY, USA: Association for Computing Machinery, Mar. 2012, pp. 749–754, doi: [10.1145/2245276.2245419](https://doi.org/10.1145/2245276.2245419).
- [24] *Distributed Allocation and Scheduling of Tasks With Cross-schedule Dependencies for Heterogeneous Multi-robot Teams*. Accessed: Feb. 19, 2021. [Online]. Available: <https://sites.google.com/view/vrp-task-planning/home>
- [25] F. Li, B. Golden, and E. Wasil, "The open vehicle routing problem: Algorithms, large-scale test problems, and computational results," *Comput. Oper. Res.*, vol. 34, no. 10, pp. 2918–2930, Oct. 2007.
- [26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.
- [27] B. Ombuki-Berman and F. T. Hanshar, "Using genetic algorithms for multi-depot vehicle routing," in *Bio-Inspired Algorithms for the Vehicle Routing Problem*, F. B. Pereira and J. Tavares, Eds. Berlin, Germany: Springer, 2009, pp. 77–99, doi: [10.1007/978-3-540-85152-3\\_4](https://doi.org/10.1007/978-3-540-85152-3_4).
- [28] A. Sprecher, R. Kolisch, and A. Drexl, "Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 80, no. 1, pp. 94–102, Jan. 1995.
- [29] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. Hoboken, NJ, USA: Wiley, 2001.
- [30] F. Baptista Pereira and J. Tavares, *Bio-inspired Algorithms for the Vehicle Routing Problem*, vol. 161, 1st ed. Berlin, Germany: Springer, 2009.
- [31] Stanford Artificial Intelligence Laboratory. (2018). *Robotic Operating System*. [Online]. Available: <https://www.ros.org>
- [32] C. Lehnert, I. Sa, C. McCool, B. Upcroft, and T. Perez, "Sweet pepper pose detection and grasping for automated crop harvesting," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2428–2434.
- [33] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications (MOS-SIAM Series on Optimization)*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2015. [Online]. Available: [https://books.google.hr/books?id=YL\\_CswEACAAJ](https://books.google.hr/books?id=YL_CswEACAAJ)



**BARBARA ARBANAS FERREIRA** (Member, IEEE) received the Ph.D. degree from the Faculty of Electrical Engineering and Computing (FER), University of Zagreb (UNIZG), in February 2022, focusing on the decentralized coordination of heterogeneous multi-robot systems under the co-supervision of Prof. Stjepan Bogdan and Prof. Martínez-de Dios. She is currently a Postdoctoral Researcher with the Laboratory for Underwater Systems and Technologies (LABUST),

UNIZG-FER. To date, she has authored or coauthored one book chapter, six journal articles, and nine conference papers. Her research interests include multi-robot coordination and planning, distributed artificial intelligence, scheduling, and optimization. She has been involved in several international and national research projects, including H2020 project subCULTron, FP7 EuRoC, and the Croatian Science Foundation project SPECULARIA. She also participated in the ERL Emergency Robots 2019 and MBZIRC 2020 Robotics Competitions as a member of the LARICS Team. She is currently working on the Horizon Europe project SeaTechHub and was the Team Leader of UNIZG-FER, which won the MBZIRC 2023 Competition.



**TAMARA PETROVIĆ** (Member, IEEE) is currently an Assistant Professor with the Laboratory for Robotics and Intelligent Control Systems (LARICS), Faculty of Electrical Engineering and Computing, University of Zagreb. She has published two book chapters, six journal articles, 17 conference papers, and participates as a reviewer in several international scientific journals and conferences. Her scientific and research interests include multi-robot systems and discrete event systems. In 2016 and 2017, she was the Faculty Erasmus Coordinator in traineeships and control engineering and automation students' mobility. She was the Vice President (2012–2013) and the President (2014–2015) of the Croatian Section of the IEEE Society for Robotics and Automation.



**MATKO ORSAG** (Member, IEEE) is currently an Associate Professor with the Faculty of Electrical Engineering and Computing (FER), University of Zagreb (UNIZG). In 2011 and 2012, he was a Visiting Researcher with Drexel University, Philadelphia, USA, as a recipient of the Fulbright exchange grant. Currently, he is a Researcher in several EU projects: AeRoTwin, ENCORE, RoboCom, and NATO project MORUS. He is the PI of SPECULARIA and H2020 ENDORSE

Project. He has authored or coauthored 30 scientific and professional articles, a monograph, and a book chapter on unmanned aerial systems and robotics. He is an Associate Editor of *Automatika* journal. He is currently the Co-Chair of the Croatian Section of IEEE RAS.



**J. RAMIRO MARTÍNEZ-DE DIOS** (Member, IEEE) is currently a Full Professor with the University of Seville. His research and development activities are focused on aerial robot perception, multi-robot cooperation, robot localization and mapping, and sensor fusion. He has authored or coauthored >130 publications. He has coordinated >14 research and development projects and participated in other >60 research and development projects, including >18 projects funded by the

European Commission in FP4V, FPV, FPVI, FP6, and H2020. He led and participated in >12 technology transfer actions to companies, such as AIRBUS, BR&TE, or IBERDROLA. He is a member of the editorial board of >seven journals, an Associate Editor of IEEE RA-L, and has been a TPC member in >60 conferences and workshops. He was a recipient or co-recipient of five international awards.



**STJEPAN BOGDAN** (Senior Member, IEEE) is currently a Full Professor with the Laboratory for Robotics and Intelligent Control Systems (LARICS), Faculty of Electrical Engineering and Computing, University of Zagreb. He is the coauthor of four books and has published more than 200 conferences and journal articles. He was the Principal Investigator and a Researcher on 30 national and international scientific projects.

His research interests include autonomous systems, aerial robotics, multi-agent systems, intelligent control systems, bio-inspired systems, and discrete event systems. He serves as an associate editor for several scientific journals and conferences and was a program and organizing committee member of major control and robotics conferences.

...