ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Robotics and Autonomous Systems 0 (2021) 1–18

Robotics and Autonomous Systems

www.elsevier.com/locate/procedia

Multi-Robot Task Allocation in Disaster Response: Addressing Dynamic Tasks with Deadlines and Robots with Range and Payload Constraints

Payam Ghassemi[a], Souma Chowdhury[a,b]

[a]*Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY 14260, USA.*
[b]*Address all correspondence to this author. Email: soumacho@buffalo.edu*

Abstract

This paper tackles a class of multi-robot task allocation (MRTA) problems called “Single-Task Robots and Single-Robot Tasks” or SR-ST problems, subject to the following additional characteristics: tasks with deadlines, tasks that are generated during the mission, and robots with range and payload constraints (thus requiring multiple tours per robot). While these characteristics are typical of various disaster response and commercial applications, there is a lack of online MRTA solutions to address them. To solve this class of complex MRTA problems, an efficient online method (which is also suitable for decentralized deployment) is developed based on the construction and weighted matching of bipartite graphs. An exact integer linear programming (ILP) formulation of this class of MRTA problems is also developed, the solution of which serves both as an offline MRTA approach and as a provably optimal benchmark against which the online method is compared. The new methods are applied to a flood response problem where multiple unmanned aerial vehicles must respond to victims spread out over a large area. The results show that the new online algorithm closely trails the offline ILP method in terms of task completion performance, while being $> 10^3$ times more computationally efficient compared to the ILP method. Dedicated case studies provide further insights into the favorable scalability of the online method with an increasing number of UAVs – offering up to 46% higher task completion compared to a random walk baseline in huge 1000-task problems. Lastly, application to a slightly different class of SR-ST problems and comparison of the ensuing results with that of corresponding state-of-the-art methods demonstrate the potential wider applicability of the proposed online MRTA method.

Keywords: Bipartite graph, Integer Linear Programming, Multi-UAV Flood response, Multi-robot task allocation, Unmanned aerial vehicles.

1. Introduction

1.1. Multi-robot Task Allocation

Coordinating a large number of tasks among robots in a team calls for efficient *multi-robot task allocation* or MRTA methods [1]. Potential real-life MRTA applications such as disaster response [2], environment monitoring [3, 4], and reconnaissance [5]) present unique challenges in the form of dynamically occurring tasks that have deadlines and robots with payload-capacity and ferry-range constraints. This paper develops and tests a new computationally-efficient online algorithm, based on weighted bipartite graphs, that can be potentially deployed onboard robots to perform asynchronous selection of tasks in MRTA problems. As an example application, we consider task allocation for a team of cooperative unmanned aerial vehicles (UAVs) that are executing a flood response mission by delivering survival kits to spatially distributed victim locations.

Broadly speaking, here we consider MRTA problems that can be formulated as finding a set of optimal task sequences to be assigned to each robot in a manner that maximizes the completion rate of all tasks with deadlines. These problems fall into the Single-task Robots, and Single-robot Tasks (SR-ST) class defined in [1, 6]. In this case, assigning any task to more than one robot leads to a conflict, aka an infeasible allocation. The specific complex characteristics of SR-ST problems that we aim to tackle in this paper include:
1) limited payload and operating range of robots;
2) generation of new tasks while the multi-robotic response mission is ongoing, e.g., driven by external discovery of victims;
3) tasks with completion deadlines; and
4) unbalanced robot-task scenarios [7], where tasks in the mission greatly outnumber robots in the team, thereby demanding the planning of multiple tours per robot.
These characteristics are central to many disaster response, logistics, and military applications [8, 9], and remain challenging to be concurrently addressed by existing MRTA methods.

Along with our new online approach, we also formulate and solve an integer linear programming (ILP) representation of the targeted class of SR-ST problems with the above described characteristics. This is to serve as a provably-optimal benchmark for evaluating the quality of the solutions provided by our new online MRTA approach, as well as judge the computational-efficiency gains offered by the latter. Due to its computational complexity, this ILP solution approach is suited for centralized and offline deployment. The new MRTA approaches are tested and compared over a set of simulated multi-UAV flood-response experiments involving varying numbers of UAVs and tasks, as well as compared with two state-of-the-art approaches over a set of general MRTA benchmarks. The remainder of this section briefly surveys the literature on related MRTA approaches, and converges on the objectives of this paper.

1.2. MRTA: Optimal Offline Approaches

To solve the MRTA problem, a wide variety of methods have been studied, which can be broadly categorized as: **1)** optimal or near-optimal methods, which are typically computationally expensive and suitable for offline or (at intermediate scales) centralized offboard deployment [8]; and **2)** online (usually approximate) methods that are computationally efficient and potentially suitable for onboard decentralized deployment [14]. In the first category, a single entity is assumed to gather information from all robots to perform task allocation, i.e., assign a set of tasks to each robot. There, the problem is often formulated as an Integer Linear Programming (ILP). When tasks are defined in terms of location, the MRTA problem becomes analogical to the Multi-Traveling Salesmen Problem (mTSP) [15, 16] or the more generalized Vehicle Route Planning (VRP) problem [17]. Existing solutions to mTSP and VRP problems in the literature [18, 19] have addressed analogical problem characteristics of interest to MRTA, albeit in a disparate manner; these characteristics include limited vehicle capacity, tasks with deadlines, and multiple tours per vehicle, with applications in the operations research and logistics communities [20, 21, 22, 23]. ILP-based solutions to mTSP-type problems have also been extended to MRTA implementations [24, 25, 26].

To provision the offline MRTA benchmark, in this paper we make important extensions to our recent ILP-based MRTA formulation [27]. Unlike existing ILP-based MRTA solutions, our approach aims to *simultaneously* capture the following complexities presented by disaster response applications: tasks with deadlines, robots with limited ferry range and limited payload, and allowance of multiple tours per robot. These complexities are modeled as additional constraints in the ILP formulation. Table A.3 in Appendix A provides a characteristic comparison of our offline MRTA approach, called *Constrained-Linear Integer Programming for MRTA* or **CLIP-MRTA**, to a representative set of existing ILP-based implementations.

1.3. MRTA: Approximate Online Approaches

Although the ILP-based approaches can in theory provide optimal solutions, they are characterized by exploding computational effort as the number of robots and tasks increases [28, 29, 30, 31, 27, 24]. In contrast, the second class of methods, namely online MRTA methods seek to provide more computationally tractable (while often not provably optimal) solutions [15]. This includes heuristics-based methods to find approximated online solutions for MRTA problems that are related to capacitated VRP (CVRP) problems [32, 33]. Majority of the classical online and decentralized MRTA solutions fall into the class of market-based and consensus-based auction methods, where robots place bids on services or resources that must be allocated [34, 35]. Decentralized auction-based methods typically require multiple biddings to yield conflict-free decisions [14, 36, 12, 7]. This class of methods has been shown to provide promising scalability with an increasing number of robots and tasks [34, 30]. This scalability can be in-part attributed to their ability to use only local information (e.g., information from other agents in a neighborhood) to converge to efficient solutions to the much larger overall problem [34, 30]. Auction-based approaches rely heavily on communication among agents especially during the consensus phase, and (with very few exceptions, e.g., ACBBA [37]) demand synchronous decision-making. Hence their performance can significantly degrade in terms of solution quality (e.g., traveled distance) or feasibility when the quality of communication decreases [30, 38]. Moreover, in the case of problems needing dynamic allocations over time (as opposed to static allocation where all tasks are known at the beginning of the mission), the performance of some of the best auction-based methods have been shown to significantly deteriorate in terms of typical metrics such as traveled distance and completion time [30, 39].

To tackle dynamic allocation of tasks, while allowing a computationally tractable and asynchronous decentralized deployment, we instead represent the MRTA problem as a bigraph. Graph matching is then performed to assign tasks to robots in a conflict-free manner. Previously, Ismail and Sun [40] have shown that in simple MRTA problems, a Hungarian (graph-matching) algorithm can provide superior computational performance/scalability compared to that of the conventional Consensus-Based Bundle Algorithm (CBBA)-type auction approach, with an increasing number of robots; which partly motivates our approach. In our proposed bigraph based method, communication needs are limited to broadcasting the task choices made individually by each robot, and no synchronization is needed. This is with the assumption that once created tasks are fully observable across the team. This new bigraph based online MRTA method has been named *Bi-Graph MRTA* or **BiG-MRTA**.

With regards to handling tasks with deadlines, only a fraction of auction-based decentralized approaches are applicable. Notable examples include the Temporal Sequential Single-Item auction approach by Nunes and Gini [10] and the distributed auction-based approach by Luo et al. [11]. However, these approaches do not account for constraints on the robot's capabilities, namely ferry-range and payload constraints, that strongly impact allocation decisions. In our proposed BiG-MRTA method, *a new incentive model is incorporated to determine the edge weights of the bigraph that drive the task allocation process; this model simultaneously accounts for task deadlines, the cost of selecting a new task given the robot's current task commitment, and robot's range constraints.* Table 1 provides a problem-characteristic comparison of BiG-MRTA with a representative set of other decentralized methods for solving the SR-ST class

Table 1. Comparison of BiG-MRTA with other well-known decentralized methods for solving Single-task Robots, Single-robot Tasks problems with hard deadline, in terms of problem characteristics.

Study	Dynamic Tasks	Task Deadline	Constrained Range	Constrained Payload	Scale (#robots:#tasks)
BiG-MRTA	✓	✓	✓	✓	100:1000
Liu and Shell [7]	✓	-	-	-	100:100
Nunes and Gini [10]	✓	✓	-	-	50:100
Luo et al. [11]	✓	✓	-	-	20:100
Lee [12]	✓	-	-	✓	11:150
Su et al.[13]	✓	-	-	-	10:100
Choi et al. [14]	✓	-	-	-	5:40

of task allocation problems. In our case, scalability with the number of robots/tasks (up to 100/1000) is of interest.

1.4. Objectives of this Paper

To provide a clear context for the SR-ST class of problems we are tackling and allow insightful evaluation of our proposed MRTA approaches, we use "*multi-UAV delivery of survival kits to flood victims*" as the demonstrative application. This is motivated by the growing body of research on multi-UAV operations in emergency response applications, e.g., post-nuclear-meltdown radiation tracking [41], survivor search [42, 43], and offshore oil spill mapping [44]. There also exist a few recent studies on the usage of a single/multiple UAVs for flood response, e.g., to locate victims or provide flood warning [45]. In our problem definition, each task is given by the victim(s) location to be visited by the UAV. We solve the ensuing MRTA problem subject to the following assumptions (at par with existing work in MRTA): **i)** all robots are identical and start/end at the same depot [10, 12]; **ii)** there are no environmental uncertainties that affect the robots' motion or the task properties [7, 10, 33]; **iii)** task information (e.g., location and deadline) is readily available to all robots without any restriction [7, 10, 12, 33].

The primary contributions of this paper thus lie in developing: **1)** an online MRTA method (BiG-MRTA) that can tackle dynamic tasks, task deadlines and robot's constraints; **2)** an ILP formulation that can furnish optimal (offline) MRTA solutions for evaluating the BiG-MRTA method over static task scenarios; and **3)** a simulation environment for testing these MRTA methods on the multi-UAV flood response application, with provisions for asynchronous task selection, dynamic task creation, and multiple tours per UAV. The performances of the new methods are compared with that of a *feasibility-preserving random-walk* baseline. We also conduct scalability analysis of BiG-MRTA, and parametric analysis of the impact of robot's capability constraints, incentive model heuristics, and communication delays on the performance of BiG-MRTA. Lastly, we present a competitive analysis to evaluate whether our new online method is more broadly applicable to related SR-ST type of problems, by comparing its performance with three existing MRTA methods, namely Iterative Local Search (ILS) [46], enhanced ILS [33]), and Earliest Deadline First (EDF) [47], over a suite of 96 benchmark MRTA problems.

The remainder of the paper is organized as follows: The next section describes the mathematical elements of the SR-ST problem with the complexities that we aim to tackle, and develops the ILP formulation of this MRTA problem. Section 3 presents our proposed online MRTA method. Section 4 describes the flood simulation and associated case studies. Results and discussion of performance of our methods, and further parametric analyses, are presented in Section 5, followed by the competitive analyses of the new method's broader applicability in Section 6. The paper ends with concluding remarks.

2. The Offline CLIP-MRTA Formulation

2.1. MRTA: Defining Problem Components

The multi-robot task allocation (MRTA) problem is defined as the allocation of tasks and resources among several robots that act together without conflict in the same environment to accomplish a common mission. Each robot can share its state and its world view with other robots. Here, the MRTA problem state can be expressed as a tuple, $\mathsf{T} = <\mathcal{R}, \{\mathcal{S}_r\}, \mathcal{T}, \{\mathcal{A}_r\}, \{C_r\}, \mathbb{M}, \mathcal{G}>$; the components of this tuple are defined below.

- $\mathcal{R} = \{1, \ldots, m\}$ is a finite non-empty set of m robots.
- $\mathcal{S}_r$ is a set of variables representing the state of robot r, e.g., its current location, decision (allocated task), battery state or remaining range, and payload. Each robot can share its state variables with its peer robots. The state of all the peers of robot r is represented by $\mathcal{S}_{-r}$.
- $\mathcal{T} = \{0, \ldots, n\}$ is a finite non-empty set of active tasks that each robot is allowed to take, with the depot location denoted by the index 0. Active tasks are defined as a set of tasks with a valid (unexpired) time deadline; any tasks with a missed deadline will be discarded.
- $\mathcal{A}_r : \mathcal{T} \times (\mathbf{H} + \mathbf{Q})$ is the set of decisions taken by robot r during the whole mission, with $\mathcal{A}_r = \cup_k \mathcal{A}_r^k$, where $\mathcal{A}_r^k$ is the decision made by the robot r at the k^{th} decision sequence. Here, $\mathbf{H}$ and $\mathbf{Q}$ respectively represent the maximum number of tours allowed to be undertaken by each robot, and the maximum payload of each robot.
- $C_r : \mathcal{A}_r \times \mathcal{T}$ is a finite set of variables describing the relationship (e.g., cost and feasibility) of each task i with respect to robot r.

- $\mathbb{M}$ is a decision function that maps the state of robot-r and its current relation to active tasks, i.e., maps $\mathcal{S}_r$ and C_r to $\mathcal{A}_r^k$ at the k^{th} decision-making sequence; and
- $\mathcal{G}$ is a metric for evaluating the mission outcome (e.g., inverse of completion time), which needs to be maximized.

While the above definitions provide a generic description of the problem components, in practice, most of these components are programmed as pertinent vectors and matrices. The MRTA problem is defined as finding the decision function $\mathbb{M}$ that generates the optimum decision set $\mathcal{A}_r^*$. Here $\mathcal{A}_r^*$ maximizes the mission outcome $\mathcal{G}$ while satisfying $\max(\cup_{r=1}^m \mathcal{A}_r^* \cap \mathcal{T})$ (i.e., completion of the maximum feasible number of tasks, subject to the robots' capacity/trip bounds, $\mathbf{Q}$ and $\mathbf{H}$) and $\cap_{r=1}^m \mathcal{A}_r^* = \emptyset$ (i.e., ensuring allocations are conflict-free). Next, we develop a centralized ILP formulation of this MRTA problem.

As a problem context, we consider the multi-UAV mission of delivering survival kits to flood victims, where the locations of victims are assumed to be known apriori in the case of the offline/centralized CLIP-MRTA method. The online approach, described later, also considers dynamic tasks, which are generated while the mission is ongoing. The degree of mission success is defined by the number of completed tasks divided by the total number of tasks – the so called "*Completion Rate*". This objective function, similar to that used in [33, 48, 49], is better aligned with the goals of disaster response or related critical applications, where collective travel time or travel cost is of significantly lower priority than completing the tasks. In our demonstrative application, the main goal is thus to maximize the number survival kit of deliveries to victims.

2.2. CLIP-MRTA: ILP Problem Formulation

Variables and Parameters: The offline MRTA problem is formulated as an Integer Linear Programming (ILP) problem, as summarized by Eqs. (1) to (10). Here, UAVs are allowed to make multiple tours. The planning process must strictly meet the deadline of each task, while subject to the range and payload constraints of each UAV. In Eqs. (1) to (10), m and n respectively represent the number of robots and number of tasks. The decision-space of the ILP comprises two types of binary decision variables, $x_{ijs}^r \in \{0,1\}$ and $y_{is}^r \in \{0,1\}$ with $i, j = 0, 1, \ldots, n+1$.

- The variable x_{ijs}^r is 1 if robot r takes task j right after finishing task i during its s^{th} tour, and is 0 otherwise. Here, $x_{iis}^r, x_{i0s}^r, x_{(n+1)0s}^r = 0$, where the task index 0 and $n+1$ both represent the depot.
- The variable y_{is}^r is 1 if task i is allocated to robot r during its s^{th} tour, and is 0 otherwise.

Each tour is defined as departing from the depot, undertaking at least one task, and returning to the depot. Each robot has a limited payload Q (i.e., maximum tasks per tour) and a limited range Δ_{range}. Here, δ_i and t_{ij} respectively represent the time deadline of task i, and the time required to finish task j after performing task i; d_{ij} is the cost metric for taking task j after performing task i; it can represent energy consumption, distance, time, or any other problem-specific metric.

Constraints: In Eqs. (1) to (10), $\hat{\mathcal{T}} = \mathcal{T} - \{0\}, \mathcal{T}^e = \{n+1\} \cup \mathcal{T}, \hat{\mathcal{T}}^e = \mathcal{T}^e - \{0\}, \mathbf{H} = \{1, \ldots, h\}, \hat{\mathbf{H}} = \mathbf{H} - \{h\}$, where h represents the maximum number of tours each robot is allowed to undertake. In the ILP problem formulation equations, constraints (2)-(3) ensure that each task is allocated to only one robot and one tour, and each tour is a fully connected single loop. Constraints (4)-(5) eliminate any sub-tour, i.e., each tour starts from the depot (node 0) and ends at the depot (node $n+1$). Constraints (6)-(7) imply that each task is taken and finished by only one unique robot in the team, which also takes the next task on the same tour. Constraints (8)-(9) enforce the limited payload (e.g., maximum number of survival kits that each UAV can carry) and the limited range of robots (or UAVs). Finally, constraint (10) restricts each robot to select only those tasks that they can complete within the designated time deadline. While we do not impose any load balancing constraint across robots (as it is of low direct relevance to disaster response), it can be readily added to this ILP formulation for other applications.

Objective Function and Problem Formulation: Here we construct an objective function that is aligned with the goals of the demonstrative application in this paper, namely multi-UAV delivery of survival kits to victim locations in an evolving flood scenario [9] that imposes time deadlines on the delivery. Mission completion, i.e., the feasibility of responding to all victims (completing all tasks), is not guaranteed under the given constraints on robot range/payload and the allowed number of tours per robot. On the other hand, if all tasks are addressable (i.e., all victims can be responded to) prior to their deadlines, it is desirable to finish these tasks sooner. However, it becomes challenging to construct any exact linear objective function (i.e., without approximations), which can directly consider the average time to finish each task. Therefore, to take into account the above factors, while retaining the integer *linear* programming characteristics of the problem, we represent the objective function as a summation of the number of tasks completed by the robots per tour, scaled by the sequence s of the tour, aggregated over all tours. Note that, this objective function formulation leads to the possibility of multiple global optimal solutions. The overall ILP problem formulation is given below.

$$\max_{x_{ijs}^r, y_{is}^r} \sum_{s \in \mathbf{H}} \frac{1}{s} \sum_{r \in \mathcal{R}} \sum_{i \in \hat{\mathcal{T}}} y_{is}^r \tag{1}$$

subject to

$$\sum_{j \in \hat{\mathcal{T}}^e} x_{ijs}^r = y_{is}^r; \quad i \in \mathcal{T}, s \in \mathbf{H}, r \in \mathcal{R} \tag{2}$$

$$\sum_{i \in \mathcal{T}} x_{iks}^r - \sum_{j \in \hat{\mathcal{T}}^e} x_{kjs}^r = 0; \quad k \in \hat{\mathcal{T}}, s \in \mathbf{H}, r \in \mathcal{R} \tag{3}$$

$$\sum_{j \in \hat{\mathcal{T}}^e} x_{0js}^r = 1; \quad s \in \mathbf{H}, r \in \mathcal{R} \tag{4}$$

$$\sum_{i,j \in \mathcal{T}^e} x_{ijs}^r \leq \sum_{i \in \hat{\mathcal{T}}} y_{is}^r + 1; \quad s \in \mathbf{H}, r \in \mathcal{R} \tag{5}$$

$$\sum_{r \in \mathcal{R}} \sum_{s \in \mathbf{H}} y_{is}^r \leq 1; \quad i \in \hat{\mathcal{T}} \tag{6}$$

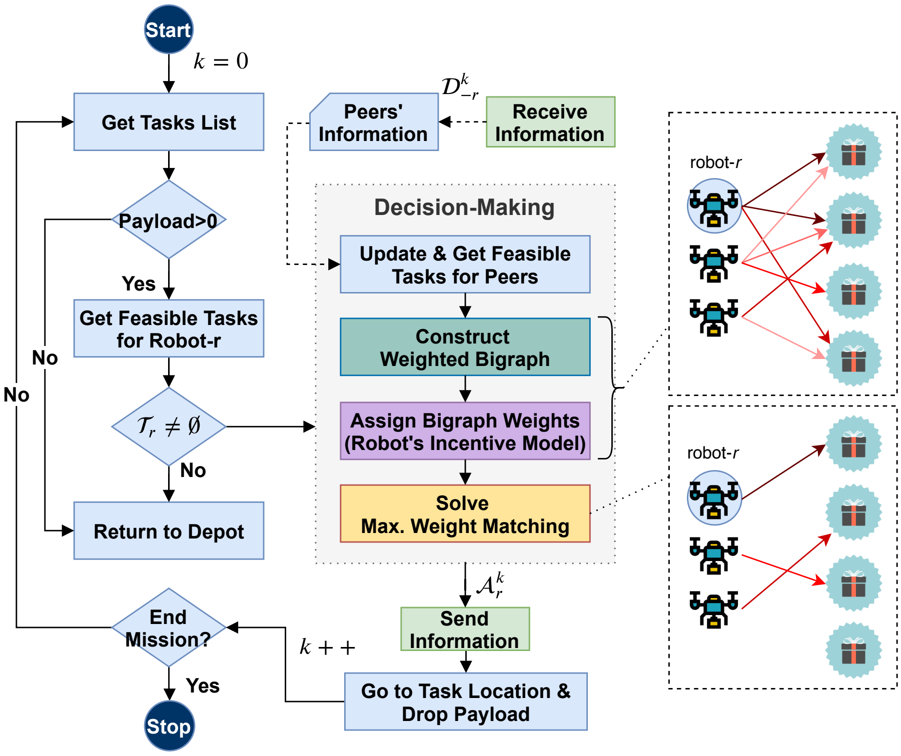

Start
$k = 0$
Get Tasks List
Payload>0
Yes
No
No
Get Feasible Tasks for Robot-r
$\mathcal{T}_r \neq \emptyset$
No
Return to Depot
End Mission?
Yes
Stop
$k++$
Peers' Information
$\mathcal{D}^k_{-r}$
Receive Information
Decision-Making
Update & Get Feasible Tasks for Peers
Construct Weighted Bigraph
Assign Bigraph Weights (Robot's Incentive Model)
Solve Max. Weight Matching
$\mathcal{A}^k_r$
Send Information
Go to Task Location & Drop Payload
robot-r
robot-r

Fig. 1. BiG-MRTA algorithm under online decentralized deployment: sequence of processes for each robot (robot-r) in the team. [Legend of block colors: Light blue indicates simple operations/actions (e.g., getting a list, comparing conditions, etc.). Light green indicates communication processes (broadcasting/receiving information). Dark green, purple, and yellow respectively show three key computational steps of the algorithm, namely the bigraph construction, bigraph weight assignment, and maximum weight matching.]

$$\sum_{r\in\mathcal{R}}\sum_{s\in\mathbf{H}} x^r_{ijs} \leq 1; \quad i,j\in\hat{\mathcal{T}} \tag{7}$$

$$\sum_{i\in\hat{\mathcal{T}}} y^r_{is} \leq \mathbf{Q}; \quad s\in\mathbf{H}, r\in\mathcal{R} \tag{8}$$

$$\sum_{i,j\in\mathcal{T}^e} d_{ij}x^r_{ijs} \leq \Delta_{\text{range}}; \quad s\in\mathbf{H}, r\in\mathcal{R} \tag{9}$$

$$\sum_{i,j\in\mathcal{T}^e}\sum_{s\in\{1..s'\}} t_{ij}x^r_{ijs} \geq \delta_{i'}y^r_{i'(s'+1)}; i'\in\hat{\mathcal{T}}, s'\in\hat{\mathbf{H}}, r\in\mathcal{R} \tag{10}$$

The complexity of this offline ILP formulation is derived to be $O(n^3m^2h^2)$. The overall cost incurred by the robot team can be expressed as: $c_{\text{overall}} = \sum_{r=1}^{m}\sum_{s=1}^{h}\sum_{i,j=0}^{n+1} d_{ij}x^r_{ijs}$, which can be used as an objective function in other applications where cost is of primary importance.

3. The Online BiG-MRTA Algorithm

3.1. BiG-MRTA: Process Overview

Figure 1 illustrates the sequence of processes and associated flow of information, encapsulating the behavior and online planning of each robot/UAV under a decentralized deployment. Onboard task selection by individual UAVs is assumed to occur in a myopic manner, i.e., at each decision-making instance a UAV only selects the next task to undertake. This is done partly due to the occurrence of randomly generated dynamic tasks, which undermine the benefits of non-myopic planning, and to preserve the computational tractability of the onboard task selection process. Importantly, these decision-making instances need not be synchronized across robots, unlike many existing decentralized implementations. For the flood response case study, the decision-making instance for any UAV is simply chosen to be a minute before completing an already committed task (we will call this the *prior committed task*). The decision-making process is instantiated only if the UAV has any remaining payload (survival kit to deliver to victims). At each of these instances, the UAV first retrieves the list of tasks, where dynamic tasks get added by an assumed external information source as soon as they are created (thus all UAVs have access to the same task information). The UAV then filters feasible tasks to undertake based on its remaining range, and if no other tasks are feasible to be undertaken, it decides to return to the depot. This task information (when feasible tasks are available), along with the latest task selection decisions broadcasted by peer UAVs, is then used by the BiG-MRTA algorithm to perform the task selection for this UAV. The ensuing decision (i.e., the information of the selected task) is immediately broadcasted by the given UAV to peer UAVs. In addition, after the given UAV finishes the immediate prior committed task, it automatically sets the selected task as its next destination and starts flying towards it. It should be mentioned that the tasks are not allowed to be preempted. The pseudocode of our online BiG-MRTA algorithm is given in Alg. 1. The principal components of the BiG-MRTA algorithm are described next. Subsequently, we provide a brief description of the communication model used to provide some degree of realism to the information exchange framework and its impact on the performance of the BiG-MRTA approach.

The BiG-MRTA algorithm is composed of three processes: **1)** *Bigraph construction*: a bipartite graph or bigraph is constructed to connect robots to tasks; **2)** *Bigraph weights assignment*: the weights of edges connecting robots and tasks are determined by an incentive model as a function of the tasks' features and robots' states, to allow problem decomposition and yield a measure of robot-task pairing suitability; and **3)** *Graph matching*: a maximum weighted (bigraph) matching problem is then solved by the individual robot to identify the optimal distribution of active tasks (one task per robot) among robots such that it maximizes the net incentive captured by the team.

3.2. Bipartite Graph Construction

In order to represent and analyze the task-robot relations, we use the concept of bipartite graphs, so-called bigraphs (popularly used in recommender systems [50] and social network analysis [51]). A bigraph is a graph whose vertices can be divided into two sets such that no two vertices in the same set are joined by an edge [52]. In this paper, we define our problem as a weighted bigraph $(\mathcal{R}, \mathcal{T}, \mathcal{E})$ during each decision-making instance, where $\mathcal{R}$ and $\mathcal{T}$ are a set of robots and a set of tasks, respectively, a representative example of which is shown on the top right portion of Fig. 1. Here, $\mathcal{E}$ represents a set of weighted edges that connect robots to available tasks, with the weight assignment process discussed in the next subsection. This bigraph definition, which facilitates one-to-one mapping between robots and tasks, is applicable to SR-ST type problems in general, i.e., where each task must be done by only a single robot.

Algorithm 1 BiG-MRTA Algorithm

Input: $\mathcal{T}^k, \mathcal{S}^k$ - the recent states of active tasks and the robots, including robot-r ($\mathcal{S}^k_r$) and its peers ($\mathcal{S}^k_{-r}$).
Output: $\mathcal{A}^k_r$ - the next decision of robot-r at its iteration k.

```
if robot-payload = 0 then
    $\mathbf{A}^k_r \leftarrow 0$                                  ▷ return to the depot
else if available-range − dist-from-Depot ≥ $\epsilon$ then
    $\mathcal{T}^{k+1}_r \leftarrow$ GETFEASIBLETASK($\mathcal{T}^k, \mathcal{S}^k_r$)
    if $\mathcal{T}^{k+1}_r \neq \emptyset$ then
        for $1 \le i \le m,\ i \neq r$ do
            $\mathcal{T}^{k+1}_i \leftarrow$ GETFEASIBLETASK($\mathcal{T}^k, \mathcal{S}^k_i$)
        $\hat{\mathcal{T}}^{k+1} \leftarrow \cup_{i=1}^{m} \mathcal{T}^{k+1}_i$
        $\mathbf{G} \leftarrow$ CONSTRUCTGRAPH($\hat{\mathcal{T}}^{k+1}, \mathcal{S}^k$)
        if k = 0 then
            $\mathcal{A} \leftarrow$ GETLARGESTEDGES($\mathbf{G}$)
        else
            $\mathcal{A} \leftarrow$ MAXMATCHGRAPH($\mathbf{G}$)
        $\mathbf{A}^k_r \leftarrow \mathcal{A}[r]$            ▷ $\mathcal{A}$ shows decisions of all robots
return $\mathbf{A}^k_r$
procedure GETFEASIBLETASK($\mathcal{T}, \mathcal{S}_r$)
    $\mathcal{T}_{\text{feasible}} \leftarrow \emptyset$
    for $i \in \mathcal{T}$ do
        $t^r_i \leftarrow$ global time that robot-$r$ finishes task-$i$
        $\Delta_r \leftarrow$ avail. range of robot-$r$ after doing task-$i$
        $w_{ri} \leftarrow$ using robots' incentive model, Eq.(11)
        if $t^r_i \le \delta_i$ and $\epsilon \le \Delta_r$ then
            $\mathcal{T}_{\text{feasible}} \leftarrow \mathcal{T}_{\text{feasible}} \cup \{\mathcal{T}_i, t^r_i, w_{ri}\}$
    return $\mathcal{T}_{\text{feasible}}$
```

3.3. Incentive Model for Bigraph Weight Assignment

The purpose of constructing a *weighted* bigraph is to systematically represent the incentive of robots for doing each task, given the task features and the state of the robots. Thus, the assignment of edge weights, incentivizing robot-task pairing, should be done in a manner that considers the following *three criteria*: **1)** the feasibility of completing the task before its time deadline; **2)** the robot must be left with enough battery capacity to return to the depot after completing the task (this can be relaxed for life-critical missions); and **3)** maximizing overall mission outcomes as encapsulated by the objective function used in the ILP (Eq. 1). Here, we hand-craft an incentive model that guarantees the satisfaction of the first two criteria, and offers a heuristic decomposition of mission level objective.

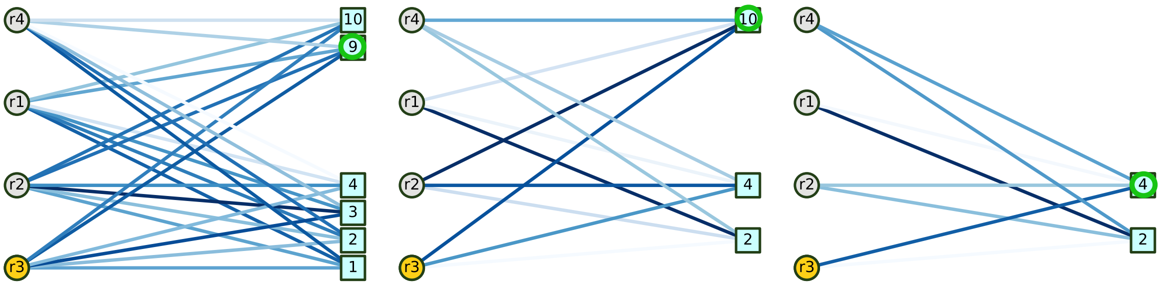
The incentive function for a robot-r (or UAV-r in our case) to choose task-i is expressed as a product of two terms. The first term gives a measure of the flight range that will be remaining (based on battery capacity remaining) if the UAV chooses and undertakes task-i and thereafter returns to the depot; this term becomes zero if the UAV wont have enough battery to return to the depot after completing task-i. This first term essentially captures the remaining potential of the UAV-r to undertake more tasks after choosing/completing task-i. The second term of the incentive function is a negative exponential function of the time (t^r_i) by which the robot r can accomplish the concerned task i if chosen next, i.e, if the task can be completed before the deadline δ_i. If the task-i cannot be completed by UAV-r before the deadline, the incentive function and thus the edge weight (w_{ri}) becomes zero. Thus, the weight, w_{ri}, of a bigraph edge (r, i) can be expressed as:

$$w_{ri} = \begin{cases} \max\left(0, \Delta_r - \epsilon\right) \cdot \exp\left(-\frac{t^r_i}{\alpha}\right) & \text{if } t^r_i \le \delta_i \\ 0 & \text{Otherwise} \end{cases} \tag{11}$$

$$\text{where} \quad \Delta_r = l^r - (d_{ri} + d_{i0})$$

In Eq. (11), l^r, d_{ri} and d_{i0} respectively represent the remaining range of the UAV-r at that time instant, the distance to be traveled by UAV-r to get to task i (subject to completing the already committed task), and the distance between task i and the depot. The parameter α is a constant used for scaling the time deadline, thereby allowing heuristic balancing of its impact on the incentive. The tolerance parameter ϵ is included to compensate for any localization errors and minimum battery level requirement of the UAV.

At the beginning, the robots' labels (r) are randomly assigned, since all robots are assumed to start from the same depot at the same time; and hence for any given task identical edge weights are assigned to each robot at the start of the mission. For later task decision instances by any UAV-r during the mission, to compute the incentive functions, it estimates the state of other peer UAVs based on the last communication received



(a) Robot-3 at t = 30 selects task 9

(b) Robot-3 at t = 120 selects task 10

(c) Robot-3 at t = 150 selects task 4

Fig. 2. Snapshots of the evolving bigraph structures for a small SR-ST case study with 10 tasks and 4 robots that run BiG-MRTA. The circle and square symbols respectively represent robots and tasks. The shown bigaphs correspond to the decision process (at different time instances) of the robot marked by an yellow circle, and its estimated decision (chosen task) is encircled with green. Darker edges indicate higher weight assignment, i.e., higher incentive for that robot-task pairing.

from them. In a deterministic environment with no communication delays/interruptions, this asynchronous approach ensures that the incentive functions (edge weight assignments) satisfy criteria 1 and 2 stated before. During our numerical experiments (in Section), we also analyze the impact of communication delays on the fidelity of the weight assignments and any degradation of mission-performance caused thereof. Studying and addressing the impact of environmental uncertainties, such as stochasticity of task features or robot states, is on the other hand considered to be a critical direction of future work.

3.4. Maximum Weight Matching

Once the weighted bipartite graph has been constructed, the final step is to solve the task selection problem as a maximum weighted matching problem [53]. This problem is defined as drawing the largest possible set of edges such that they do not share any vertices and the summation of the weights of the selected edges is maximized. To this end, we use the Karp modified maximum matching algorithm [54], which produces a conflict-free task allocation. This algorithm advances on the more classical Blossom and Hungarian graph matching algorithms by identifying a maximal set of shortest augmenting paths per iteration, and increases the augmenting path by the maximum flow instead of one at a time. A pseudocode (Alg. 3) for the graph matching algorithm is given in Appendix C. For our BiG-MRTA formulation, the computational complexity of this graph matching process is $O(\hat{m}\hat{n}\log\hat{n})$ for $\hat{m}$ robots and $\hat{n}$ tasks. Note that the size of the graph varies across each decision-making instance, as tasks get selected/completed or expire due to deadline, and new dynamic tasks get created over time during the mission. The effective number of active tasks, $\hat{m}$, typically decreases over time, unless the rate of the dynamic task creation exceeds that of task completion/expiry. In addition, the design of the incentive model automatically filters out any infeasible robot-task pairings subject to robot state and task features; and thus $\hat{n} \leq n$, where n is the total number of robots in the team. An illustration of this bigraph based task selection process, across sequential decision-making instances of a given robot, is shown in Fig. 2. This diagram shows how the structure of the graph and the weights of robot-task edges change over time.

3.5. Information Sharing

In multi-robot systems, reliable inter-robot communication is required for sharing information and maintaining a collaborative mission. Communication capabilities under real-world settings are usually subject to range limitations, delays, and packet losses [55]. In this paper, we assume the following inter-robot communication platform: a 900MHz frequency band, e.g., XBee Pro 900HP [56]. This communication platform has been previously used for multi-UAV applications by [57], with a reported range of 5 km. To consider an online decentralized deployment we use an information-sharing policy where robots broadcast a compact data package after every decision instance, as shown in Fig. 1. This package includes: current positional state, next selected task, and its remaining range (estimate) and payload.

The simulated communication range restriction of 5 km is significantly smaller than the area covered by the UAVs in our case studies (20 × 30 sq. km), a deliberate choice. Hence, at times, UAVs are outside the communication range of their peers and may not receive the latest broadcast from their peers. Assuming a multi-hop communication capability is available, the effect of the range limitation is modeled as communication delays, when studying its impact on the performance of the BiG-MRTA algorithm under decentralized deployment (in Section 5.3). For this purpose, we consider the worst-case scenario, where two robots are located at the two farthest corner points of the environment. Hence, they need 8 hops to share information. The information exchange latency is computed based on the formulations reported in [56, 57], with an assumed baud rate of 9600kbps, a success rate of 90%, a latency per hop of 25ms, an estimated data size of 15 Bytes. Based on these assumptions, the worst-case latency is computed to be 300 milliseconds. In order to allow for additional safety factor, in our case studies analyzing the effect of communication latency on MRTA performance, we consider the following two settings: a fixed 1 min and 5 min latency.

4. Case Study: Multi-UAV Flood Response

4.1. Overview

We execute a set of numerical experiments to investigate the performance and scalability of the online **BiG-MRTA** approach, and compare it with the offline **CLIP-MRTA** approach

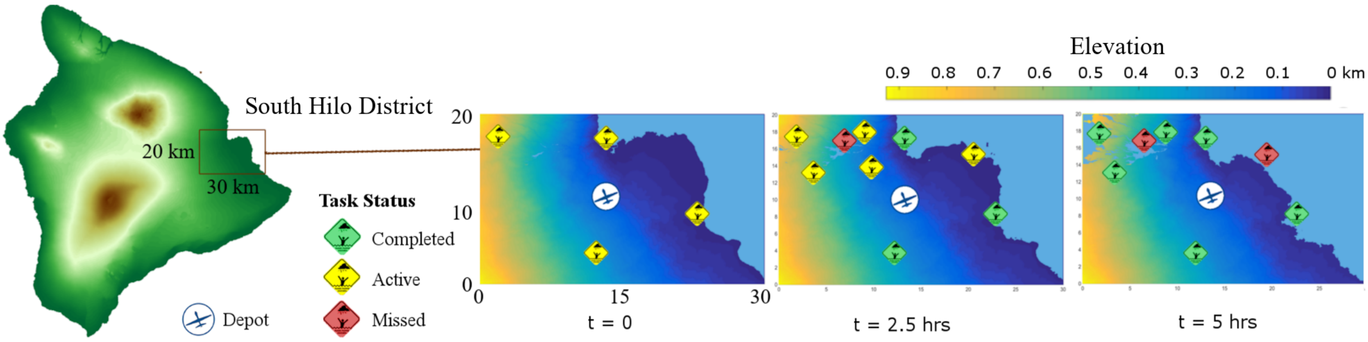


Fig. 3. Case study area showing altitude information, and example of evolution of the simulated flood, and task locations and their varying status over time.

and a *Feasibility-preserving Random-Walk MRTA* or **Feas-RND** approach. Owing to the intractability of re-running the ILP solution process every time new tasks appear, CLIP-MRTA is only applied to the cases with static tasks. Here, the Feas-RND method is introduced to provide a baseline for all instances that are solved by BiG-MRTA, since solutions obtainable by a randomized approach provides a measure of the difficulty of an optimal planning problem. However, note that Feas-RND is not a fully random method. In the Feas-RND approach, each robot randomly chooses tasks from among those that do not conflict with the decisions of other robots, and are feasible to be undertaken by this robot in terms of criteria 1 and 2 stated in Section 3.3. This is performed by a constrained random allocation process, as described by Algorithm 2 in Appendix B.

In order to evaluate the MRTA approaches, a simulation environment for multi-UAV flood response is developed, which is described next in Section 4.2. Subsequently, we describe our design of experiments with static and dynamic task case studies and different robot/task ratios, followed by a summary of our simulation and computational settings.

Here, the results of the MRTA approaches are evaluated and compared in terms of two metrics: i) completion rate and ii) computational efficiency. Completion rate can be expressed in % as: $f_{\text{CR}} = 100 \times \left(\sum_{s\in\mathbf{H}} \sum_{r\in\mathcal{R}} \sum_{i\in\hat{\mathcal{T}}} y^r_{is}\right)/n$. In addition, a scalability analysis is performed to study how team size affects the performance of the MRTA methods. A parametric analysis is then performed to study the impact of the incentive model heuristics. Finally, we perform analyses of how robot capabilities, e.g., ferry range, speed, and communication latency, affect the team performance.

4.2. Flood Response Simulation

Selection of Geography: Due to the recent flood devastation from Tropical Storm Lane in August 2018 [58] and its flood-critical characteristics, the East side of the Big Island of Hawaii, South Hilo district, is chosen as the case study area. Using a 10-meter USGS Digital Elevation Model [59], the topographical map of the island of Hawaii is obtained. From this map, we select a 30 km by 20 km well-populated region, as shown in Fig. 3, to represents the geographical area over which the multi-UAV missions are simulated, as shown in Fig. 3.

Flood Simulation: The application being mainly demonstrative, a simple flood simulation is used here. An aggressive rate of floodwater rise is considered here, which is comparable to extreme flash flooding scenarios [60, 61], and allows posing challenging dynamic task scenarios where 100% completion rate may not necessarily be achievable. The following set of assumptions is used in creating the flood simulation: i) there are two floodwater levels; ii) the first level is a horizontal plane, which starts at the elevation of the ocean water and rises uniformly at a user-defined rate (4 meters per hour in our case studies); iii) the second level is an inclined plane underneath the river drainage of Hilo ($0 \leq x \leq 10$ and $14 \leq y \leq 20$), where the water level rises at a higher averaged rate of 6 meters per hour; and iv) floodwater does not recede during the mission.

Tasks – Flood Victims: Tasks are defined by their location and time deadline. In this environment, the task locations are specified to be initially above the water level, and at least 1 km away from each other. It is assumed that a different process, e.g., a team of scouting UAVs identifies the task locations, and passes on this information to the multi-UAV response team without any delay or loss of information (i.e., immediate and complete observation of the dynamic task space is assumed). For our case studies, a distribution based on the population density of each region and the FEMA flood zones [62] is used by a random generator to create the representative task scenarios. After generating the location of tasks, the flood simulation is executed, and the time deadline of each task is defined in a deterministic manner to be the time when the water level reaches 0.5 meters above ground level at the task location. The tasks with missed deadlines are not allowed to be selected and they are thus removed from the set of available tasks. Tasks, once generated, can take three different statuses: active, completed, and missed (i.e., deadline is passed), as shown in Fig. 3

Assumed UAV Platform and Payload: Here we assume the use of hybrid UAVs, which are capable of VTOL while providing sufficient range and payload for such disaster response applications [63]. Specifically, we simulate the use of a typical tilt-rotor type UAV that offers a 2 kg payload, a 140 km flight range, and a maximum speed of 40 km/h. The payload unit carried by each UAV, to be delivered to each task location, is assumed to be a survival kit that is stocked with a first aid kit and a radio, weighing a total of 400g. Therefore, each UAV can carry a total of 5 survival kits.

4.3. Design of Experiments

Based on the simulated application, we define four different MRTA case studies [1]. These cases correspond to different combinations of numbers of UAVs and tasks, and whether dynamic tasks are included, as defined here: 1) *small static problem*: 2 UAVs and 10 tasks; 2) *large static problem*: 10 UAVs and 50 tasks; 3) *huge static problem*: 1000 tasks, with the multi-UAV team size varying from 1 to 100; and 4) *huge dynamic problem*: 1000 tasks including dynamics ones, and with the multi-UAV team size varying from 5 to 100. The fourth case starts with 500 initial tasks (that are known at the start of the mission) and up to 500 new tasks are added to the environment during the mission simulation, at a rate of 10 new tasks or locations every 12 minutes. In order to provide a statistically insightful evaluation and comparison, ten random test scenarios are generated for the first two static task cases, where the number of flood victims remain fixed while their locations are randomized across scenarios. The location of the UAV depot is fixed across all case studies and is selected to coincide with the Haihai fire station at $(10, 14)$ in Fig. 3. It is assumed that the depot provides near instantaneous battery swap and payload replenishment, which is close to what is achievable with state-of-the-art platforms.

4.4. Simulation and Framework Settings

The "Python" 3.6.0 language and the 64-bit distribution of "Anaconda" 4.3.0 are used to implement the BiG-MRTA and Feas-RND approaches and the flood simulation. The "*Gurobi v8.0*" [64] library is used as the ILP solver for implementing CLIP-MRTA. For implementing BiG-MRTA, the "*networkx*" library is used to perform graph-based computing. The MRTA simulations are executed on a workstation with Intel® i7-6820HQ 2.70 GHz 4 Cores CPU and 16 GB RAM. The offline ILP solver (Gurobi) exploits all 8 processors, while the Python implementation of BiG-MRTA does not exploit all the cores, as "*networkx*" does not offer multi-threading. Here the parameters of the Gurobi's solver are set at: time limit of 3600s, and absolute MIP optimality gap of $1e-4$. In order to promote equitable distribution of load across the robots and retain the computational tractability of the ILP solution process, the maximum number of tours per robot is set as: $h = \lfloor n/m \rfloor + 2$. For the BiG-MRTA implementation, we set the scaling length parameter in the incentive model at $\alpha = 300$ (which is the total mission time in minutes) and the margin parameter at $\epsilon = 0$.

5. Results and Discussion

5.1. Comparative Analysis of BiG-MRTA

Figure 4 summarizes the performances of the offline ILP (CILP-MRTA), the online BiG-MRTA, and the random-walk baseline (Feas-RND) approaches for the *small* and *large* test problems. From the left plot in Fig. 4, it can be observed that the completion rate of CILP-MRTA and BiG-MRTA algorithms is found to be 100% in all scenarios across both test cases, i.e., UAVs can respond to all victims before their respective deadlines. The completion rate of the baseline Feas-RND is found to vary from 94% to 100% across the case scenarios. Note that the CLIP-MRTA solutions were found to have converged to the true optima (i.e., 0.00% optimally gap) to the ILP problem defined by Eqs. (1) to (10), thereby providing a well-suited benchmark in these studies.

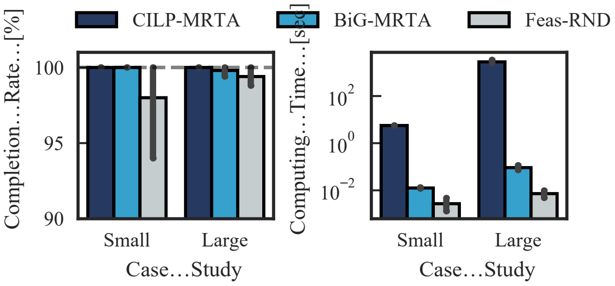


Fig. 4. The performance of the three algorithms for the *small* and *large* static-task cases. The computing time (shown in log-scale) reported for CILP-MRTA is the time required by it to reach the completion rate achieved by BiG-MRTA for that scenario. The computing time reported for BiG-MRTA and Feas-RND is the cumulative computing time over all task selection decisions.

The comparison of the computational efficiency is critical to analyzing the efficacy of the online approaches in multi-robot operations. While BiG-MRTA and Feas-RND computes one decision per run (i.e., select the next task), the offline CLIP-MRTA approach computes the entire sequence of tasks for all robots. Thus, to allow fair comparison of computational efficiency, we consider the cumulative computing time of BiG-MRTA and Feas-RND, i.e., aggregated over all decisions taken by a robot over the whole mission. Then, we report the average of the individual robots' cumulative computing times across the team for the online approaches, and the computing time of a single run of CLIP-MRTA. The computing times for the two static-task test cases, across 10 randomized missions scenarios each, are illustrated in the right plot of Fig. 4. As expected, Feas-RND is the fastest, where the only tangible computing is attributed to checking the feasibility of returning to the depot after selecting the next task. *More importantly, as seen from Fig. 4, the cumulative computing time of BiG-MRTA is 3-5 orders of magnitude smaller than that of the offline ILP approach.* The time taken by BiG-MRTA for each decision-making instance is found to be a mere 1-5 milliseconds and 2-40 milliseconds in the small and large problems, respectively.

Figure 5 depicts how the graph size (i.e., the number of edges in the bigraph) and the computing time changes over the decision-making history of the UAVs in a representative run for the huge dynamic case with 50 UAVs and 500+500 tasks. The mission duration is 5 hours, but no active tasks remain after 140 minutes. The bigraph size increases up to 22,400 and eventually decreases to 0, while exhibiting some fluctuations. The computing time scales with the size of the bigraph, as the majority of the computing cost is associated with the maximum weighted matching process. The fluctuations are due to variations caused by the dynamic generation of new tasks and the varying (help-

[1] To aid the replication of results, benchmarking and further adoption of the proposed method, the implementations of BiG-MRTA and the comparative methods, the flood simulation, and associated case study data have been made available at the following repository: `https://github.com/adamslab-ub/BiG-MRTA`.

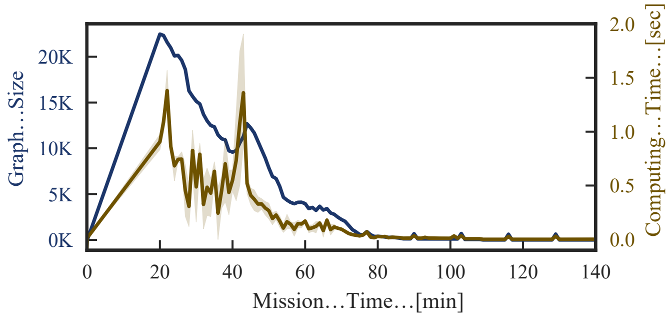


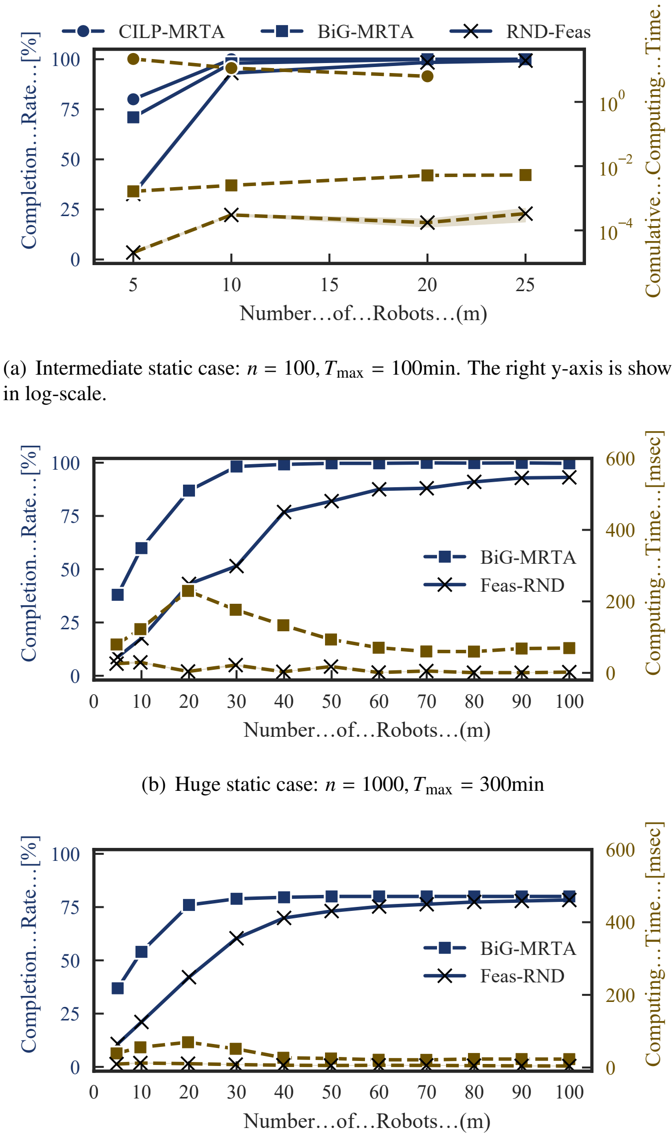
Fig. 5. BiG-MRTA (dynamic case study: $m = 50$, $n = 500 + 500$): computing time and graph size per decision-making instance per robot, plotted versus the progressive (simulated) physical time.

ful) sparsity of the bigraph constructed by each robot, where the bigraph only contains feasible edges based on the UAV's battery and remaining-payload state and the task state (remaining time till deadline). Note that, even at this substantial robot-task scale, the computing time stays lower than 1 second at all times. Adjusting for the computing capacity difference between the workstation used here and computing nodes available onboard state-of-the-art UAVs, this efficiency is considered attractive for planning purposes.

5.2. *Scalability Analysis of BiG-MRTA*

In order to study the scalability of BiG-MRTA, i.e., the impact of the number of robots on the performance and computational tractability of the algorithm, we use the *huge static* and *huge dynamic* cases, where the number robots in the team is changed as $m = 5, 10, 15, \ldots, 100$. The baseline Feas-RND is also applied for these cases, executed 10 times per case, with results averaged across them. However, owing to the extensive computational cost of running large size ILPs with 1000 tasks, especially intractable when incited repeatedly during dynamic missions), the CLIP-MRTA outcomes are generated only for a smaller 100 static tasks case for this scalability study, and compared with the online algorithms on the same case.

It can be seen from Fig. 6 that for all the three cases the completion rate increases with increasing number of robots, and saturates after a certain point, i.e., after $m = 10$, $m = 80$, and $m = 50$ in the static large, static huge and dynamic huge cases, respectively – this is due to the decreasing marginal utility of additional team members. The performance of BiG-MRTA closely trails that of the optimal CLIP-MRTA approach in the intermediate static problem, as observed from Fig. 6(a). The computing time of BiG-MRTA remains three orders of magnitude smaller than that of CLIP-MRTA. The substantial performance advantage of BiG-MRTA over Feas-RND in the three cases shown in Fig. 6 directly demonstrates the effectiveness of the BiG-MRTA algorithm. Favorable scalability of BiG-MRTA is particularly evident from the increasing completion rate margin (before saturation) of BiG-MRTA over Feas-RND, as seen from the results of the larger case (Figs. 6(b) and 6(c)) – for example, in the huge static case, this completion rate margin goes up from 29% to 46% as the number of UAVs in the team increases from 5 to 30.



(a) Intermediate static case: $n = 100, T_{\max} = 100$min. The right y-axis is shown in log-scale.

(b) Huge static case: $n = 1000, T_{\max} = 300$min

(c) Huge dynamic case: $n = 500 + 500, T_{\max} = 300$min

Fig. 6. Scalability analysis of the online BiG-MRTA approach with increasing team size (m) for three cases with different number of tasks (n) and mission duration ($T_{\max}$). The computing times are in terms of the average of individual robots' cumulative computing time over the mission.

For the huge dynamic case (Fig. 6(c)), the performances of both online algorithms saturate to completion rates that are noticeably below 100%, unlike in the static cases. This gap is due to the shorter time between the generation and expiry of the dynamic tasks (refer Fig. D.10), with the task deadlines being the same as that in the huge static case; deadlines are linked with rising localized flood levels given by the simulation. Lastly, note that the robot-averaged cumulative computing times increase up to a point (e.g., the team of 20 in the huge static case) and then decreases – the latter effect is because the increased rate of collective task completion with a larger team leads to smaller effective bigraph sizes, thus lower cost of maximum weighted matching as the mission progresses, eventually resulting in reduced cumulative computing costs per robot.

5.3. *Parametric Analysis of BiG-MRTA*

In the BiG-MRTA algorithm, there is one notable parameter (heuristics) that needs to be tuned – the time scaling parameter α. We run experiments to study how this scaling parameter (α varying from 10 to 450) affects the performance of BiG-MRTA for the huge static and dynamics cases, with a team of $m = 50$ UAVs. The performance outcomes in terms of completion rate are summarized in Fig. 7. From this figure, it can be seen that a performance sweet spot occurs around $\alpha = 300$ (expressed in the same units (minutes) as the mission completion time), and remains mostly at the same level there onwards in the huge static case – exhibiting the normalizing effect of α on “time to do task-i next, if chosen”. For the huge dynamic case, performance seems to be relatively insensitive to α, likely due to the widely different distribution of time periods over which tasks remain active.

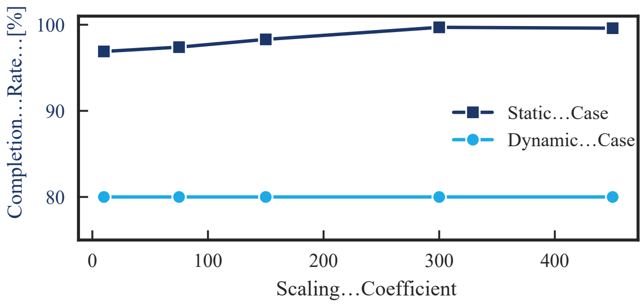


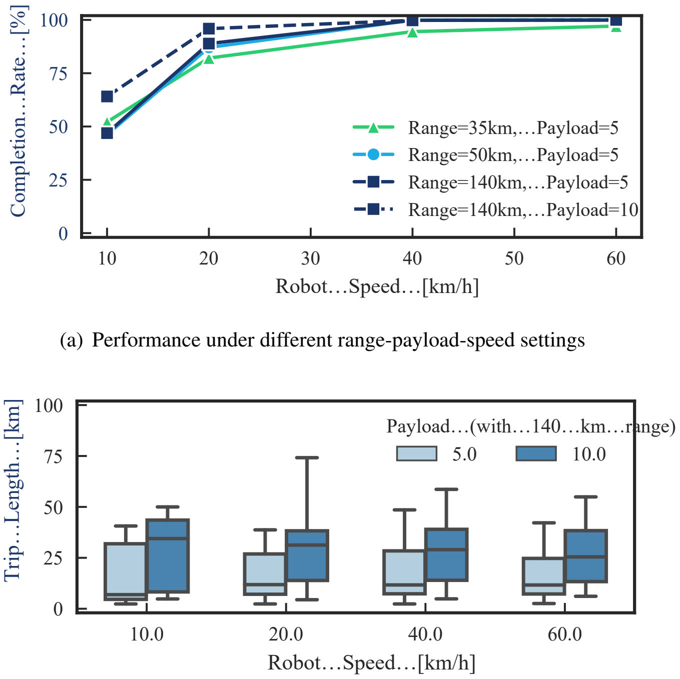
Fig. 7. Parametric analysis of BiG-MRTA: studying impact of the incentive model parameter α on completion rate, for the huge static ($n = 1000$, $m = 50$) and dynamic ($n = 500 + 500$, $m = 50$) cases.

5.4. *Impact Analysis of Robot Capabilities*

Here, we analyze how robot or UAV capabilities, namely ferry range, nominal speed, payload, and (separately) communication latency, affect the performance of the team of UAVs using the BiG-MRTA algorithm.

Ferry Range, UAV Speed, & payload: The huge static case with 50 UAVs is used to perform the impact analysis. We perform numerical experiments at different ferry ranges between 35 – 140 km with the standard payload of 5 survival kits and a double payload of 10 kits. For ease of illustration and mitigate redundancy, here we report the outcomes for four settings – 35, 50, 140 km ranges with 5 kits and 140 km range with 10 kits – that clearly portray the key observations. These range/payload settings are combined with the following UAV speed settings: 10, 20, 40, 60 km/hr. Figure 8(a) shows the completion rate achieved by BiG-MRTA under these settings. As expected performance improves with increasing UAV speed, given the increased ability to meet task deadlines. Some spurious behavior is observed at the very uncommon low speed of 10 km/hr, where a 35 km range fares slightly better than the higher range settings. With myopic planning, the spatial distribution of tasks can lead to cases involving the immediate choice of tasks that are members of sparser clusters, affecting the eventual completion rate.

From Fig. 8(a) we also observe that, at the standard payload of 5 kits, ferry range has a marginal impact on the performance. This prompted the insight that the UAVs are not exploiting the higher ferry range when available, probably because they keep running out of payloads (kits) and return to the depot with tour lengths that are significantly smaller than the ferry range. To validate this insight, we plot the distribution of tour lengths per UAV across the multiple tours they undertake over the mission, under the two different payload settings of 5 and 10 with 140 km range. From the resulting boxplots of actual tour lengths, shown in Fig. 8(b), it is readily evident that, with a double payload of 10, the team of UAVs is able to better exploit the higher range – actual tour lengths tend to be higher. This then allows achieving higher completion rates with 140 km / 10 kits settings, as seen from Fig. 8(a).



(a) Performance under different range-payload-speed settings

(b) Length of tours per UAV across the team over the entire mission with a range setting of 140 km

Fig. 8. Analyses of the impact of UAV capabilities on the performance of BiG-MRTA: for the huge static case ($n = 1000$ and $m = 50$).

Communication Latency: To analyze the impact of communication latency, we run BiG-MRTA on the huge static case ($m = 1000$ tasks). The number of robots is varied from 5 to 100 to study how the effect of latency on performance is tied to the team size, given its direct impact on the spatial distribution of robots over the mission area and the likelihood of conflicting decisions due to the information gap caused by latency. We use three different settings for this study, no latency, 1 min latency (a worst-case scenario; refer Section 3.5), and an extreme 5 min latency. This is to demonstrate that even with resource-scarce communication settings, BiG-MRTA performs significantly better than the random walk baseline (which requires no inter-UAV communication). Note that, BiG-MRTA can asynchronously plan conflict-free decisions under full observability. Hence, what matters from the perspective of one robot is having the latest information at its disposal regarding peers’ decisions, before instantiating its own planning. Latency impacts this in-

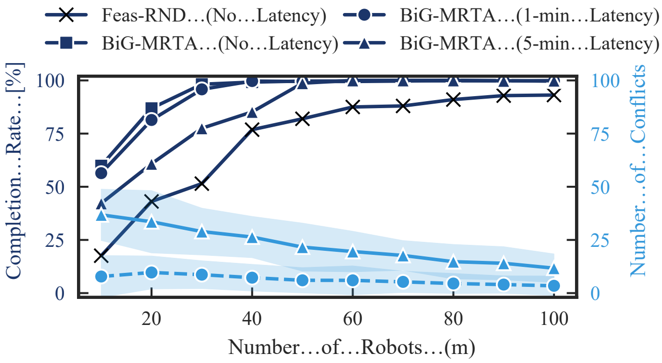


Fig. 9. Analyses of the impact of latency on the performance of BiG-MRTA: for the huge static case ($n = 1000$). The right y-axis denotes the number of conflicting decisions per UAV over the mission, shown as shaded distributions.

formation availability, thereby leaving scope for conflicts in task selections, unless robots deliberately delay their decision-making, which could then negatively impact the mission completion rate. This issue differs from the imperfect-communication-related issues plaguing synchronous planning concepts that are central to many auction-based methods.

The performance outcomes under the above communication settings, shown in Figure 9, clearly illustrates this competitive advantage of BiG-MRTA. Latency does impact the performance of BiG-MRTA, with a relatively stable offset in performance with varying team size. However, even with 5 min latency, BiG-MRTA is remarkably able to complete up to ~25% more tasks (roughly 250 more tasks) than Feas-RND with perfect communication (i.e., no latency). Figure 9 also shows the distribution (mean and 95% CIs) of the number of conflicting decisions per UAV over each mission. With one minute latency, team size appears to have a negligible impact on the number of conflicting decisions, since conflicts are in general rare. Note that, with further modifications (to BiG-MRTA in the future) that allow probabilistic estimations over the decisions of peer robots, it would be viable to mitigate these conflicts.

5.5. *Further Discussion of Findings & Potential Improvements*

In our scalability analysis, the computing time of BiG-MRTA remained tractable even for the huge static case with 1000 tasks and 100 UAVs. Further analysis of the impact of dynamic tasks earlier in this section also showed that their presence reduced the effective time periods for which tasks are active, as a result of which the overall completion rate asymptotically leveled off at a value ~25% lower than that obtained in a similarly sized static-task case. In addition, the various case studies and scalability analyses clearly point to the competitive potential of BiG-MRTA to serve in the role of an *asynchronous online* planning method that is also amenable to *decentralized deployment* over small robots such as UAVs or UGVs. The communication latency analysis did provide indirect insights into likely performance losses (while retaining scalability) under scenarios with partial observation across the team. Thus BiG-MRTA's potential for decentralized deployment requires further investigations in the future. For example, future work should explore how to adapt the bigraph construction or edge-weight assignment model, so as to estimate the likelihood of task-selections of peer robots, especially that are out of communication range of the concerned robot. Along the same lines, further adaptation of the incentive model is needed to also account for real-world uncertainties associated with robot localization and the state and observability of tasks; and study how these factors influence the sparsity of the bigraphs constructed by each robot at each decision-making instance.

6. Competitive Analysis of BiG-MRTA

6.1. *Choice of Benchmark Problems and SOTA Methods*

A direct comparison with other existing MRTA/VRP approaches is challenging, since most existing methods address some, but not exactly all of the problem complexities that we consider in this paper, namely robot ferry-range/payload constraints (and ensuing allowance of multi-tours), time deadlines on tasks, and dynamic tasks. However, to analyze the broader potential of the incentive-weighted bigraph concept underlying BiG-MRTA, we need to identify a related set of benchmark SR-ST problems (that at least share some of the major complexities considered for BiG-MRTA) with reported solutions based on well-known existing online method(s). To this end, we choose the following three existing MRTA methods for comparison:

1. the Iterative Local Search (ILS) method [46];
2. the enhanced ILS (eILS) method [33]; and
3. the Earliest Deadline First (EDF) [47]

The first two methods are based on a meta-heuristic local search approach and the third method is a heuristic method that has been used for MRTA and real-time multi-processor scheduling problems. Comparisons are performed on a comprehensive suite of problems that are relatively close (albeit not exactly the same) to the class of problems that BiG-MRTA is designed to tackle. These benchmark problems were classified as Task Allocation Problem with Time and Capacity[2] constraints (TAPTC) [46]. The TAPTC problems thus do share the "tasks with time deadlines" characteristics with the class of SR-ST problems that BiG-MRTA is designed to solve. In addition, note that, the ILS, eILS, and EDF methods can use the *completion rate* objective function, similar to that used in this paper. While the programmatic implementation of these methods were not publicly available, the performance data on all the 96 instances of the TAPTC problem is available for ILS and eILS, and the EDF method is reported [47] in sufficient detail allowing us to implement it programmatically ourselves. Appendix E provides further details on these benchmark problems. These problems are divided into two 48-instance groups: **Group 1** includes tasks with tight time deadlines (thus the tour length of solutions is small to moderate); and **Group 2** includes tasks with wider time deadlines (thus the tour length of solutions is moderate to large).

The characteristics of these benchmark SR-ST problems that differ from those considered in this paper can be summarized

[2] The term "Capacity" in [46] refers to the task achievement rate of robots, which is different from the term "Payload Capacity" that we use in our paper.

Group 1					***Group 2***				
Instance	**Score [%]**	**Score Gap [%]**			**Instance**	**Score [%]**	**Score Gap [%]**		
		ILS	**eILS**	**EDF**			**ILS**	**eILS**	**EDF**
r11a200	24	**4.35**	**4.35**	**118.18**	r21a200	47	**2.17**	**2.17**	**123.81**
r11a201	23	**0.00**	**0.00**	**91.67**	r21a201	47	**4.44**	**2.17**	**123.81**
r11a202	23	**0.00**	**0.00**	**130.00**	r21a202	44	**0.00**	-2.22	**91.30**
r11a300	31	**3.33**	**0.00**	**93.75**	r21a300	61	**3.39**	**3.39**	**134.62**
r11a301	37	**0.00**	-2.63	**105.56**	r21a301	73	**2.82**	**1.39**	**160.71**
r11a302	31	**3.33**	**0.00**	**138.46**	r21a302	63	**5.00**	**3.28**	**162.50**
r11a500	51	**0.00**	-1.92	**131.82**	r21a500	87	-8.42	-10.31	**27.94**
r11a501	50	-1.96	-3.85	**100.00**	r21a501	89	-4.30	-8.25	**36.92**
r11a502	53	-3.63	-7.02	**120.83**	r21a502	96	-3.03	-4.00	**18.52**
r11a700	80	**5.26**	**2.56**	**45.45**	r21a700	99	-1.00	-1.00	-1.00
r11a701	78	**0.00**	-1.27	**41.82**	r21a701	99	-1.00	-1.00	-1.00
r11a702	74	-2.63	-2.63	**51.02**	r21a702	97	-3.00	-3.00	-3.00
r12a200	24	**14.28**	**9.09**	**100.00**	r22a200	46	**12.20**	**4.55**	**155.56**
r12a201	22	**4.76**	**4.76**	**100.00**	r22a201	44	**2.33**	**2.33**	**144.44**
r12a202	24	**4.35**	**4.35**	**118.18**	r22a202	46	**6.98**	**2.22**	**142.11**
r12a300	30	**3.45**	**0.00**	**130.77**	r22a300	58	**1.75**	-3.33	**114.81**
r12a301	36	**2.86**	0.00	**89.47**	r22a301	71	**2.90**	**0.00**	**136.67**
r12a302	30	**7.14**	**0.00**	**114.29**	r22a302	62	**5.08**	**5.08**	**113.79**
r12a500	51	**0.00**	**2.00**	**88.89**	r22a500	84	-5.62	-9.68	**86.67**
r12a501	52	**1.96**	**0.00**	**116.67**	r22a501	89	-2.19	-5.32	**71.15**
r12a502	57	**3.64**	**1.79**	**111.11**	r22a502	89	-6.32	-7.29	**81.63**
r12a700	70	-1.41	-4.11	**75.00**	r22a700	95	-5.00	-5.00	**18.75**
r12a701	74	-2.63	-2.63	**124.24**	r22a701	97	-3.00	-3.00	**15.48**
r12a702	68	-8.11	-6.85	**83.78**	r22a702	98	-2.00	-2.00	**22.50**
r13a200	22	**0.00**	**0.00**	**69.23**	r23a200	43	**4.88**	**4.88**	**126.32**
r13a201	22	**4.76**	**0.00**	**120.00**	r23a201	41	**2.50**	**0.00**	**141.18**
r13a202	19	-5.00	-5.00	**90.00**	r23a202	42	**5.00**	**2.44**	**110.00**
r13a300	31	**3.33**	**0.00**	**93.75**	r23a300	54	**0.00**	-1.82	**63.64**
r13a301	35	**6.06**	**0.00**	**84.21**	r23a301	63	-4.54	-5.97	**90.91**
r13a302	29	**0.00**	-6.45	**70.59**	r23a302	54	-1.82	-3.57	**80.00**
r13a500	48	**0.00**	-4.00	**92.00**	r23a500	73	-6.41	-9.88	**65.91**
r13a501	48	-4.00	-7.69	**71.43**	r23a501	83	-2.35	-6.74	**56.60**
r13a502	51	-3.77	-5.56	**82.14**	r23a502	81	-7.95	-8.99	**68.75**
r13a700	72	-2.70	-1.37	**80.00**	r23a700	96	-4.00	-4.00	**52.38**
r13a701	72	-1.37	-4.00	**80.00**	r23a701	94	-6.00	-6.00	**36.23**
r13a702	69	-1.43	-2.82	**81.58**	r23a702	95	-5.00	-5.00	**43.94**
r14a200	20	**5.27**	-4.76	**81.82**	r24a200	35	**6.06**	-2.78	**118.75**
r14a201	20	**0.00**	**0.00**	**66.67**	r24a201	36	**5.88**	**2.86**	**100.00**
r14a202	20	**5.26**	**0.00**	**66.67**	r24a202	32	-3.03	-11.11	**68.42**
r14a300	24	**4.35**	-4.00	**50.00**	r24a300	47	**2.17**	**2.17**	**56.67**
r14a301	29	-3.34	-6.45	**31.82**	r24a301	53	-5.36	-5.36	**65.63**
r14a302	27	**3.85**	**3.85**	**80.00**	r24a302	50	**2.04**	**0.00**	**117.39**
r14a500	43	**4.88**	**2.38**	**95.45**	r24a500	70	-7.89	-7.89	**70.73**
r14a501	41	-2.38	-4.65	**28.13**	r24a501	73	-1.36	-5.19	**62.22**
r14a502	43	-2.27	-6.52	**72.00**	r24a502	76	-3.80	-6.17	**61.70**
r14a700	67	**3.08**	**0.00**	**81.08**	r24a700	94	-3.09	-5.05	**59.32**
r14a701	64	**1.59**	**0.00**	**77.78**	r24a701	92	-3.16	-6.12	**41.54**
r14a702	62	-4.61	-7.46	**63.16**	r24a702	89	-8.25	-10.10	**53.45**

Table 2. Performance of BiG-MRTA vs. ILS, eILS, and EDF over TAPTC problems. Note that the results of ILS and eILS has been taken from [33]. Both ILS and eILS are stochastic in nature and their performance is thus reported in terms of average values. Instances where BiG-MRTA performs better than a competing algorithm (ILD, eILS or EDF), the corresponding score-gap % is shown in **bold** font in this table.

as: **(i)** no provision for multi-tour planning, no depot location and different starting point for each robot; **ii)** robot ferry-range and payload constraints are not considered; **iii)** robots have heterogeneous capacity, namely different task achievement rates; and **iv)** tasks are associated with different workloads, i.e., the amount of time that a robot requires to execute it. In order to make our BiG-MRTA method applicable under these changed problem characteristics, we had to make minor modifications to the implementation of BiG-MRTA (its decision-making logic remained unchanged), which are further described in Appendix E. Note that, no tuning of the incentive model is performed in BiG-MRTA to heuristically improve its instance-specific performance on the TAPTC problems. For comparison with the ILS and eILS methods, we use their results reported in [33]. In that work, the heuristic parameters in ILS and eILS were specifically tuned for each group (Group 1 and Group 2). In addition, note that, both of these methods are stochastic, and were thus run 10 times and the average values of the performance were reported in [33]. To allow comparison with the EDF method [1], we implemented it in Python based on the heuristic described in [47].

6.2. *Results of Competitive Analysis on TAPTC Problems*

Table 2 summarizes the results of BiG-MRTA on the TAPTC problems, in terms of the completion rate (*score*), and its relative performance in comparison to ILS, eILS, and EDF expressed in terms of the *score gap*. This score gap represents the relative difference between the achieved completion rate of BiG-MRTA and that of ILS, eILS and EDF, with a *positive score gap* (shown in **bold font** in Table 2) indicating where BiG-MRTA outperforms the competing methods. It is observed from Table 2 that BiG-MRTA provides similar or better solutions than ILS, eILS, and EDF respectively for 60%, 52%, and 100% of Group 1 problems. For Group 2 problems, BiG-MRTA shows similar or better performance than ILS, eILS, and EDF respectively in 38%, 33%, and 94% of the instances. These results show that BiG-MRTA performs better on Group 1's instances, which have tasks with tighter time deadlines; this gain is likely attributed to the formulation of the incentive function of BiG-MRTA. Even for the instances where BiG-MRTA's performance falls behind that of ILS/eILS, the score gap remains generally within -10%. The overall completion rate performance across the benchmark problems thus demonstrates the applicability of BiG-MRTA to broader classes of SR-ST type MRTA problems, without requiring problem-specific tuning of the incentive model parameters.

In terms of computing time, EDF is found to be the fastest method with the average computing time of 4 milliseconds, which is about 70 times faster than BiG-MRTA (whose average computing time being in the $<$ 300ms level can however be considered to be adequate for most practical applications of online MRTA). A direct comparison between the computation costs of BiG-MRTA, ILS, and eILS is not feasible due to the unavailability of the original programmatic implementations of ILS and eILS. Thus we simply compare our computation time measurements with those reported in [33]. It was reported in [33] that the ILS and eILS methods were implemented in Java, and run on a Ubuntu 14.04 workstation with Intel i7 2.20 GHz and 8GB RAM to solve the TAPTC problems. For this competitive analysis, we run our BiG-MRTA's Python implementation on a Windows 10 workstation with Intel i5 2.30 GHz CPU and 16 GB RAM. Based on these settings and the reported computation times for ILS/eILS, we observed that BiG-MRTA is on average roughly 4 times faster than ILS and eILS over the 96 instances of the TAPTC problems. While a direct comparison was not feasible, these observations do point to the promising real-time performance of BiG-MRTA as an online MRTA method.

7. Conclusion

In this paper, we developed an efficient online method called BiG-MRTA for solving SR-ST type MRTA problems where tasks have deadlines, new tasks could appear during the mission, and robots are subject to range and payload constraints. BiG-MRTA uses a novel combination of a bipartite graph construction, an incentive model to assign edge weights in the bigraph, and maximum weighted matching over the bigraph to allocate tasks to robots. To benchmark the performance of BiG-MRTA, optimal solutions were generated by solving a special ILP formulation of this SR-ST problem, which allows multiple tours per robot. In order to evaluate the effectiveness of the new SR-ST methods, we considered the problem of multi-UAV response to flood victims, where performance is assessed in terms of task completion rate. For this application, BiG-MRTA was found to be $> 10^3$ times faster than the ILP approach, while its task completion rate was close to that of the optimal solutions provided by the ILP method. Significant performance superiority over a feasibility-preserving random-walk baseline was also demonstrated. By exploring up to huge 1000-task scenarios, BiG-MRTA was observed to provide remarkable scalability with increasing number of robots, while retaining its computational efficiency – cumulative computational costs of task selections per robot remained below 1 second.

In addition, we provided a competitive analysis of BiG-MRTA with respect to three existing methods, namely ILS, eILS, and EDF, over 96 SR-ST benchmark problems involving heterogeneous robot capacity and task execution effort. The results showed that BiG-MRTA provides competitive performance over these SR-ST problems (especially when tighter task-deadlines are involved), compared to ILS, eILS, and EDF, even though it is designed to tackle a slightly different class of SR-ST problems. This analysis therefore points to the wider applicability of the incentive-weighted bigraph concept in BiG-MRTA, and the need for future work on potential integration of such graph-based and meta-heuristic local search-based approaches to solve complex MRTA problems with varying characteristics. A particular limitation of the current online BiG-MRTA method is the myopic nature of selecting one task at a time, which promotes computational efficiency at a compromise in the optimality of decisions. A learning-based approach or an ILP decomposition approach (unprecedented when dynamic tasks are involved) can be explored to mitigate this limitation in the future.

Overall, the BiG-MRTA formulation provides a flexible efficient representation of the constrained SR-ST problem, which is expected to allow further extensions for wider applicability to multi-robot and multi-agent problems, and translation to physical implementations.

Acknowledgement

Support from the DARPA award HR00111920030 and the NSF award IIS-1927462 is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the DARPA or the NSF.

Appendix A. Integer Linear Programming Methods for MRTA

Table A.3 provides a characteristic comparison of our offline MRTA approach based on ILP, **CLIP-MRTA**, to a representative set of existing ILP-based implementations.

Appendix B. Feasibility-preserving Random-Walk MRTA

Algorithm 2 describes the Feas-RND approach, where each robot randomly chooses available tasks that are feasible to be undertaken by the UAV, in terms of criteria 1 and 2 stated in Section 3.3.

Algorithm 2 Feas-RND Algorithm

```
Input: $\mathcal{T}^k, \mathcal{S}^k$ - the recent states of active tasks and the robots, including robot-$r$ ($\mathcal{S}_r^k$).
Output: $\mathcal{A}_r^k$ - the next decision of robot-$r$ at its iteration $k$.
if robot-payload = 0 then
    $\mathbf{A}_r^k \leftarrow 0$                ▷ return to the depot
else if available-range − dist-from-Depot $\geq \epsilon$ then
    $\mathcal{T}_r^{k+1} \leftarrow$ GETFEASIBLETASK($\mathcal{T}^k, \mathcal{S}_r^k$)
    if $\mathcal{T}_r^{k+1} \neq \emptyset$ then
        $\mathbf{A}_r^k \leftarrow$ SELECT-RANDOMLY($\mathcal{T}_r^{k+1}$)
return $\mathbf{A}_r^k$
procedure GETFEASIBLETASK($\mathcal{T}, \mathcal{S}_r$)
    $\mathcal{T}_{\text{feasible}} \leftarrow \emptyset$
    for $i \in \mathcal{T}$ do
        $t_i^r \leftarrow$ global time that robot-$r$ finishes task-$i$
        $\Delta_r \leftarrow$ avail. range of robot-$r$ after doing task-$i$
        $w_{ri} \leftarrow$ using robots' incentive model, Eq.(11)
        if $t_i^r \leq \delta_i$ and $\epsilon \leq \Delta_r$ then
            $\mathcal{T}_{\text{feasible}} \leftarrow \mathcal{T}_{\text{feasible}} \cup \{\mathcal{T}_i, t_i^r, w_{ri}\}$
    return $\mathcal{T}_{\text{feasible}}$
```

Appendix C. Karp algorithm for Maximum Matching

Algorithm 3 shows a psuedo code of the Karp maximum matching algorithm that has been used in this work, which is based on [54].

Algorithm 3 Karp Maximum Weight Matching Algorithm [54]

```
for $x \in X$ do
    LIST($x$) $\leftarrow$ an array containing the set of elements $\{(x, y) | y \in Y\}$ in increasing order of $w(x, y)$
$M \leftarrow \emptyset$
for $v \in V$ do
    $c(v) \leftarrow 0$
while $|M| < |X|$ do
    PATHSET $\leftarrow \emptyset$
    $Q \leftarrow \emptyset$
    $S \leftarrow$ {free sources}
    for $y \in R$ do
        $\gamma(x) \leftarrow 0$
        $(x, y) \leftarrow$ first element of LIST($x$)
        $Q \leftarrow Q \cup ((x, y), \gamma(x) w^*(x, y))$
    while $R \cap$ free destination $= \emptyset$ do
        $y \leftarrow$ SELECT procedure
        if $y \notin R$ then
            PATHSET $\leftarrow$ PATHSET $\cup (x, y)$
            $R \leftarrow R \cup y$
            $\gamma(y) \leftarrow \gamma(x) + \bar{w}(x, y)$
            if $y$ is not free then
                $\{y, v\} \leftarrow$ the edge of $M$ incident with $y$
                PATHSET $\leftarrow$ PATHSET $\cup (y, v)$
                $R \leftarrow R \cup v$
                $\gamma(v) \leftarrow \gamma(y)$
                $(v, l) \leftarrow$ first element of LIST(v)
                $Q \leftarrow Q \cup ((v, l), \gamma(v) w^*(v, l))$
        for $v \notin R$ do
            $\gamma(v) \leftarrow \gamma(y)$
        for $v \in V$ do
            $\alpha(v) \leftarrow \alpha(v) + \gamma(v)$
        Let $\hat{P}$ be the unique directed path from a free source to $y$ whose edges are all in PATHSET
        Let $P$ be the set of edges in $G$ corresponding to directed edges in $\hat{P}$
        $M \leftarrow M \oplus P$
return $M$
```

Appendix D. Temporal Information of Dynamic Case

Figure D.10 shows the time period between the generation and expiry of the huge dynamic tasks, with the task deadlines being the same as that in the huge static case; deadlines are linked with rising localized flood levels given by the simulation.

Appendix E. Competitive Analysis: Description of TAPTC Problems and Implementation Changes

Appendix E.1. The TAPTC Benchmark Problems

TAPTC includes two 48-instance groups of problems. Each instance of the TAPTC benchmark includes 100 tasks and is assigned a name, as shown in Table 2. For example, the instance named as ”r*gdam*0*i*” depicts the following: 1) “*g*” indicates the

Table A.3. Comparison of features of CLIP-MRTA with other well-known centralized methods for solving "Single-task Robots Single-robot Tasks" problems.

Study	Objective Function	Task Deadline	Constrained Payload	Constrained Range
Our CLIP-MRTA	Task Completion Rate	✓	✓	✓
Kamra & Ayanian [25]	Time & Distance	✓	✓	-
Azi et al. [20]	Distance	✓	✓	-
Baldacci et al. [21]	Distance	✓	✓	-
Mingozzi et al. [22]	Distance	✓	✓	-
Jose & Kumar [24]	Distance	-	-	-

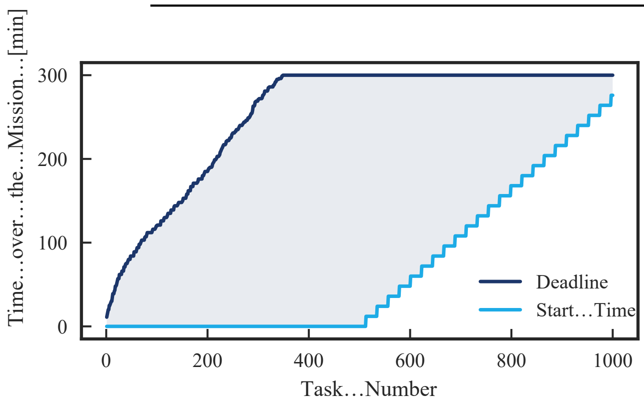


Fig. D.10. The beginning and deadline time of each task for the huge dynamic problem

group id ($g \in \{1,2\}$); 2) "d" indicates that the time deadlines of $d/4$ of tasks have been randomly drawn from $[\delta_{\text{low}}, \delta_{\text{high}}]$, where $d \in \{1,2,3,4\}$, and the time deadlines of the remaining tasks are set at δ_{high}; 3) "m" is the number of robots, where $m \in \{2,3,5,7\}$; and 4) Based on the above three features, three random instances have been generated that they are distinguished by the index "$i \in \{0,1,2\}$". A detailed explanation of the TAPTC benchmark problems can be found in [33].

Appendix E.2. Modifications to BiG-MRTA for Application on the TAPTC Problems

In order to apply our BiG-MRTA algorithm on the TAPTC benchmark problems with their original characteristics as assumed by the ILS, eILS, and EDF methods (and described in Section 6), we implemented the following modifications:

1. We set the maximum allowed payload (Q) and the ferry-range (Δ_{range}) of robots at very large values ($Q = 100$, $\Delta_{\text{range}} = 10^6$) making them practically unconstrained.
2. Since robots start from different locations in the TAPTC problems, we set $k = 1$ in Algorithm 1, to override the execution of line 11 of the algorithm (which otherwise seeks to efficiently allocate tasks based on the assumption of the same starting depot).
3. To address the heterogeneity of robots' capacity, i.e., differing task achievement rates (c_r), we replace Eq. (E.1) with Eq. (E.2). These equations essentially compute the total time required by any robot-r to reach and complete task-i, if selected, given the current location status of the robot.

$$t_i^r = \frac{d_{ri}}{V_r} \tag{E.1}$$

$$t_i^r = \frac{d_{ri}}{V_r} + \lceil \frac{w_i}{c_r} \rceil \tag{E.2}$$

Here V_r and w_i represent the speed of robot-r and the workload of task-i, respectively. Here, V_r is set at 1 m/s.

References

[1] B. P. Gerkey, M. J. Matarić, A formal analysis and taxonomy of task allocation in multi-robot systems, The International Journal of Robotics Research 23 (9) (2004) 939–954.

[2] K. Nagatani, Y. Okada, N. Tokunaga, S. Kiribayashi, K. Yoshida, K. Ohno, E. Takeuchi, S. Tadokoro, H. Akiyama, I. Noda, et al., Multi-robot exploration for search and rescue missions: A report on map building in robocuprescue 2009, Journal of Field Robotics 28 (3) (2011) 373–387.

[3] M. V. Espina, R. Grech, D. De Jager, P. Remagnino, L. Iocchi, L. Marchetti, D. Nardi, D. Monekosso, M. Nicolescu, C. King, Multi-robot teams for environmental monitoring, in: Innovations in Defence Support Systems–3, Springer, 2011, pp. 183–209.

[4] L. Collins, P. Ghassemi, E. T. Esfahani, D. Doermann, K. Dantu, S. Chowdhury, Scalable coverage path planning of multi-robot teams for monitoring non-convex areas, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021.

[5] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, B. Marinier, Progress toward multi-robot reconnaissance and the magic 2010 competition, Journal of Field Robotics 29 (5) (2012) 762–792.

[6] E. Nunes, M. Manner, H. Mitiche, M. Gini, A taxonomy for task allocation problems with temporal and ordering constraints, Robotics and Autonomous Systems 90 (2017) 55–70.

[7] L. Liu, D. A. Shell, Optimal market-based multi-robot task allocation via strategic pricing., in: Robotics: Science and Systems, Vol. 9, 2013, pp. 33–40.

[8] G. A. Korsah, A. Stentz, M. B. Dias, A comprehensive taxonomy for multi-robot task allocation, The International Journal of Robotics Research 32 (12) (2013) 1495–1512.

[9] P. Ghassemi, D. DePauw, S. Chowdhury, Decentralized dynamic task allocation in swarm robotic systems for disaster response, in: 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), IEEE, 2019, pp. 83–85.

[10] E. Nunes, M. Gini, Multi-robot auctions for allocation of tasks with temporal constraints, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2110–2116.

[11] L. Luo, N. Chakraborty, K. Sycara, Distributed algorithms for multirobot task assignment with task deadline constraints, IEEE Transactions on Automation Science and Engineering 12 (3) (2015) 876–888.

[12] D.-H. Lee, Resource-based task allocation for multi-robot systems, Robotics and Autonomous Systems 103 (2018) 151–161.

[13] X. Su, M. Zhang, Q. Bai, Coordination for dynamic weighted task allocation in disaster environments with time, space and communication

constraints, Journal of Parallel and Distributed Computing 97 (2016) 47–56.

[14] H.-L. Choi, L. Brunet, J. P. How, Consensus-based decentralized auctions for robust task allocation, IEEE transactions on robotics 25 (4) (2009) 912–926.

[15] A. Khamis, A. Hussein, A. Elmogy, Multi-robot task allocation: A review of the state-of-the-art, in: Cooperative Robots and Sensor Networks 2015, Springer, 2015, pp. 31–51.

[16] J. A. Svestka, V. E. Huckfeldt, Computational experience with an m-salesman traveling salesman algorithm, Management Science 19 (7) (1973) 790–799.

[17] G. B. Dantzig, J. H. Ramser, The truck dispatching problem, Management science 6 (1) (1959) 80–91.

[18] T. Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures, Omega 34 (3) (2006) 209–219.

[19] K. Braekers, K. Ramaekers, I. Van Nieuwenhuyse, The vehicle routing problem: State of the art classification and review, Computers & Industrial Engineering 99 (2016) 300–313.

[20] N. Azi, M. Gendreau, J.-Y. Potvin, An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles, European Journal of Operational Research 202 (3) (2010) 756–763.

[21] R. Baldacci, A. Mingozzi, R. Roberti, New route relaxation and pricing strategies for the vehicle routing problem, Operations research 59 (5) (2011) 1269–1283.

[22] A. Mingozzi, R. Roberti, P. Toth, An exact algorithm for the multitrip vehicle routing problem, INFORMS Journal on Computing 25 (2) (2013) 193–207.

[23] D. Wang, M. Hu, Y. Gao, Multi-criteria mission planning for a solar-powered multi-robot system, in: ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, 2018, p. V02AT03A026.

[24] K. Jose, D. K. Pratihar, Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods, Robotics and Autonomous Systems 80 (2016) 34–42.

[25] N. Kamra, N. Ayanian, A mixed integer programming model for timed deliveries in multirobot systems, in: 2015 IEEE International Conference on Automation Science and Engineering (CASE), IEEE, 2015, pp. 612–617.

[26] L. Liu, D. A. Shell, Physically routing robots in a multi-robot network: Flexibility through a three-dimensional matching graph, The International Journal of Robotics Research 32 (12) (2013) 1475–1494.

[27] P. Ghassemi, S. Chowdhury, Decentralized task allocation in multi-robot systems via bipartite graph matching augmented with fuzzy clustering, in: The ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2018), American Society of Mechanical Engineers (ASME), Quebec City, Canada, 2018, p. V02AT03A014.

[28] R. Nallusamy, K. Duraiswamy, R. Dhanalaksmi, P. Parthiban, Optimization of non-linear multiple traveling salesman problem using k-means clustering, shrink wrap algorithm and meta-heuristics, International Journal of Nonlinear Science 8 (4) (2009) 480–487.

[29] P. Toth, D. Vigo, Vehicle routing: problems, methods, and applications, SIAM, 2014.

[30] E. Schneider, E. I. Sklar, S. Parsons, A. T. Özgelen, Auction-based task allocation for multi-robot teams in dynamic environments, in: Conference Towards Autonomous Robotic Systems, Springer, 2015, pp. 246–257.

[31] D. Cattaruzza, N. Absi, D. Feillet, Vehicle routing problems with multiple trips, 4OR 14 (3) (2016) 223–259.

[32] C. Sarkar, H. S. Paul, A. Pal, A scalable multi-robot task allocation algorithm, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 5022–5027.

[33] H. Mitiche, D. Boughaci, M. Gini, Iterated local search for time-extended multi-robot task allocation with spatio-temporal and capacity constraints, Journal of Intelligent Systems 28 (2) (2019) 347–360.

[34] M. B. Dias, R. Zlot, N. Kalra, A. Stentz, Market-based multirobot coordination: A survey and analysis, Proceedings of the IEEE 94 (7) (2006) 1257–1270.

[35] Y. Shoham, K. Leyton-Brown, Multiagent systems: Algorithmic, game-theoretic, and logical foundations, Cambridge University Press, 2008.

[36] B. Dai, H. Chen, A multi-agent and auction-based framework and approach for carrier collaboration, Logistics Research 3 (2-3) (2011) 101–120.

[37] L. Johnson, S. Ponda, H.-L. Choi, J. How, Asynchronous decentralized task allocation for dynamic environments, in: Infotech@Aerospace 2011, American Institute of Aeronautics and Astronautics, 2011, p. 1441.

[38] M. Otte, M. J. Kuhlman, D. Sofge, Auctions for multi-robot task allocation in communication limited environments, Autonomous Robots 44 (3) (2020) 547–584.

[39] N. Buckman, H.-L. Choi, J. P. How, Partial replanning for decentralized dynamic task allocation, in: AIAA Scitech 2019 Forum, 2019, p. 0915.

[40] S. Ismail, L. Sun, Decentralized hungarian-based approach for fast and scalable task allocation, in: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2017, pp. 23–28.

[41] J. Han, Y. Xu, L. Di, Y. Chen, Low-cost multi-UAV technologies for contour mapping of nuclear radiation field, Journal of Intelligent & Robotic Systems (2013) 1–10.

[42] H. H. Choi, S. H. Nam, T. Shon, Two tier search scheme using micro UAV swarm, Wireless Personal Communications 93 (2) (2017) 349–363.

[43] P. Ghassemi, S. Chowdhury, An extended bayesian optimization approach to decentralized swarm robotic search, Journal of Computing and Information Science in Engineering 20 (5) (2020) 051003.

[44] P. Odonkor, Z. Ball, S. Chowdhury, Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping, Swarm and Evolutionary Computation 46 (2019) 52–68.

[45] M. Abdelkader, M. Shaqura, M. Ghommem, N. Collier, V. Calo, C. Claudel, Optimal multi-agent path planning for fast inverse modeling in UAV-based flood sensing applications, in: Unmanned Aircraft Systems (ICUAS), 2014 International Conference on, IEEE, 2014, pp. 64–71.

[46] P. Vansteenwegen, W. Souffriau, G. V. Berghe, D. Van Oudheusden, Iterated local search for the team orienteering problem with time windows, Computers & Operations Research 36 (12) (2009) 3281–3290.

[47] H. Mitiche, D. Boughaci, M. Gini, Efficient heuristics for a time-extended multi-robot task allocation problem, in: 2015 First International Conference on New Technologies of Information and Communication (NTIC), 2015, pp. 1–6. doi:10.1109/NTIC.2015.7368756.

[48] H. C. Lau, M. Sim, K. M. Teo, Vehicle routing problem with time windows and a limited number of vehicles, European journal of operational research 148 (3) (2003) 559–569.

[49] S. Hasgül, I. Saricicek, M. Ozkan, O. Parlaktuna, Project-oriented task scheduling for mobile robot team, Journal of Intelligent Manufacturing 20 (2) (2009) 151.

[50] Z. Huang, D. D. Zeng, H. Chen, Analyzing consumer-product graphs: Empirical findings and applications in recommender systems, Management science 53 (7) (2007) 1146–1164.

[51] Z. Zhu, J. Su, L. Kong, Measuring influence in online social network based on the user-content bipartite graph, Computers in Human Behavior 52 (2015) 184–189.

[52] A. S. Asratian, T. M. Denley, R. Häggkvist, Bipartite graphs and their applications, Vol. 131, Cambridge University Press, 1998.

[53] D. B. West, et al., Introduction to graph theory, Vol. 2, Prentice hall Upper Saddle River, 2001.

[54] R. M. Karp, An algorithm to solve the m× n assignment problem in expected time o (mn log n), Networks 10 (2) (1980) 143–152.

[55] R. W. Beard, T. W. McLain, Multiple uav cooperative search under collision avoidance and limited range communication constraints, in: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, Vol. 1, IEEE, 2003, pp. 25–30.

[56] Datasheet, Digi xbee-pro 900hp, Digi International Inc.

[57] M. R. Silva, E. S. Souza, P. J. Alsina, D. L. Leite, M. R. Morais, D. S. Pereira, L. B. Nascimento, A. A. Medeiros, F. H. C. Junior, M. B. Nogueira, et al., Performance evaluation of multi-uav network applied to scanning rocket impact area, Sensors 19 (22) (2019) 4895.

[58] S. Cullinane, Hurricane lane dumped 52 inches of rain on hawaii and there might be more on the way (Aug 2018).

[59] Geomorphological research group.
URL http://gis.ess.washington.edu/grg/index.html

[60] S. N. Jonkman, B. Maaskant, E. Boyd, M. L. Levitan, Loss of life caused by the flooding of new orleans after hurricane katrina: analysis of the relationship between flood characteristics and mortality, Risk analysis 29 (5) (2009) 676–698.

[61] H. Lamb, K. Frydendahl, Historic Storms of the North Sea, British Isles

and Northwest Europe, Cambridge University Press, 1991.

[62] State of hawaii, department of land and natural resources, flood hazard assessment tool.
URL http://gis.hawaiinfip.org/fhat/

[63] A. S. Saeed, A. B. Younes, C. Cai, G. Cai, A survey of hybrid unmanned aerial vehicles, Progress in Aerospace Sciences.

[64] I. Gurobi Optimization, Gurobi optimizer reference manual (2016).