A Unified Framework for Reliable Multi-Drone Tasking in Emergency Response Missions

Maria Terzi, Panayiotis Kolios, Christos Panayiotou, Theocharis Theocharides

***Abstract*— In this paper a unified framework is presented for coordinated multi-drone tasking in emergency response missions. As elaborated hereafter, response missions consist of a number of distinct tasks that can be assigned among the available agents to expedite the response operations. The proposed framework enables the development and execution of algorithms that jointly schedule and route drone agents across the field to complete their tasks and successfully address the mission goals considering the agent limitations. The key design challenges of implementing the proposed framework are discussed. Finally, initial simulation and experimental results are presented providing evidence of the real life applicability and reliability of the proposed framework.**

***Index terms*—** Multi-agent tasking, mathematical optimisation, constraint programming, emergency response, UAS swarm test-bed

I. Introduction

Unmanned Aerial Vehicles (UAVs) or simply drones, have been deployed in several emergency response missions in the recent past including firefighting incidents [1], weather and hurricane monitoring [2], [3]. However, their current use is mainly focused in providing a "bird's eye view" of the situation using camera payloads. Decisions are taken solely based on the human operator, who analyses and gives an expert opinion based on the images received.

Evidently, the rapid advances in drone technologies have greatly improved their capacity to aid first responders in a more substantial way. Take for instance an emergency evacuation mission where the safe and timely movement of people away from threats and hazards, is critical. To achieve this mission's objectives, the whole process is split into a sequence of phases (including detection, decision, alarm and people movement) in which specific operations take place. The better these phases are interlinked with each other (through mutual visibility, timely information exchange, and early situational awareness) the better they can be planned and interweaved, which eventually leads to better response efficiency and significantly reduced evacuation times.

Under this setting, the mission of one or more UAVs is to assist first responders by conducting, among others, the following important tasks: (i) provide fast situational assessment, (ii) guide teams on-site safely by detecting and avoiding hazards, (iii) search, detect and track victims/survivors, and (iv) provide a temporary infrastructure network (e.g., supporting distribution of urgent supplies, radio connectivity).

In this work we build a mathematical framework that can model the aforementioned tasks in a unified way. Using this framework, we then build joint scheduling and routing algorithms that are used to allocated and execute tasks considering the available drone agents and their capabilities. The proposed framework and the developed algorithms are then implemented as a robust and scalable ground control platform that can be used by a single operator to monitor and control a fleet of drone agents to conduct their mission.

The rest of the paper is split in the following sections. Related work is included in Section II while Section III elaborates on the proposed framework and derives the multi-agent tasking algorithms. In the later case, we first formulate an optimal mathematical programme for the coordinated multi-tasking problem that arises to demonstrate the complexity of the envisioned multi-agent tasking problem. We then provide heuristic approximations to solve the problem that arises in real-time. In Section IV we detail on the system architecture of the ground control platform and Section V-B provides experimental results with a fleet of 4 UAVs to demonstrate the applicability and reliability of the architecture and the algorithms developed.

II. Background and Related work

Mobile autonomous agents are becoming increasingly capable of achieving complex missions mainly due to miniaturisation and improvements of electronic components (including processors, sensors, actuators, batteries and communications circuitry). Existing works have looked into the challenges of tasking individual UAVs to accomplish autonomous functionalities in diverse applications ranging from infrastructure inspections, logistics, security, as well as emergency response [4].

The coverage problem is among the most cited problems to tackle whereby a fleet of agents need to spread across an area over the shortest time interval to obtain situational awareness or when searching for particular objects. The literature in this domain deals with task assignment, scheduling and path planning of the multiple agents, considering physical resource constraints such as the total number of agents, their communication ranges [5], [6] and their battery levels. Rescheduling methods have been introduced to ensure the safe return of a drones [7] and the trade-off between mission performance and battery consumption has been thoroughly investigated [8]. Solutions consider variants of the travelling salesman (TSP) and vehicle routing problems (VRP) [9],

The authors are with the KIOS Research and Innovation Centre of Excellence (KIOS CoE) and the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus. e-mail: {terzi.maria, pkolios, ttheocharides, christosp, mpolycar}@ucy.ac.cy

[10], [11], while heuristic [12] and meta-heuristic algorithms [13] have been proposed to solve practical instances of the problems with short computation times. To deal with uncertainty in the model, a number of studies have looked into stochastic modelling formulations with partially observable Markov decision processes being the most prominent example, [14]. Distribution of urgent supplies has also been considered in [15], with the objective of minimizing delivery times or monetary cost subject to delivery time deadlines.

While the majority of existing literature focuses on addressing the issues for one of the aforementioned tasks, in real emergency scenarios, several of the aforementioned tasks arise concurrently, and thus need to be dealt with jointly. Take for example a plausible deployment of a fleet of UAV in a search and rescue mission where the task is to search and track survivors out of shipwrecks when migrants attempt treacherous boat crossings. In such a scenario, teams of agents unlock significantly greater capabilities than what is possible by individual agents (due, for example, to the limited field of view of single agents and their endurance), but coordination is necessary for effective search and track [16]. This coordination problem is particularly challenging since it involves, in many cases, tasks with competing objectives, and agents with heterogeneous capabilities, as emphasized in [17].

The basic problem of task allocation consists of a set of tasks, a set of agents and a set of utilities of the agents to execute the tasks. The objective is to find the allocation of the agents that maximises the utility of the multi-agent system considering application-specific constraints such as resource availability and the individual capabilities of possibly heterogeneous agents [18], [19], [20]. The work in [18] discusses a deterministic, centralised task allocation algorithm for a fleet of UAV using the "stable marriage problem"; a problem of finding a stable matching between two equally sized sets of elements given an ordering of preferences for each element. Embracing the underlying uncertainties, the work in [19] employs a genetic-algorithm to solve the complex multi-object problem considering a fleet of UAV. The work in [20] also considers a genetic-algorithm approach to address specific task requirements and the heterogeneity in the fleet.

Recent work [21] provided initial investigations to the use of fully distributed control algorithms in a UAV swarm, taking into account the battery of the UAVs [22], [23]. The efficiency of the algorithms and frameworks proposed are tested in simulations. We argue on the need of providing evidence of the applicability of the suggested algorithms and frameworks in a real work setting.

Importantly, the lack of realism in the underlying assumptions of the aforementioned works limits their practicality. Firstly, UAVs operate in resource restricted environments with limited communication and computation resources.Secondly, the computational complexity of the aforementioned methods may pose a challenge for real-world applications. Thirdly, an effective approach for task allocation should definitely take into account the uncertain and dynamic environment, the varying capabilities of the UAV swarm and their different attributes such as the flight speed, fuel consumption and specific abilities.

A key contribution of our work is that we develop a unified framework that addresses all three aforementioned challenges by considering the practical limitations of using off-the-shelf hardware (including UAVs and mobile computing units) and the contraints posed by the uncertain environment. In addition to the novel mathematical modelling of the unified framework that encompasses these challenges, we also implement the proposed framework as part of a ground control platform that is then used to conduct a detailed experimental evaluation to demonstrate the applicability and effectiveness of the proposed solution.

III. Proposed framework and tasking algorithms

In a nutshell, the objective of our multi-drone tasking system is to schedule and route multiple UAVs taking into consideration that:

- a UAV must have the equipment required to execute a particular task,
- the total fly time of a UAV never exceeds its maximum fly time supported by the available battery capacity, and
- the total duration of the mission is minimised.

Moreover, each task has the following characteristics:

- Consists of a set of locations where the task should be performed.
- Requires the UAV to stay over each location for a particular duration of time. We define this requirement as the demand of the task.
- It has an order. For example, searching precedes tracking and distribution of supplies.

For each UAV we make the following assumptions:

- It has a defined maximum flight time.
- It has a battery consumption per minute ratio.
- Should take-off and land at the depot.
- Travels at a constant predefined speed.

Given the set of tasks $t \in \mathscr{T}$ and the set of available UAV agents $k \in \mathscr{K}$, the problem that arises is how to assign those tasks to agents and which routes should be followed by each agent, in order to complete all tasks using the least number of resources. To address this problem we consider a variation of the Vehicle Routing Problem (VRP) where routes are computed on a graph of nodes and edges, and task requirements are modelled as costs on the edges and as demands on nodes visited. Mathematically, the problem can be defined as follows: Given K UAVs initially placed at some depot s, find closed walks that visit nodes in the network with some demand b^t, $t \in \mathscr{T}$ and return back to s with the smallest cost (i.e., consuming the least battery consumption).

To formulated this problem, the affected area is first modelled as a fully connected weighted graph G where edge costs c_{ij} represent fly times for traversing edge $i \mapsto j$, and a linear integer program is derived on G to construct paths that will cover the field without exceeding fly time constrains.

Let $G = (\mathscr{N}, \mathscr{E})$ be a quantized version of the field where $\mathscr{N} = \{1, 2, \ldots, N\}$ is the set of nodes (i.e., discrete location) and $\mathscr{E}$ is the set of edges. By moving through edge $(i, j) \in \mathscr{E}$, a UAV spends $c_{ij} \geq 0$ in flight time (and consumes the relevant battery resources). We assume that we also have $(j, i) \in A$ with possibly different cost c_{ji}. Each agent is initially located at a source node $s \in \mathscr{N}$ and its initial flight time availability is $B(k), k \in \mathscr{K}, k = 1, \ldots, K$. We would like to find a route for each k that starts at the source node s visits some other nodes in N with specific task demand $b_i^t \leq 0, i \in \mathscr{N}, t \in \mathscr{T}$ and returns to s within its total flight time availability. To allow for recharging of the agents, we assume that the source s has a finite resupply capability where agents can wait for their batteries to replenish. To model this on the graph, we assume that there is an edge $s \mapsto s$ with cost $c_{ss} > 0$ and positive supply $b_i^k > 0$. Agents using these specific edges have a traversal cost that reflects the waiting that takes places in recharging while the positive supply $b_i > 0$ adds to their available fly time to continue their routes.

The following linear integer problem, $(P1)$, has the objective of minimizing the total fly time cost under the limited fly time duration of each agent and the demands set by the set of tasks.

$$(\text{P1}) \min \sum_{k=1}^{K} \sum_{(i,j)\in\mathscr{E}} c_{ij}^{kt} x_{ij}^{kt} \tag{1}$$

$$\text{s.t.} \sum_{t=1}^{T} \sum_{(i,j)\in\mathscr{E}} c_{ij}^{kt} x_{ij}^{kt} \leq B(k) + \sum_{k=1}^{K} b_i^{kt} \sum_{j:(i,j)\in\mathscr{E}} x_{ji}^{kt} \, \forall k \in \mathscr{K} \tag{2}$$

$$\sum_{k=1}^{K} \sum_{j:(i,j)\in\mathscr{E}} x_{ij}^{k} \geq \begin{cases} 0, & b_i^{kt} = 0 \\ 1, & b_i^{kt} > 0, \end{cases} \quad \forall i \in \mathscr{N} \tag{3}$$

$$\sum_{j:(i,j)\in\mathscr{E}} x_{ij}^{kt} - \sum_{j:(j,i)\in\mathscr{E}} x_{ji}^{kt} = 0 \, \forall k \in \mathscr{K}, t \in \mathscr{T}, i \in \mathscr{N} \tag{4}$$

$$\sum_{t=1}^{T} \sum_{j:(s,j)\in\mathscr{E}} x_{sj}^{kt} = 1 \, \forall k \in \mathscr{K} \tag{5}$$

$$M \sum_{(i,j)\in A(Q)} x_{ij}^{k} \geq \sum_{(i,j)\in A'(Q)} x_{ij}^{k} \, \forall Q \subset N, Q \neq \emptyset, k \in \mathscr{K} \tag{6}$$

$$0 \leq x_{ij}^{kt} \leq u_{ij}^{kt}, \; x_{ij}^{kt} \in Z \tag{7}$$

In $(P1)$, integer variable x_{ij}^{kt} denotes a route legs for UAV k indicating the number of times that agent traverses edge (i, j). The objective function in (1) minimizes the total fly time cost for all agents to complete their routes and respective tasks. Constraints (2) ensure that all computes paths adhere to the fly time limitations of the available capacity $B(k)$ and the replenishing that may have occurred at the source. Constraints (3) ensure that all nodes with non-zero task demands are visited by at least a single agent. Note that through this constraint we allow each agent to visit a node more than once if that is necessary to minimize the total route cost. The conservation of flow constraints at each node i are given in constraint eq. (4). In order to ensure that the computed route for each UAV will be a connected cycle that leaves s and returns to s we use equation (5) and (6). While the former constraint simply ensures that any path starts from s, the latter constraint is novel with respect to the basic subtour elimination constraints (based on the Miller-Tuck-Zemlin) that are used in the travelling salesman and vehicle routing problems.

The MTZ subtour elimination constraints are used to find a minimum-cost Hamiltonian tour (a tour that visits all the nodes exactly once). These constraints eliminate disconnected subtours with the only exception of any cycle formed with s included. Of course this is not necessary in the problem considered here and thus the alternative subtour elimination constraints that we present in (6) are used. In these constraints, for any subset $Q \subset N$, we define $\mathscr{E}(Q)$ to be the set of edges with only one end in Q and let $\mathscr{E}'(Q)$ to be the set of arcs with both ends in Q. The right hand side of the constraint is the total flow in Q with respect to UAV k and the sum on the left hand side is the total flow in and out of Q. Thus if there is some flow in Q with respect to k then the right hand side is positive which forces the left hand side to be positive which means that the flow in Q is connected to nodes outside of Q. This eliminates disconnected flow cycles but does not eliminate flow cycles that are connected to the source.

When solved to optimality $(P1)$ computes the best alternative routes that complete the set tasks at the minimum cost. However due to the exponential number of constraints in eq. (6), $(P1)$ is computationally hard to solve in practice. Hence, in the next section we seek to device an alternative heuristic solution that can solve instances of $(P1)$ efficiently and be used in practice to construct and update routes in real-time.

A. *Multi-Drone Tasking Algorithm*

To develop an online multi-drone tasking algorithm we consider: a) the demand for the requested tasks and b) the capacity of each drone to complete that task. The algorithm accepts as input the number of UAVs, the number of tasks, the maximum total mission time if there is one, the location of the depot, the speed of the UAV, and the recharging time required to replenish battery levels by some amount. In the proposed algorithm, detailed in Alg. 1, each UAV k and a fly time capacity (associate with its battery level) $B(k)$. Each task has a demand $b_i^t, t \in \mathscr{T}$ at location $i \in \mathscr{N}$ from a set of locations that must be visited, an order number, and an indicator demonstrating if the task can be undertaken by multiple UAVs simultaneously. Considering the tasks in reverse order, the algorithm will first check if the task can be executed by a single or multiple units. If a feasible assignment is found (i.e., the UAVs have enough battery to undertake their allocated tasks) then the algorithm proceeds with updating the constraints of the problem including the remaining maximum fly time of a UAV for the next task to be considered. If the solution identified is not feasible, the algorithm will check if recharging is possible and add a recharging slack to the problem. If no recharging is available, then the algorithm will exit with no solution. If all the tasks are executed successfully then the algorithm will output the vehicle routing paths and terminate.

Algorithm 1 Multi-Drone Tasking algorithm

Require: $B(k), b_i^t, c_{ij}^{kt}, \forall k, t, i, j$

Sort tasks based on reverse order sequence
top:
if $all\_tasks\_executed$ **then**
  $output\_drone\_routes()$ **return**;
else
  top1:
  $feasible \leftarrow Alg.2$
  if $feasible$ **then**
    $update\_constrains()$
    $mark\_task\_as\_executed()$
    goto top
  else
    if $recharge$ **then**
      $add\_recharging\_slack()$
      goto top1
    else
      No solution found **return**;

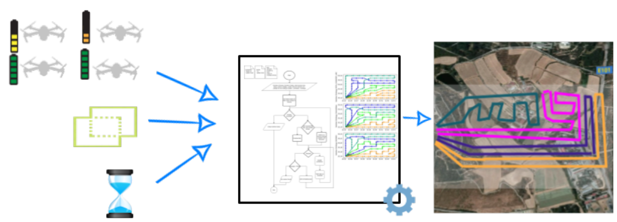

Fig. 1. Task allocation taking into consideration the field parameters, the drones constraints and the task requirements.

To compute solutions, Contraint Programming (CP) is employed through the OR-tools algorithmic toolkit [24]. OR-tools [24] is a highly optimised and sophisticated operations-research open source library provided by Google. OR-Tools provides a routing module which is capable of solving instances of the VRP using a plethora of different heuristics. In our case, initial solutions are being generated using three popular solution strategies: 1) Path cheapest arc, 2) Christophides [25], and 3) Savings [26]. Based on the minimum objective value, the platform chooses the most efficient solution, i.e., the solution that has the minimum total total flight time for all participating UAVs.

Note that to accommodate for the multi-drone tasking, we extend the standard VRP heuristic approach used by the CP solver by adding constrains on the flight time of each drone. To explicitly demonstrate the applicability of these constrains, consider as an example the case of collaboratively mapping the field by a fleet of agents. Note that in this case, the number of node in $\mathscr{N}$ representing the field is governed by the flying altitude, the onboard camera resolution, and the the percentage of overlap between consecutive images [1]. Alg. 2 elaborates of the steps taken.

[1]https://www.melown.com/products/photogrammetry/flight-calculator/

Algorithm 2 CP with collaborative agents

Require: $G(\mathscr{N}, \mathscr{E}), B(k), b_i^t, c_{ij}^{kt}, \forall k, t, i, j$

if $\sum_{i,t} b_i^t > \sum_k B(k)$ **then return**;
else
  $start\_fn \leftarrow sameStartFinish{=}True$
  $model\_parameters \leftarrow default\ parameters$
  $parameters \leftarrow default\ search\ parameters$
  $parameters.set\_fist\_solution\_strategy()$
  $parameters.time\_limit\_ms \leftarrow limit\ to\ three\ seconds$
  $routing.SetArcCostEvaluatorOfAllVehicles \leftarrow dist\_fn$
  $routing.addDimension \leftarrow total\ time$
  $routing.addDimension \leftarrow flight\ time\ of\ each\ drone$
  $timeDimension.setUpperboundforVehicle \leftarrow max\ flight\ time\ for\ specific\ drone$
  $routing \leftarrow CP(G(\mathscr{N}, \mathscr{E}), B(k), b_i^t, c_{ij}^{kt}, parameters)$
  $assignment \leftarrow\ routing.SolveWithParameters()$
  return assignment;

Algorithm 3 Flight time calculation from location a to b

$speed\_mph \leftarrow The\ speed\ of\ drones\ \text{-}\ 30mph$
$distance[a][b] \leftarrow distance\ between\ a\ and\ b$
$speed\_mpm \leftarrow speed\_mph/60$
$flight\_time \leftarrow distance/speed\_mpm$
return flight_time; =0

IV. Ground Control Platform Implementation

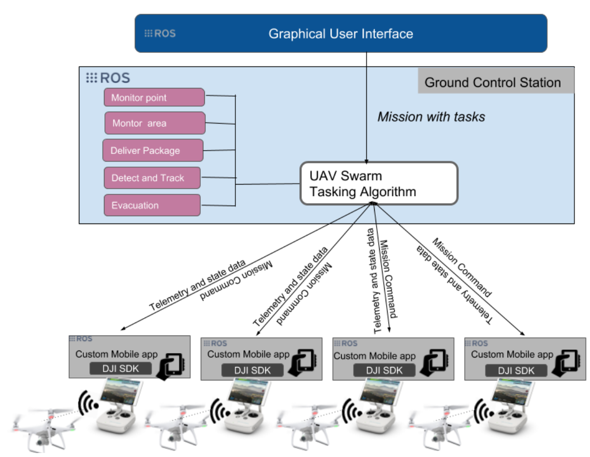

Fig. 2. GCP system architecture

Both the unified framework as well as the developed algorithms were implemented as a Ground Control Platform (GCP): a platform designed to assist first responders in their operations by providing planning and deployment capabilities of a fleet of UAVs. Such capabilities include task allocation to individual UAVs of the swarm such as monitoring emergency event progress, identifying stranded survivors and path planning for each UAV.

GCP consist of three essential components: a central ground control station (GCS), a user interface and UAV

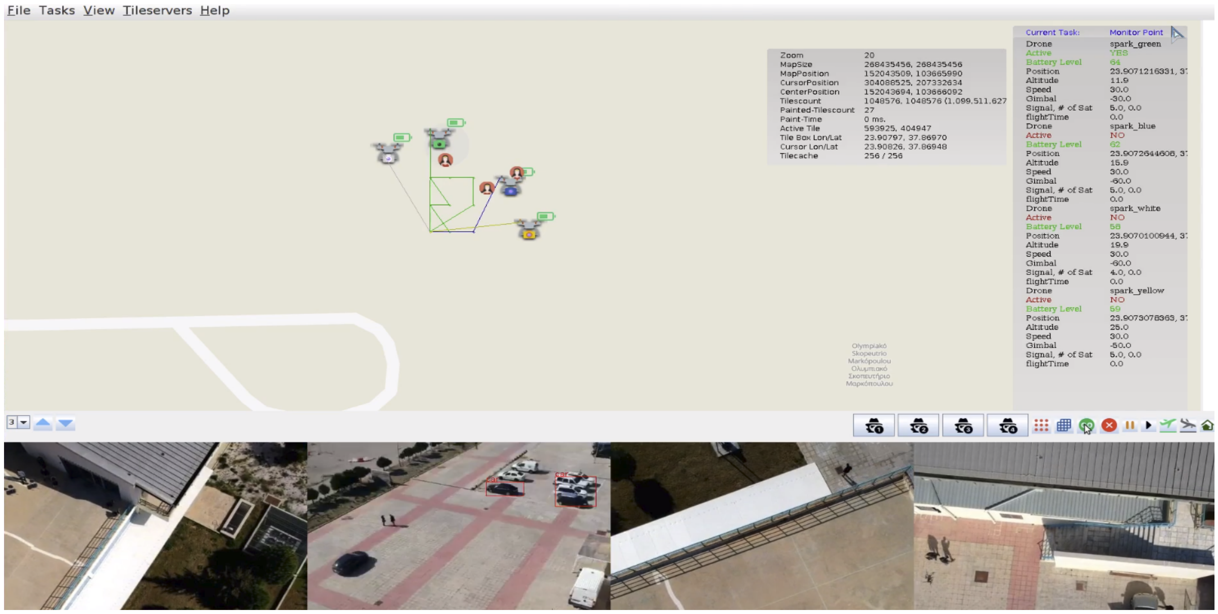

Fig. 3. The user interface of the Ground control platform

interfaces. The GCS receives telemetry data from all the UAV in the swarm and it sends commands back to each UAV. The GCP as shown in Figure 2 accepts tasks from the user interface, the tasking algorithms then produce mission commands for each of the UAVs to achieve those tasks considering the real-time telemetry and status data from the UAVs. The Tasking algorithms component is developed using OR-Tools library and the communication between the User interface, the GCS and the UAVs is established via the ROS library [27]. The platform has been tested using DJI drones and interfaces used to enable the communication with the drones were implemented based on the SDK provided by DJI [28].

The user interface of the platform, shown in Figure 3 allows the operator to design a mission consisting of one or more tasks. The platform consist of a top menu, a map, info panels, command buttons and the live feeds of the cameras. The top menu includes between others the available tasks such as the monitor area, monitor point, deliver package. The main map shows the location of each drone and the current battery level of the drone. In addition info panels are provided presenting important information regarding the state of the drones as well as information regarding the area. At the button, quick access command buttons are provided to enable the detection of people and vehicles on the drones camera footage, to monitor an area, to synchronised take off, land and return home. The implemented GCP was used to check the applicability and reliability of the proposed framework. The next section presents results from simulation and real life experiments.

V. Simulation and Experimental Results

The applicability and reliability of the proposed framework has been tested both in simulations and experiments outside the lab. In this section the results of a specific emergency response mission scenario are presented. In the selected use case, four UAVs are asked to monitor an area to detect survivors and upon detecting a survivor, one of the agents must presumably deliver a first-aid kit to one of the survivors.

A. Simulations

As discussed in Section III, the multi-tasking algorithm considers tasks in reverse order sequence. Therefore, in the mission scenario discussed above the algorithm first calculates the battery requirements for the delivery task and

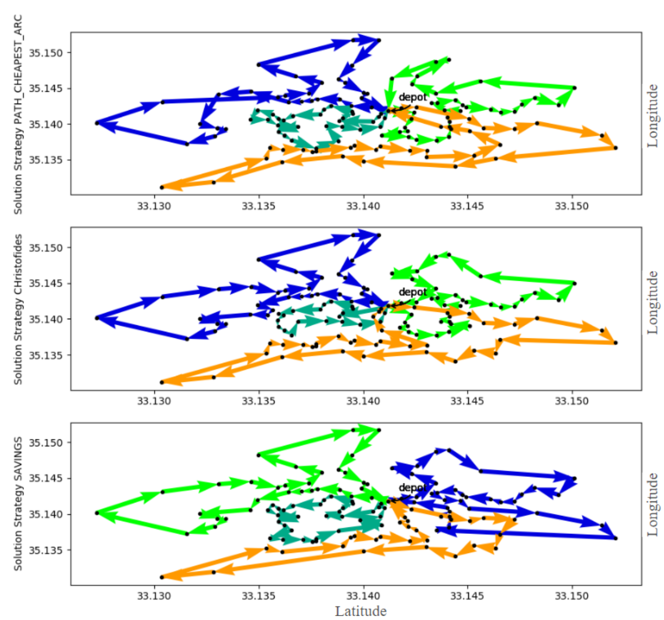

Fig. 4. Comparison of routes produced for each drone produced by Alg. 1 for a fleet of UAV using the Path Cheapest Arc, Christophides and Savings heuristics.

then uses this requirement as a constrained to searching the field.

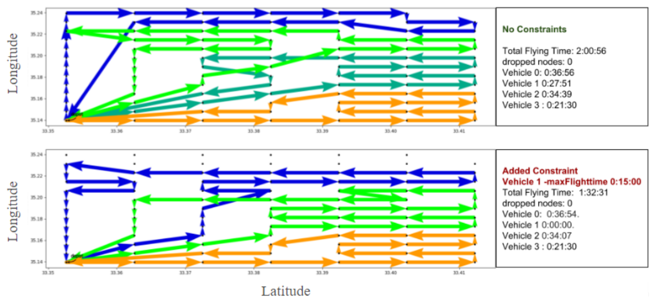

Fig. 5. Computed mapping routes with and without the delivery task constrain.

For this experiment, an operation area of 1 squared kilometer is considered, around a depot point s with latitude 33.1395926 and longitude 33.526203. As indicated above, four UAVs have been considered having a constant speed speed of 61Kmph when in motion, with maximum flight time of 38 minutes and with altitude between 15 and 25 meters (i.e., space separation is used between UAVs to ensure collision-free operation). The UAVs are asked to stay on top of each monitoring point for 3sec. In total, 127 points were normally distributed within the area with standard deviation of $\pm 3m$ and a random point within the area is selected for the delivery task.

The multi-drone tasking algorithm presented in this paper follows a bottom-up approach by adding constrained to the top tasks based on the resources required for full-filing the last tasks. Therefore, in this scenario, the algorithm first estimates the time required for the delivery task. This estimation is currently set to the maximum required time to visit the most distant point and return to the deport. Further work will investigate methodologies to predict the location of the potential delivery point such as by introducing additional knowledge into the system or exploring machine learning techniques.

As discussed, OR-tools enables the use and evaluation of a plethora of solution strategies. To select a solution strategy, the platform first produces solutions with three popular solution strategies: 1) Path cheapest arc, 2) Christophides [25], and 3) Savings [26]. An illustration of the solutions produced for this specific task is presented in 4. Based on the minimum objective value, the platform chooses the most efficient solution, i.e., the solution that has the minimum total total flight time for all participating UAVs.

The time required to complete the delivery task is then added as a constrained to the searching task as exemplified in Alg. 2. Figure 5 presents the output of the CP with collaborative agents algorithm using the flight time constrain. As shown, some points are left behind since the not all tasks can be completed given the UAV resource limitations and the constrains set. While these simulations demonstrate the applicability of the proposed solution, real experiments are required to provide a realistic performance comparison as we demonstrate in the next section.

B. *Experimental Results*

The proposed algorithms and framework were evaluated in a real world setting. In this experiment the drones are instructed to collaborate in monitoring an area. The experiment evaluates the applicability and the reliability of the platform. The applicability is evaluated in terms of successfully applying the proposed platform, and algorithms in a real world setting. The reliability of the platform is evaluated in terms of accurately predicting the time required by a drone to complete a given mission. For the experiment, we used four DJI Spark drones. DJI Spark are 400g drones equipped with GPS satellite positioning system and a 12 MP camera. They offer a flight time of 15 minutes, operating at a maximum altitude of 30m and have a maximum transition distance of 2Km.

We investigated the performance of the tasking algorithms in a grid of a 165 square meters field with no obstacles. To avoid any collisions between the drones during operation, we preset the altitudes of the four drones to 15m, 18m, 21m, 25m. UAVs were then set to fly on a constant speed of 30mph. Additionally, we set a demand (b) of three seconds to each location - meaning that the UAV should stay for three seconds on top of each location and take a photo.

Flight time constrains were added for all UAVs in the CP solver.Their maximum flight time for this scenario was set to six minutes. Hence the algorithm considered only solutions that were within that time span. The drones were commanded to complete the paths produced by the CP algorithm, presented in Figure 6. As shown, Drone A was instructed to visit only five locations, Drone B and Drone C to visit 30 locations each, and Drone D to visit 31 locations - totalling to 97 locations. During the experiment we logged the state of the drones including their battery level, latitude and longitude, and the time they reached each of their allocated locations. All the raw data of the experiments,

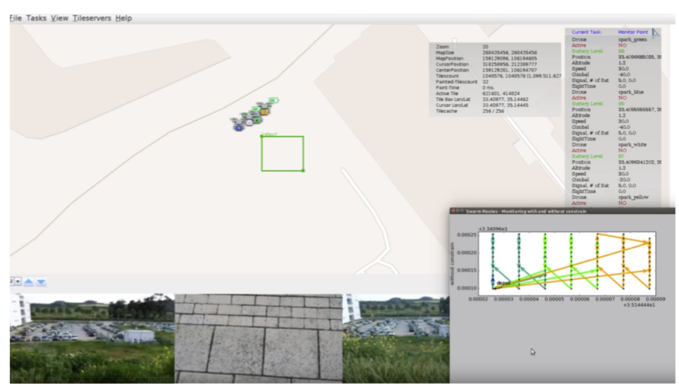

Fig. 6. Screen capture during the experiment showing the paths allocated by the Flight constrained VRP algorithm for UAV swarm area monitoring

including the output of the algorithms, the photos captured by the drones as well as the screen recording of the platform and additional video from the experiment are publicly available[].

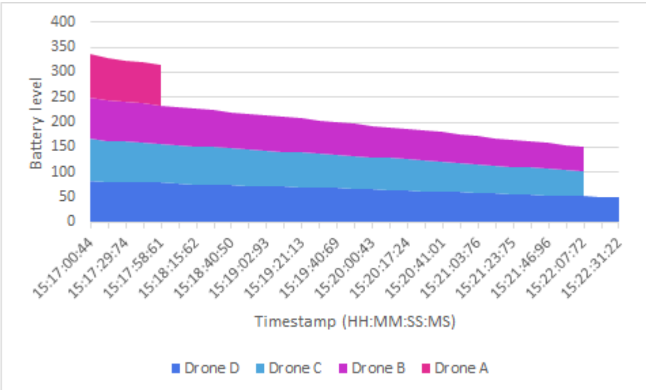

Fig. 7. Battery level of each drone during mission

Figure 7 shows the battery consumption of each UAV and the respective area covered during the mission. As shown, Drone A consumed the least battery, followed by Drone B, Drone D and then Drone C. In total, Drone A consumed 6% of its battery during the misison, Drone B consumed 36%, and Drone C and Drone D consumed 32% and 34% respectively. The low battery consumption for Drone A was expected since Drone A was allocated only five locations. The graph also illustrates that there was a linear use of battery over time by all drones and there were no unexpected behaviour during the flight. Figure 8 shows the area covered

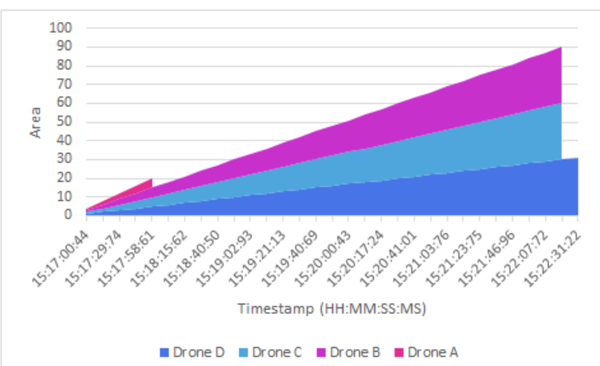

Fig. 8. Area covered by each drone during mission

by each UAV during the mission. The area covered by each UAV is calculated by dividing the number of the locations visited by the the number of locations allocated by the algorithm during the mission. We can observe that Drone A finished first the mission followed by Drone B and Drone C and then Drone D, which was also expected since all the drones were instructed to fly with the same speed (that is 30mph.)

Figure 9 demonstrates the observed time and the estimated time each of the drones reached each of the points allocated to the drones during the mission in milliseconds.

Finally, Table I summarises the estimation results. As shown, the maximum difference between the observed and predicted duration is almost five seconds and the average difference is three seconds. The calculated Root Mean Square Error (RMSE) is 3.4 seconds.

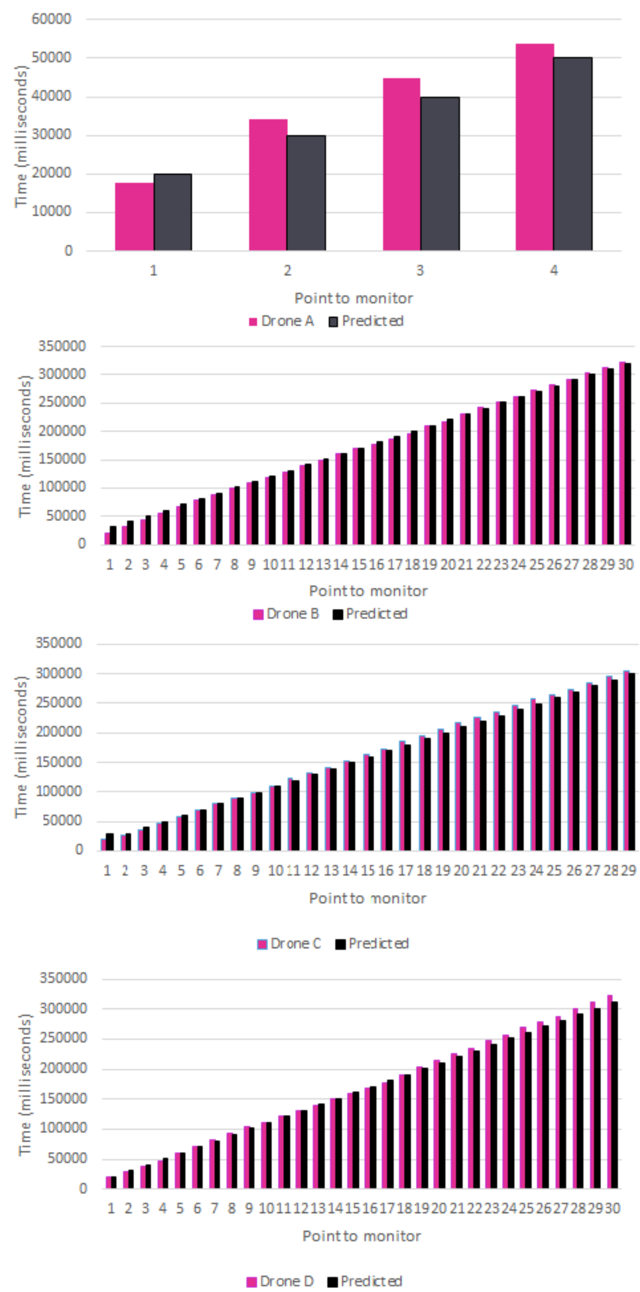

Fig. 9. Observed versus estimated arrival time for each UAV at each point during mission.

TABLE I
ESTIMATED FLIGHT DURATION MEASURED IN MILLISECONDS.

	Actual	Predicted	Accuracy
Drone A	53700	50000	93.1%
Drone B	322220	320930	99.6%
Drone C	304360	299390	98.4%
Drone D	323340	320930	99.3%

The results presented in this section provide initial evidence of the applicability of the proposed framework in a real world setting. The battery level and area coverage logged data over time indicate the stability of the proposed framework. Most importantly, The low estimation error, with RMSE of 3.4 seconds, provides evidence of the reliability and accuracy of proposed framework. The correct estimation of the time duration of a mission/ allocated path to a drone is crucial for the safe and reliable operation of drones in the field, and for emergency response in particular.

VI. Conclusions

This paper presented a unified framework for intelligent multi-drone tasking in emergency response missions. The paper presented a multi-drone tasking algorithm which takes into account both the status of the drones and the requirements of the tasks to be executed in an emergency situation. The algorithm makes the CP solver with adaptations made to address the application-specific challenges and constraints. The proposed framework was evaluated in computer simulations and in a real world setting in the field with four drones. The investigation presented in this paper provides evidence of its applicability and reliability of the proposed framework in a real world setting. While initially designed from emergency response, this framework is applicable to a plethora of other UAV swarm applications and applications sensitive to correct flight estimations in particular.

Known limitations of this work include the lack of extensive evaluation of the framework with multiple and reproducible scenarios in varying weather conditions, and the lack of evidence of for the efficiency and the scalability of the framework to multiple tasks and multiple drones. Further work should investigate the performance of the test-bed platform presented and provide results regarding its scalability. Evidence shall be provided for the efficiency of the proposed multi-drone tasking framework using more complex scenarios and potentially including the refuelling of drones. In addition, further research shall be conducted on optimising the CP with collaborative agents using local search is used to improve on those solutions by navigating through the search space and reach better solutions (using, for example, one of different meta-heuristics such as Greedy Descent, Simulated Annealing, and Tabu Search). The algorithms presented in this paper, the raw experimental data including the logs and photos captured, as well as the screen recording of the platform and additional video from the experiment are publicly available at [29].

ACKNOWLEDGMENT

This work is funded by the European Union Civil Protection under grant agreement No 783299 (SWIFTERS) and supported by the European Unions Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

References

[1] T. Zajkowski, S. Dunagan, and J. Eilers, "Small uas communications mission," in *Eleventh Biennial USDA Forest Service Remote Sensing Applications Conference, Salt Lake City, UT*, vol. 37, 2006.

[2] J. Curry, J. Maslanik, G. Holland, and J. Pinto, "Applications of aerosondes in the arctic," *Bulletin of the American Meteorological Society*, vol. 85, no. 12, pp. 1855–1862, 2004.

[3] P. Daponte, L. De Vito, G. Mazzilli, F. Picariello, S. Rapuano, and M. Riccio, "Metrology for drone and drone for metrology: Measurement systems on small civilian drones," in *Metrology for Aerospace (MetroAeroSpace), 2015 IEEE*, pp. 306–311, IEEE, 2015.

[4] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, p. 460, 2015.

[5] V. Mersheeva and G. Friedrich, "Multi-uav monitoring with priorities and limited energy resources," in *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.

[6] G. Avellar, G. Pereira, L. Pimenta, and P. Iscold, "Multi-uav routing for area coverage and remote sensing with minimum time," *Sensors*, vol. 15, no. 11, pp. 27783–27803, 2015.

[7] S. J. Kim, N. Ahmadian, G. J. Lim, and M. Torabbeigi, "A rescheduling method of drone flights under insufficient remaining battery duration," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 468–472, IEEE, 2018.

[8] A. R. Hovenburg, T. A. Johansen, and R. Storvold, "Mission performance trade-offs of battery-powered suas," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 601–608, IEEE, 2017.

[9] E. Yanmaz, R. Kuschnig, M. Quaritsch, C. Bettstetter, and B. Rinner, "On path planning strategies for networked unmanned aerial vehicles," in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 212–216, IEEE, 2011.

[10] H. Wang, B. Yan, X. Li, X. Luo, Q. Yang, and W. Yan, "On optimal path planning for uav based patrolling in complex 3d topographies," in *2016 IEEE International Conference on Information and Automation (ICIA)*, pp. 986–990, IEEE, 2016.

[11] H. Bai, S. Shao, and H. Wang, "A vtol quadrotor platform for multi-uav path planning," in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, vol. 6, pp. 3079–3081, IEEE, 2011.

[12] J. Hu, L. Xie, J. Xu, and Z. Xu, "Multi-agent cooperative target search," *Sensors*, vol. 14, no. 6, pp. 9408–9428, 2014.

[13] D. Zhang, Y. Xian, J. Li, G. Lei, and Y. Chang, "Uav path planning based on chaos ant colony algorithm," in *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, pp. 81–85, IEEE, 2015.

[14] G. Murtaza, S. Kanhere, and S. Jha, "Priority-based coverage path planning for aerial wireless sensor networks," in *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 219–224, IEEE, 2013.

[15] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2017.

[16] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets: Review," *IEEE Transactions on Cybernetics*, vol. 48, pp. 187–198, Jan 2018.

[17] M. Campion, P. Ranganathan, and S. Faruque, "Uav swarm communication and control architectures: A review," *Journal of Unmanned Vehicle Systems*, vol. 0, no. ja, p. null, 0.

[18] J. J. Roldán, B. Lansac, J. del Cerro, and A. Barrientos, "A proposal of multi-uav mission coordination and control architecture," in *Robot 2015: Second Iberian robotics conference*, pp. 597–608, Springer, 2016.

[19] Q. Deng, J. Yu, and N. Wang, "Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes," *Chinese Journal of Aeronautics*, vol. 26, no. 5, pp. 1238–1250, 2013.

[20] F. Çakıcı, H. Ergezer, U. Irmak, and M. K. Leblebicioğlu, "Coordinated guidance for multiple uavs," *Transactions of the Institute of Measurement and Control*, vol. 38, no. 5, pp. 593–601, 2016.

[21] T. Sherman and S. Boskovich, "Uav swarm mapping using a fully distributed control approach," in *AIAA Scitech 2019 Forum*, p. 2287, 2019.

[22] M. Monwar, O. Semiari, and W. Saad, "Optimized path planning for inspection by unmanned aerial vehicles swarm with energy constraints," *arXiv preprint arXiv:1808.06018*, 2018.

[23] A. Zhang, D. Zhou, M. Yang, and P. Yang, "Finite-time formation control for unmanned aerial vehicle swarm system with time-delay and input saturation," *IEEE Access*, vol. 7, pp. 5853–5864, 2019.

[24] L. Perron, "Operations research and constraint programming at google," in *International Conference on Principles and Practice of Constraint Programming*, pp. 2–2, Springer, 2011.

[25] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.

[26] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations research*, vol. 12, no. 4, pp. 568–581, 1964.

[27] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[28] "Dji developer documentation." https://developer.dji.com/. Accessed: 2019-03-03.

[29] "Multi-drone tasking algorithm for emergency response." https://github.com/mariankh1/A-Unified-Framework-for-Reliable-Multi-Drone-Tasking-in-Emergency-Response-Missions-. Accessed: 2019-05-06.