Collaborative Online Planning for Automated Victim Search in Disaster Response

Zoltán Beck[a], W. T. Luke Teacy[a], Alex Rogers[b], Nicholas R. Jennings[c,d]

[a] *University of Southampton, UK*
[b] *University of Oxford, UK*
[c] *Imperial College London, UK*
[d] *King Abdulaziz University, Saudi Arabia*

Abstract

Collaboration is essential for effective performance by groups of robots in disaster response settings. Here we are particularly interested in *heterogeneous* robots that collaborate in complex scenarios with incomplete, dynamically changing information. In detail, we consider an automated victim search setting, where unmanned aerial vehicles (UAVs) with different capabilities work together to scan for mobile phones and find and provide information about possible victims near these phone locations. The state of the art for such collaboration is robot control based on independent planning for robots with different tasks and typically incorporates uncertainty with only a limited scope. In contrast, in this paper, we take into account complex relations between robots with different tasks. As a result, we create a *joint*, *full-horizon* plan for the whole robot team by optimising over the uncertainty of future information gain using an online planner with hindsight optimisation. This joint plan is also used for further optimisation of individual UAV paths based on the long-term plans of all robots. We evaluate our planner's performance in a realistic simulation environment based on a real disaster and find that our approach finds victims 25% faster compared to current state-of-the-art approaches.

Keywords: Search and rescue, Task allocation, Hindsight optimization, Path planning, Multi-robot teams, Particle filter

1. Introduction

In recent years, professional first responders have started to use novel technologies at the scene of disasters in order to save more lives. Increasingly, they use robots to search disaster sites [1, 2, 3, 4] and mobile phones are used for both localising and communicating with casualties [5, 6, 7].

One of the most widely and successfully used robot platforms in the disaster response domain are *unmanned aerial vehicles* (UAVs) [2, 4, 7]. UAVs allow remote inspection and mapping. They are able to provide high resolution imagery and often need minimal infrastructure to fly. To speed up the information gathering process, multiple UAVs can be airborne at the same time. This increases the area that can be observed in a given amount of time, but requires coordination to ensure the resources are deployed effectively. Currently, however, such deployments use labour intensive, individually teleoperated UAVs. Given this, there is a drive toward using multiple robots operating with a certain level of autonomy, in order to decrease the operators' workload. One approach for utilising multiple robots in this way is semi-autonomous operation supervised by a small number of professionals; only requiring human experts for crucial decisions [8]. Current commercial UAV platforms also allow the deployment of a diverse group of robots, allowing them to combine their individual capabilities to be more efficient [9]. For example, *fixed-wing* UAVs are capable of flying faster and carrying larger payloads, but when they do so, they should be deployed with higher safety measures (safety pilots are required for non-lightweight aircraft). On the other hand, small *rotary-wing* UAVs are more agile and can approach and provide imagery about objects on the ground.

Against this background, we consider an automated victim search scenario in a disaster area with semi-autonomous UAVs using the mobile phones of casualties to detect their presence. Some systems rely on the global positioning capabilities of the mobile phones or assume an operational mobile network after the disaster[1]. We do neither. We assume that we have fixed-wing UAVs that are capable of carrying equipment for locating mobile phones and setting up temporary communication with them [10, 11] (*mobile phone scanning*). In addition, we also have small rotary-wing UAVs that can be easily teleoperated by professionals to inspect the surroundings of a mobile

[1]A large-scale disaster often destroys local infrastructure or at least overloads it causing unreliable operation.

phone to find detailed information about the state of a possible victim (*victim search*). This provides an opportunity to utilise the advantages of different UAV platforms in a collaborative setting. Of course, the problem complexity increases when collaborating with different robots. It is not different in this setting. Locating mobile phones and inspecting their surroundings involves an underlying dependency between these actions; a specific mobile phone needs to be located before its surrounding can be inspected.

To date, research on collaboration between multiple robots has typically focused on known settings, where the possible robot actions are defined as a set of tasks [12, 13, 14]. However, in most real-world settings, there is a significant amount of uncertainty present. For example, information about a disaster site develops gradually during disaster relief. Thus initially there is often very little certainty about the locations of people requiring assistance (e.g. damaged buildings, trapped victims, or supply shortages). Existing solutions that tackle collaboration in the face of uncertain information are typically limited to simple exploration or target search problems [15, 16, 17, 18]. Moreover, the use of generic temporal planners rapidly becomes intractable for such problems [19] unless applied in a domain-specific manner [15]. Finally, domain specific approaches rarely involve complex action relations, such as task dependencies where the actions of some robots are built on the actions of others. When they do so, decomposition techniques are applied to decrease the problem complexity [20, 21] or simple heuristics are applied to enhance similar collaboration [22]. Such approaches often lead to low quality solutions, because vital action dependencies across different roles are not taken into account during the optimisation.

Particularly in our scenario, the collaboration between the *mobile phone scanning* and *victim search* robots is vital when operating at the same time. When using decomposition planning, *victim search* robots cannot build on the plans of *mobile phone scanning* to identify valuable actions, and similarly, *mobile phone scanning* robots cannot prioritise their search according to the planned actions of the *victim search* robots. The main challenge is that the problem of multi-robot exploration (mobile phone scan) and multi-robot task allocation (victim search) are both complex optimisation problems [23]; especially when combined within an uncertain domain. For this reason *joint* planning[2] for multiple robots involving action dependencies in an uncertain

[2]Combined planning for both partial planning problems that optimises the plan of the

setting has not been achieved. When researchers tackle a similarly complex planning problem, often the planning horizon is limited to a number of actions in order to decrease the complexity. Unfortunately in this setting, the collaboration has a strong spatial bound (victims are located near mobile phone locations) and less strict temporal constraints (a victim search can start anytime after a mobile phone is detected). Subsequently, some actions may have an effect on other actions that are performed much later. Therefore *full-horizon* planning is necessary to fully capture the collaboration between the robots.

Given this context, and building on our preliminary work in [24], we offer a novel online planning approach for heterogeneous multi-robot collaboration under uncertainty that provides the benefits of *joint*, *full-horizon* planning. Uncertain information is incorporated in our planning process using *hindsight optimisation* (HOP) [25] that allows us to apply computationally efficient deterministic planning techniques for uncertain optimisation. To create a joint plan for different robots, the individual plans of homogeneous robot groups are optimised, with regards to the plans of the other groups as constraints, in an iterative process. In particular, we propagate the effects of the temporal constraints over the stochastic action domain between the plans of the different robot groups. As a result, we can generate plans for robots assigned to *victim search* even before the final locations of the mobile phones are confirmed. Similarly, we can generate plans for robots assigned to *scanning for mobile phones*, taking account of the current uncertain plans of *victim search* robots that will be tasked with searching victims near mobile phones. We evaluate the performance in a realistic physical simulation of mobile phone signals and UAV flight trajectories[3] in a disaster response setting set after the Haiti earthquake in 2010. Our results show that joint planning allows us to find victims faster compared to state-of-the-art approaches that use decomposition planning. Additionally, we successfully exploit the knowledge about the complete plans of the robot team by applying further optimisation in the *mobile phone scanning* using short-term path planning. In so doing, we make the following contributions to the state of the art:

---

entire robot group.

[3]The mobile phone signals are simulated using a log distance path loss model with added noise, UAV flight trajectories are simulated using 2D acceleration (rotary-wing) or angular speed limited model (fixed-wing) controlled by a simple simulated autopilot.

- We are the first to propose a method for multi-robot task allocation (MRTA) given an arbitrary task distribution and continuous action space (UMRTA problem, see Section 3.2). In detail, we use HOP to assign a task or a motion direction for each robot using a standard MRTA scheduler.

- We offer the first online planning approach to create a *joint* plan under uncertainty for distinct groups of robots in a complex collaborative setting (described above). As a result, *victim search* plans are made over an uncertain set of victim locations and iteratively optimised along with *mobile phone scanning* plans.

- In the current setting, we exemplify the value of robots being aware of the full-horizon plan of all other robots by applying further optimisation in the flight path of the *scanning* robots using Monte Carlo tree search (MCTS). We do this by showing that the same optimisation is not beneficial when applied using heuristics instead of the robot plans.

- We evaluate the performance of the joint planning approach against the state-of-the-art decomposition planning approach in a realistic simulation setting detailed in Section 7. In particular, we use realistic UAV flight path simulation, model mobile phone locations using particle filtering, while running robust, decentralised computation on the UAVs using message passing. The details of the message passing are discussed and communication requirements are determined for a physical deployment.

In the following we discuss the background of this paper (Section 2). This is followed by the problem definition (Section 3). After that in Section 4, we detail the long-term planning approaches used in the evaluation. In Section 5, the additional short-term path planning optimisation is detailed. Section 6 details the communication between robots. The following two sections detail the experiments (Section 7) and evaluate the results gained from the experiments (Section 8). Finally, we conclude the paper and provide directions for future work in Section 9.

2. Background

In this section, first the related literature is discussed, then relevant algorithms are described about task allocation and scheduling, optimisation under uncertainty and tree-search based optimisation techniques.

2.1. Related work

There are two main methodologies for utilising mobile robots (or autonomous vehicles) that are relevant to our setting. The first is task allocation, where the problem is allocating a set of robots to accomplish tasks, and it is well defined in the literature as the *multi-robot task allocation* (MRTA) problem [23]. Specifically, when coordinating mobile robots, tasks typically consist of going to a specific location and performing an operation in its proximity. This problem can be solved using centralised or decentralised approaches [23, 14], but relies on a known set of tasks (more details in Section 2.2 and Section 3). The second main methodology is searching an area, where the problem is defining specific paths for the robots in the area to decrease uncertainty by making observations. In these settings, an importance or *intensity map* contains the current state of the search and the knowledge about the area. This map also incorporates the amount of uncertainty in the current information, therefore optimisation under uncertainty can be applied (see Section 2.3). The robots travel on this map and update the intensity values according to their observations. A simple and robust way to guide the robots is to apply gradient descent on this map [26, 18]. Also several biologically inspired methods exist [27, 28, 29], and some use potential field based approaches in unstructured environments [30] that is beneficial when unexpected obstacles appear, but requires the tuning of several parameters to adopt to different settings. When the motion capabilities are limited as for fixed-wing UAVs, performing a tree search within the feasible paths can be used to control the vehicles [17, 31, 32] (detailed in Section 2.4). Unfortunately, to use such techniques for long-term online planning, the action space has to be significantly reduced by either discretising the problem [16] or limiting the action space to a set of predefined search patterns [15]. When discretising the search problem, it is often defined as a coverage problem that shows some similarity to the task allocation formulation. In coverage problems, the search problem is divided into neighbouring cells that have to be covered by the robots. The robot actions are defined as traversing between neighbouring cells to cover cells. This approach can be beneficial when the traversability of the area is non-trivial (e.g. no-fly zones, obstacles or walls divide regions in the search area). However when using UAVs to fly autonomously above a disaster site, the flight height can be chosen to be high enough to fly above any possible obstacle if the sensing capabilities allow it to do so (as in the setting described in this paper). In such cases, the same can be achieved using task allocation where each cell represents a task. Several papers show the capabilities of partially observable Markov decision processes (POMDP) to solve the coverage problem, but a correctly chosen greedy algorithm often performs very similarly to the computationally heavy POMDP approach in realistic problem settings [18, 33].

As discussed in Section 1, a full-horizon plan is necessary to capture the collaboration in a similar scenario. From the planning approaches mentioned earlier, full-horizon plans can be created using MRTA (in case of the time-extended assignment variation) [23] or temporal planning [16, 15]. However, because of the computational overhead of temporal planners, UAV search is impossible to compute in an online fashion for reasonable sized problems (10×10 grid, > 2 UAVs) as shown in [19]. For this reason, we approach the *mobile phone scan* problem as a MRTA problem with tasks on a grid.

An efficient tool to create full-horizon plans for problems containing uncertainty is determinisation. This method is detailed in Section 2.3. In this work, we use *hindsight optimisation* (HOP) for planning under uncertainty, that has been successfully applied in a multi-UAV context in [16].

When multiple groups of robots collaborate at the disaster site, the collaboration of groups becomes important in addition to the task allocation for the individual groups. Some approaches investigate multiple task types and their relations and creates an allocation for the whole group. For example, the relations between tasks can be represented as a task tree and subtrees are allocated to individual robots [13]. However, this assumes that the robots are capable of performing all tasks within a subtree and this is not the case when the different robot platforms are specific for a certain type of task. When using aerial robots, the amount of instrumentation on a platform is very limited, so distinct groups of robots performing tasks at different levels is more likely, as exemplified by [34]. In such settings, the most common approach of collaboration is independent action planning. In this vein, [35] performs a search and rescue mission using multiple UAVs, but search and rescue are separated in time. There are examples of robots with different capabilities operating at the same time in a search and surveillance setting [20, 21]. However, the planning for the different robot platforms is separate, only the outcome of their actions are shared between all robots. Some examples exist for collaborative planning with different robots. In particular, [22] uses the probabilistic plans of a ground robot to find areas to scout with a UAV and in [36] the allocations of different tasks are planned sequentially to allocate robots for a complex mission. These approaches allow different robots to plan according to the intentions of the collaborating robots, but does not adjust the plans of these collaborating robots.

In contrast, we provide a planner that adjusts the plans of all collaborating robots in order to optimise their behaviour. This creates a *joint plan* for all robots, that is able to capture action dependencies as discussed in Section 1.

2.2. Task-allocation and scheduling

The MRTA problem in our setting can be defined as a tardiness scheduling problem with common due date. Rescue tasks are time-critical, each needs to be completed as soon as possible, the common due date can be defined as $t = 0$. This means the utility of each task decreases over time, as later defined in Equation 1. Moreover, there are sequence dependent setup times for each task, that is, the time spent by a robot to travel to the specific task. Besides this, the scheduling problem is defined for a group of robots that operate at the same time, therefore it is a parallel machine scheduling problem. To summarize, the MRTA problem in our setting is a sequence dependent parallel machine tardiness scheduling problem with a common due date $P/\mathrm{ST}_{\mathrm{sd}}/\sum T_j$ [37] (detailed in Section 3.1).

In this work, the optimisation problem of the mobile robot group behaviour is of much higher complexity, as detailed in Section 3.3. For this reason, we apply a heuristic to solve a single MRTA problem, the shortest adjusted processing time first (SAPT) algorithm [38]. This heuristic assigns tasks in a sequence every time choosing the shortest adjusted processing time task, the one that can be finished the soonest from all available tasks. The same approach is referred to as iterated assignment, or broadcast of local eligibility for the MRTA problem in [23]. The SAPT scheduling is optimal for the sequence-independent single machine tardiness scheduling problem $(1/\mathrm{ST}_{\mathrm{si}}/\sum T_j)$ but nevertheless provides a good heuristic in the sequence-dependent settings [39, 40].

2.3. Optimisation under uncertainty

Uncertainty in a problem means that parts of the problem are unknown or random. These problems are called non-deterministic, and usually the unknown or random parts of the problem can be represented with a distribution of possible outcomes.

Optimising action plans in a non-deterministic problem is challenging, especially when full-horizon plans are necessary as in this work. A common approach to produce long-term plans for a non-deterministic problem is determinisation. This way, the non-deterministic problem can be solved as multiple deterministic problems. One common method is using all-outcomes determinisation that considers all possible random outcomes [41]. This approach is beneficial when the uncertainty is represented by discrete uncertain events, however it becomes intractable for a continuous random variable. Another possibility is hindsight optimisation (HOP), where instead of considering all possibilities, a Monte-Carlo simulation is applied to consider a limited number of samples of the possible random outcomes [25]. The deterministic problem is solved for each sample as if the outcome was known in hindsight, providing an upper bound to the solution quality for that specific outcome. After that, an aggregated solution is determined based on the solution for each problem sample. The average utility of these solutions will provide an upper bound to the solution for the non-deterministic problem.

2.4. Tree-search based optimisation

When optimisation is applied to problems with a large decision space often tree-search is applied. Using a tree representation, the decision space can be organised into a sequence of choices. Each node in the tree represents a choice and its children represent different outcomes of that choice. The most beneficial feature of the tree representation of the decision space is the separation into branches that consist of related decisions. If an algorithm can tell if a specific branch is suboptimal, it can be excluded from further investigation.

In many cases, it is very difficult to determine the suboptimality of a branch. Stochastic tree search techniques such as Monte Carlo tree search (MCTS) [42] or rapidly exploring random tree (RRT) provide a way to randomly expand the search tree towards more promising parts of the decision space. The value of each branch is calculated as the tree is grown randomly according to these values.

The tree representation is very effective for path planning. The tree nodes represent states of the vehicle, while different children of a node represent different control inputs (e.g. flight curvatures, accelerations or joint speeds). This ensures the restriction of the decision space to trajectories that are possible to follow by the vehicle.

Drawing this all together, we present an approach that provides a *full-horizon joint* plan for the collaborative automated victim search problem, optimised over the *uncertainty* in the victim locations. In more detail, *full-horizon* plans are created by considering the *mobile phone scanning* as a MRTA and the *victim search* as an UMRTA (uncertain multi-robot task allocation, defined in Section 3.2) problem. Specifically, full-horizon plans are created over the *uncertain* problem using a determinisation technique, HOP, that extends the solution in [16]. Moreover, *joint* planning is achieved by solving partial planning problems in an iterative process. We use realistic sensor modelling for the *mobile phone scan* process, based on the model in a very similar RF emitter search problem presented in [17]. As this paper also presents a solution for the coordination of the scanning UAVs, we use their method as a benchmark for the independent *mobile phone scan* planning. To incorporate past and simulate future observations of mobile phone signals, an observation model is applied that relies on a particle filter method. This method is often applied in the state-of-the-art target search and capture literature [43].

3. Task allocation problem

As introduced below, a fundamental problem underlying multi-robot collaboration is task allocation. In this section, we first define the details of the *multi-robot task allocation* (MRTA) problem in our victim search setting, then we propose an extension to it, the *uncertain multi-robot task allocation* (UMRTA) problem introducing uncertainty in the tasks. By building on these, we present an overall solution to the combined victim search problem.

3.1. Multi-Robot Task Allocation

As described earlier, both the *victim search scheduling* and the *mobile phone scanning* problem can be formulated as a task allocation problem. Specifically, we formulate them as a MRTA with single-task robots, single-robot tasks, and time-extended assignment (ST-SR-TA) [23]. This means that each robot can do a single task at once, each task can be done by a single robot, and the robot actions are considered in an extended time horizon (task schedule instead of a single task).

In our context, tasks are divided into two types: *scan* tasks (visiting a certain location to scan for mobile phones) and *victim search* tasks (finding a victim near the estimated mobile phone location). By performing these tasks, the robots aim to maximise the number of successful rescues resulting from finding and collecting information about the victims in a disaster. As the chance of a successful rescue decreases with time [44], we model this as a linearly decreasing utility function[4]:

$$U(\tau) = U_0 - \gamma t(\tau), \tag{1}$$

$$\sum_{\tau \in \mathbf{T}} U(\tau) = U_0 |\mathbf{T}| - \sum_{\tau \in \mathbf{T}} \gamma t(\tau), \tag{2}$$

where $\mathbf{T}$ is the set of tasks, $t(\tau)$ represents the time of completion of task τ, U_0 is the initial utility of a task, and γ is the utility decrease factor. U_0 and γ are chosen so that the utility does not go below 0 within the mission time (just as the chance of a successful rescue should not reach 0 during search and rescue). Moreover, each of these tasks has a specific location which the robots have to travel to in order to complete it. The time required to travel between the specific tasks can be derived from the motion model of the robot and the traversability of the area. These times can be regarded as necessary setup times to execute a task.

This MRTA problem can be formulated as a parallel machine scheduling problem (PMSP) of the following format: $P/\mathrm{ST}_{\mathrm{sd}}/\sum T_j$ [37]. In particular it is an identical machine scheduling problem with sequence dependent setup times where the aim is to minimise the total tardiness of the jobs with a common due-date at $t = 0$ ($T_j = t(\tau_j)$). The complexity of the problem for a single machine and sequence independent setup times ($1/\mathrm{ST}_{\mathrm{si}}/\sum T_j$) is polynomial when the due-date is common, however given sequence dependent setup times ($1/\mathrm{ST}_{\mathrm{sd}}/\sum T_j$) it becomes NP-hard [39]. Therefore the MRTA problem, that is the parallel machine version of the above, is also NP-hard.

In the following, we define the MRTA problem as a tuple $\langle \mathbf{R}, \mathbf{T} \rangle$, where $\mathbf{R}$ is the set of robots and $\mathbf{T}$ is the set of tasks. The solution consists of an assigned task (x_R) for each robot (R):

$$\mathbf{X} = \{x_R \in \mathbf{T} : \forall R \in \mathbf{R}\}.$$

3.2. Uncertain Multi-Robot Task Allocation

In realistic settings there might not be complete information about the tasks' location or other parameters. In this case, the uncertainty about the tasks

[4] *scan* tasks are not directly associated with a utility, but instead are responsible for discovering mobile phones and associated victim search tasks (detailed in Section 3.3).

can be represented as a probability distribution. We define the UMRTA problem as an MRTA problem with a random variable representing the set of tasks: $\langle \mathbf{R}, \mathcal{T} \rangle$. The random task set ($\mathcal{T}$) breaks into a set of known tasks ($\mathbf{T}$) and a set of uncertain tasks[5] ($\mathcal{T}^+$): $\mathcal{T} = \mathbf{T} \cup \mathcal{T}^+$.

In this case, the optimisation maximises the expected overall utility gain. Accordingly, Equation 2 changes as follows:

$$\mathop{\mathbf{E}}_{\mathcal{T}^+} \left[\sum\nolimits_{\tau \in \mathcal{T}} U(\tau) \right] \tag{3}$$

In contrast to the MRTA problem, uncertain tasks cannot be assigned directly because their location is unknown. Therefore we allow the solution to contain a general motion direction for a robot ($\mathbf{D} = [0, 2\pi)$) besides the set of known tasks ($\mathbf{T}$). This allows us to move the robots to optimise their position given the distribution of uncertain tasks, so that they are able to reach tasks soon after they are discovered. Of course, these directions need to be reassigned in a timely manner to navigate the robots efficiently. As a result, the solutions will have the following form:

$$\mathbf{X} = \{x_R \in \mathbf{T} \cup \mathbf{D} : \forall R \in \mathbf{R}\}.$$

3.3. Collaborative victim search problem

Having described the task allocation elements of the automated victim search planning problem (MRTA and UMRTA), we now present the combined heterogeneous robot planning problem in the automated UAV victim search setting. In this scenario, different tasks require a different set of skills and sensors from the UAVs (high speed and RF sensors for *mobile phone scan*, good maneuverability and live video for *finding victims*), so they split into a group of *mobile phone scanning* UAVs and a group of *victim search* UAVs.

Scanning UAVs find mobile phones and estimate their locations, while *victim search* UAVs move around the disaster area and find victims near the mobile phone locations. As described earlier, long-term planning is necessary to fully capture the collaboration aspect of the automated victim search problem. To achieve this, the problem of *scanning for mobile phones* can be formulated as a set of locations necessary to visit in order to cover the area.

[5]This means both the properties of the tasks in the set and the cardinality of the set can be uncertain.

To this end, the collaborative planning breaks into a classical task allocation problem (MRTA) for the *scanning* UAVs and another task allocation problem with possibly unknown task locations (UMRTA) for the *victim search* UAVs.

With this in mind, we define each *victim search* task with a 2D location of a mobile phone and a time length (process time) based on how long it takes to locate a victim (i.e. $\mathcal{T}_v \subset \mathbb{R}^2 \times \mathbb{R}^+$). As all these parameters are estimated using the observation model, the mobile phone location is the mean of the phone location distribution and the task length is the expected search time. Assuming a fixed search speed, the *victim search* process time is proportional to the area of the possible locations of the mobile phone.

As the victim search task is to gather information about a possible victim, there is very little known before this happens. For this reason, we use uniform utility for the *victim search* tasks and their process time is according to the detection area of a mobile phone (detailed in Section 7). Assuming complete spatial randomness of the mobile phone locations, the task locations are the outcome of a non-homogeneous spatial Poisson process [45] with intensity function λ. This distribution frequently changes when a region is scanned or new information is introduced about the disaster site.

Formally, the *mobile phone scan* MRTA problem is determined by the set of *scanning* robots ($\mathbf{R}_s$) and *scan* tasks (locations on a grid, $\mathbf{T}_s$), while the *victim search* UMRTA problem is determined by the set of *victim finding* robots ($\mathbf{R}_v$) and the random set of *victim search* tasks ($\mathcal{T}_v$). The *victim search* tasks can be divided into a set of known ($\mathbf{T}_v$) and a random set of unknown tasks, that can be modelled as a Poisson point process ($\mathcal{T}_v^+ \sim Poisson(\lambda)$). Additionally, the connection between the *scan* and the *victim search* problems can be given by the $closest : \mathcal{T}_v \to \mathbf{T}_s$ function that associates the closest *scan* task to a *victim search* task. This indicates the most likely *scan* task that would discover the mobile phone associated to a *victim search* task.

Accordingly, the collaborative victim search problem is defined by $\langle \mathbf{R}_s, \mathbf{T}_s, \mathbf{R}_v, \mathcal{T}_v, closest \rangle$, and the solutions will have the following form:

$$\mathbf{X} = \{x_R \in \mathbf{T}_s : \forall R \in \mathbf{R}_s\} \cup \{x_R \in \mathbf{T}_v \cup \mathbf{D} : \forall R \in \mathbf{R}_v\} .$$

This describes the specific planning problem in the introduced setting. If the problem contains multiple homogeneous robot groups, and their decision making can be formulated into UMRTA (or MRTA) problems, the definition can be formulated in a very similar way, and the planning method can be easily adapted.

4. Multi-robot planning

We now detail the planning approaches used in the evaluation. Some algorithms are selected or modified to suit the UAV platforms used in the evaluation (see Section 7). Specifically, we intend to use fixed-wing UAVs for scanning for mobile phones, and rotary-wing UAVs for victim search. These roles are chosen to best suit the capabilities of the platforms and best practice in the literature [1, 2] following consultations with Rescue Global [46].

4.1. MRTA planning with SAPT scheduling

After considering several (many of them identical) heuristics to solve the MRTA problem [23] we chose the shortest adjusted processing time first (SAPT) algorithm, the heuristic that is tailored for tardiness scheduling problems with common due date (detailed in Section 2.2). In detail, we define the adjusted processing time as follows:

$$AP\left(\tau_i, [s, \tau_j]\right) = AP\left(\tau_j, s\right) + trav(\tau_i, \tau_j) + P(\tau_i). \tag{4}$$

$AP\left(\tau, s\right)$ represents the adjusted processing time of task τ after schedule s (equals $t(\tau)$ in Equation 1 assuming the schedule is followed by the robot), $[s, \tau]$ stands for a schedule where task τ is inserted at the end of schedule s, $trav(\tau_i, \tau_j)$ is the necessary travel time between task τ_i and τ_j (sequence dependent setup time), and $P(\tau)$ is the process time of task τ.

4.2. Victim search planning with gradient descent

Without planning under the uncertainty of expected mobile phone locations, the best choice is to direct the victim search robots closer to the likely mobile phone task locations in order to decrease the time necessary to approach them when the phones are identified. For this purpose, we use gradient descent on the task intensity to make use of the task distribution knowledge. This robust approach does not need the tuning of additional parameters that potential fields, the other commonly used approach for guiding robots towards beneficial locations [30], requires (as discussed in Section 2.1).

In more detail, we apply task allocation for the set of known *victim search* tasks ($\mathbf{T}_v$), and use gradient descent to direct the robots when there are no *victim search* tasks assigned to them. In several settings there are large constant intensity areas where the gradient descent is unable to provide a direction. In these cases, robots are directed towards central areas in order to minimise the expected distance from upcoming tasks. In detail, victim search UAVs with no assigned tasks will travel:

- towards the nearest nonzero intensity location if outside the affected area,
- otherwise along the gradient of the task intensity, or
- towards the mean of the task distribution if the gradient is zero.

4.3. Victim search planning with with HOP

In this approach, instead of ignoring the task distribution for the planning of the *victim search* allocation, HOP (described in Section 2.3) is used to solve the UMRTA problem. HOP has been shown to effectively incorporate probabilistic information for a similar problem domain for a generic temporal planner (see discussion in Section 2.1). Unfortunately the introduced temporal planner in [16] does not scale well with the size of the problem due to the iteration through the possible resulting states. By contrast, we present a novel HOP planner that is able to solve the planning problem without discretising the state space. Our method provides a computationally efficient solution in an obstacle-free environment. Specifically, it does not create plans for all the possible future states, it rather finds the most beneficial direction of travel using the solutions for the current state. This approach simplifies the computation and is able to provide a solution in a continuous state space (as per the first contribution in Section 1).

The HOP planner incorporates the distribution of unknown task locations, using a Monte Carlo simulation. It provides solutions for independent samples of the distribution as if it was a deterministic planning problem (MRTA in this case). As a result, the maximisation criterion (Equation 3) is approximated as follows:

$$\underset{\mathcal{T}_v^+}{\mathbf{E}}\left[\sum_{\tau\in\mathcal{T}_v} U(\tau)\right] \approx \frac{1}{N}\sum_{i}^{N}\sum_{\tau\in\mathbf{T}_v\cup\mathbf{t}_v^i} U_{sch}(\tau) \tag{5}$$

In detail, the expected utility is estimated by the average of the utilities ($U_{sch}(\tau)$) determined by the SAPT MRTA scheduler's (Section 4.1) resulting in schedules ($s_{R,i}$) for N samples ($\mathbf{t}_v^i$) of the Poisson process ($\mathcal{T}_v^+$). Note that the MRTA scheduler can be replaced with any MRTA solver for a specific application (e.g. an auction-based negotiation combined with RRT path planning [14] or simulated annealing [38]).

The maximisation is applied on this utility estimation, and as a result, the rotary-wing robot actions are determined. This process is detailed in Algorithm 1.

Algorithm 1 HOP UMRTA Solver

Require: $\mathbf{T}_v$: set of known *victim search* tasks
Require: $\mathbf{R}_v$: set of *victim search* robots
Require: N: size of the Monte Carlo simulation
Require: $s_{R,i}, \forall R \in \mathbf{R}_v, i \in \{1..N\}$: hindsight schedules
1: **for all** $R \in \mathbf{R_v}$ **do**
2: $\quad \tau^* = \arg\max_{\tau \in \mathbf{T}_v} \mathrm{count}_{i=1}^{N} \left(\tau = first(s_{R,i})\right)$
3: $\quad$ **if** $\mathrm{count}_{i=1}^{N} \left(\tau^* = first(s_{R,i})\right) > \frac{N}{2}$ **then**
4: $\quad\quad x_R \leftarrow \tau^*$ $\triangleright$ assign task τ^* to robot R
5: $\quad$ **else**
6: $\quad\quad d = \mathrm{mean}_{i=1}^{N} \left[w(s_{R,i}) * dir(R, first(s_{R,i}))\right]$
7: $\quad\quad x_R \leftarrow d$ $\triangleright$ assign direction d to robot R
8: $\quad$ **end if**
9: **end for**
Ensure: $\mathbf{X} = \{x_R : \forall R \in \mathbf{R}_v\}$: chosen actions for robots

Specifically, function $first(s)$ returns the first task in schedule s, and $dir(R, \tau)$ calculates the direction that robot R has to take to move towards task τ. In general, for each robot, the aggregation will assign either the most commonly assigned first task in the schedules (τ^*) to the robot or the weighted average of the direction of the first assigned tasks. Specifically, the algorithm iterates over all robots, and finds the most commonly assigned first task ($\tau^* \in \mathbf{T}_v$) in its schedules (Line 2). If it is assigned as first in the majority of the schedules ($> \frac{N}{2}$), the robot's instruction will be to execute task $x_R = \tau^*$, otherwise it will be a general heading direction ($d \in \mathbf{D}$) as in Line 6. This direction is a weighted average of the first assigned tasks' directions, where the weight represents the number of tasks in a schedule ($w(s) = |s|$). This weight indicates how many tasks are going to be delayed if the execution of the first task is delayed.

This weighted average will provide the *optimal direction of movement* to maximise the utility estimate in Equation 5. This, of course, only provides optimality assuming perfect hindsight knowledge of the random task set for each sample ($\mathbf{t}_v^i$). In particular, this means it is only optimal given that the outcome of the random variable is known by the next time step. The proof of this hindsight optimality is given in Appendix A.

4.4. Mobile phone scanning with MCTS path planning

This approach builds upon the work in [17] where multiple fixed-wing UAVs locate RF emitters and decrease the uncertainty of their locations. This problem formulation is very similar to our *mobile phone scanning* problem, therefore their approach can be applied directly for the path planning of the *scanning* robots. In our simulation, we use a Monte Carlo tree search (MCTS) [42] tool (rather than a rapidly exploring random tree as mentioned in the paper) that reproduces the behaviour of the path planner detailed in this work. In detail, the MCTS engine uses the immediate reward default policy described in [17], and uses the UCB as a tree policy that provides a better performance than a flat tree policy suggested by the original algorithm. Because the task in our case is to find all mobile phones in the disaster area, rather than decreasing the uncertainty of specific RF emitters, we use the intensity of the posterior distribution of unobserved targets (detailed in Section 7) as the cost map.

4.5. Mobile phone scanning using MRTA

Producing a long-term plan is crucial in order to make strategic decisions with the robots that brings benefits at a later stage. As discussed in the related work, long-term planning is not possible with classical path planning using motion primitives, a set of goals need to be defined instead. This is why we use task allocation for the scanning problem as well, defining a set of locations that will ensure the full coverage of the area.

To that end, we use the SAPT scheduling approach explained in Section 4.1, although the standard tardiness scheduling problem is slightly modified to suit the motion model of a fixed-wing. In particular, when maneuvering a fixed-wing aircraft, the path length can be highly affected by the direction of passing through a specific location. This means not only the task schedule, but also the desired heading for each task needs to be determined. In order to cope with this, each *scan* task is split into a number of tasks with different headings (we used 8 for the simulation). The adjusted processing time (AP) is determined for each heading, but for each location only the heading with the shortest adjusted processing time is considered for execution at each scheduling step.

4.6. Joint planning with HOP

The joint planning approach presented below is the first that is able to produce a joint plan in a collaborative setting while planning under uncertainty and not limiting the planning horizon (as per the second contribution in Section 1). It shows a way to produce online plans under uncertainty for groups of robots when their tasks impose temporal constraints on each other. At the same time, it incorporates long-term temporal effects between tasks by achieving a full-horizon plan.

Specifically, our joint planning approach creates a combined plan for the *scanning* and *victim finding* robots via an iterative process[6]. At each iteration step, one of the robot groups (*mobile phone scanning* UAVs or *victim finding* UAVs) optimises their plan according to the current plan of the other group. This way, the solution quality can be further improved by taking the relations between the *scan* and *victim search* activities into account (*scan* tasks result in discovering nearby mobile phones that create *victim search* tasks).

Victim search plan optimisation uses the *scan* plan to introduce a temporal constraint on the scheduling problem of *victim search* tasks. Accordingly, the adjusted processing time in Equation 4 changes as follows:

$$AP'(\tau, s) = \max\left(AP(\tau, s), cstr(\tau)\right). \tag{6}$$

Function $cstr(\tau)$ represents the temporal constraint, it is the time when task τ is discovered according to the current *scan* plan. In brief, the execution time of each "hindsight task" is delayed until discovered according to the current *scan* plan. The MRTA scheduling problem is solved using SAPT (Section 4.1) using the adjusted processing time in Equation 6.

Also, these constraints need to be considered when solving the UMRTA problem using HOP (Section 4.3). The weights (w) in Line 6 of Algorithm 1 need to be adjusted to further produce the *optimal direction of movement* as per Theorem 2 (Appendix A). Tasks that are delayed by the introduced temporal constraints do not increase the urgency of a schedule, because a small delay in the schedule will not delay their execution. For these reasons, these tasks should not be counted in the weights. Accordingly, here $w(s)$ is the number of consecutive non-delayed tasks at the beginning of schedule s.

[6]Our empirical studies show that an iteration lengths above 5 do not produce significant changes in the overall result, so we chose 5 iterations in the evaluation to minimise the computation time.

Fixed-wing planning optimisation does not simply minimise the execution time of the *scan* tasks using the plan of the other group. Rather the subject of optimisation is to improve the *victim search*, as the utility comes from the time when victims are found. Therefore, the optimisation is applied to minimise the delay introduced on the *victim search* tasks by the temporal constraints (introduced in Equation 6). Due to the nonlinearity of this measure, building an increasing plan by adding tasks to the end of the schedules (as in SAPT scheduling) would not result in a good quality solution. For this reason, we need a method that can cope with this characteristic. Our approach is to sequentially introduce tasks and insert them into a position of a schedule with minimal cost, similarly to the MURDOCH negotiation process [47], detailed in Algorithm 2.

Algorithm 2 Fixed-wing Plan Optimisation

Require: $\mathbf{T}_s$: set of scan tasks
Require: $\mathbf{I} = \{\mathbf{t}_\tau : \forall\tau \in \mathbf{T}_s\}$: timings from rotary-wing plans
Require: $\mathbf{R}_s$: set of fixed-wing scanning robots
  $\mathbf{T}' \leftarrow \mathbf{T}_s$: unassigned scan tasks
  $s_r \leftarrow \emptyset, \forall r \in \mathbf{R}_s$: fixed-wing robot schedules
  while $\mathbf{T}' \neq \emptyset$ **do**
      $\tau^* = \arg\min_{\forall\tau\in\mathbf{T}'} \min \mathbf{t}_\tau$
      $\langle r^*, i^*\rangle = \arg\min_{\langle r,i\rangle;\forall r\in\mathbf{R}_s,\forall i\leq|s_r|\in\mathbb{N}} \Delta\left(s_r, \tau^*, i\right)$
      $s_{r^*} \leftarrow \mathit{insert}\left(s_{r^*}, \tau^*, i^*\right)$ ▷ Insert τ^* to schedule
      $\mathbf{T}' \leftarrow \mathbf{T}' \setminus \{\tau^*\}$ ▷ Remove from unassigned tasks
  end while
Ensure: $\mathbf{S} = \{s_r : \forall r \in \mathbf{R}_s\}$

Here, $\mathbf{t}_\tau$ is a set of times when *victim search* could be initiated (according to the current rotary-wing plans) at locations that are the closest to the *scan* task τ: $\mathbf{t}_\tau = \{AP(\tau_i) - P(\tau_i) : \forall\tau_i \text{ where } \mathit{closest}(\tau_i) = \tau\}$. The $\mathit{insert}\left(s, \tau, i\right)$ function inserts task τ into schedule s in the i^{th} location. Besides that, $D\left(s\right)$ estimates the delay caused by schedule s:

$$D\left(s\right) = \sum_{\tau\in s}\sum_{q\in\mathbf{t}_\tau} \max\left(0, t(\tau, s) - q\right), \tag{7}$$

$$\Delta\left(s_r, \tau, i\right) = D\left(\mathit{insert}\left(s_r, \tau, i\right)\right) - D\left(s_r\right). \tag{8}$$

In Equation 7, $t(\tau, s)$ denotes the execution time of task τ within schedule s. In brief, Algorithm 2 inserts available tasks – starting with the most urgent ones – into the *mobile phone scanning* robots' schedule using minimal insertion. Specifically, the most urgent task is selected as the one with the soonest *victim search* initiation time (Line 4). Having selected this task, the best insert location is determined within the current schedules (Line 5). The best position is determined by minimising the delay estimate's increase (Equation 8) computed based on the information from the rotary-wing robots' plan ($\mathbf{I} = \{\mathbf{t}_\tau : \forall \tau \in \mathbf{T}\}$).

Although this specific joint planning method applies for the planning problem presented in Section 3.3, it can easily be applied for joint planning for problems with different structures as suggested at the end of the same section. Specifically, after determining how the behaviour or the utility gain is modified for each group through the task relations (as per Equations 6 and 7) an incremental planning algorithm can be applied to maximise the utility in the given context for each robot group resulting in a joint plan.

5. Fixed-wing short-term path optimisation

A further benefit of long-term joint planning is being able to use the other robot's plans for behaviour optimisation (as mentioned in the third contribution in Section 1). By solving the *mobile phone scanning* as a MRTA, we restrict the actions of the *mobile phone scanning* UAVs to only pass over a grid of locations. On the other hand, a path planner algorithm can take advantage of the manoeuvrability of the UAVs and find more informative routes for them.

Given this, in this section we show a way to preserve the long-term plans produced by the joint planner, and combine them with a short-term path planner to improve the victim search performance. The empirical evaluation (Section 8) shows that doing so improves the overall performance by decreasing the length of *victim search* tasks, but this is only beneficial when the robots are aware of the plans of other robots (the joint plan).

Initially when using the task allocation based planning (detailed in the previous section), the *mobile phone scanning* UAVs take the shortest path between the scheduled scan tasks (locations on a grid). However, sometimes it is beneficial to deviate from this path in order to make more valuable observations about the nearby mobile phones' signals. Of course this means that the rest of the scanning schedule is going to suffer delays due to the longer path taken to the next scan task. In order to assess this benefit and drawback of taking a specific route deviation, the differences have to be translated into utilities.

In order to optimise possible deviations from the shortest path, we use a standard MCTS method with the information provided by the observation model. This method provides individual optimisation for search agents that can make use of the joint plan of all agents. In more detail, when taking more informative observations about a mobile phone's location, its location uncertainty (the victim search area) decreases. In order to approximate this decrease, we use a Monte Carlo method using a three-step process. First, possible mobile phone locations are sampled from the current belief from the observation model of observed mobile phones. After that, mobile phone signals are simulated from these locations along the specific path. Finally, the area decrease is estimated by an update step of the particle population based on these simulated signals. This decrease in the search area advances the schedule of the assigned *victim search* robot. It therefore causes this *victim search* task and the following tasks in the UAV's schedule (the ones not delayed by the *scanning*) to finish sooner. If the area decrease is of small increments, we can assume that each of the affected tasks finish the same $\Delta t = \Delta A / v_s$ earlier, where ΔA is the decrease in the victim search area and v_s is the victim search speed of the UAV. In this case the only information needed from the *victim search* UAV plans is $\mathbf{E}[w(\tau)]$, the expected number of consecutive tasks after a specific task that are not delayed by the scan process according to Equation 6:

$$\Delta U_\tau^+(\Delta t) = (1 + \mathbf{E}[w(\tau)]) * \gamma * \Delta t. \tag{9}$$

On the other hand, when making a detour in the route of a *scanning* UAV, it is going to cause delays in the further *scan* process. The deviation causes $\Delta t = (s' - s)/v_f$ delay, where s' and s is the length of the deviated path and the original path respectively, and v_f is the cruising speed of the *mobile phone scanning* UAVs. This delay is the same for all scan tasks in the UAV's schedule. The utility decrease comes from increasing the imposed delays on the victim search of the currently unknown mobile phone locations (D in Equation 7). If this Δt (the delay from the original plan) increases with small δt increments, the delay increase can be estimated by the number of delayed tasks by *scan* schedule s ($N_{d,s}(\Delta t)$): $\Delta D \approx \delta t * N_{d,s}(\Delta t)$. Because the number of delayed tasks comes from N independent samples of the possible mobile phone locations, division by N is necessary to get the expectation:

$$\Delta U_s^-(\delta t, \Delta t) = -\frac{N_{d,s}(\Delta t)}{N} * \gamma * \delta t. \tag{10}$$

Using these two measures, we can quantify the utility difference between two slightly different routes:

$$\Delta U = \sum_{\tau \in \mathbf{T}_v} \Delta U_\tau^+ \left(\frac{\Delta A_\tau}{v_s} \right) + \Delta U_s^- \left(\delta t, \Delta t \right), \tag{11}$$

where ΔA_τ is the difference in the victim search area estimate for task τ for the two routes, $\mathbf{T}_v$ is the set of known *victim search* tasks, and δt ($\ll \Delta t$) is the time difference between completing the two routes. This formula is used as an immediate cumulative reward for a Monte Carlo tree search (MCTS) engine that is run on a time-limited basis [42]. The tree search uses the standard UCT tree policy that compares the achieved utility gain to the utility gain of taking the shortest path ($c = \sqrt{2} * U_{shortest}$). The outcome of the MCTS is then verified against the original planned path and applied in case it increases the overall utility.

The utility gain calculation is highly dependent on data from the plans of other agents. However, when using a decomposition technique, such data is not available and the robots are not aware of the plans of other robot groups. In these settings, we have substituted the plan-based values with constants and simple heuristics. Specifically, the expected number of delayed tasks is approximated with an average value in the current scenario[7] $\mathbf{E}\left[w(\tau)\right] = w_o$, and the number of delayed tasks is substituted with a simple linear function[8] $N_d(\Delta t) = N_{d0} + \alpha \Delta t$.

6. Robot communication

In this section we detail the communication between the robots, detailing the requirement of perfect communication and the required bandwidth. These requirements are crucial for a physical deployment of the system in the future.

[7] Chosen to be 10 for 12 rescue UAVs and 200 rescue tasks. This means about 17 tasks on average for each UAV, so the average number of tasks following a task is around 8, but we expect already observed tasks to be sooner in a schedule.

[8] The value of N_{d0} does not affect the result, as the delay is compared to the initial (shortest path) solution, α is chosen 7 task/s.

The communication can be split into two categories, broadcast communication and point to point communication. These two categories are discussed in the following sections.

6.1. Broadcast communication

Broadcast type messages serve the purpose of updating the state of the victim search problem and synchronising the robot group's behaviour.

The update of the victim search problem consists of the update of the state of each robot[9], the state of each task[10], and the update of the available information about undiscovered mobile phone locations (distribution of $\mathcal{T}^+$). These updates are not essential for the robot collaboration, if some messages are lost due to imperfect communication, the collaboration is still achieved. When some messages are lost, the solution might be suboptimal in some cases, but the impact on the overall solution quality is low according to our preliminary tests [48]. As for the required bandwidth, the robot and task position update requires minimal bandwidth, while the update of the distribution of undiscovered mobile phone locations requires the transmission of a larger amount of data. In the presented evaluation, the distribution is an array of approximately one million elements, stored as 8 MB of data. However, the amount of data can be easily compressed[11] as high precision is not required and many of the values are 0.

The current implementation of the synchronisation of the behaviour of robots is achieved using broadcast messages. This assumes prefect communication, and would fail to update the robot decisions if some messages are lost. However, this synchronisation can be achieved through different means on physical robots. One possibility is using synchronised clocks to time the robot decision making behaviour [49]. An alternative would be not using fix intervals for decision making, but starting a new decision making process as soon as the robot and the task states are updated. Our evaluation uses fix intervals for decision making for easier comparison of different decision making approaches. The synchronisation messages do not convey data, therefore do not represent a significant portion in the overall communication bandwidth.

[9]A 2D position for each robot.

[10]A 2D position for each *mobile phone scan* task, and a 2D position and process time for each *victim search* task.

[11]For example, the initial value occupies only 40kB of space when saved as a png image.

6.2. Point to point communication

Point to point messages provide information exchange between robots while running the collaboration algorithms. In the following, the message types are listed and the requirement for perfect communication and bandwidth requirement is detailed for each:

1. Messages between *mobile phone scanning* robots: There are no messages between *mobile phone scanning* robots for the collaboration algorithm, the same algorithm is recomputed independently on each robot.

2. Messages between *victim search* robots: The decentralised algorithm processes independent samples on each robot, and the end result is shared between the robots. This means each robot sends the relevant part of each solution of the HOP process to every other robot. The transferred data consists of the first task or its direction and the length of the schedule, so the message consists of a 2D vector (direction and schedule length) and an integer (task ID) for each HOP sample. In case of imperfect communication, some message data would be lost, but that only means that the HOP solution uses fewer samples, that has only a minor effect on the solution quality.

3. Messages from *victim search* robots to *mobile phone scanning* robots: These messages convey the information about the timings of the *victim search* robots for the *mobile phone scanning* robots, **I** in Algorithm 2. The information about the number of delayed tasks is also necessary for the short-term path-planning optimisation ($N_{d,s}(\Delta t)$ in Equation 10). The data includes the location, HOP schedule timing and the number of consecutive tasks for every HOP task. If some messages are lost due to imperfect communication, the number of HOP samples decrease in the timing information. Similarly to the previous case, this only has a minor effect on the solution quality.

4. Messages from *mobile phone scanning* robots to *victim search* robots: These messages provide the constraints for the victim search MRTA scheduling problems, $cstr(\tau)$ in Equation 6. The message data includes the location and planned execution time of mobile phone scanning tasks. If some of these messages are lost, the MRTA problems can be computed using information received earlier. In this case the *victim search* robots might react with a little delay when *mobile phone*

scanning robots change their plan and there is a temporary loss in communication. Another possibility is that the number of iterations for the joint planning (Section 4.6) decrease if some messages are randomly dropped.

None of these messages require perfect communication, as a temporary communication loss or some dropped messages do not result in a major defect in the collaboration.

Message	Size [byte]	Optimised size	Frequency
Robot update	72	10	$\|R_s\| + \|R_v\|$
Task update	$24\,(\|\mathbf{T}_v\| + \|\mathbf{T}_s\|)$	$12\ (\|\mathbf{T}_s\| + \|\mathbf{T}_v\|)$	1
Distribution u.	7 964 128	$\approx 41\ 000$	1
Type 2	$24 * N_{hop}$	$12 * N_{hop}$	$3 * \|\mathbf{R}_v\| * (\|\mathbf{R}_v\| - 1)$
Type 3	$32 * N_{hop} * \|\mathcal{T}_v\|$	$7 * N_{hop} * \|\mathcal{T}_v\|$	$3 * \|\mathbf{R}_v\| * \|\mathbf{R}_s\|$
Type 4	$24 * \|\mathbf{T}_s\|$	$6 * \|\mathbf{T}_s\|$	$3 * \|\mathbf{R}_v\|$

Table 1: Summary table of communication details

Table 1 summarizes the message sizes in the implementation for the evaluation, and an approximate message size after optimisation of the message content besides the frequency of messages. The rows represent different message types, three broadcast update messages detailed in Section 6.1 and three robot-to-robot messages listed in Section 6.2. The message size of the current implementation, the approximate message size of a bandwidth optimised implementation, and the frequency of messages, the number of messages sent each cycle[12], are detailed for each message type.

Here N_{hop} represents the number of HOP samples processed by a single *victim search* robot and the overall number of HOP samples processed is $N = N_{hop} * |R_v|$ (detailed in Section 4.3). Substituting the values used in the evaluation, the amount of communication between the robots can be calculated: $|R_s| = 2$, $|R_v| = 16$, $|\mathbf{T}_s| = 108$, $|\mathbf{T}_v| \lessapprox 200$, $|\mathcal{T}_v| \lessapprox 200$, $N_{hop} = 9$. Overall sent data each cycle in the current implementation is < 13.2 MB, while the same number with bandwidth optimised messages decrease to < 1.3 MB. This amount of data is sent in the system every 10 s cycle, resulting in an average bandwidth of 1.3 MB/s and 0.13 MB/s for all

[12]The cycle time in the evaluation is 10 s.

messages in the system. This is not the bandwidth requirement for a single link of the system, it counts all sent messages. Point to point messages may not need to be transferred over the same communication link depending on the means of robot communication. Point to point messages decrease the required bandwidth if independent communication links are present between robots.

7. Experimental setup

To evaluate the performance of our automated victim search approach, we chose the 2010 Haiti earthquake. We did this for the following reasons. Firstly, the official disaster assessment data is available from the United Nations [50] and this provides high resolution data about the destruction after the earthquake that allows us to create a realistic simulation. This earthquake is also an illustration of poor organisation and information distribution between first responders. For example, only half of the *search and rescue* (SAR) sectors could be completed in the first week [51] and the poor information management made the collaboration of different agencies very difficult [52]. Given this, we would like to show how our automated victim search approach could have helped save lives by providing crucial information about some of the victims within a couple of hours using a smaller team of first responders.

In more detail, our experiments simulate a first response scenario in Carrefour after the earthquake. As disaster responders arrive at the site, they send an automated group of UAVs to identify the first rescue locations for the SAR team. The UAV group consists of fixed-wing *mobile phone scanning* UAVs and rotary-wing *victim search* UAVs. The fixed wing aircraft carry equipment to create an ad hoc mobile network that allows emergency responders to push message/calls to the mobile phones in range [10, 11]. The equipment is also able to locate the mobile phones by their signal strength using a similar approach to [17]. The emergency responders select the mobile phones that they prioritise given the result of the communication with the individual phones. These selected phones' locations are then automatically visited by one of the rotary-wing UAVs. As a mobile phone location is visited, the victim search process starts around the location. The victim search is carried out by the teleoperation of a disaster responder professional to collect the necessary information for the SAR triage process.

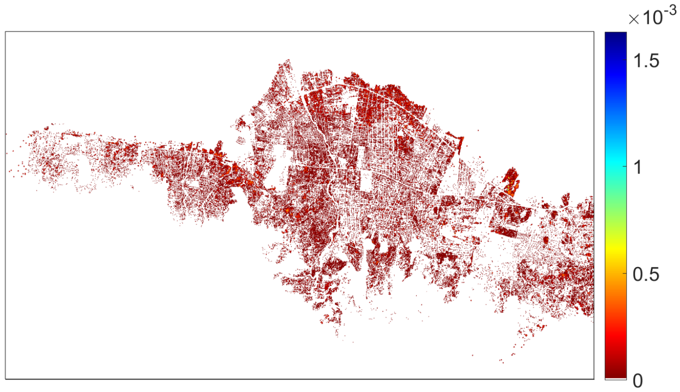
Grade	Description	P(phone)	#buildings	#phones
1	Negligible to slight damage	0.000780	59 377	46.29
2	Moderate damage	-	0	0
3	Substantial to heavy damage	0.004873	2 118	10.32
4	Very heavy damage	0.012182	3 988	48.58
5	Destruction	0.030455	3 113	94.81
Σ			68 596	200.0

Table 2: Assessment damage grades, probability and expected number of mobile phones

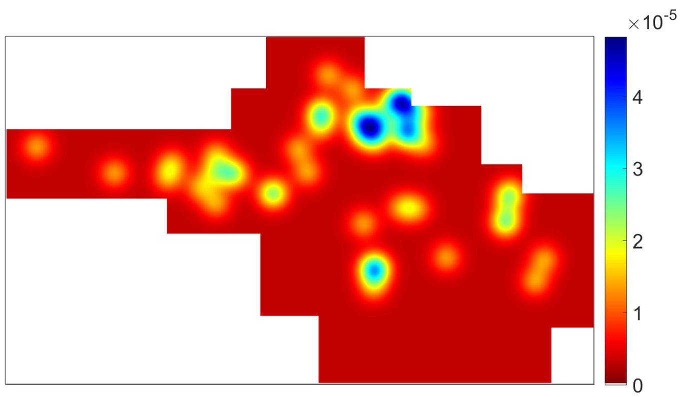
For the mobile phone detection, we apply a particle filter-based probability distribution model using the sensor modelling in [17] but additionally incorporate information from negative observations as well as positive ones. This allows us to determine the posterior for unobserved targets and the distribution of the uncertain tasks ($\mathcal{T}^+$) in the UMRTA problem of the collaborative victim search problem (Section 3.2 and Section 3.3). More details about the sensor model can be found in [48]. The initial value of this posterior is the belief distribution of undetected mobile phones on the disaster site when the mobile phone is first detected ($Poisson(\lambda)$). This distribution is periodically updated based on the recent observations.

The UN dataset contains the location and the damage status of each building in the area. The damage grading is explained in Table 2. We have assigned a chance for each building for a mobile phone to request help in its proximity. There were over 15 000 messages that were made available to the Ushahidi Haiti Project, where the frequency of tags falling in the category of emergency is 5% [53] and Carrefour contains 25% of the tagged buildings in Haiti. These statistics result in about 200 mobile phones requesting emergency help from the 68 596 building locations in Carrefour. The mobile phone locations were determined by a two dimensional random distribution with standard deviation $\sigma = 9.1$ m (the average closest building distance to a building) added to the building location from the dataset.

The resulting random process to generate the ground truth mobile phone locations for the simulations is an inhomogeneous 2D spatial Poisson point process [45]. The intensity of the spatial Poisson process based on the description above (*ground truth intensity*) can be seen in Figure 1a. The robots' initial *belief* of the intensity of the mobile phones at different locations can be seen in Figure 1b. This map is generated based on a rough perimeter area of Carrefour and 50 simulated reported locations (e.g. first incoming reports via social media or emergency channels). The locations of these reports are drawn from the *ground truth* Poisson process, and added on the *belief intensity* as 20 times wider Gaussian functions than the ones used for the individual buildings. The resulting intensity imitates a basic intensity map manually generated by a first response team using the available information of the approximate area perimeters and 50 report locations. There is a large difference between the scale of the mobile phone density in the two intensity functions. This is because the high detail in the *ground truth* intensity function includes much higher peaks than the less detailed, smoother belief intensity function. Although, the expected number of mobile phones (the integral of the intensity function) is 200 in both cases.



(a) Haiti ground truth intensity [phone/m^2]



(b) Haiti initial belief intensity [phone/m^2]

Figure 1: Intensity maps of the Poisson point processes

Parameter	Fixed-wing UAVs	Rotary-wing UAVs
Maximum speed	22 m/s	10 m/s
Maximum acceleration	-	3 m/s^2
Minimum turn radius	60 m	-
Cruise height	50 m	varying
Mobile phone detector range	300 m	-
Victim search speed (v_s)	-	40 m^2/s
Number of platforms	2	12
Number of tasks	108	avg. 200

Table 3: UAV parameters

The performance of the planning approaches is evaluated empirically with 128 different possible disaster outcomes that is sufficient to show statistical significance. Each outcome is an independent sample from a Poisson process with the *ground truth* intensity introduced above, and the approaches are run with the same set of disaster outcomes.

We have chosen realistic parameters for the fixed-wing and rotary-wing aircraft[13] that can be seen in Table 3. In our scenario, once the estimated mobile phone location is automatically reached by rotary-wing UAVs, they are teleoperated by emergency responders to find a victim and collect information for the triage process. The time length of the teleoperated victim search process is estimated during the simulation based on the location error of the mobile phone estimate. We assume the victim search starts at the location estimate of the mobile phone and spirals out from there with constant speed. Using this approach, finding a victim will take $t_f = e_{loc}^2/v_s$ time, where e_{loc} is the location estimate error distance and v_s is the victim search speed[14].

We chose a small number of fixed-wing aircraft due to the expensive sensor equipment and the safety pilot requirement for operating these platforms We have a higher number of inexpensive small rotary-wing camera UAVs that are easier to deploy. Their number is only limited by the number of possible disaster responders teleoperating them in order to find information about

[13]The specifications are based on the ING Robotic Aviation Serenity fixed-wing [54] and senseFly Albris rotary-wing UAVs [55].

[14]Based on our analysis of a post-disaster UAV footage, we chose the scanning speed equivalent to searching a 20 m wide area with 2 m/s speed.

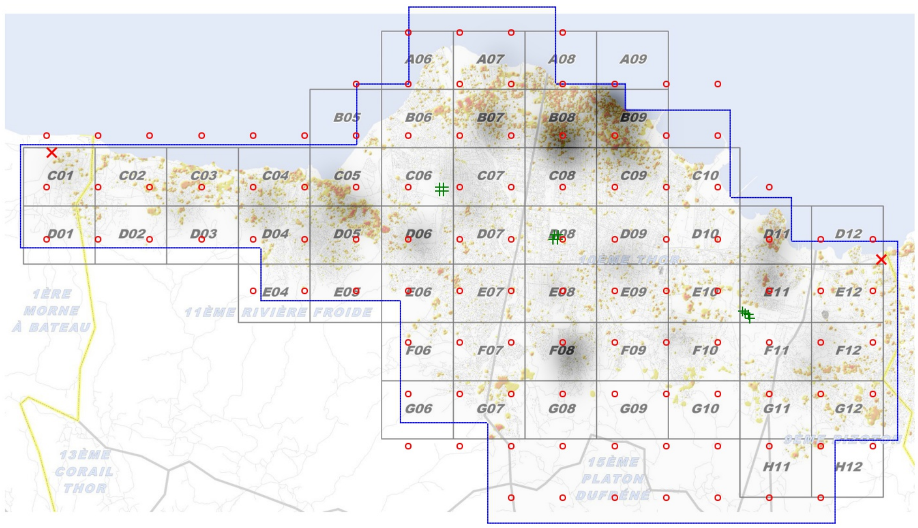


Figure 2: UAV start locations and scan task locations

possible victims. The initial mission setup can be seen in Figure 2 over the overview of the UN dataset (damaged buildings marked with orange and yellow). In detail, the blue edge marks the perimeter of the belief intensity (Figure 1b) and the dark areas are the high intensity regions in the same map. There are two fixed-wing UAVs that take off on larger roads near the perimeter of the area. These locations are marked with red crosses in Figure 2 (in sectors *C01* and *D12*). There are 12 rotary-wing UAVs that take off from 3 locations in manageable-size groups of 4. These locations are larger fields where infrastructure can be set up for controlling them such as a university garden, a college park and a stadium. The locations are marked with green crosses in Figure 2 (in sectors *C06*, *D08*, and *E11*). The predefined tasks for fixed-wing *mobile phone scanning* UAVs are marked with red circles and are placed on a grid with 592 m spacing that is twice the detection range (measured on the ground) of a fixed-wing UAV flying at 50 m height, higher than the tallest building in the region[15].

The chosen environment[16] is portable, it can be run under a range of op-

[15] We ran simulations with different variations of these parameters, but the results were broadly similar.

[16] A Python and NumPy based environment using ZeroMQ for communication.

erating systems, and popular on embedded platforms on robots both in terms of the runtime and the communication. The UMRTA planning is distributed by running only N/N_{rw} Monte Carlo simulations, where N is the number of intended samples, and N_{rw} is the number of rotary-wing UAVs. The relevant results of these Monte Carlo simulations are then exchanged between the UAVs. In this way message losses are not crucial, they only lead to a smaller number of samples processed, and the computational load is distributed between the UAVs. The fixed-wing plan optimisation (Section 4.6) is computed independently on each process, but due to its deterministic nature, the resulting schedule is identical. The state of the UAVs are broadcast periodically and all plans are computed accordingly. This ensures the system to rapidly adapt to UAV dropouts. The UAV simulators accept waypoint commands that are the standard for open-air UAV coordination. To sum it up, these features result in a system that is very close to physical deployment. The software system is released here: `https://bitbucket.org/zbeck/thesischapter5`.

In the following, the results of the conducted experiments are introduced. The statistical significance is tested by comparing the relative performance against our approach for each run that used the same victim locations as a one-sample t-test with 95% confidence level. Specifically, we use the average time of finding a victim as a performance indicator. This gives a direct comparison of the overall utility gained by the different approaches according to the utility definition (Equation 2). The lower this metric, the higher the SAR performance.

8. Experimental results

In this section the performance is evaluated for all possible combinations of the previously detailed approaches in the above settings. Consequently, there are eight different planning methods that show different approaches for the *mobile phone scanning* and for how *victim search* UAVs cope with uncertainty. In detail, the planning methods for fixed- and rotary-wing UAVs are compared for the evaluated approaches in Table 4.

Our baseline approach, *MCTS Grad*, uses independent *mobile phone scanning* using a MCTS implementation based on the state-of-the-art for RF emitter scanning with multiple fixed-wing UAVs [17] and deterministic MRTA for scheduling *finding victims* using gradient descent when no tasks are assigned to an agent (see Section 4.2). We evaluate the effects of different planning features in the previously detailed realistic setting:

Name	Mobile phone scan	Victim search	Joint
MCTS Grad	MCTS based on [17]	Gradient descent	No
MCTS HOP	MCTS based on [17]	HOP	No
Indep Grad	SAPT MRTA	Gradient descent	No
Indep HOP	SAPT MRTA	HOP	No
Collab	Collaborative MRTA	HOP	Yes
Indep Grad Man	SAPT MRTA & MCTS	Gradient descent	No
Indep HOP Man	SAPT MRTA & MCTS	HOP	No
Collab Man	Collab. MRTA & MCTS	HOP	Yes

Table 4: Planning methods used in the compared approaches

1. Long-term task allocation planning (SAPT MRTA) for the fixed-wing *scan* (Section 4.2),

2. Hindsight optimisation (HOP) for planning under the uncertainty of mobile phone locations with rotary-wing UAVs (Section 4.3),

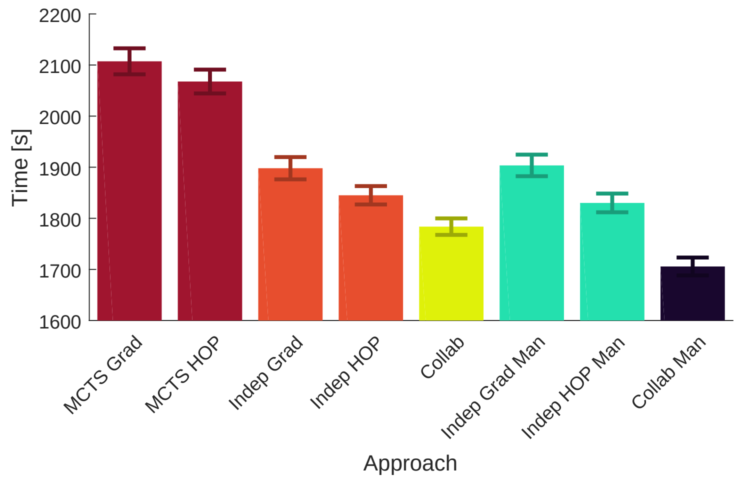
3. Collaborative (joint) planning with fixed- and rotary-wing UAVs (Section 4.6),

4. Long-term planning with short-term path optimisation using Monte Carlo tree search (MRTA & MCTS) for fixed-wing *scan* (Section 5).

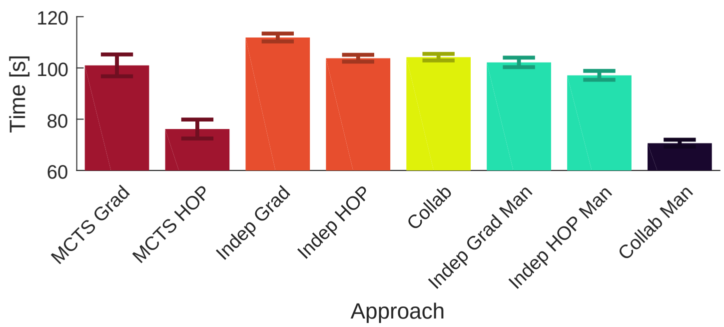
The overall performance comparison can be seen in Figure 3a. Based on our utility definition (Equation 2), the average time when victims are found gives an appropriate comparison between different approaches for the same simulation outcome, and it is also independent of the number of victims at the specific simulation setting. All the presented planning features improved the overall performance compared to the baseline approach:

1. The long-term planning allowed the search to explore the area more efficiently, not leaving isolated areas behind. This improved the overall performance by 12% (*Indep Grad* vs *MCTS Grad*).

2. Hindsight optimisation significantly improved the performance in the independent planning settings compared to the gradient descent approach (*Indep Grad* vs *Indep HOP*: 3.3% and *Indep Grad Man* vs *Indep HOP Man*: 5.1%). This is because long-term planning under uncertainty provides a possibility for *victim search* UAVs to split up



(a) Average of time when victims are found.



(b) Average time a rotary-wing UAV takes to find a victim from arriving to the estimate mobile phone location.

Figure 3: Overall performance comparison with 95% confidence intervals.

and cover different areas rather than all travel towards the first mobile phone locations. HOP is also a necessary step for collaborative planning with the UAV teams, because the connection between the search and the rescue problem lies within the not yet searched regions.

3. Joint planning of the long-term plans also increases the performance. There is a 6.9% improvement without the fixed-wing path optimisation (*Indep Grad* vs *Collab*) and 12.5% with the optimisation enabled (*Indep Grad Man* vs *Collab Man*). This is in line with the results in [24].

4. The short-term fixed-wing path optimisation significantly improved (by

5.4%) the performance when it could take advantage of the plans of the teams of the UAVs to prioritise important *scan* or *victim search* tasks (*Collab* vs *Collab Man*). Also, there is no significant improvement when the plans are substituted with heuristics (*Indep Grad* vs *Indep Grad Man* and *Indep HOP* vs *Indep HOP Man*). This shows how being aware of the full plan can be beneficial when using individual optimisation techniques. This information can identify the priority of actions in relation to the actions of others, therefore makes such optimisation techniques more efficient.

Altogether, there was an 24.7% improvement between the baseline approach and the approach with all enhancements. All computation was done using a single core of an Intel Xeon E5-2670 processor per UAV while meeting the computational time limit that a real-time system would face. The long-term planning was done every 10 s, while the short term planning every 1 s; therefore long-term planning had a 10 s, and short-term planning has 1 s computational time limit. The tree-search techniques are anytime approaches, so they were stopped after the computational limit was reached, while the collaborative planning has a fixed computation time depending on the complexity of the problem (cubic in terms of the number of scanning tasks). The computation time of the collaborative planning is well below the real-time limit, and this limit is only reached when having over 400 scanning tasks based on our tests.

Providing some insight into the behaviour of the approaches, Figure 3b shows the average process time of *victim search* tasks and the average time rotary-wing UAVs took to find victims near the sensed location of mobile phones. There is a clear improvement in this measure when the fixed-wing path optimisation is added, but only when it can make use of the joint plans of the UAVs (*Collab Man*). This shows the importance of knowing the long-term plans of the agents rather than relying on general heuristics when applying further optimisation in their behaviour. It also explains the lack of improvement when the path optimisation is used in other approaches. The difference in the rotary-wing *victim search* length for the baseline and the tree-search with hindsight optimisation approaches (*MCTS Grad* vs *MCTS HOP*) emphasizes the main difference between the gradient descent and HOP behaviour. Specifically, when using gradient descent, *victim search* tasks are immediately scheduled as mobile phones are observed, while during HOP *victim search* UAVs wait at high mobile phone density locations until they

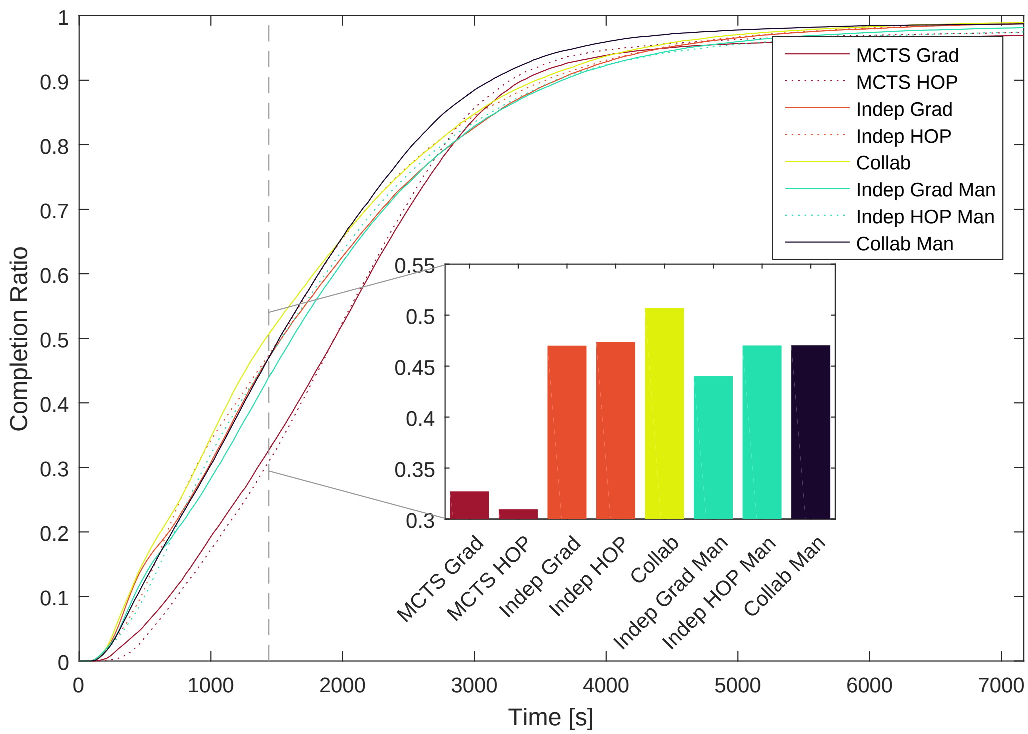


Figure 4: Completion rate of finding victims over time. Bars show the progress after 24 minutes.

are scanned. Having waited, there are more observations made about other mobile phones that decreases the time the rotary-wings need to find a victim at these locations, but waiting imposes a large delay causing a low overall performance.

The progress of the automated victim search can be observed over time in Figure 4. The completion ratio shows the ratio between found and not yet found victims. The bar graph shows a cross section after 24 minutes, the maximal flight time of a popular professional small rotary-wing camera drone (DJI Phantom 3 Professional). If the victim search would stop after this time, the collaborative approach without path optimisation (*Collab*) would find the most victims, 101 on average. The fact that only half of the victims are found at this point shows that the long battery life of the platforms is crucial for a successful application of automated victim search in a similar setting.

In more detail, the progress using tree search based mobile phone scanning

(*MCTS Grad* and *MCTS HOP*) shows a significantly less steep completion curve in the first 20 minutes due to the slower coverage of the area by mobile phone scanning. It can also be seen that when using deterministic planning for the rotary-wing planning (*MCTS Grad*, *Indep Grad*, *Indep Grad Man*) the immediately scheduled tasks accelerate the initial progress in the first 500 seconds, but the progress rapidly falls back as this causes the UAVs to move to the same regions that is suboptimal for covering all areas. The progress graph also shows the differences between the joint planning with and without the fixed-wing path optimisation (*Collab* and *Collab Man*). In particular, we see the path optimisation will slightly delay the mobile phone scanning process in order to make critical mobile phone locations more accurate. In line with this, the progress has an approximate 2 minutes delay at t=1000 s. However, after that the completion stays steep for a longer time due to the more accurate mobile phone locations; the completion is 3.7% higher (about 7 more victims found) at t=3000 s.

9. Conclusions and future work

Challenging planning problems are often simplified with relaxed constraints in order to apply complex algorithms that optimise system performance. Such simplifications are often necessary to avoid an optimal or near-optimal solution being intractable. These simplifications may include replacing direct dependencies of actions with general goals or applying the same constraints to a group of actions. Specifically, when multiple actions create a workflow processes, this would mean optimising each stage independently in order to optimise the whole process. However, there are many problems where the dependencies between the individual actions are very strong, which in turn, requires a specific sequence to reach a goal. These problems can be dealt with by using temporal planning, that searches for beneficial action sequences given a general problem description. Although, it is often difficult to provide a problem description in a realistic setting with such approaches. This is especially true in a multi-robot setting, where there is significant uncertainty present.

Against this background, in this paper we show a different approach. We make compromises in the individual solution quality, but make use of the complex connections in the uncertain planning problem. As a result, we construct an online planner that manages to improve the performance compared to the state-of-the-art.

Our approach is presented through an application in disaster response, in an automated victim search system. The automated victim search consists of scanning an area in order to locate mobile phones and then finding victims near the mobile phone locations with two specialised groups of robots (UAVs) for each tasks. The complex planning problem is finding a *joint* solution for the multi-robot search problem (mobile phone scanning) and for the multi-robot task allocation problem (finding victims), while taking the connection of these activities into account. Moreover, the connection lies within the unexplored mobile phone locations, therefore has a significant uncertainty.

To this end, we present an online planner that creates a joint plan for both groups of robots, while planning under uncertainty for the victim finding robots. We show the benefits of full-horizon planning, so long-term dependencies can be taken into account. Moreover, we combine long-term planning with a short-term path planning approach that further improves the performance, but only when the robots are aware of the plans of the other robots. This shows another advantage of long-term joint planning, when it is possible to produce local changes and see their effect on a mission level.

Although we illustrated our approach in a particular scenario, it is a general approach. The joint planner has been applied in different SAR problem settings in our previous work in small-scale SAR settings[17] and limited detail simulation of an imagery-based victim search scenario [24]. It could also easily be applied in other domains, such as wildlife reservation [56] or critical power line inspection and recovery [57]. In both settings multiple actors can be present at a time critical and uncertain scenario (e.g. surveillance or inspection UAVs, law enforcement or repair vehicles). The dependencies between the actions of these actors can be taken into account using uncertain joint planning of their actions improving their response time, thus the success rate of anti-poaching or decreasing system downtime.

In this work, we evaluated the performance of the planner in a highly realistic simulation system. We apply an accurate sensor model of mobile phone signal strength, simulate the control of the UAVs resulting in realistic flight paths, and also simulate the synchronisation and communication between the UAVs. The resulting system provides a portable, easy to deploy solution for embedded hardware on physical robots. The simulation is based

[17] SAR of a small sector (0.9 km^2) after the Haiti earthquake and after an industrial spill in Hungary.

on the Haiti earthquake in 2010. We use high resolution disaster assessment data to produce realistic victim locations. During the evaluation, our optimised joint planning approach finds victims 25% faster than the state-of-the-art approach. The software system where the evaluation has been run is released[18].

The communication requirements are discussed in Section 6. While collaboration works best with perfect global communication, the collaboration can be maintained with missing messages or communication links as well. The required bandwidth is reasonable and can be reduced to one tenth using message optimisation.

Besides the optimisation of communication data, there are many aspects of this work that needs further development before a physical deployment becomes possible. The most important is the mobile phone sensing and communication equipment. There are prototypes present for this purpose [10, 11], but these are far from robust products that can easily fit as an UAV payload. Bringing this technology closer to deployment in disaster settings would contribute into an extremely useful tool in disaster response.

As mentioned in Section 8, the computation time of the joint planning is sufficiently low in the presented setting, but could reach the computational budget in other settings. To address this shortcoming, we plan to investigate online learning techniques to improve plans using the experience from previously produced plans as future work. The current planner establishes a new plan discarding the information from previous planning steps. This provides high flexibility when the information of the disaster site suddenly changes, but results in redundant computation when such change does not happen. Using online learning, important information can be extracted from earlier plans to create higher quality plans in later planning steps. This allows the planner to cope with complex problems more efficiently (e.g. problems with more tasks, more kinds of dependencies, or more types of actions or actors).

10. Acknowledgments

This research was undertaken as part of the ORCHID project funded by the EPSRC (EP/I011587/1). We also acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

[18] Can be found at `https://bitbucket.org/zbeck/thesischapter5`.

Appendix A. Decision making to determine the optimal motion direction of *victim search* robots given hindsight plans

In this section, the HOP solution aggregation method is detailed that is applied in Algorithm 1 in Section 4.3. The problem is directing *victim search* robots while mobile phones (and therefore *victim search* tasks) are not explored yet – the *scanning* process is ongoing – in order to maximise the overall utility by finding victims sooner.

In this case, the optimisation should rely on the distribution of the possible mobile phone locations. As detailed in Section 4.3, the distribution is sampled and several solutions are given using HOP. Consequently, the aggregation problem is to find a movement direction for each robot given the hindsight schedules for each sample of the tasks in order to maximise the expected utility gain of the rescue agents.

At first, we solve the problem for a single step look ahead planning problem, then we extend the solution to multiple step look ahead planning.

Theorem 1. *The optimal aggregated direction for a robot to maximise the utility of executing the closest task from a random distribution is the average of the directions of closest tasks from independent samples of the task distribution.*

Proof. First the proof is shown for an example of two tasks. In Figure A.5, we can see an agent choosing a direction for proactive movement to execute one of the two possible tasks appearing with p_1 and p_2 independent probability respectively. The distance from the tasks are d_1 and d_2 respectively, $d_1 < d_2$. The unit vector to task 1, task 2 and the chosen direction of proactive movement is $\mathbf{v_1}, \mathbf{v_2}$ and $\mathbf{v_a}$ respectively. We assume a linearly decreasing utility function by the distance from the task, $U = U_0 - d * \lambda$ (in line with Equation 1). The optimal $\mathbf{v_a}$ vector can be derived as follows:

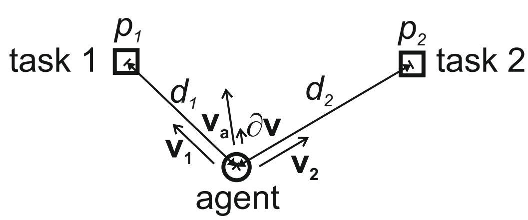


Figure A.5: Simple proactive movement example.

$$E[U] = p_1 * (U_0 - d_1 * \lambda) + p_2 * (1 - p_1) * (U_0 - d_2 * \lambda), \tag{A.1}$$

$$\frac{\partial E[U]}{\partial \mathbf{v}} = \lambda * \left(-p_1 * \frac{\partial d_1}{\partial \mathbf{v}} - p_2 * (1 - p_1) * \frac{\partial d_2}{\partial \mathbf{v}} \right), \tag{A.2}$$

$$\frac{\partial E[U]}{\partial \mathbf{v}} * \frac{1}{\lambda} = -p_1 * \frac{\partial |d_1 \mathbf{v_1} - \partial \mathbf{v}|}{\partial \mathbf{v}} - p_2 * (1 - p_1) * \frac{\partial |d_2 \mathbf{v_2} - \partial \mathbf{v}|}{\partial \mathbf{v}}. \tag{A.3}$$

Now, as the length of $\partial \mathbf{v}$ converges to zero, we can make the following approximation:

$$\partial |\mathbf{v_n} - \partial \mathbf{v}| \approx -\frac{\mathbf{v_n}}{|\mathbf{v_n}|} \cdot \partial \mathbf{v}, \tag{A.4}$$

that is the length of the parallel component of $\partial \mathbf{v}$ with $\mathbf{v_n}$. Therefore,

$$\frac{\partial E[U]}{\partial \mathbf{v}} * \frac{1}{\lambda} = p_1 * \frac{\mathbf{v_1} \cdot \partial \mathbf{v}}{\partial \mathbf{v}} + p_2 * (1 - p_1) * \frac{\mathbf{v_2} \cdot \partial \mathbf{v}}{\partial \mathbf{v}}, \tag{A.5}$$

$$\frac{\partial E[U]}{\partial \mathbf{v}} * \frac{1}{\lambda} = \frac{(p_1 \mathbf{v_1} + p_2 * (1 - p_1) \mathbf{v_2}) \cdot \partial \mathbf{v}}{\partial \mathbf{v}}, \tag{A.6}$$

$$|\partial \mathbf{v}| = \partial d, \quad \partial \mathbf{v} = \mathbf{v_a} \partial d, \tag{A.7}$$

$$\frac{\partial E[U]}{\mathbf{v_a} \partial d} * \frac{1}{\lambda} = \frac{(p_1 \mathbf{v_1} + p_2 * (1 - p_1) \mathbf{v_2}) \cdot \mathbf{v_a} \partial d}{\mathbf{v_a} \partial d}, \tag{A.8}$$

$$\frac{\partial E[U]}{\partial d} = \frac{(p_1 \mathbf{v_1} + p_2 * (1 - p_1) \mathbf{v_2}) \cdot \mathbf{v_a}}{\lambda}. \tag{A.9}$$

Now, to maximize the expected utility gain, we have to choose a direction of movement $\mathbf{v_a}$ that makes $\frac{\partial E[U]}{\partial d}$ the highest, that will point to the same direction as $p_1 \mathbf{v_1} + p_2 * (1 - p_1) \mathbf{v_2}$, as it will maximize the scalar product in Equation A.9:

$$\arg\max_{\mathbf{v_a}} \frac{\partial E[U]}{\partial d} = \frac{p_1 \mathbf{v_1} + p_2 * (1 - p_1) \mathbf{v_2}}{|p_1 \mathbf{v_1} + p_2 * (1 - p_1) \mathbf{v_2}|}. \tag{A.10}$$

If we take infinite number of samples of the possible outcomes, task 1 will be chosen with a ration of p_1, and task 2 will be chosen with a ratio of $p_2 * (1 - p_1)$, and no tasks will be in the $(1 - p_1) * (1 - p_2)$ ratio of the samples. The average direction vector in this example will be identical to the optimal direction in Equation A.10. The average of the vectors similarly lead to the optimal proactive movement direction for more tasks or agents when only one task can be performed by an agent. ■

Theorem 2. *The optimal aggregated direction for a robot to maximise the utility of a series of tasks is the weighted average of initial directions from* *hindsight optimisation solutions for independent samples drawn from the distribution. Using weight values that are the number of non-delayed tasks in the corresponding schedule.*

Proof. The method is very similar to the previous case (Theorem 1). The difference is that not only one task utility is decreasing as the first task is delayed, but later tasks in the schedule as well. Of course, if a task has a temporal constraint causes it to start later than when the robot arrives there, there is no extra delay caused by arriving a little later. For this reason, *critical* tasks are distinguished: a critical (non-delayed) task has no delay from time constraints compared to the corresponding schedule ($AP\,(\tau, s) > cstr(\tau)$ in Equation 6). Time constraints can be imposed from the scan plan as detailed in Section 4.6. Because the delay in all the critical tasks in a schedule is the same as the delay in the first task, Equation A.5 changes as follows:

$$\frac{\partial E[U]}{\partial \mathbf{v}} * \frac{1}{\lambda} = \sum_i p_i * c_i \frac{\mathbf{v_i} \cdot \partial \mathbf{v}}{\partial \mathbf{v}}. \tag{A.11}$$

Here the sum is over all possible task outcomes. p_i, c_i and $\mathbf{v_i}$ stands for the chance of the i^{th} possibility, the number of critical task for the solution schedule, and the direction vector of the first task of the schedule. This will change the average direction in Equation A.10 to a weighted average with the weighting factors of the number of critical tasks for each schedule:

$$\arg\max_{\mathbf{v_a}} \frac{\partial E[U]}{\partial d} = \frac{\sum_i p_i c_i \mathbf{v_i}}{|\sum_i p_i c_i \mathbf{v_i}|}. \tag{A.12}$$

∎

References

[1] J. Qi, D. Song, H. Shang, N. Wang, C. Hua, C. Wu, X. Qi, J. Han, Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake, Journal of Field Robotics 33 (3) (2016) 290–321. doi:10.1002/rob.21615.

[2] E. Lubrano, Drone adventure in haiti, http://www.droneadventures.org/2013/05/29/haiti/ (May 2013).

[3] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, S. Kawatsuma, Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots, Journal of Field Robotics 30 (1) (2013) 44–63. doi:10.1002/rob.21439.

[4] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, C. Humphrey, Supporting wilderness search and rescue using a camera-equipped mini uav, Journal of Field Robotics 25 (1-2) (2008) 89–110. doi:10.1002/rob.20226.

[5] D. Pastor-Escuredo, A. Morales-Guzmán, Y. Torres-Fernández, J. M. Bauer, A. Wadhwa, C. Castro-Correa, L. Romanoff, J. G. Lee, A. Rutherford, V. Frias-Martinez, N. Oliver, E. Frias-Martinez, M. Luengo-Oroz, Flooding through the lens of mobile phone activity, in: Global Humanitarian Technology Conference (GHTC), 2014 IEEE, 2014, pp. 279–286. doi:10.1109/GHTC.2014.6970293.

[6] F. M. Grath, Mobile network restoration & humanitarian response, Tech. rep., GSMA, http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2014/02/Mobile-Network-Restoration-and-Humanitarian-Response.pdf (2014).

[7] A. Nakanishi, "119-Ban de 'kyūmei dorōn', kusuri ya AED todokeru... konshū ni jisshō jikken" １１９番で「救命ドローン」、薬やＡＥＤ届ける...今秋に実証実験 ["Lifesaving drone" delivers medicine and AED when you call 119... experiments this fall], Yomiuri Online News commentary.

[8] S. Ramchurn, J. Fischer, Y. Ikuno, F. Wu, J. Flann, A. Waldock, A study of human-agent collaboration for multi-uav task allocation in dynamic environments, in: International Joint Conference on Artificial Intelligence, 2015.

[9] A. Brutschy, N.-L. Tran, N. Baiboun, M. Frison, G. Pini, A. Roli, M. Dorigo, M. Birattari, Costs and benefits of behavioral specialization, Robotics and Autonomous Systems 60 (11) (2012) 1408 – 1420, towards Autonomous Robotic Systems 2011. doi:http://dx.doi.org/10.1016/j.robot.2012.05.018.

[10] J. Munoz-Castaner, P. C. Soto, F. Gil-Castineira, F. J. Gonzalez-Castano, I. Ballesteros, A. di Giovanni, P. C. Villar, Your phone as a personal emergency beacon: A portable gsm base station to locate lost persons, IEEE Industrial Electronics Magazine 9 (4) (2015) 49–57. doi:10.1109/MIE.2015.2484922.

[11] O. Andryeyev, A. Rubina, O. Golokolenko, O. Artemenko, A. Mitschele-Thiel, Skysail: A flexible software-defined radio enabled micro aerial vehicle, in: 2016 25th International Conference on Computer Communication and Networks (ICCCN), 2016, pp. 1–6. doi:10.1109/ICCCN.2016.7568575.

[12] F. M. D. Fave, A. Farinelli, A. Rogers, N. Jennings, A methodology for deploying the max-sum algorithm and a case study on unmanned aerial vehicles, in: IAAI 2012: The Twenty-Fourth Innovative Applications of Artificial Intelligence Conference, 2012, pp. 2275–2280.

[13] Y. Liu, J. Yang, Y. Zheng, Z. Wu, M. Yao, Multi-robot coordination in complex environment with task and communication constraints, International Journal of Advanced Robotic Systems 10 (229).

[14] M. Nanjanath, M. Gini, Repeated auctions for robust task execution by a robot team, Robotics and Autonomous Systems 58 (7) (2010) 900 – 909, advances in Autonomous Robots for Service and Entertainment. doi:http://dx.doi.org/10.1016/j.robot.2010.03.011.

[15] S. Bernardini, M. Fox, D. Long, J. Bookless, Autonomous search and tracking via temporal planning, in: Proceedings of the 23st International Conference on Automated Planning and Scheduling (ICAPS-13), 2013.

[16] E. Burns, J. Benton, W. Ruml, S. Yoon, M. Do, Anticipatory on-line planning, in: International Conference on Automated Planning and Scheduling, 2012.

[17] P. Scerri, R. Glinton, S. Owens, D. Scerri, K. Sycara, Geolocation of RF emitters by many UAVs, in: "AIAA InfotechAerospace 2007 Conference and Exhibit", American Institute of Aeronautics and Astronautics, 2007. doi:doi:10.2514/6.2007-2858.

[18] S. Waharte, N. Trigoni, Supporting search and rescue operations with uavs, in: Emerging Security Technologies (EST), 2010 International Conference on, 2010, pp. 142–147. doi:10.1109/EST.2010.31.

[19] A. I. Coles, D. Long, P. Rendell, Experiences with temporal planning, in: Proceedings of the Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG), 2010.

[20] B. Grocholsky, J. Keller, V. Kumar, G. Pappas, Cooperative air and ground surveillance, Robotics Automation Magazine, IEEE 13 (3) (2006) 16–25. doi:10.1109/MRA.2006.1678135.

[21] R. Zivan, K. Sycara, Cooperation between search and surveillance agents in dcop_mst, in: Workshop 18 The Twelfth International Workshop on Distributed Constraint Reasoning DCR 2010, 2010.

[22] R. Boumghar, S. Lacroix, Over the hill and far away: aerial/ground cooperation for long range navigation, in: Proceedings of the 14th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Paris (France), World Scientific, 2011, pp. 215–222.

[23] 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. arXiv:http://ijr.sagepub.com/content/23/9/939.full.pdf+html, doi:10.1177/0278364904045564.

[24] Z. Beck, W. L. Teacy, A. Rogers, N. R. Jennings, Online planning for collaborative search and rescue by heterogeneous robot teams, in: AAMAS 16: 15th Int. Conf. on Autonomous Agents and Multi-Agent Systems, 2016.

[25] S. Yoon, A. Fern, R. Givan, S. Kambhampati, Probabilistic planning via determinization in hindsight, in: Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2, AAAI'08, AAAI Press, 2008, pp. 1010–1016.

[26] S. K. Gan, R. Fitch, S. Sukkarieh, Real-time decentralized search with inter-agent collision avoidance, in: Robotics and Automation (ICRA),

2012 IEEE International Conference on, 2012, pp. 504–510. doi:10.1109/ICRA.2012.6224975.

[27] E. Gelenbe, N. Schmajuk, J. Staddon, J. Reif, Biologically inspired autonomous systems autonomous search by robots and animals: A survey, Robotics and Autonomous Systems 22 (1) (1997) 23 – 34. doi:http://dx.doi.org/10.1016/S0921-8890(97)00014-6.

[28] A. Marjovi, L. Marques, Multi-robot olfactory search in structured environments, Robotics and Autonomous Systems 59 (11) (2011) 867 – 881. doi:http://dx.doi.org/10.1016/j.robot.2011.07.010.

[29] B. Yang, Y. Ding, Y. Jin, K. Hao, Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis, Robotics and Autonomous Systems 72 (2015) 83 – 92. doi:http://dx.doi.org/10.1016/j.robot.2015.05.001.

[30] J. L. Baxter, E. K. Burke, J. M. Garibaldi, M. Norman, Shared potential fields and their place in a multi-robot co-ordination taxonomy, Robotics and Autonomous Systems 57 (10) (2009) 1048 – 1055, 5th International Conference on Computational Intelligence, Robotics and Autonomous Systems (5th CIRAS). doi:http://dx.doi.org/10.1016/j.robot.2009.07.023.

[31] C. Crispin, A. Sobester, An intelligent, heuristic path planner for multiple agent unmanned air systems, in: AIAA SciTech 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, 2015.

[32] D. J. Grymin, C. B. Neas, M. Farhood, A hierarchical approach for primitive-based motion planning and control of autonomous vehicles, Robotics and Autonomous Systems 62 (2) (2014) 214 – 228. doi:http://dx.doi.org/10.1016/j.robot.2013.10.003.

[33] G. Murtaza, S. Kanhere, S. Jha, Priority-based coverage path planning for aerial wireless sensor networks, in: Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on, 2013, pp. 219–224. doi:10.1109/ISSNIP.2013.6529792.

[34] L. Chaimowicz, V. Kumar, A framework for the scalable control of swarms of unmanned ground vehicles with unmanned aerial vehicles, in: In Proceedings of the 10th International Conference on Robotics and Remote Systems for Hazardous Environments, 2004.

[35] P. Doherty, P. Rudol, A uav search and rescue scenario with human body detection and geolocalization, in: M. Orgun, J. Thornton (Eds.), AI 2007: Advances in Artificial Intelligence, Vol. 4830 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 1–13. `doi:10.1007/978-3-540-76928-6\_1`.

[36] D. Jing, C. Jian, S. Min, Cooperative task assignment for heterogeneous multi-uavs based on differential evolution algorithm, in: Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on, Vol. 2, 2009, pp. 163–167. `doi:10.1109/ICICISYS.2009.5358296`.

[37] A. Allahverdi, C. Ng, T. Cheng, M. Y. Kovalyov, A survey of scheduling problems with setup times or costs, European Journal of Operational Research 187 (3) (2008) 985 – 1032. `doi:http://dx.doi.org/10.1016/j.ejor.2006.06.060`.

[38] G. Rabadi, G. C. Anagnostopoulos, M. Mollaghasemi, A heuristic algorithm for the just-in-time single machine scheduling problem with setups: a comparison with simulated annealing, The International Journal of Advanced Manufacturing Technology 32 (3) (2007) 326–335. `doi:10.1007/s00170-005-0333-1`.

[39] G. Rabadi, M. Mollaghasemi, G. C. Anagnostopoulos, A branch-and-bound algorithm for the early/tardy machine scheduling problem with a common due-date and sequence-dependent setup time, Computers & Operations Research 31 (10) (2004) 1727 – 1751. `doi:http://dx.doi.org/10.1016/S0305-0548(03)00117-5`.

[40] S. Radhakrishnan, J. A. Ventura, Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times, International Journal of Production Research 38 (10) (2000) 2233–2252. `arXiv:http://dx.doi.org/10.1080/00207540050028070`, `doi:10.1080/00207540050028070`.

[41] S. Yoon, A. Fern, R. Givan, Ff-replan: A baseline for probabilistic planning, in: ICAPS, Vol. 7, 2007, pp. 352–359.

[42] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, A survey of monte carlo tree search methods, Computational Intelligence and AI in Games, IEEE Transactions on 4 (1) (2012) 1–43. doi:10.1109/TCIAIG.2012.2186810.

[43] C. F. Chung, T. Furukawa, Coordinated pursuer control using particle filters for autonomous search-and-capture, Robotics and Autonomous Systems 57 (6–7) (2009) 700 – 711. doi:http://dx.doi.org/10.1016/j.robot.2008.11.002.

[44] Y. Kawata, The great hanshin-awaji earthquake disaster : Damage, social response, and recovery, Journal of natural disaster science 17 (2) (1995) 1–12.

[45] J. F. C. Kingman, Poisson processes, Vol. 3, Oxford university press, 1992.

[46] D. Jones, Multi-agency command support, Crisis Response 9 (2013) 20–23.

[47] B. Gerkey, M. Mataric, Sold!: auction methods for multirobot coordination, Robotics and Automation, IEEE Transactions on 18 (5) (2002) 758–768. doi:10.1109/TRA.2002.803462.

[48] Z. Beck, Collaborative search and rescue by autonomous robots, Ph.D. thesis, University of Southampton, University Road, Southampton, SO17 1BJ, UK (12 2016).

[49] S. H. Song, D. H. Kim, C. H. Chang, Experimental reliability analysis of multi-uav simulation with tmo-based distributed architecture and global time synchronization, in: 2010 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, 2010, pp. 211–218. doi:10.1109/ISORCW.2010.23.

[50] UNOSAT, Haiti earthquake 2010: Remote sensing based building damage assessment data, http://www.unitar.org/unosat/

haiti-earthquake-2010-remote-sensing-based-building-damage-assessment-data (March 2010).

[51] MapAction, Haiti: Earthquake – sar competed sectors port-au-prince (January 2010).

[52] C. De Ville de Goyet, F. Grünewald, J. P. Sarmiento, Health responses to the earthquake in haiti january 2010: Lessons to be learned for the next massive sudden-onset disaster, Tech. rep., Pan American Health Organization (PAHO) (2011).

[53] N. Morrow, N. Mock, A. Papendieck, N. Kocmich, Independent evaluation of the ushahidi haiti project, Development Information Systems International 8.

[54] ING Robotic Aviation Serenity, continuous eyes in the sky, http://ingrobotic.com/drones/serenity-fixed-wing-aircraft/, accessed: 2017-08-19.

[55] senseFly Albris, the intelligent mapping & inspection drone, https://www.sensefly.com/drones/albris.html, accessed: 2017-08-19.

[56] M. Mulero-Pázmány, R. Stolper, L. D. van Essen, J. J. Negro, T. Sassen, Remotely piloted aircraft systems as a rhinoceros anti-poaching tool in africa, PLoS ONE 9 (1) (2014) 1–10. doi:10.1371/journal.pone.0083873.

[57] J. Katrasnik, F. Pernus, B. Likar, A survey of mobile robots for distribution power line inspection, IEEE Transactions on Power Delivery 25 (1) (2010) 485–493. doi:10.1109/TPWRD.2009.2035427.