The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

Ramchurn, Sarvapali D. and Fischer, Joel E. and Ikuno, Yuki and Wu, Feng and Flann, Jack and Waldock, Antony (2015) A study of human-agent collaboration for multi-UAV task allocation in dynamic environments. In: International Joint Conference on Artificial Intelligence (IJCAI-15), 25-31 July 2015, Buenos Aires, Argentina.

A Study of Human-Agent Collaboration for Multi-UAV Task Allocation in Dynamic Environments*

S . D. Ramchurn,[1] **Joel E. Fischer,**[2] **Yuki Ikuno,**[1] **Feng Wu,**[3] **Jack Flann,**[1] **Antony Waldock**[4]
[1]Electronics and Computer Science, University of Southampton, UK
[2]Mixed Reality Lab, Dept. of Computer Science, University of Nottingham, UK
[3]School of Computer Science and Technology, USTC, China
[4]BAE Systems, UK

Abstract

We consider a setting where a team of humans oversee the coordination of multiple Unmanned Aerial Vehicles (UAVs) to perform a number of search tasks in dynamic environments that may cause the UAVs to drop out. Hence, we develop a set of multi-UAV supervisory control interfaces and a multi-agent coordination algorithm to support human decision making in this setting. To elucidate the resulting interactional issues, we compare manual and mixed-initiative task allocation in both static and dynamic environments in lab studies with 40 participants and observe that our mixed-initiative system results in lower workloads and better performance in re-planning tasks than one which only involves manual task allocation. Our analysis points to new insights into the way humans appropriate flexible autonomy.

1 Introduction

Unmanned aerial vehicles (UAVs) provide a unique capability to emergency responders in the aftermath of major disasters to carry out situational awareness tasks. Thus, they are able to fly quickly over large areas, provide live video and thermal imagery, and collect valuable environmental data (e.g., radiation or temperature) in order to identify casualties, damaged infrastructure, or radiation sources. In order to deploy as many of UAVs as possible over a large area, *human supervisory control* mechanisms have recently been developed to allow multiple UAVs to be deployed in $m < n$ operator-to-vehicle ratio (where m is the number of operators and n is the number of UAVs) [Cummings *et al.*, 2010a]. However, such deployments come with a number of challenges. First, operators have to determine, in real-time, the shortest paths and schedules for individual UAVs. Second, they have to decide on the number of UAVs to send to each area of importance to ensure they cover the most important areas as quickly as possible. Third, given that UAVs operate in a dynamic and uncertain (possibly adversarial) environment, they may suffer damage from environmental hazards, run out of fuel, and eventually drop out of the system. In such situations, the operators may need to quickly re-plan paths for the remaining UAVs in order to achieve their mission objectives.

To address these challenges, agent-based decentralised coordination algorithms have been developed for multi-UAV coordination and task allocation [Delle Fave *et al.*, 2012; Ramchurn *et al.*, 2010] while decision support systems have been developed to structure the interactions between humans and UAVs and help humans cope with high levels of workload in managing sets of UAVs that execute simple waypoint following tasks [Cummings *et al.*, 2007; de Greef *et al.*, 2010; Mekdeci and Cummings, 2009] (see Section 2 for more details). However, very few of these approaches have addressed the interactional issues that arise when human operators are supported by agents that autonomously coordinate (negotiate and agree on a plan) to allocate tasks among themselves. An exception to this is the work by [Cummings *et al.*, 2010b; de Greef *et al.*, 2010] and [Goodrich *et al.*, 2007] but they focus on re-planning triggers and frequency and how different levels of autonomy given to individual robots can affect team performance respectively. Hence, they do not consider how flight paths can be collaboratively constructed (by multiple humans and agents) nor how to deal with random UAV dropouts that can severely impact the operators' plans.

Against this background, we develop and study a set of multi-UAV coordination interfaces that integrate the notion of flexible autonomy, that is, where control may dynamically shift between a team of humans and a team of agents. In more detail, we develop a set of interfaces for UAV task allocation in disaster response situations based on requirements given by emergency responders[1] and go on to develop novel interaction mechanisms for team of humans to manage multi-UAV task allocation. Specifically, we develop interfaces that allow operators to influence plans computed by a state-of-the art decentralised coordination algorithm, called max-sum [Rogers *et al.*, 2011; Delle Fave *et al.*, 2012]. In particular, we extend max-sum, to incorporate human input by transforming human preferences for specific task allocations into constraints that the algorithm needs to work against. Given this, through a number of lab studies, we evaluate the performance of human-agent teams with different levels of autonomy (with and without max-sum) and dynamism (where

*We acknowledge support from the ASUR-DSTL Programme and the ORCHID Project (EP/I011587/1).

[1]Hampshire Fire&Rescue (http://www.hantsfire.gov.uk) and RescueGlobal (http://www.rescueglobal.org).

UAVs may or may not drop out). Based on our analysis of interaction logs, NASA TLX workload metrics, 30 hours of videos, and interviews with 40 participants, we show that our flexible autonomy interfaces have a positive impact on user performance and workload in re-planning tasks when UAVs may drop out.

The rest of this paper is structured as follows. Section 2 presents related work while Section 3 describes the human-agent collaboration challenges faced when coordinating multiple UAVs. Section 4 then describes the system while Section 5 presents our experimental setup and results of our experiments. Section 6 analyses the results and presents novel design guidelines for human-agent collaboration. Finally Section 7 concludes.

2 Background

A number of decentralised coordination algorithms for task allocation have been developed by the multi-agent systems community recently [Amador *et al.*, 2014; Delle Fave *et al.*, 2012; Macarthur *et al.*, 2011]. However, to our knowledge, none of these approaches have been studied with human operators. Hence, to date, it is unclear how they can actually be used in real-world settings, and specifically, in multi-UAV task allocation settings. In this paper, we set out to demonstrate how such coordination techniques can be embedded within a supervisory control interface.

Supervisory control interfaces for autonomous systems has instead been an active area of research in the Human Factors (HF), Human Robot Interaction (HRI), and Human-Computer Interaction (HCI) domains. A key issue they try to address is that plans computed by autonomous systems are typically brittle as they strictly conform to initially set design decisions and ignore the contextual decisions that humans need to make (e.g., weather conditions that may lead to UAVs dropping out, or the changing priorities of the mission) [Silverman, 1992; Smith *et al.*, 1997]. For example, [Miller and Parasuraman, 2007] developed a 'Playbook' of tasks for automated agents to perform when faced with certain situations (upon request from human controllers). Moreover, [Lewis *et al.*, 2009] developed interfaces to help an operator interact with large numbers of UAVs (hundreds). Bertuccelli et al. [Bertuccelli *et al.*, 2010] instead, developed operator models for UAV control and, under simulations, study the performance of their 'human-in-the-loop' algorithms whereby operators are unreliable detectors and the algorithm may not perform well in search tasks. While these approaches relate to our context, they do not specifically study how decentralised coordination algorithms can be embedded in such systems.

Closer to our work, Cummings et al. [Cummings *et al.*, 2010b] evaluate a mixed-initiative system with a *single* pilot and with an auction-based task allocation scheme. However, they focus on how *often* an operator should be asked to re-plan, and through a set of lab studies, show that operator performance drops with too many frequent re-planning requests. In turn, our approach considers re-planning as a human-driven exercise and lets the operator decide on the level of autonomy given to agents, from waypoint specification to task allocation.

In a similar vein [Goodrich *et al.*, 2007] study how different levels of autonomy (as per [Sheridan and Verplank, 1978]) given to teams of agents can impact on performance and workload. Specifically, they show that reliance on team autonomy (i.e., a team allocates tasks amongst its members independently of human control) results in neglect from the operator though it reduces workload. Instead they suggest there should be shifts between adjustable (where the autonomy of manually controlled), adaptive autonomy (where the agent changes its behaviour by itself), and team autonomy. In line with [Jennings *et al.*, 2014], we capture these shifts in autonomy levels in through the notion of flexible autonomy. In particular, in the next section we describe how different autonomy levels fit the different situations encountered when multiple humans need to interact with multiple UAVs.

3 Managing Autonomy in Multi-UAV Deployments

In typical disaster response settings (e.g., Haiti in 2010 or Typhoon Haiyan in 2013) where buildings have collapsed or roads are blocked by debris, UAVs are deployed by emergency response teams to gather situational awareness (through the use of video cameras, thermal imaging) about the location of key emergencies (i.e., casualties, fires, resources to be collected) so that the responders can prioritise their tasks.[2] Such deployments will typically fall under the responsibility of tactical commanders (aka Silver) and operational (aka Bronze) teams. While Silver commanders (one to monitor the camera feed and one to allocate tasks) decide on and monitor a tactical allocation of UAVs to specific areas of the disaster struck region, Bronze operators monitor the physical state of individual UAVs (in one-to-one or one-to-many operator ratios) and sometimes take control of individual UAVs. We elaborate on these interactions in the next subsection and describe how they impact on UAV autonomy. Building upon this, we elaborate on the human-agent collaboration challenges they induce.

3.1 UAV Autonomy

UAVs may be given different levels of autonomy to carry out imagery collection tasks and or to support humans in allocating tasks [Goodrich *et al.*, 2007]:

1. **Teleoperation**: a Bronze operator can tele-operate a UAV in order to further investigate an area (from its video feed) or bring the UAV back to base along a safe route if it cannot complete its mission (e.g., due to lack of fuel or damage). Here, UAVs are only given navigational autonomy (i.e., maintain altitude, speed).
2. **Waypoints**: geo-locations can be specified by human operators to scan an area in potentially complex ways. Here we assume that Silver commanders are responsible for specifying waypoints. In this case, UAVs have more advanced navigational autonomy (i.e., adjust speed and altitude, obstacle avoidance).
3. **Region scanning**: an area is specified (by Silver) for a set of UAVs to plan paths around and scan autonomously. Here, we specialise the definition of region scanning to require multiple UAVs. The selection

[2]Our scenarios draw upon the experience of Rescue Global (http://www.rescueglobal.org) and Hampshire Fire and Rescuel (http://www.hantsfire.gov.uk).

of UAVs to scan an area can be automated through the use of a multi-agent coordination algorithm (e.g., in Section 4.1), and hence have path planning autonomy.

4. **Flexible Coordination**: given a set of tasks (waypoints or regions) specified by Silver commanders, the UAVs run a decentralised coordination algorithm to allocate tasks among themselves [Delle Fave *et al.*, 2012]. Tasks can be given different priorities if needed. UAVs may also re-allocate tasks dynamically (i.e., on-the-fly) should any of them fail or should tasks be changed by Silver operators. This is the highest level of autonomy (or team autonomy as in [Goodrich *et al.*, 2007; Sheridan and Verplank, 1978]).

These different modes of delegation of tasks to UAVs result in different levels of workload and collaboration challenges for the operators. We discuss these next.

3.2 Human-Agent Collaboration Challenges

The goal of increased UAV autonomy is to reduce the workload of human operators [Goodrich and Schultz, 2007]. However, UAVs that autonomously coordinate and their supervisory control by different actors (Bronze and Silver) is likely to increase collaboration complexity if not carefully managed [Cummings *et al.*, 2007; Smith *et al.*, 1997]. For example, Silver commanders can request Bronze operators to teleoperate a UAV once it has reached its target location to get better imagery. In turn, however, the Bronze operator can decide, at short notice, to bring back a UAV should weather conditions deteriorate. In general, the fact that multiple humans can take full or partial control of a UAV can significantly affect the tactical objectives of Silver commanders and may require them to revise their plans continually, wasting time and effort. In addition to this, if UAVs are allowed to decide on flight paths for region scanning or allocate tasks among themselves using complex coordination algorithms, it is unclear how Silver commanders would interpret such behaviours. Moreover, if Bronze operators can control of individual UAVs, the plans recalculated on-the-fly by the UAVs may not fit the expectation of Silver commanders. Hence, our aim is to try and answer the following questions, with a focus on the task of Silver commanders, in order to address some of these interactional challenges. Specifically, (i) How do Silver operators interpret and appropriate the decentralised coordination algorithm used by the agents/UAVs? (ii) Does the use of a decentralised coordination algorithm help Silver commanders perform better (in terms of tasks completed, time taken, targets found) after suffering drop-outs (i.e., loss of UAVs)? (iii) Is collaboration within human teams affected by autonomy?

Thus, in what follows, we develop and evaluate a multi-UAV mixed-initiative coordination system.[3] In particular, we focus on the work by Silver operators: (i) assigning tasks and flight paths to UAVs, (ii) monitoring the video feeds from the UAVs, and (iii) reacting to drop-outs. To this end, we develop a novel multi-UAV coordination system and go on to evaluate it through a number of lab studies.

[3] Based on requirements drawn from a number of workshops and meetings with Silver operators and Bronze level pilots from Hampshire Fire and Rescue and Rescue Global.

4 System Description

The system consists of a simulation engine that generates the movement of UAVs,[4] a set of user interfaces (UIs) for tactical UAV task allocation, and a flexible coordination module. In more detail, the simulation engine stores positioning data and simulates the behaviours (flying via waypoints, hovering, dropped-out) of the UAVs based on the plans chosen by the users through the interfaces and simulated events in the environment. The flexible coordination module allows agents, representing the UAVs in the back-end, to exchange messages as per the max-sum algorithm and receives input from users to modify the plan computed by max-sum [Rogers *et al.*, 2011]. A *camera view* and a *planner view*, allow two operators to allocate tasks to the simulated UAVs, using both manual control and automatic optimisation, and monitor their performance. We detail these components in the following subsections.

4.1 Flexible Coordination Module

The flexible coordination module continuously monitors the state of the UAVs and tasks defined in the system and dynamically determines a task allocation plan so as to minimise the time that the UAVs take to complete their allocated task(s). We employ the *max-sum* algorithm for decentralised coordination. As shown in [Rogers *et al.*, 2011; Delle Fave *et al.*, 2012], max-sum provides good approximate solutions to challenging dynamic decentralised optimisation.[5] However, max-sum does not explicitly handle constraints imposed by human operators. For example, if after running max-sum, agent A is tasked to go to point X, agent B to point Y, and agent C to point Z, there is no explicit method for human operators to partially modify the plan such that agent A goes to point Y, and B and C *automatically* re-allocate points Y and Z among themselves in the best way possible. Hence, to cater for such situations, in what follows we first provide a basic description of the max-sum algorithm (see details in [Macarthur *et al.*, 2011; Delle Fave *et al.*, 2012]) and then explain how it can be modified to take into account human input.

The max-sum Algorithm

The max-sum algorithm works by first constructing a factor graph representation of a set of tasks (each representing a point or waypoints UAVs are meant to fly to) and the set of agents (each representing a UAV) and then sets a protocol for an exchange of messages between different nodes in the factor-graph. The factor graph is a bi-partite graph where vertices represent agents and tasks, and edges the dependencies between them (see [Rogers *et al.*, 2011] for details on how this graph is constructed). Given this, max-sum defines two types of messages that are exchanged between agent (variable) nodes and task (factor) nodes:

- From agent i to task j:

$$\forall_{x_i \in D_i}\ q_{i \to j}(x_i) = \alpha_{i \to j} + \sum_{k \in M(i) \setminus j} r_{k \to i}(x_i) \qquad (1)$$

[4] Our system simulates the movement of quadcopters that can hover and follow waypoints such as A. R. Drones.

[5] Other decentralised coordination algorithms could also be used here (e.g., ADOPT or BnB-ADOPT [Gutierrez *et al.*, 2011]) as we only adapt the graph over which they exchange messages to compute a solution.

where x_i is a task assignment of agent i, D_i is a set of all possible assignments of agent i, $M(i)$ denotes the set of indices of the task nodes connected to agent i, and $\alpha_{i\to j}$ is a scalar chosen such that $\sum_{x_i\in D_i} q_{i\to j}(x_i) = 0$.

- From task j to agent i:

$$\forall_{x_i\in D_i}\ r_{j\to i}(x_i) = \max_{\mathbf{x_j}\setminus x_i}\left[U_j(\mathbf{x_j}) + \sum_{k\in N(j)\setminus i} q_{k\to j}(x_k)\right] \quad (2)$$

where $U_j \in \Re$ is the utility function of task j, $N(j)$ denotes the set of indices of the agent nodes connected to task j, and $\mathbf{x_j}$ is a vector of task assignments $\langle x_{j1}, \cdots, x_{jk}\rangle$ for the agents that are relevant to task j. As in [Delle Fave *et al.*, 2012] the utility function in our scenario includes the travel time to a task, the priority level of the task, and the suitability of the agent(s) for the task (e.g., different agents may have different affinities for a task given its onboard sensors). Each task may take more than one agent.

Notice that both $q_{i\to j}(x_i)$ and $r_{j\to i}(x_i)$ are scalar functions of assignment $x_i \in D_i$. The largest calculation that any agent performs, as shown in Eq. 2, is exponential only in the number of its neighbours, which is typically much less than the total number of agents. Thus, the max-sum algorithm can scale to relatively large problems with many agents depending on their interaction structures.

For an acyclic factor graph (which can be constructed using techniques defined in [Rogers *et al.*, 2011]), these messages represent the maximum aggregated value for each assignment $x_i \in D_i$ of agent i over the respective components of the graph formed by removing the dependency between task j and agent i. Thus, the marginal function of each assignment x_i is calculated by:

$$z_i(x_i) = \sum_{j\in M(i)} r_{j\to i}(x_i) = \max_{\mathbf{x}\setminus x_i}\sum_{j=1}^{m} U_j(\mathbf{x_j}) \quad (3)$$

after which the assignment of x_i can be selected by:

$$x_i^* = \arg\max_{x_i\in D_i} z_i(x_i) \quad (4)$$

In our system, Eq. 1 is computed by the UAV agents while the more computational-intensive calculations (i.e., Eq. 2) are processed by a base station (simulated in the system). This is critical because the computational resources of the UAVs are usually very limited. In each step, the base station submit tasks to the UAVs so they know what are their possible assignments and who are their neighbours (i.e., the task nodes) in the factor graph. Then, the UAVs and the base station exchange messages as detailed by Eqs. 1 and 2. Finally, the best assignment is selected by each UAV using Eq. 4.

Integrating Human Input and Coping with Drop-Outs

Given a set of tasks and agents, max-sum computes shortest paths for each UAV to the tasks as well as taking into account the priorities set for each task and the type of UAV required. However, this assignment may not be preferred by the human operators as it may conflict with their priorities and expectations about flight paths. For example, a UAV may be allocated by max-sum to fly from its position in the East to a task in the West but the human operators may, instead, prefer a UAV to fly from the South to the same task in order to provide imagery over the area covered by that path, which may be more important than the lateral traversal from East to West. Moreover, if the priority levels are too coarse, max-sum will not differentiate between tasks of very similar priority levels.

Hence, given a plan computed by max-sum, using our planner interfaces (see Section 4.2), users can *manually* allocate UAVs. These manual allocations specify a task-agent pair (i, j). Given this, for a given agent i, we then define $D_i = \{j\}$. This effectively results in the deletion of all edges in the factor graph that connect the agent node i with other task nodes apart from that of j. This, in turn, forces max-sum to only allocate agent i to task j (as per Eq. (1) and (2)) messages, and if two (or more) agents are required by task j, another agent is chosen based on this restriction.

Now, a key property of max-sum is its ability to recover from changes to the factor graph (nodes and edges). Crucially, any addition (e.g., a new task is added) or removal (e.g., a drop-out is suffered) of a node in the factor graph will simply result in new messages being exchanged across the graph until the nodes converge on a new solution. However, such a solution may not always be acceptable to the human commanders and hence, as we show next, we explicitly hand-over control to humans to check, modify (if need be), and enact the plan suggested by max-sum following a drop-out. As we discuss later, such handing over of control is crucial to ensuring humans can trust plans computed by max-sum.

4.2 User Interfaces

Our interfaces are designed to allow human operators, at any point during a mission, to request plans from the max-sum algorithm, visualise the plans, and modify these plans.[6] As in [Cummings *et al.*, 2010b], we do not show the max-sum messages exchanged between the agents. Rather, participants were explained how it generally works and told that there may be a discrepancy between their plans and those of the agent.[7] We next elaborate on the two views (on two full HD screens) where each one is operated by one Silver commander (i.e., two sitting next to each other).

Planner View

This is the main planning tool that provides both monitoring and planning capabilities (see Figure 1), which we separated as in [Cummings *et al.*, 2010b]. These capabilities are accessible in two modes (accessible through the tabs on the top right), namely 'Task Edit' and 'Monitor'. We describe each of these modes in turn.

Task Edit Mode

This mode provides the user with a number of planning options (see Figure 1):

1. add/delete tasks (region or point): Users can create two types of tasks: (i) region tasks – this task requires two[8] UAVs to carry out a sweep search of the area selected

[6] See system operation video with users here: `http://bit.ly/uavstudyvideo` and a screen capture here: `http://bit.ly/uavscreen`.

[7] Unless the operators understand the limitations of the algorithm, they are likely to take up most agent suggestions even if wrong [Smith *et al.*, 1997].

[8] Pilot studies revealed users very rarely allocated more than two UAVs to one region task.

by the user using mouse clicks (ii) point tasks – a point selected in the map.

2. change/adapt the allocation of tasks to agents: In the manual mode, the user has to click on the UAV and click on a task. In the mixed-initiative setting, max-sum produces an allocation (on request or following a drop-out) that can be changed by clicking on a UAV and allocating it to another task. max-sum then adapts its allocation to fit to the constraint set by the user (as per Sec. 4.1).

3. add way points to the paths taken by the UAVs: this applies to paths chosen to point tasks, whereby users can adjust the path taken by a UAV.

Once an allocation of UAVs to tasks has been chosen, the user can verify the completion time of the tasks using the side bar widgets and then decide to execute the plan.

Monitor Mode

This mode shows the current status of the allocation (see Figure 1). Paths chosen by the max-sum algorithm are differentiated from those manually specified by the users are shown in orange. Once a region scanning task has been completed, the region scanned turns green. A region task is deemed completed when UAVs have covered its area and a point task is considered completed when the allocated UAV has reached that task. Once a point task is completed, the task disappears from the map.

Camera View

The camera view provides multiple live streams of the aerial view from GoogleMaps *aerial* view (see Figure 1). The images displayed are taken at real GPS locations of the UAVs in the disaster space (as defined by the scenario). The centre of the display shows an expanded view of a chosen UAV (when one side tile is clicked).

Targets, identified by specific blue icons are positioned at specific points in the space considered and displayed on the aerial view whenever the UAV flies over it. The user can then click on the icon and annotate it with the matching description. Once a target has been identified (either correctly or incorrectly), the colour of the icon changes to its detected state both on the Camera view and the Planner view. The two operators can also collaborate to identify targets by sharing different parts of the screen.

5 Experimental design and Results

We designed a lab study in order to evaluate our multi-UAV control interfaces in use. A particular challenge we are interested in evaluating is how well our mixed-initiative task allocation method supports operators in responding to unexpected UAV drop-outs, which are caused by technical failures or non-compliance by individual UAV supervising pilots (which may be legitimate due to a more immediate understanding of the situation 'on the ground'). Drop-outs require Silver to adjust the current operation.

Participants and Conditions. 40 participants were recruited from different University departments with 15 females and 25 males (26 Computer Science undergraduates and others from non-CS subject areas). We randomly created teams of two operators from this set (1 camera operator and 1 planner view operator). We adopted a within-subjects (counterbalanced repeated measures) 2X2 factorial design to evaluate how our interface designs fare in practice, i.e., how they affect performance (number of targets discovered) and collaboration between operators (number of turns-at-talk). A within-subjects design was used to reduce error variance (the effects of individual differences). We manipulate two independent variables, *allocation method* (manual vs. mixed-initiative) and *UAV drop-outs* (none v.s. three at random). Users use two versions of our planner view r that are identical apart from the allocation method:

1. Manual Allocation (MA) – Users can manually assign tasks to individual UAVs using region and point tasks.
2. Mixed-Initiative Task Allocation (MI) – Users use recommendations from max-sum to allocate tasks.

Further, we have manipulated the experimental design so that we can evaluate the effects of UAV drop-outs, as follows:[9]

1. No drop outs (ND) – UAVs function as planned.
2. Drop-outs (DO)– three UAVs drop out one-by-one at unexpected (pseudo-random) times (we divide the duration of the experiment in three and drop out each UAV randomly in one of these slots) during execution of a run.

Given this, counterbalancing was achieved with two groups: Group A (10 pairs of operators) experienced MA-ND, MA-DO, then MI-ND, MI-DO, Group B (10 pairs of operators) experienced MI-ND, MI-DO, MA-ND, then MA-DO.

Hypotheses. We are particularly interested in how Silver operators perform using our mixed-initiative allocation (i.e., with max-sum) method in comparison to a control method (manual), in dealing with UAV drop outs. We postulate the following set of testable hypotheses.

- H1 When UAVs drop out, Silver operators perform tasks significantly better, in terms targets found, efficiency, effort spent, and number of tasks completed, when provided with a mixed-initiative task allocation compared to manual task allocation.
- H2 If there are no UAV drop outs, the effort (defined by operator clicks) will be lower for the mixed-initiative task allocation compared to manual task allocation.
- H3 When UAVs drop out, we expect significantly more collaboration (defined in terms of number of turns taken) between the operators compared to when there are no UAV drop outs.
- H4 We expect that self-reported task workload is more correlated with MA than with MI task allocation.

Task design. The participants' task is to search and find targets collaboratively, using the interfaces provided (see Section 4). Given that in real-world conditions, UAVs are deployed based on reports from people on the ground, the participants are initially given hints as to where these targets might be. For example, (i) *A large number of casualties have been reported around the North-East of the base (Centre of*

[9]Counterbalancing for the order of ND/DO would require doubling the number of participants to 80 with, potentially, minimal impact on the results. Hence, we leave this as future work.

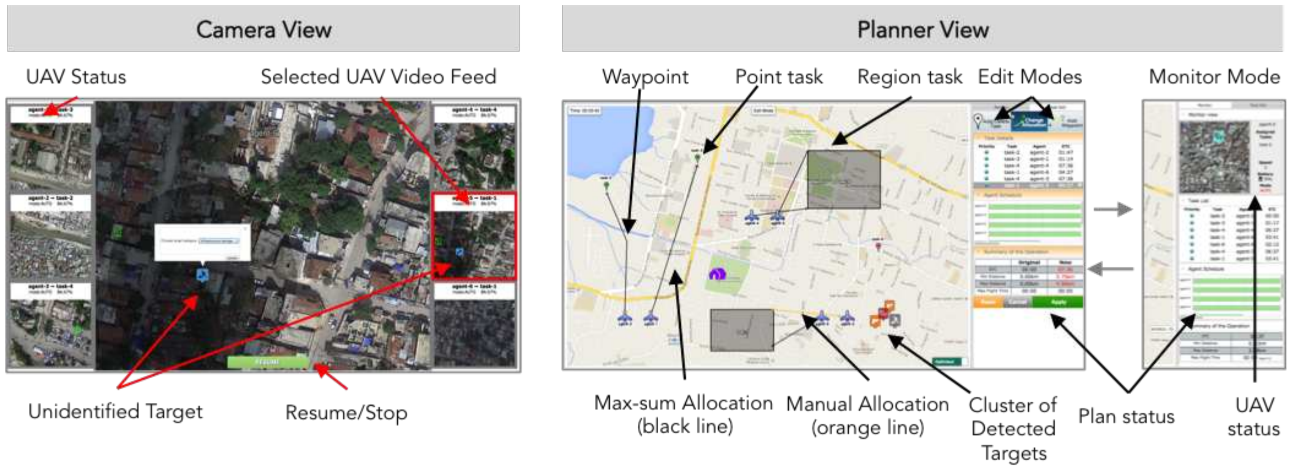

Figure 1: Camera view (left) for six UAVs, Planner View (right) in task edit mode and the monitor mode bar.

map) — to signify spread targets, (ii) *There is a group of targets in the East of the map (i.e, far right)* — to signify a cluster of targets, (iii) *There 'may' be a few targets in the South of the base but we are not sure* — to indicate low certainty of targets present. In each scenario there are 20 targets distributed across the map. Targets are spread such that there are two clusters of targets in different parts of the map, with a few dispersed in other parts. At the start of each run, the targets are 'hidden' (i.e., invisible) from the map and only unveiled through discovery of these using the camera view. The operators can individually tag these targets using a predefined list of tags. We also introduced a time-limited *discussion phase* prior to the mission starting. This was found to be useful for the users to collaboratively decide on where to place the tasks across the map. Stationery was provided to them to write down their thoughts.

Procedure. For each run of the experiment, two participants (one for each of the two interfaces) collaborate to view and control six UAVs at once to accomplish the given task. Prior to the experiment, the participants are asked to fill a consent form and a survey to capture demographic information and relevant previous experience, e.g., in flying real UAVs, and familiarity with strategy video games that exhibit similar properties to our scenario. They are then given a simple learning task to perform (search for three targets in the environment close to a point of interest), in order to induce some familiarity with the controls provided by the system and with the procedure.[10]

After each run, each participant filled out a NASA TLX form to collect self-reported measures of workload. Each run lasts a maximum of 10 minutes. We conduct pre and post interviews and all runs are recorded using a video camera to enable analysis of verbal collaboration between the two operators. Interactions with the system are logged and later triangulated with the video feed and audio data.

[10] In the three pilot runs we performed, users found the interface quite complex to understand initially.

5.1 Results

We report on our statistical analysis of study data in order to test our hypotheses. We present measures of performance (from system logs) and then turn to collaboration measures (from 30 hours of video analysis). Unless otherwise stated, we use repeated measures ANOVA (Analysis of Variance) to test the effects of independent variables (IVs): *allocation method* (IV1) (i.e., MA and MI) and *drop-outs* (IV2) (i.e., ND and DO) on parametric dependent variables ($\alpha = 0.05$).
Performance. Performance in the experiment was measured in terms of:

- *Number of targets found* — higher is better.
- *Efficiency*: number of targets found per task created — the more targets found per task created, the more efficient the task allocation method is.
- *Effort*: number of mouse clicks by the operator — the more mouse clicks (create waypoints, re-allocate UAVs, create tasks), indicates more effort to create plans.
- *Tasks completed*: a further efficiency metric, to complete a task the operators need to (i) create the task (ii) assign a UAV to it, (iii) the UAV needs to reach the task.

With Drop-outs
In the following we outline the results from repeated-measures ANOVA on the dependent variables (DVs).
No. of Targets Found. The allocation method (i.e., MI v/s MA) had no significant effect on the number of targets found ($p = 0.642$, $F(1, 19) = 0.223$) while *drop-outs* had a significant effect ($p < 0.001$, $F(1, 19) = 16.6$). The interaction effect *between allocation* method and *drop-outs* was not significant ($p = 0.2, F(1, 19) = 1.7$).
Efficiency. Neither the allocation method nor the number of drop-outs had a significant effect on the efficiency of the teams. For the allocation method, $p = 0.321, F(1, 19) = 1.037$, and for drop-out effect $p = 0.113, F(1, 19) = 2.769$. There is also no significant interaction effect between the factors ($p = 0.234, F(1, 19) = 1.153$).
Effort. Allocation method had a significant effect on the number of mouse clicks, with $p < 0.01, F(1, 19) = 33.22$. Furthermore, a paired-samples t-test on MA-DO and MI-DO

confirms that the number of clicks in the MI conditions are significantly lower than in the MA conditions (mean of 32 fewer clicks than MA, $p < 0.01$).

Tasks Completed. *Drop-outs* had a significant effect on tasks completed ($p = 0.03$, $F(1, 19) = 5.5$), while neither *allocation method* nor the interaction of the two IVs were significant. Further investigation showed that the trend for number of tasks completed in the MI-DO condition was higher ($M = 5.9, SD = 8.1$) than in MA-DO ($M = 2.6, SD = 1.8$). However, the difference was significant only for point tasks ($p = 0.041$), but not for region tasks ($p = 0.163$) or both task types combined ($p = 0.052$).

Without Drop-outs

There is no difference between MA and MI ($p = 0.186$ for targets/task, $p = 0.338$ for targets, $p = 0.915$ for tasks completed) in terms of targets found, in line with our expectations. However, there was a significant difference in *effort* ($p = 0.001, t = 4.169$).

Collaboration

We analysed 30 hours of video recordings of lab sessions to evaluate the collaboration between the plan editor and the camera view operator. We note the following results.

Collaborative Planning. We counted the turns taken by participants when discussing task allocation plans (on the planner view). *Turns* are defined according to Conversation Analysis as utterances in a sequentially organised conversation. We found that even if drop-out scenarios induced more collaboration than no drop-out scenarios, this effect was not significant ($p = 0.087$, $F(1, 18) = 3.262$). The allocation method also did not have a significant effect ($p = 0.17, F(1, 18) = 2.04$) on the number of turns taken.[11]

Collaborative Annotation. We counted the number of times the planner view operator helped the camera view operator annotate targets (i.e., they shared the annotation task). In this respect, we find that drop-outs have a significant effect on the number of annotation turns taken $p = 0.002, F(1, 18) = 13.49$. We also noted a significant interaction effect between allocation method and drop-outs with $p = 0.023$, $F(1, 18) = 6.194$.

Pre-Launch Planning. We counted the time taken by teams to plan where to send UAVs prior to starting the mission. Here we found that the allocation method had a significant effect ($p = 0.049$, $F(1, 19) = 4.434$). Thus, MI was found to induce less collaborative pre-launch planning.

Workload

Using NASA TLX workload metrics for the planner view operator we found that neither the allocation method nor the drop-outs individually had a significant effect but there was a significant interaction effect between the two factors ($p = 0.024$, $F(1, 19) = 6.025$). Thus, workload was found to be on average lower for MA-ND compared to MI-ND (28.9 v.s. 35.2) while workload was found to be higher for MA-DO than MI-DO (31.0 v.s. 29.9).

Summary and hypotheses

The analysis of performance metrics partially supports hypothesis H1, and fully supports hypothesis H2. While *allocation method* (IV1) and *drop-outs* (IV2) had no significant effect on *number of targets found* and *efficiency*, when drop-outs occurred we observed a significant effect on *effort* (more effort in the MA condition) and *tasks completed* (more tasks completed in the MI condition).

The analysis of collaboration between operators partially supports hypothesis H3, and partially supports H4. While no significant effects were observed on the *number of turns taken*, *drop-outs* had a significant effect on *collaborative annotations*, leading to significantly more collaborative annotating when drop-outs occurred; and *allocation method* had a significant effect on *pre-launch planning*, leading to more pre-launch planning in the manual condition, thus partially supporting H3. The IVs had a significant interaction effect on *self-reported workload*; participants reported lower workload for MA than MI when no drop outs occurred, but when drop-outs occurred, the workload for MA was reported as higher than for MI; thus partially supporting (and qualifying) H4.

Validity

We conducted tests to check the validity of our experimental design focusing on allocation method and order effects.

Manipulation check: task allocation. In order to test adoption (use) of the max-sum (mixed-initiative) task allocations we looked at manual task allocations in detail. To be deemed successful, we expect significantly fewer manual allocations for MI than for MA — absence of significance on the other hand would show that participants effectively override the MI task allocations, which would jeopardise the internal validity of the study. Manual task allocations are defined as selections of specific UAVs and assigning them to specific tasks individually. Participants made use of the option to manually edit task allocations in the mixed-initiative conditions occasionally (MI-ND: μ=5.7, MI-DO:μ=4.9), but significantly less often than in the manual conditions (MA-ND:μ=27.4, MA-DO:μ=24.5). The fact that allocation method has a significant effect ($p < 0.01$, $F(1, 19) = 75.3$) on manual allocation validates that participants adopted (i.e., accepted and used) MI to control the UAVs. Drop-outs did not have a significant effect on manual allocation.

Order effects. Finally, we tested for order effects between groups A and B using *order* as a between-subjects factor. We found significant interaction effects for *order* and *allocation method* on *number of targets found* ($p < 0.001$, $F(1, 18) = 18.8$), as well as on *effort* ($p = 0.16$, $F(1, 18) = 7.01$). There was a significant three-way interaction of *order*, *allocation method*, and *drop-outs*, again on both *number of targets found* ($p = 0.041$, $F(1, 18) = 4.6$), and on *effort* ($p = 0.011$, $F(1, 18) = 8.1$). The presence of order effects thus established, an inspection of descriptive statistics shows decreasing performance over time (hence the order effect). This suggests a fatigue effect (in line with [Mekdeci and Cummings, 2009]), and corroborates our counterbalancing approach.

6 Discussion

Here we further discuss the results of the user study, contextualising these with findings from the interview in order to draw out the key lessons learnt.

6.1 Responding to Autonomous Planning

The results (in terms of performance, workload, and reliance on max-sum plans) suggest that participants understood

[11] One data point was lost due to video recording issues.

and used the interaction mechanisms we provided them to interact with task allocations computed by max-sum.

Managing drop-outs. A key objective of our design was to study how operators manage drop outs. In particular, we hypothesised that the mixed-initiative task allocation would outperform manual allocation when drop outs occur (H1). Although there was no effect on overall number of targets found and efficiency, there were *significant effects on effort and tasks* completed; thus the results (partially) support hypothesis H1. Moreover, effort was significantly higher even when there were no drop outs, thus supporting H2.

This suggests that participants experienced less effort and completed more tasks when supported by mixed-initiative planning (max-sum), which supports our overall design rationale. However, the interviews showed that some participants questioned the usefulness of max-sum in drop-out settings. In particular, as max-sum re-allocates UAVs automatically following a task completion or a drop-out, some participants mentioned they did not always agree with the new allocations. One participant mentioned *"I would watch the next allocation from the corner of my eye, just to make sure..."*. Related to this, the findings on self-reported workload (partially supporting H4) suggest a trade-off: while participants experienced higher workload for MI when no drop-outs occurred, when drop-outs occurred, the reported workload for MI was lower than for MA.

In the interviews, participants also requested features to communicate preferences and priorities to MI at a more fine-grained level, one said *"you should be able to tell it, if this UAV drops out then do this.."*. This suggests a desire for more direct control at times, beyond monitoring UAVs, including preference specification and modification of UAV plans, as elaborated in the following.

Modifying plans. Aside from 'keeping an eye on it', participants also actively intervened in the max-sum plans, as was reflected in the (modest) number of changes to max-sum plans (see section 5.1). Another participant stated that *"it's right most of the time but it's not perfect, you still have to go and change it sometimes"*.

Modifications of plans 'at run-time' were typically occasioned by UAV drop-outs. We hypothesised that this would affect operator workload and increase collaboration particularly in the manual task allocation setting. In addition to the aforementioned findings on workload; interestingly, increased collaboration in the manual condition was manifested in *significant effects on collaborative annotations and pre-launch planning* (supporting H3). While the manipulation check also showed that operators modified plans significantly less often in the MI condition, the modifications in the MI condition were not negligible. Users employed the whole range of modifications provided to them, validating suggestions in the literature they should be given a range of options when advised by an agent [Smith *et al.*, 1997].

Intelligibility of max-sum. Interviews also revealed the participants' mental model of max-sum: *"It chooses the shortest path for each UAV to each task and it plans the next step"*. Also, they realised that *"you can let it do the easy tasks and focus on the hard ones"*, suggesting delegation of control to an autonomous system is effective for routine tasks in that it frees up operator cycles to manage more difficult tasks. Also, *"it does not know what tasks we want to prioritise"* and that *"It does not avoid areas we have already explored"*, thus identifying the shortcomings of autonomous (re)planning. In particular, when users modified plans (e.g., manually replace a UAV in an allocation with another UAV), they would sometimes disagree with the consequent re-allocation computed by max-sum for the remaining UAVs and again reconfigure the allocations manually.

These observations and the associated empirical results suggest that, while users tend to rely on max-sum for routine task allocations, there is a need for them to express tactical priorities (in terms of areas to be searched, what should be done after a drop-out) to max-sum. In fact, such needs may not require new interface elements. For example, participants worked out higher-order tactics to react to drop-outs. Thus, they would sometimes send two UAVs to search an important area, *"just in case the other one dropped out"*, as one participant put it, such that max-sum would automatically allocate the dropped UAV's task to the other UAV. This, in addition to the fact that participants used tasks as waypoints, indicates that rather than trying to encode all the mission priorities and conditional behaviours within max-sum, it may be better left to the users to develop truly elaborate mixed-initiative tactics that combine, potentially simple, capabilities of the agents with an understanding of the most likely contingencies at run-time (rather than design-time).

Impact of max-sum on human collaboration. To reiterate, our analysis of human collaboration showed that the allocation method (i.e., MI v.s. MA) had no impact on collaborative planning but that increased collaborative annotation was observed in the MI conditions. Given that the camera and planner views are on different screens, the planner view operator necessarily loses sight of the planner view while annotating collaboratively. This suggests autonomous agents may be difficult to monitor effectively. This shortcoming, however, was counterbalanced by a higher number of tasks completed in MI conditions which suggests that operators exploited the autonomous planning ability of the UAVs in order to spend more time collaborating on finding targets in the maps. Particularly, this occurred when these targets were not easy to spot within aerial imagery of buildings of different colours and sizes. In fact, some participants explicitly decided on sharing the camera view (given the multiple video feeds) during the pre-launch mission planning session.

In contrast, the camera view operators often commented that their task was boring. As one participant put it, *"it would be good if the interface flashed when a target was identified by the UAV or if it completed its task"*, showing an expectation that the UAVs had advanced vision capabilities (which supports our earlier comments on expectations of autonomy). Hence, we noted they would turn to helping the planner view operator at the expense of finding targets in the video feeds.

6.2 Implications for Design

Here we summarise the implications for the design of mixed-initiative supervisory control systems.

Interacting with Algorithms. Users found autonomous planning useful for the mundane parts of task allocation. They made use of the different types of tasks to exploit the UAVs' autonomy (in terms of waypoint following, region scanning, and task allocation). Moreover, users found it useful to modify plans computed by max-sum for the 'hard' parts of task allocation. This suggests the interaction mechanisms with max-sum were effective. This was achieved with the intuitive specification of constraints (see Section 4.1) as click actions with immediate on-screen feedback of re-allocations computed by max-sum. Thus, the users' need to control the inner workings of the algorithm, and their understanding how the algorithm reacts to user-driven constraints, suggest that coordination algorithms need to be designed with the user in mind. By this, we mean that controls on the algorithm's objective function, and feedback about the effect of these controls should be carefully developed to ensure users understand and correctly anticipate the impact of their inputs on multiple agents.

Mixed-Initiative Re-planning. It was clear that participants wanted greater control of re-planning events. Max-sum plans seemed to add to their workload and they did not always trust it. Also, users came up with their own workarounds for the way that max-sum was re-planning. However, they did appreciate the fact that some parts of the plan were good. Hence, instead of computing a plan and inviting the operator to modify it (through manual allocations and waypoints – adding more effort), we suggest that autonomous planning tools should provide more control over the types of plans suggested, without necessarily increasing effort. One way of doing this would be to offer dynamic plan libraries that are adapted to the context, in a similar vein to the 'playbook' approach [Miller and Parasuraman, 2007].

7 Conclusions

Our study of multi-UAV task allocation interfaces in lab studies with 40 participants reveals that they found max-sum useful, requiring less effort and resulting in more task completed and lower workload. However, max-sum plans are not always trusted and this may result in higher workload for the operators than purely manual conditions. Our results point to the need to design algorithms with the users' needs for control in mind. Future work will aim to improve the way max-sum plans can be modified and evaluate the impact of inefficient plans on operators' trust in automation.

References

[Amador *et al.*, 2014] S. Amador, S. Okamoto, and R. Zivan. Dynamic multi-agent task allocation with spatial and temporal constraints. In *AAAI*, pages 1384–1390, 2014.

[Bertuccelli *et al.*, 2010] L F Bertuccelli, N W M Beckers, and M L Cummings. Developing operator models for uav search scheduling. *Proc. Conf. on Guidance, Navigation and Control*, 2010.

[Cummings *et al.*, 2007] M. L. Cummings, A. S. Brzezinski, and J. D. Lee. The impact of intelligent aiding for multiple unmanned aerial vehicle schedule management. *IEEE Intelligent Systems*, 22(2):52–59, 2007.

[Cummings *et al.*, 2010a] M. L. Cummings, S. Bruni, and P. J. Mitchell. Human supervisory control challenges in network-centric operations. *Reviews of Human Factors and Ergonomics*, 6(1):34–78, 2010.

[Cummings *et al.*, 2010b] M. L. Cummings, A. Clare, and C. Hart. The role of human-automation consensus in multiple unmanned vehicle scheduling. *Human Factors*, 2010.

[de Greef *et al.*, 2010] T. de Greef, H. F. R. Arciszewski, and M. A Neerincx. Adaptive automation based on an object-oriented task model: Implementation and evaluation in a realistic c2 environment. *Journal of Cognitive Engineering and Decision Making*, 4(2):152–182, 2010.

[Delle Fave *et al.*, 2012] F. M. Delle Fave, A. Rogers, Z. Xu, S. Sukkarieh, and N. R. Jennings. Deploying the max-sum algorithm for decentralised coordination and task allocation of unmanned aerial vehicles for live aerial imagery collection. In *Proc. of ICRA*, pages 469–476, 2012.

[Goodrich and Schultz, 2007] M. A Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[Goodrich *et al.*, 2007] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall. Managing autonomy in robot teams: observations from four experiments. In *Proc. of HRI*, pages 25–32. ACM, 2007.

[Gutierrez *et al.*, 2011] P. Gutierrez, P. Meseguer, and W. Yeoh. Generalizing adopt and bnb-adopt. In *Proc. of IJCAI*, pages 554–559, 2011.

[Jennings *et al.*, 2014] N. R. Jennings, L. Moreau, D Nicholson, S. D. Ramchurn, S Roberts, T Rodden, and A. Rogers. On human-agent collectives. *Communications of the ACM*, 57(12):33–42, 2014.

[Lewis *et al.*, 2009] M. Lewis, K. Sycara, and P. Scerri. Scaling up wide-area-search-munition teams. *IEEE Intelligent Systems*, 24(3):10–13, 2009.

[Macarthur *et al.*, 2011] K. S. Macarthur, R. Stranders, S. D. Ramchurn, and N. R. Jennings. A distributed anytime algorithm for dynamic task allocation in multi-agent systems. In *AAAI*, 2011.

[Mekdeci and Cummings, 2009] B. Mekdeci and M. L. Cummings. Modeling multiple human operators in the supervisory control of heterogeneous unmanned vehicles. In *Proc. Workshop on Performance Metrics for Intelligent Systems*, pages 1–8, 2009.

[Miller and Parasuraman, 2007] C. A. Miller and R. Parasuraman. Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors*, 49(1):57–75, 2007.

[Ramchurn *et al.*, 2010] S. D. Ramchurn, A. Farinelli, K. S. Macarthur, and N. R. Jennings. Decentralized coordination in robocup rescue. *The Computer Journal*, 53(9):1447–1461, 2010.

[Rogers *et al.*, 2011] A. Rogers, A. Farinelli, R. Stranders, and N. R. Jennings. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence*, 175(2):730–759, 2011.

[Sheridan and Verplank, 1978] Thomas B Sheridan and William L Verplank. Human and computer control of undersea teleoperators. Technical report, DTIC, 1978.

[Silverman, 1992] B. G. Silverman. Human-computer collaboration. *Hum.-Comput. Interact.*, 7(2):165–196, June 1992.

[Smith *et al.*, 1997] P.J. Smith, C.E. McCoy, and C. Layton. Brittleness in the design of cooperative problem-solving systems: the effects on user performance. *IEEE Trans. on Systems, Man and Cybernetics: Part A*, 27(3):360–371, May 1997.