Online Planning for Collaborative Search and Rescue by Heterogeneous Robot Teams

Zoltán Beck
Electronics and Computer Science
University of Southampton
Southampton, UK
zb1f12@ecs.soton.ac.uk

Luke Teacy
Electronics and Computer Science
University of Southampton
Southampton, UK
wtlt@ecs.soton.ac.uk

Alex Rogers
Department of Computer Science
University of Oxford
Oxford, UK
alex.rogers@cs.ox.ac.uk

Nicholas R. Jennings
Electronics and Computer Science
University of Southampton
Southampton, UK
nrj@ecs.soton.ac.uk

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 a search and rescue setting, where robots with different capabilities work together to accomplish tasks (rescue) and find information about further tasks (search) at the same time. The state of the art for such collaboration is robot control based on independent planning for robots with different capabilities and typically incorporates uncertainty with only a limited scope. In contrast, in this paper, we create a *joint plan* to optimise all robots' actions incorporating uncertainty about the future information gain of the robots. We evaluate our planner's performance in settings based on real disasters and find that our approach decreases the response time by 20-25% compared to state-of-the-art approaches. In addition, practical constraints are met in terms of time and resource utilisation.

Keywords

multi-robot systems, robot coordination, online planning, disaster response, search and rescue, hindsight optimisation

1. INTRODUCTION

In recent years, there has been an ongoing focus within the rescue robotics research domain to use robots in professional disaster response [20, 21]. These initial deployments use individual robots that are labour intensive to control and are operated independently. This places a high demand on human expertise, which is a limited resource in disaster response. Thus, to decrease this workload, there is a need to develop semi-autonomous teams of robots, that can make their own decisions and coordinate to make best use of their combined resources [27]. To tackle this challenge, researchers typically focus on multi-robot collaboration within known settings, where the possible robot actions are defined as a set of tasks [11, 19, 22]. 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). In addition, as new technologies allow a diverse group of robots with differing capabilities to be deployed, they are required to work together, combining their individual capabilities to be more efficient [6]. A good example for such complex settings is *Search And Rescue* (SAR), where robots cooperate to find and aid victims [18]. In this setting, some robots may be specialised for fast exploration (e.g. by having advanced mobility or high resolution sensors), while others are capable of aiding victims (e.g. by having good manoeuvrability or specific actuators).

Existing solutions that tackle collaboration in the face of uncertain information are typically limited to simple exploration or target search problems [3, 7, 29, 31]. Moreover, the use of generic temporal planners rapidly becomes intractable for such problems [9] unless applied in a domain-specific manner [3]. Finally, domain specific approaches rarely involve complex action relations, such as task workflows 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 [15, 33], or simple heuristics are applied to enhance similar collaboration [5]. 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 SAR, collaboration between different robots is vital for good performance when using a diverse group of robots to perform search operations and rescue actions at the same time. When using decomposition planning, rescue robots cannot build on the plans of search robots to identify valuable actions, and similarly, search robots cannot priori-

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, S. Marsella, C. Jonker (eds.), May 9–13, 2016, Singapore.*

tise their search according to the actions of the rescue robots. The main challenge is that the problem of multi-robot exploration (search) and multi-robot task allocation (rescue) are both complex optimisation problems [14]; especially when combined within an uncertain domain. For this reason joint planning for multiple robots involving action dependencies in an uncertain setting has not been achieved. In addition, in line with the dynamic nature of disaster response, incoming information from first responders can significantly change the situation from one moment to another. For this reason, a computationally efficient online planning approach is necessary to ensure rapid response from the system.

To address these shortcomings, we offer a novel planning approach for multi-role robot collaboration under uncertainty that provides the benefits of collaborative planning while keeping the computation time at an acceptable level. Uncertain information is incorporated in our planning process using *hindsight optimisation* (HOP) [32] that allows us to apply computationally efficient deterministic planning techniques. 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 introduce temporal constraints (search precedes rescue) over the stochastic action domain. As a result, we can generate plans for robots assigned to rescue operations even before the final locations of those rescues are confirmed. Similarly, we can generate plans for robots assigned to searching for casualties, taking account of the current plans of rescue robots that will be tasked with rescuing them. We show that joint planning improves the overall SAR performance against the state-of-the-art decomposition approach in simulations based on two real-world disaster scenarios (on an earthquake in Haiti and an industrial spill in Hungary). In so doing, we make the following contributions to the state of the art:

- We are the first to propose a method for homogeneous Multi-Robot Task Allocation (MRTA) given an arbitrary task distribution and continuous action space (UMRTA problem). In detail, we use HOP to assign a task or a motion direction for each robot using an MRTA scheduler.

- We offer the first online planning approach to create a joint plan under uncertainty for distinct groups of search robots and rescue robots in a SAR scenario. As a result, rescue plans are made over an uncertain set of rescue locations and iteratively optimised along with search plans.

- We evaluate the performance of the planning approaches against the state-of-the-art decomposition planning approach in realistic SAR settings [2, 24, 30]. In particular, we find that the time of the rescues are reduced by 20-25% with our joint planning compared to decomposition planning.

In the following, we introduce the background of this work (Section 2). Next, the underlying basic task allocation problems are explained (Section 3), then the SAR collaboration problem itself is defined (Section 4). After that, we present the state-of-the-art approach and our joint planning approach to solve this problem (Section 5). This is followed by the experimental evaluation of these approaches (Section 6). Finally, we provide concluding remarks and outline directions for future work (Section 7).

2. BACKGROUND

There are two main methodologies for mobile robot (or autonomous vehicle) collaboration. 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. In particular, [14] introduces the family of *Multi-Robot Task Allocation* (MRTA) problems. Specifically, when coordinating mobile robots, tasks typically consist of going to a specific location and perform an operation in its proximity. The second formulation of the spatial collaboration of robots is during searching an area. In these settings, an importance or intensity map contains the current state of the search and the knowledge about the area. The robots travel on this map and update the intensity values according to the sensed values. A simple and robust way to guide the robots is to apply gradient descent on this map [12, 31]. When the motion capabilities are limited as for fixed-wing *Unmanned Aerial Vehicles* (UAVs), performing a tree search within the feasible paths can be performed to control the vehicles [29]. To better capture the random outcomes of actions probabilistic planning can be applied. Unfortunately, to use such techniques in an online fashion, the action space is significantly reduced by either discretising the problem [7], or limiting the action space to a set of fixed search patterns [3].

When multiple groups of robots collaborate at the disaster site, the collaboration of groups becomes important besides the task allocation for the individual groups. Some approaches investigate multiple task types and their relations and apply the task allocation for the whole group. For example, [19] represents the complex relations between tasks as a tree, and subtrees are allocated to individual robots. However, this assumes that the robots are capable of performing all tasks within a subtree, that 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 a different level is more likely, as exemplified by [8].

In such settings, the most common approach of collaboration is independent action planning. In this vein, [10] performs a search and rescue like 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 [33, 15]. 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, [5] uses the probabilistic plans of a ground robot to find areas to scout with a UAV, and in [16] the allocations of different tasks are planned sequentially to allocate robots for a complex mission. However, these approaches are limited in terms of taking vital action dependencies into account by all robot groups, as they do not create a *joint plan* for all kinds of robots in a way that all plans are optimised regarding the plans of all other robots. In the rest of the paper, we introduce an online planner that creates a joint plan for search and rescue robots, and meets practical constraints in terms of computation time and resource utilisation.

3. TASK ALLOCATION

As introduced below, a fundamental problem underlying multi-robot collaboration is task allocation. In this section, we first define the *Multi-Robot Task Allocation* (MRTA) problem in a SAR setting, then we propose an extension to it, the *Uncertain Multi-Robot Task Allocation* (UMRTA) problem introducing uncertainty in the tasks.

3.1 MRTA

In this paper, we investigate MRTA with Single-Task robots, Single-Robot tasks, and Time-extended Assignment (ST-SR-TA) [14]. 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: *search tasks* (e.g. searching a specific area for victims, or assessing collapsed buildings) or *rescue tasks* (e.g. rescue a victim, or secure an unsafe structure). By performing these tasks, the robots aim to maximise the number of successful rescues, which must first be discovered by undertaking search tasks. We model this as a linearly decreasing utility function[1]:

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

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

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. 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: $P/\mathrm{ST_{sd}}/\sum T_j$ [1]. 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 ($1/\mathrm{ST_{si}}/\sum T_j$) is polynomial when the due-date is common, however given sequence dependent setup times ($1/\mathrm{ST_{sd}}/\sum T_j$) it becomes NP-hard [25]. 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 UMRTA

In realistic settings, however, there might not be complete information about the tasks. In this case, the uncertainty about the tasks can be represented as a probability distribution. We define the UMRTA problem as an MRTA problem ($\langle \mathbf{R}, \mathbf{T} \rangle$) 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 ($\mathcal{T}^+$): $\mathcal{T} = \mathbf{T} \cup \mathcal{T}^+$.

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, the solution may 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 optimise the robot's position given the distribution of uncertain tasks to be closer to these tasks when 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:

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

4. SAR COLLABORATION PROBLEM

Having described the components of the collaboration problem, we now present the complete heterogeneous robot collaboration problem in a SAR setting. Because search and rescue require a different set of skills from the robots (high speed and advanced sensors for search, good manoeuvrability and actuators for rescue), they split into a group of search robots and a group of rescue robots. Search robots find rescue locations during their search process, while rescue robots move around the disaster area and perform rescues at these locations.

In this setting, the collaboration breaks into a task allocation problem for the search robots and another one for the rescue robots. The search tasks are known beforehand, while rescue tasks are initially unknown, and will be gradually discovered by search tasks. This means that search can be formulated as a MRTA problem, while rescue as an UMRTA problem. For simplicity, we assume uniform utility and process time for rescue tasks with random 2D locations. Assuming complete spatial randomness, the task locations are the outcome of a non-homogeneous spatial Poisson process [17] with intensity function λ. During SAR, this distribution frequently changes as new information is introduced about the disaster site, or as a region is searched. Within our evaluation we chose a simplistic observation model, meaning that when a region is searched, all tasks within that region are discovered, and the intensity for undiscovered tasks within that region thus drops to zero.

Formally, the search MRTA problem is determined by the set of search robots ($\mathbf{R}_s$) and search tasks ($\mathbf{T}_s$), while the rescue UMRTA problem is determined by the set of rescue robots ($\mathbf{R}_r$) and the random set of rescue tasks ($\mathcal{T}_r$). The rescue tasks can be divided into a set of known ($\mathbf{T}_r$) and a random set of unknown tasks, that is a Poisson point process ($\mathcal{T}_r^+ \sim \mathit{Poisson}(\lambda)$). Additionally, the connection between the search and the rescue problem can be given by the $\mathit{exploredby} : \mathbf{T}_s \times \mathcal{T}_r \to \{0, 1\}$ function that indicates if a rescue task is found when a search task is performed (if it falls within the associated area).

Accordingly, the SAR collaboration problem is defined by $\langle \mathbf{R}_s, \mathbf{T}_s, \mathbf{R}_r, \mathcal{T}_r, \mathit{exploredby} \rangle$ and the solutions will have the following form:

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

[1] Search tasks are not directly associated with a utility, but instead are responsible for discovering rescue tasks (details below).

5. ONLINE PLANNING APPROACHES

We now introduce three planning methods for the above problem: the first approach represents the state-of-the-art in collaborative online planning, the second uses HOP to solve the UMRTA problem, and the third approach creates a joint plan for the linked MRTA and UMRTA problem.

5.1 Decomposition Planning with Gradient Descent

This approach represents our benchmark, where the MRTA problem of search and the UMRTA problem of rescue are solved independently using existing approaches. We use Shortest Adjusted Processing Time first (SAPT) algorithm to solve the task allocation as tardiness scheduling problem ($P/\mathrm{ST}_{\mathrm{sd}}/\sum T_j$). The SAPT scheduling is optimal for the sequence-independent single machine case ($1/\mathrm{ST}_{\mathrm{si}}/\sum T_j$), but nevertheless provides a good heuristic in the sequence-dependent settings [25, 26]. In detail, we define the adjusted processing time as follows:

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

$AP\left(\tau, s\right)$ represents the adjusted processing time of task τ after schedule s ($t(\tau)$ in Equation 1, if 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, and $P(\tau)$ is the process time of task τ.

As a standard online planner, the SAPT scheduler is applied to provide a solution both for the search MRTA problem $\langle \mathbf{R}_s, \mathbf{T}_s \rangle$ and the known part of the rescue problem $\langle \mathbf{R}_r, \mathbf{T}_r \rangle$. This is to prioritise the known tasks in the optimisation, as can be found in the literature for similar problem settings [28, 33]. In case of idle rescue robots (no known tasks are assigned by the SAPT scheduler), gradient descent is applied to maximise the intensity at the robot's location, and to orientate the robots towards central areas. In detail, rescue robots with no assigned tasks will travel:

- towards the nearest point of the disaster area if outside,
- along the gradient of the task distribution intensity,
- towards the mean of the task distribution at zero gradient.

5.2 Decomposition Planning with HOP

In this approach, instead of ignoring the task distribution for the planning of the rescue, HOP 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 [7]. Unfortunately the introduced temporal planner does not scale well with the size of the problem due to the iteration through the possible resulting states. By contrast, we present a HOP planner that can solve the above presented UMRTA problem with continuous state space. This planner, instead of replanning for all the possible future states, determines the best action using only a single solution for the current state.

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:

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

In detail, the expected utility is estimated by the average of the utilities ($U_{sch}(\tau)$) determined by the SAPT MRTA scheduler output schedules ($s_{R,i}$) for N samples ($\mathbf{t}_r^i$) of the Poisson process ($\mathcal{T}_r^+$). 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 [22]).

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

Algorithm 1 HOP UMRTA Solver

```
Require: $\mathbf{T}_r$: set of known rescue tasks
Require: $\mathbf{R}_r$: set of rescue robots
Require: $N$: size of the Monte Carlo simulation
Require: $s_{R,i}, \forall R \in \mathbf{R}_r, i \in \{1..N\}$: hindsight schedules
 1: for all $R \in \mathbf{R_r}$ do
 2:     $\tau^* = \arg\max_{\tau\in\mathbf{T}_r} \mathrm{count}_{i=1}^{N}\left(\tau = first(s_{R,i})\right)$
 3:     if $\mathrm{count}_{i=1}^{N}\left(\tau^* = first(s_{R,i})\right) > \frac{N}{2}$ then
 4:         $x_R \leftarrow \tau^*$                ▷ assign task $\tau^*$ to robot $R$
 5:     else
 6:         $d = \mathrm{mean}_{i=1}^{N}\left[w(s_{R,i}) * dir(R, first(s_{R,i}))\right]$
 7:         $x_R \leftarrow d$               ▷ assign direction $d$ to robot $R$
 8:     end if
 9: end for
Ensure: $\mathbf{X} = \{x_R : \forall R \in \mathbf{R}_r\}$: 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. In detail, the algorithm iterates over all robots, and finds the most commonly assigned first task ($\tau^* \in \mathbf{T}_r$) in its schedules (Line 2). If it is assigned as first in the majority of the schedules, 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 tells 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, assumes hindsight knowledge of the random task set for each sample ($\mathbf{t}_r^i$), as it is a HOP technique. The proof of the optimality is omitted due to space limitations[2].

5.3 Joint Planning with HOP

This approach creates a joint plan for the search and rescue robots via a negotiation process[3]. At each step of the nego-

[2] It can be found in http://bit.ly/20ZD3wn.

[3] The iteration length within 5 and 9 does not change the overall result significantly, so we chose 5 iterations in the evaluation.

tiation, one of the robot groups (search or rescue) 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 search and rescue activities into account (search tasks find rescue tasks within an area).

Rescue Planning Optimisation uses the search plan to introduce a constraint to the execution time of sampled rescue 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 $cstr(\tau)$ is the time when task τ is discovered according to the current search plan (temporal constraint). In brief, the execution time of each "hindsight task" is delayed until discovered according to the current search plan. The MRTA scheduling problem is still solved using SAPT with the adjusted processing time in Equation 6.

Also, the weights (w) in Line 6 of Algorithm 1 need adjustment to ensure optimality. Because the tasks that are delayed due to the temporal constraints do not increase the urgency of a schedule, they should not be counted in the weight. Accordingly, $w(s)$ equals the number of consecutive non-delayed tasks at the beginning of schedule s.

Search Planning Optimisation does not simply minimise the execution time of the search tasks using the plan of the other group. The subject of optimisation is to improve rescue, as the utility comes from the rescue tasks. Therefore, the optimisation is applied to minimise the delay introduced on the rescue 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 each schedule (as in SAPT scheduling), would not result in a good quality solution. For this reason, a greedy method sequentially introduces tasks and inserts them to a position of a schedule with minimal cost, as in the MURDOCH negotiation [13], detailed in Algorithm 2.

```
Algorithm 2 Search Plan Optimisation
Require: $\mathbf{T}_s$: set of search tasks
Require: $\mathbf{I} = \{\mathbf{t}_\tau : \forall \tau \in \mathbf{T}_s\}$: timings from rescue
Require: $\mathbf{R}_s$: set of search robots
1: $\mathbf{T}' \leftarrow \mathbf{T}_s$: unassigned search tasks
2: $s_r \leftarrow \emptyset, \forall r \in \mathbf{R}_s$: robot schedules
3: while $\mathbf{T}' \neq \emptyset$ do
4:     $\tau^* = \arg\min_{\forall \tau \in \mathbf{T}'} \min \mathbf{t}_\tau$
5:     $\langle r^*, i^* \rangle = \arg\min_{\langle r,i \rangle ; \forall r \in \mathbf{R}_s, \forall i \leq |s_r| \in \mathbb{N}} \Delta(s_r, \tau^*, i)$
6:     $s_{r^*} \leftarrow insert(s_{r^*}, \tau^*, i^*)$    ▷ Insert $\tau^*$ to schedule
7:     $\mathbf{T}' \leftarrow \mathbf{T}' \setminus \{\tau^*\}$    ▷ Remove from unassigned tasks
8: end while
Ensure: $\mathbf{S} = \{s_r : \forall r \in \mathbf{R}_s\}$
```

Here, $\mathbf{t}_\tau$ is a set of planned rescue times for sampled rescue tasks discovered by task τ, and the $insert(s, \tau, i)$ function inserts task τ into schedule s to the i^{th} location. Besides that, $D(s)$ estimates the delay caused by schedule s:

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

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

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 search robots' schedule using minimal insertion. Specifically, the most urgent task is selected as the one with the soonest rescue 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 rescue robots' plan ($\mathbf{I} = \{\mathbf{t}_\tau : \forall \tau \in \mathbf{T}\}$).

6. EXPERIMENTAL EVALUATION

The evaluation resembles a currently feasible robotic search and rescue setting based on feedback from Rescue Global[4]. In this setting we rely on aerial robots (UAVs) only, due to their successful disaster response deployments as opposed to ground robots. Specifically, fixed-wing UAVs are able to rapidly fly over the disaster site, so they take the role of search robots; small rotary-wing UAVs have high manoeuvrability and are able to approach objects on the ground, therefore they are responsible for rescue tasks. In detail, fixed-wings search the area by taking aerial images. These images are analysed (by either a professional or computer vision) to point out sights of interest (e.g. possible victim locations, dangerous buildings, or critical supplies). The sights of interest are visited by rotary-wing UAVs to provide detailed, close-up video feeds to first responders. This procedure allows first responders to assess an area and be able to locate some high priority tasks before accessing the disaster site.

In disaster response, SAR can be very different depending on the cause of the disaster (e.g. earthquake, extreme weather, or industrial disaster), the impact of the disaster or the population density of the area. For this reason, our experiments include two very different settings taken from the Ajka Alumina Plant industrial spill (2010) (Figure 3) and the Haiti earthquake (2010) (Figure 2). In detail, the former has a sparse effect spreading along a large region, while the latter represents a larger impact disaster in a smaller area. The area size and the density of tasks will determine if rescue agents spend more time moving between rescue locations or performing rescues. In the following we detail the settings and the numerical results of the conducted experiment.

6.1 Experimental Setup

For both disasters, there are three different settings with different levels of available information. In each setting, there is a rescue task intensity map (λ) that represents the prior information about the disaster impact:

- *Ground Truth*: The intensity of the rescue locations at the disaster site is based on detailed assessment.
  - *Ajka* scenario: A combination of the built-up regions and the flooded area (with decreasing intensity with the distance from the source) covering 5.1 km^2 taken from [2]. (Figure 1d)
  - *Haiti* scenario: Victim intensity in a search sector of Port au Prince based on a building damage assessment with intensities according to Table 1 covering 0.106 km^2 taken from [30]. (Figure 1a)
- *Preliminary*: There is some preliminary information about the possible outcome of the disaster containing

[4] http://www.rescueglobal.org/

Tag	Description	**E**[#rescues]
Grade 1	Negligible to slight damage	0.002
Grade 2	Moderate damage	0.004
Grade 3	Substantial to heavy damage	0.01
Grade 4	Very heavy damage	0.04
Grade 5	Destruction	0.125

Table 1: Haiti assessment [30] damage levels and expected number of rescues[5] at each tag location

Parameter	Search Robots	Rescue Robots
Robot speed	25 m/s	10 m/s
Task length	0 s	30 s

(a) Robot parameters

Parameter	Ajka Scenario	Haiti Scenario
Search grid spacing	440 m	105 m
Search imagery area	530×530 m	125×125 m
Simulation time step	3 s	1 s
#samples (N)	100	100
#search robots	2	2
#rescue robots	4	8
#search tasks	400, 120, 79	95, 95, 87
#rescue tasks	20 (avg.)	61 (avg.)

(b) Scenario parameters

Table 2: Simulation parameters

less information than the ground truth intensity. This is the realistic case when disaster impact estimations are available due to disaster resilience [4].

  - *Ajka* scenario: A combination of the built-up regions and the results of a spill simulation for a similar spill incident covering a larger possible area (12.7 km^2) taken from [24]. (Figure 1e)
  - *Haiti* scenario: Uniform inside the search sector (0.875 km^2) with higher intensities around 20 simulated report locations (drawn from the *Ground Truth* intensity). (Figure 1b)

- *Blank*: There is no information about the task density other than the search area, the intensity is uniform.
  - *Ajka* scenario: A uniform intensity over a large rectangular area (78.5 km^2) above the possible area of the industrial spill. (Figure 1c)
  - *Haiti* scenario: A uniform intensity inside a search sector (0.875 km^2) in Port au Prince. (Figure 1f)

The performance of the different planning approaches are compared in simulation for 128 different disaster outcomes for both scenarios. Each outcome is an independent sample from a Poisson process with the *Ground Truth* intensity introduced above. Each approach is tested with the three different levels of available information independently.

During the experiments, we simulate robots performing collaborative SAR in two groups: search robots and rescue robots. Table 2 details the chosen parameters for these simulations. The parameters are chosen to match realistic settings as much as possible. The parameters for the two scenarios differ due to the differences in the two disaster environments. Specifically, the Haiti sector is a densely populated urban area where the large number of features require sub-cm resolution, while large fields covered with red mud near Ajka require lower resolution imagery for victim search. The search imagery area size was chosen according to these measures to include a similar number of features. Consequently, the simulation time step is chosen in a way that the search robots need 4-6 time steps to travel between two search locations. This allows a human supervisor to intervene between actions, and also to interact with a reasonably responsive decision making system [23]. However, varying the simulation time step did not change the result significantly. Search tasks are located on grid points where the intensity is greater than zero ($\lambda > 0$) for the corresponding area.

The start locations of the UAVs can be seen in Figure 2 and in Figure 3 over the satellite imagery of the area. In detail, the area marked with blue edge is the edge of the Blank search area, yellow markers show rotary-wing start locations, and green rectangles mark the start location of fixed wing UAVs. The fixed-wing start locations are placed on possible take-off locations such as nearby airports and intact straight roads, while rotary-wing start locations are chosen at rubble-free locations where the infrastructure can be set up for controlling them.

Each simulation instance used a single core of an Intel Xeon E5-2670 processor. The computation time is measured for each approach to evaluate computational feasibility.

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 rescue locations by a one-sample t-test. Specifically, we use the average time of the rescue operations 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.

6.2 Experimental Results

In this section the performance and the computation time is evaluated for the previously detailed approaches in the above settings. Within the comparison (Figure 4 and 5), there is a two character code identifying each test case, assembled by the first letter of the information representation and the number of the planning approach. The notation of the case depending on available information is the following:

Bx: *Blank* intensity map,

Px: *Preliminary* intensity map,

Gx: *Ground Truth* intensity map.

Likewise, the three collaboration approaches are:

x1: *Decomposition Planning with Gradient Descent*,

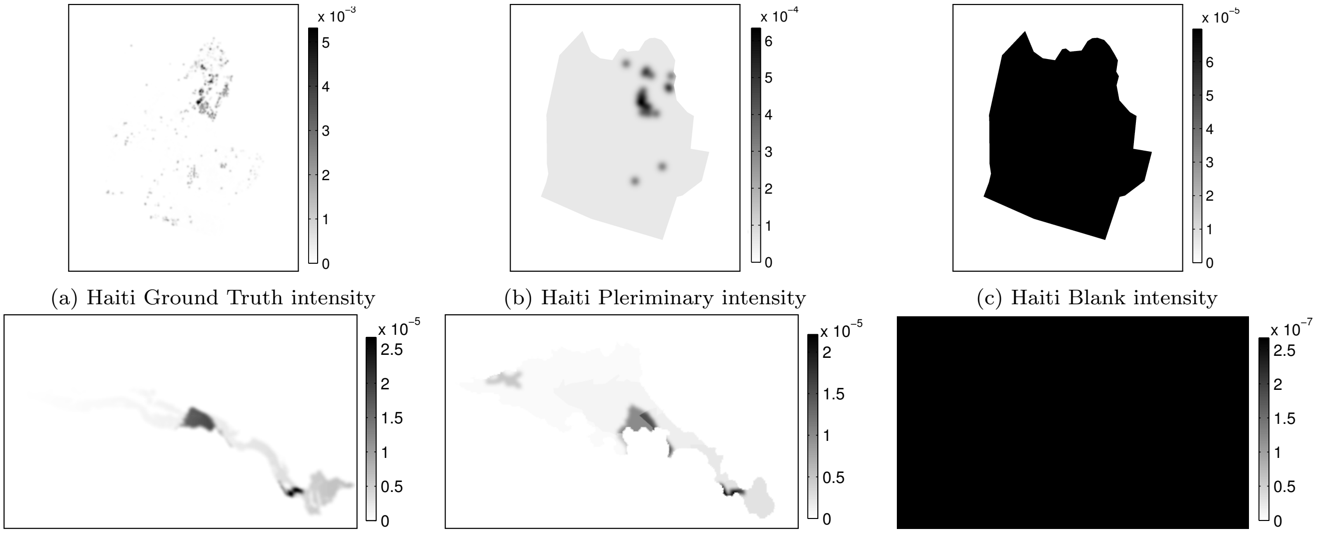
x2: *Decomposition Planning with HOP*,

x3: *Joint Planning with HOP*.

For instance, “G3” stands for the *Ground Truth* information representation and *Joint Planning with HOP* approach.

The **Overall Performance Comparison** can be seen in Figure 4, while the relative performance to our approach

[5] Expected number of points in a Poisson point process equals the integral of the intensity function within an area.



(a) Haiti Ground Truth intensity

(b) Haiti Pleriminary intensity

(c) Haiti Blank intensity

(d) Ajka Ground Truth intensity

(e) Ajka Pleriminary intensity

(f) Ajka Blank intensity

Figure 1: Intensity maps (units are task/m^2)

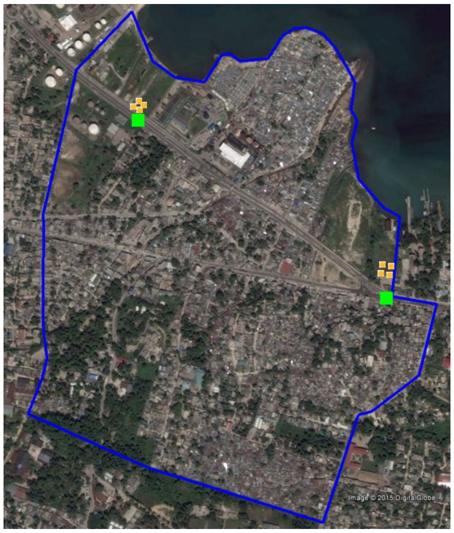


Figure 2: UAV start locations for the Haiti scenario

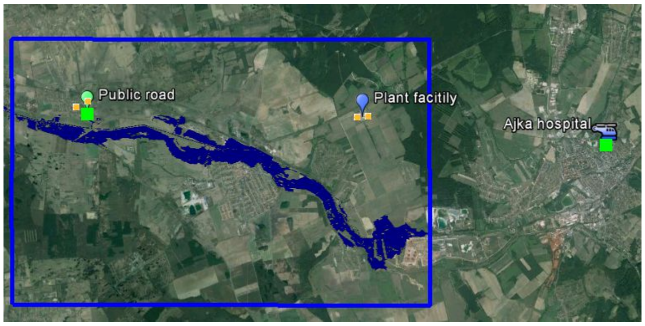


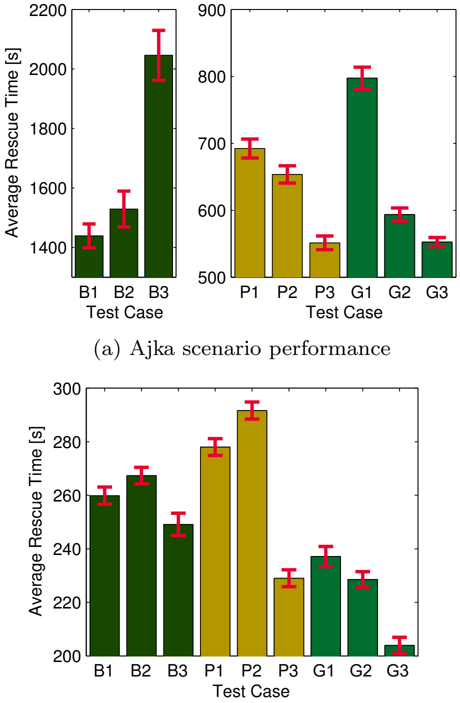
Figure 3: UAV start locations for the Ajka scenario

	Blank		Prelim.		G. Truth	
Approach	Ajka	Haiti	Ajka	Haiti	Ajka	Haiti
Dec. GD	2	<1	<1	<1	<1	<1
Dec. HOP	18	40	8	43	10	39
Joint HOP	**4541**	210	214	212	97	159

Table 3: Average computation time in each time step [ms]

can be seen in Figure 5. Firstly, an improvement of 15-45% in the average rescue time can be observed between the *Decomposition Planning with Gradient Descent* (the state-of-the-art for complex collaborative planning under uncertainty) and our *Joint Planning with HOP* approach for the cases with sufficient available information. In more detail, the *Joint Planning with HOP* approach (P3 or G3) performs significantly better than the other approaches for the same problem setting (P1-2 or G1-2) as the confidence regions are above 1. For the realistic case of inaccurate preliminary information, the rescues are performed 26% earlier for the Ajka scenario, and 22% earlier for the Haiti scenario compared to the gradient descent approach. However, the joint planning approach does not perform well for the *Blank* settings (B3). These settings do not provide enough information about the task distribution for significant improvements using sophisticated planning. In this case the solution for search becomes trivial as the lawnmower pattern (the result of SAPT scheduling) is optimal [9].

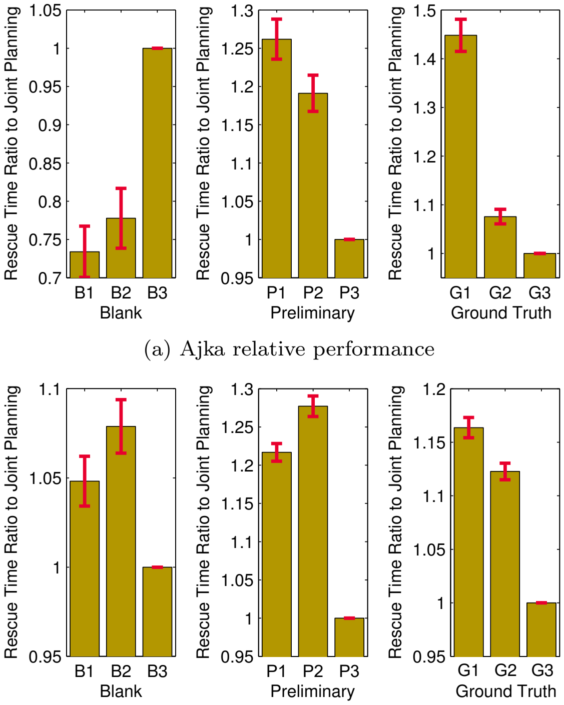
The **Computation Time** of the different approaches is listed in Table 3. The values are significantly below the couple of seconds time limit to result in a reasonably responsive system [23]. The only case when it is around the limit is in the *Blank* setting within the Ajka scenario (highlighted with bold). As discussed earlier, the search area optimisation is not efficient in this setting, because of the large amount of empty areas being searched (see Figure 3). The search plan optimisation (Algorithm 2) has linear complexity in terms of number of agents, but it is cubic in terms of the number of search tasks. The number of search tasks is 400 in the Ajka Blank case, and around 100 in the other test cases, that is reflected by the computation times in Table 3. Overall, the computation time is kept under a practical limit in realistic



(a) Ajka scenario performance

(b) Haiti scenario performance

Figure 4: Average rescue time comparison with 95% confidence intervals.



(a) Ajka relative performance

(b) Haiti relative performance

Figure 5: Relative performance comparison with 95% confidence intervals.

settings with not more than a couple hundred search tasks. For larger search missions with more tasks, we recommend grouping nearby tasks together for a feasible online planning process.

7. CONCLUSIONS

We introduced an algorithm for solving the Uncertain Multi-Robot Task Allocation (UMRTA) problem to control robots in a continuous space given an arbitrary task distribution. We also proposed an approach for heterogeneous SAR robot collaboration under uncertainty, that is the first to create a joint plan for both search and rescue robots. The resulting online planner[6] rescues 26% and 22% earlier compared to the state of the art approaches. Moreover, computation time is maintained on an acceptable level for realistic settings.

For future work, we aim to deploy this work on physical platforms where a robust, communication fault tolerant system is desired. We also plan to investigate applying online learning techniques to improve plans using the experience from previously produced plans.

8. 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.

REFERENCES

[1] A. Allahverdi, C. Ng, T. Cheng, and M. Y. Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985 – 1032, 2008.

[2] A. Ambrus. Analysis of high resolution digital aerial imagery – the mapping of the sludge spill of october 2010 and statistical analysis. Bsc thesis, Eötvös Loránd University, Faculty of Science, 2011.

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

[4] J. Birkmann, editor. *Measuring Vulnerability to Natural Hazards: Towards Disaster Resilient Societies.* United Nations University Press, Dec. 2013.

[5] R. Boumghar and 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)*, pages 215–222. World Scientific, 2011.

[6] A. Brutschy, N.-L. Tran, N. Baiboun, M. Frison, G. Pini, A. Roli, M. Dorigo, and M. Birattari. Costs and benefits of behavioral specialization. *Robotics and Autonomous Systems*, 60(11):1408 – 1420, 2012. Towards Autonomous Robotic Systems 2011.

[7] E. Burns, J. Benton, W. Ruml, S. Yoon, and M. Do. Anticipatory on-line planning. In *International*

[6] The following example videos show the system working: http://bit.ly/1Odyzv4, http://bit.ly/1NBjxMX

Conference on Automated Planning and Scheduling, 2012.

[8] L. Chaimowicz and 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.

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

[10] P. Doherty and P. Rudol. A uav search and rescue scenario with human body detection and geolocalization. In M. Orgun and J. Thornton, editors, *AI 2007: Advances in Artificial Intelligence*, volume 4830 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2007.

[11] F. M. D. Fave, A. Farinelli, A. Rogers, and 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*, pages 2275–2280, 2012.

[12] S. K. Gan, R. Fitch, and S. Sukkarieh. Real-time decentralized search with inter-agent collision avoidance. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 504–510, May 2012.

[13] B. Gerkey and M. Mataric. Sold!: auction methods for multirobot coordination. *Robotics and Automation, IEEE Transactions on*, 18(5):758–768, Oct 2002.

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

[15] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *Robotics Automation Magazine, IEEE*, 13(3):16–25, Sept 2006.

[16] D. Jing, C. Jian, and S. Min. Cooperative task assignment for heterogeneous multi-uavs based on differential evolution algorithm. In *IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009.*, volume 2, pages 163–167, Nov 2009.

[17] J. F. C. Kingman. *Poisson processes*, volume 3. Oxford university press, 1992.

[18] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE SMC '99 Conference Proceedings.*, volume 6, pages 739–743 vol.6, 1999.

[19] Y. Liu, J. Yang, Y. Zheng, Z. Wu, and M. Yao. Multi-robot coordination in complex environment with task and communication constraints. *International Journal of Advanced Robotic Systems*, 10(229), 2013.

[20] E. Lubrano. Drone adventure in haiti, May 2013.

[21] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and 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):44–63, 2013.

[22] M. Nanjanath and M. Gini. Repeated auctions for robust task execution by a robot team. *Robotics and Autonomous Systems*, 58(7):900 – 909, 2010. Advances in Autonomous Robots for Service and Entertainment.

[23] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[24] Police of Hungary. Simulation reproducing the industrial spill at the ajka alumina plant, 2010. Simulated in the OSIRIS system, using the DTA-50 and DDM-50 databases of the Hungarian Defese Froce.

[25] G. Rabadi, M. Mollaghasemi, and 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):1727 – 1751, 2004.

[26] S. Radhakrishnan and 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):2233–2252, 2000.

[27] S. Ramchurn, J. Fischer, Y. Ikuno, F. Wu, J. Flann, and A. Waldock. A study of human-agent collaboration for multi-uav task allocation in dynamic environments. In *International Joint Conference on Artificial Intelligence (IJCAI 2015)*, July 2015.

[28] C. Rasche, C. Stern, W. Richert, L. Kleinjohann, and B. Kleinjohann. Combining autonomous exploration, goal-oriented coordination and task allocation in multi-uav scenarios. In *Autonomic and Autonomous Systems (ICAS), 2010 Sixth International Conference on*, pages 52–57, March 2010.

[29] P. Scerri, T. Von Gonten, G. Fudge, S. Owens, and K. Sycara. Transitioning multiagent technology to uav applications. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track*, AAMAS '08, pages 89–96, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[30] UNOSAT. Haiti earthquake 2010: Remote sensing based building damage assessment data, March 2010.

[31] S. Waharte and N. Trigoni. Supporting search and rescue operations with uavs. In *Emerging Security Technologies (EST), 2010 International Conference on*, pages 142–147, Sept 2010.

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

[33] R. Zivan and 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.