Multi-Agent Search with Interim Positive Information

Haye Lau, Shoudong Huang and Gamini Dissanayake

ARC Centre of Excellence for Autonomous Systems (CAS)
University of Technology, Sydney
NSW, Australia

{hlau, sdhuang, gdissa}@eng.uts.edu.au

***Abstract* − A problem of searching with multiple searchers and scouts is presented. Unlike most search problems that terminate as soon as the target is found, successful detections by scouts only improve on the current knowledge of the moving target's location, such that the searchers can more effectively find and service the target in the future. The team must correspondingly plan not only to maximize the probability of the searchers directly finding the target, but also give them the best chance of exploiting any new information from potential scout detections. It is shown that this need to plan for replanning can be addressed by equivalently solving a series of simpler detection search problems that always do terminate on detection. Optimal and heuristic solution methods for this Searcher/Scout problem are derived, such that the capabilities of all the sensing platforms in a search task are harnessed even when only a subset are capable of actually servicing the target.**

Index Terms – Multiple agent search, target search, branch and bound, optimal searcher path problem.

I. Introduction

Optimizing the search for a moving target constitutes an important problem with applications in many rescue and security scenarios. Given all available knowledge of the environment structure, the likely target locations and searcher capabilities, the objective in general is to find the best trajectory for the searcher to travel such that the chance of finding the target is maximized [1]. Using multiple agents can clearly further improve the chances of finding the target within a limited amount of time [2][3]. Although many applications involve agents that are all capable of servicing (e.g. rescuing or engaging) the target, there may be situations where some of the agents lack the equipment or skill to do so [4]. Those agents should then do what they can to share the workload of the more capable searchers.

This paper considers the search of an area with a heterogeneous team in which the searchers are aided by mobile sensor scouts that can provide target information but cannot directly service the target. Envisaged scenarios include a team of fire-fighters entering a burning building with a number of scouting robots, which then collectively search for the moving victim until a fire-fighter can arrive to render aid. Unlike standard detection search problems that terminate as soon as the target is found, the target is free to continue moving after being detected by a scout and may disappear again from view. Scout detections therefore serve to improve the available target information on which future search actions can be based. Given this opportunity for the agents to react to positive target information, they should thus balance between maximising the searchers' own ability to directly find the target, and ensuring that they can fully exploit potential detections by the scouts. It is shown that this need to plan in anticipation of replanning can be addressed by equivalently solving a series of simpler detection search problems that always do terminate on detection. Building on our recently developed optimal solution for a related detection search problem [1], optimal solution methods for the above Searcher/Scout problem are derived. Given the high computational complexity of optimally solving the problem, a number of faster heuristic methods are also proposed and compared with the optimal approach.

This paper is organized as follows. Section II describes the multi-agent search problem addressed in this paper and provides a detailed definition. A strategy for obtaining optimal policies is presented in Section III and a number of heuristic methods are proposed in Section IV. Section V illustrates the optimal and heuristic solutions with the aid of an example. Section VI discusses the related work and Section VII summarises the paper.

II. Problem Description

A. Searching with the aid of scouts --- Overview

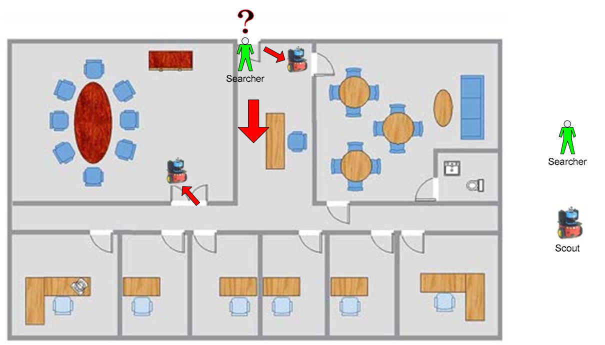

Figure 1. Searcher examining an area with the aid of scouts

Figure 1 depicts a typical scenario that motivates the problem considered in this paper. Two types of agents, searchers (e.g. humans/capable robots) and scouts (mobile sensors), are tasked with locating a target moving through an environment divided into a set of cells. Each agent may move from one cell to another at each time step, subject to the environment's structure. A searcher can detect the target with some finite probability if both it and the target occupy the same cell. The search process terminates once the searcher detects and consequently services the target. A scout may also similarly detect a target it shares a cell with, but it cannot service the target and may only update the team's knowledge to reflect the confirmed current target location. The main role of a scout is therefore to gather information and trigger a change of searcher and scout paths, where necessary, to better intercept the target in the future. A search team consists of one or more searchers working in conjunction with zero or more scouts.

The objective of the team is to maximize the probability of a searcher detecting the target within a given T number of time steps. Although the scouts cannot directly service the target, they can contribute by covering ground not reached by the searcher and revisiting already inspected cells. Clearly, capitalising on information obtained by the scouts would result a higher probability of servicing the target than just using the searchers alone.

B. Problem Statement

The searcher/scout search problem considered in this paper is described as follows:

Environment

In this paper, it is assumed that the environment can be divided into a number of uniform cells $C = \{1,...,N\}$. Figure 2 shows an example of such an environment.

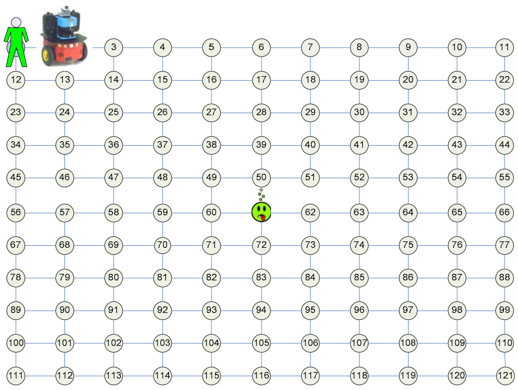

Figure 2. An environment represented by a network of connected cells

Target

The target occupies one cell in $C = \{1,...,N\}$ at a time and moves according to a specified Markov process represented by a matrix Γ at each time step. A prior probability distribution

$$p(\cdot,1) = [p(1,1), p(2,1)...p(N,1)]$$

of the target at time 1 is initially supplied. The distribution evolves according to the formula

$$p(\cdot,t+1) = p(\cdot,t)''\cdot\Gamma,$$

where $p(\cdot,t)''$ denotes the updated undetected target probability at time t, after the effects of sensing by searchers and scouts in the same time step have been taken into account.

Search team

A search team consists of $n \geq 1$ searchers and $m \geq 0$ scouts. The searchers are indexed before the scouts for convenience, i.e., searchers are numbered as agents $1,...,n$ while scouts are numbered as agents $n+1,...,n+m$. Each agent occupies one cell at each time step; let $S_s(i), i \in C, s \in \{1,...,n+m\}$ denote the set of cells that an agent s can directly move to from cell i. The agent s in cell i at time t is constrained to move to a cell $j \in S_s(i)$ at time $t+1$, then $l \in S_s(j)$ at $t+2$, and so on.

Let $X(t)$ be the set of feasible combinations of cells occupied by the agents (team state) at time step $t = 1,...,T$, and $c_s(x), 1 \leq s \leq n+m$ be the cell occupied by agent s in a given state $x \in X(t)$. Membership of $X(t)$ is constrained by $\psi(0)$, the initial state of the team. Let $\psi(1),...,\psi(T), \psi(\cdot) \in X(\cdot)$ be a fixed plan for the searcher and scouts to follow, represented by the series of cells respectively searched by each agent in one time unit increments. At the commencement of the search process, each agent s first moves to and searches cell $c_s(\psi(1))$ for one time period, then travels to search $c_s(\psi(2))$ for another time step, and so on until either the target is serviced or the time limit T is reached.

Target detection

Target detection is modelled as follows: if both an agent s and the target are in cell $c_s(x)$ during time t, detection occurs with a glimpse probability of $0 \leq g_s(c_s(x),t) \leq 1$. Concurrent searches are independent, that is if the target and both agents s and l are in cell i at time t, then the probability of detection is $1-(1-g_s(i,t))\cdot(1-g_l(i,t))$.

Assuming unsuccessful detection, the un-normalised target probability distribution is updated via:

$$p(j,t)' = p(j,t)\cdot\prod_{s=1}^{n}(1 - I_{\psi(t)}(s,j)\cdot g_s(j,t)), j \in \{1,...,N\}$$

$$p(j,t)'' = p(j,t)'\cdot\prod_{s=n+1}^{n+m}(1 - I_{\psi(t)}(s,j)\cdot g_s(j,t)), j \in \{1,...,N\}$$

$$p(\cdot,t+1) = p(\cdot,t)''\cdot\Gamma,$$

where $I_x(s,j)$ is an indicator function that equals 1 if agent s searches cell j when the team is in state x. The product of an empty set is interpreted as 1. $p(\cdot,t)'$ denotes the undetected target probability distribution at time t after the effects of only searcher actions in that time step (and all agent actions prior to that) have been taken into account.

A successful scout detection serves to concentrate the target probability in the cell in which it was found. In particular, when a scout detects a target in cell j at time t, the target distribution $p(\cdot,t)''$ is set to be Q_j, a $1\times N$ vector with element j set to 1 and the remaining elements set to 0, reflecting the confirmed target location at that time. The target is free to continue moving until it has been found by a searcher. The target is free to continue moving until it is found by a searcher.

Objective: Probability of Detection (PD)

Assuming that the scouts never detect the target, the probability of detection of a fixed plan ψ from t to T, given an initial target distribution of p, is:

$$PD'(\psi,t,p)=\sum_{\tau=t}^{T}\sum_{j=1}^{N}p(j,\tau)\cdot(1-\prod_{s=1}^{n}(1-I_{\psi(\tau)}(s,j)\cdot g_s(j,\tau)))$$

For the team to take full advantage of any newly received information, it should be able to change plans whenever the target is detected by a scout. Thus, instead of finding a fixed sequence of cells for each agent to search, a set of plans comprising the initial plan for the team to follow, as well as all the new plans triggered by each possible subsequent scout detection, is required to completely describe the team's actions.

Suppose ψ is now the initial plan for the team to follow. Let $\psi_{x,j,\tau}$ be a new plan that is selected after a scout detects the target in cell j at time τ, given that the agents are in the cells indicated by state x. This plan will then be followed from time step $\tau+1$ to time step T or until a scout detects the target once more. Given the different possible outcomes of scout actions (e.g., a scout s may or may not be successful at each of the cell it inspects) and the correspondingly different target distribution for each eventuality, the effective PD of each set of plans can be calculated recursively via:

$$PD(\psi,t,p)=PD'(\psi,t,p)+$$
$$\sum_{\tau=t}^{T-1}\sum_{j=1}^{N}\left[\begin{array}{l}p(j,\tau)'\cdot(1-\prod_{l=n+1}^{n+m}(1-I_{\psi(\tau)}(l,j)\cdot g_l(j,\tau)))\\ \cdot PD(\psi_{\psi(\tau),j,\tau},\tau+1,Q_j\cdot\Gamma)\end{array}\right]$$

The Searcher/Scout Problem

The goal of the problem considered in this paper can thus be summarised as:

$$\max_{\psi,\psi_{\psi(1),1,1},\ldots,\psi_{\psi(T-1),N,T-1}} PD(\psi,1,p(\cdot,1))$$

Subject to:

$$c_s(\psi(t))\in S_s(c_s(\psi(t-1))),\ \forall s\in\{1,\ldots,n+m\},t\in\{1\ldots T\}$$

$$c_s(\psi_{\psi(\cdot,\tau),j,\tau}(\tau+1))\in S_s(c_s(\psi(\tau)),$$
$$\forall s\in\{1,\ldots,n+m\},\tau\in\{1\ldots T-1\}$$

Remark. If the team does not contain any scouts, this problem reduces to a multi-searcher version of the Optimal Searcher Path Problem (OSP) considered in the literature [3].

III. OPTIMAL POLICIES

This section derives a method for optimally solving the Searcher/Scout problem.

A. Solving the Searcher/Scout problem as a series of smaller OSP problems

Clearly, the probability of a searcher finding the target depends not only on the agents' current paths but also on the alternative paths that will be taken in the event of target detections by scouts. Since the team must additionally plan to take advantage of potential scout detections and adjust accordingly whenever they actually occur, the problem is much more complicated to solve than the Optimal Searcher Path problem (OSP) [5].

Although the search process terminates as soon as the target is found in almost all the search problem formulations considered in the literature, the Generalised Surveillance Search Problem (GSSP) [6] provides one rare framework that assigns rewards to multiple target detections. The formulation could for instance model the surveillance of a smuggling operation in which the smuggler can only be boarded if detection occurs within national boundaries (i.e., in a specific set of cells). Detecting the smuggler outside these boundaries would nevertheless be useful for increasing the likelihood of future success. While the work only optimised the actions of a single searcher whose attention can be infinitely divided over all the cells in the environment without any consideration for path constraints, it made the key observation that the GSSP can be solved as a simpler detection search problem that always terminates as soon as the target is found, provided the best possible outcome leading from each detection in the original problem (which may or may not terminate the process) are already known.

Following similar reasoning, the Searcher/Scout problem may be solved instead as a detection search problem in which the reward for an agent s finding the target at time t when the team is in state $x\in X(t)$ is denoted by $0\le d^*(x,s,t)\le 1$. Naturally $d^*(\cdot,s,\cdot)=1$ if agent s is a searcher, since the process actually terminates on the very first searcher detection and no further reward in the form of detection probability can be gained. However, as the utility of a scout detection ultimately depends on how the team is able to make use of this new information in the future, each value of $d^*(x,s,t),s>n$ can be seen as the optimal payoff from a smaller detection problem in which the target is known to start from cell $c_s(x)$ at time t, the agents begin in state x, and the searches take place during time steps $t+1,\ldots,T$.

In particular, this utility can be computed by solving a modified OSP problem where the reward for detecting the target in the remaining time is not necessarily one. Let $L_{x,p,t+1}(d^*,\psi^*)$ denote the optimal payoff of such an OSP problem over the time periods $t+1,\ldots,T$, given that the team was in state x at time t, the initial target probability distribution at time $t+1$ is p, and the rewards for any detections are given by d^*.

Since each such problem depends only on "future" values of d^*, one can first assign $d^*(x,s,T)=0, \forall s>n$ and then solve a series of progressively larger OSP problems with specified rewards in order to obtain $d^*(x,s,t)$ successively for $t=T-1,...,1$ (Figure 3). The algorithm for finding optimal plans for the multi-agent, discrete Searcher/Scout problem is as follows.

Algorithm for the Searcher/Scout Problem:

1. Let $d^*(x,s,t)=1$ for each $x \in X(t)$, $s \in \{1,...,n\}$, $t \in \{1,...,T\}$, and $d^*(x,s,T)=0$ for each $x \in X(T)$, $s \in \{n+1,...,n+m\}$. Set $t=T-1$.
2. For each $x \in X(t)$, $s \in \{n+1,...,n+m\}$:
   Find ψ^* to maximise $L_{x,Q_{c_s(x)} \cdot \Gamma, t+1}(d^*,\psi^*)$.
   Set $\psi^*_{x,c_s(x),t} = \psi^*$ and
   $d^*(x,s,t) = L_{x,Q_{c_s(x)} \cdot \Gamma, t+1}(d^*,\psi^*)$.
3. Set $t=t-1$.
4. If $t>0$, go to Step 2, else go to Step 5.
5. Find ψ^* to maximise $L_{\psi(0),p(\cdot,1),1}(d^*,\psi^*)$. ψ^* is the optimal plan for the team to first follow. If a scout s finds the target at time t when the team is in state x, the alternative plan $\psi^*_{x,c_s(x),t}$ should then be used.

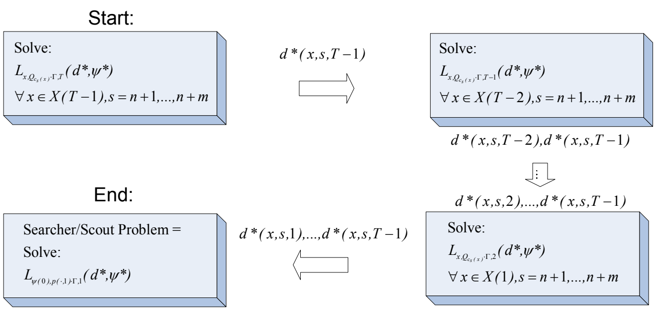

Figure 3 Solving the Searcher/Scout problem via a series of OSP problems

Apart from the use of multiple searchers, the sole difference between $L_{x,p,t}(d^*,\psi^*)$ and the original OSP problem as defined in literature lies in the arbitrary specification of detection rewards, such that:

$$L_{x,p,t}(d^*,\psi^*) = \max_{\psi} \sum_{\tau=t}^{T} \sum_{j=1}^{N} p(j,\tau) \cdot (1 - \prod_{s=1}^{n+m} (1 - I_{\psi}(s,j,\tau) \cdot g_s(j,\tau) \cdot d^*(j,s,\tau))$$

Subject to:

$$c_s(\psi(\tau)) \in S_s(c_s(\psi(\tau-1))), \forall s \in \{1,...,n+m\}, \tau \in \{t+1...T\}$$
$$c_s(\psi(t)) \in S_s(c_s(x)), \forall s \in \{1,...,n+m\}$$

The OSP branch and bound algorithm and DMEAN bounding method described in [1] can be adapted to solve this OSP problem with specified rewards. The next section describes such an approach in more detail.

B Solution to multi-agent OSP problems

Branch and bound method for solving the multi-searcher OSP problem with specified rewards

The algorithm below solves a multi-searcher OSP problem $L_{x,p,t+1}(d^*,\psi^*)$, in which the agents initially begin in the cells indicated by $x \in X(t)$ and are able to search for the target during the time steps $t+1,...,T$.

Following the approach in Section III of [1], let $K(a)$ be a set of 2-tuples $\{nextstate, upperbound\}$ representing plan continuations yet to be explored after a given branch of potential plans has been considered up to time a. The first field refers to the next combination of cells (team state) for the team to search for one time step (at time $a+1$) and the second contains the upper bound associated with this particular extension. r^* holds the best achievable reward discovered thus far. Let $S(x)$ be the set of states that are directly reachable from state x such that $c_s(y) \in S_s(c_s(x)), \forall s, y \in S(x)$.

Algorithm for Branch and Bound (Multi-Searcher OSP with specified rewards):

1. Let $\sigma_t = x$. Set $a=t, K(a)=\{\{\sigma_a, 0\}\}$ and r^* to a value below 0.
2. If $K(a)$ is empty, let $a=a-1$, else go to 4.
3. If $a=t$, go to 9, else go to 2.
4. Selection: Remove from $K(a)$ the tuple $\{\sigma_a, r_a\}$ with the highest bound.
5. If $r_a < r^*$, this extension can be fathomed. Go to 2.
6. Else Branch: For each state $\sigma_c \in S(\sigma_a)$, if $a<T$, obtain r_c, the upper reward bound for any plan beginning with the sequence $\{\sigma_t,...,\sigma_a,\sigma_c\}$. Add tuple $\{\sigma_c, r_c\}$ to $K(a+1)$.
7. If no tuples were added to $K(a+1)$, the current path extension $\{\sigma_a, r_a\}$ is a leaf and no more searches can be done within the time horizon. Let $r^* = r_a$ and store $\{\sigma_t ..., \sigma_a\}$ as the incumbent best path. Go to 2.
8. Else let $a=a+1$, go to 4.
9. Stop, the last saved plan is optimal with the maximum reward of r^*.

Since $d^*(\cdot,s,T)=0, \forall s>n$, scout positions are irrelevant for the last time step T and only states with unique combinations of searcher positions in $\sigma_c \in S(\sigma_a)$ need to be expanded. An algorithm that can be used to efficiently obtain the upper reward bound for Step 6 is described in the following.

Adapted DMEAN bound for OSP problem with specified rewards

The following algorithm computes an upper bound of the reward obtainable for any extension of a partially defined plan $\sigma_t,...,\sigma_k$ from time steps $k+1$ to T. This can then be used to fathom unpromising nodes in the *Algorithm for Branch and Bound*. Let $\pi(k)$ denote the target probability distribution at time k after the effects of searches have been taken into account and $\Gamma_{l,i}$ be the probability that a target known to be in cell l will move to cell i at the next time step.

Algorithm for the DMEAN Bound (for OSP with specified rewards $d*$)

1. For each time step from k to T, create a graph node per cell at that time. Mark node $\{\sigma_k, k\}$ as valid.
2. Use $P(\cdot,\tau) = \pi(k)\cdot\Gamma^{\tau-k}$ to calculate $P(\cdot,\tau)$ for $k < \tau \le T$.
3. From each valid node $\{x,\tau\}, \tau < T$, extend arcs to all nodes $\{y,\tau+1\}, y \in S(x)$ and mark them as valid.
4. Assign DMEAN_Reward$(y,\tau+1,x)$ as the weight for each such new arc to $\{y,\tau+1\}$.
5. Repeat 3 until arcs have been extended from all valid nodes.
6. Apply a longest path algorithm for directed acyclic graphs to find the maximum length for any path leading from node $\{\sigma_k, k\}$ to any node $\{\cdot, T\}$ at time T. Add this to the known reward of following sequence $\sigma_t,...,\sigma_k$ to form the upper reward bound of any continuation.

Function *DMEAN_Reward*(y,v,x)

1. Set $U = P(\cdot,v)$ and $R = 0$.
2. If $v = k+1$, go to step 4.
3. For each cell $i = 1,...,N$, set:
   $$U(i) = U(i) - \sum_{l=1}^{N} P(l,v-1)\cdot(1-\prod_{s=1}^{n+m}(1-g_s(l,v-1)\cdot I_x(s,l)))\cdot\Gamma_{l,i}$$
   $U(i)$ is now the probability that the undetected target is in cell i at time v, given that the cells indicated by x were searched in time $v-1$ as well.
4. For each agent s, in decreasing order of $d*(y,s,v)$:
   Set $pd = U(c_s(y))\cdot g_s(c_s(y),v)$.
   Set $U(c_s(y)) = U(c_s(y)) - pd$.
   Set $R = R + pd\cdot d*(y,s,v)$.
5. Return R.

A more detailed description of the DMEAN bounding method can be found in Section IV of [1].

IV. PRACTICAL HEURISTICS

Due to the large state space of the Searcher/Scout problem, it is presently only feasible (with a MATLAB implementation on a current desktop processor) to optimally solve problems up to 9×9 cells in size with one searcher and one scout over $T = 10$ time steps. This section describes a number of applicable heuristics that may be necessary to obtain solutions for problems with a larger number of cells, more agents, or longer time horizons. Instead of computing the complete set of possible plans, the overall approach uses a run-time heuristic to generate a single plan for the agents to follow and replans whenever a scout detects the target.

Heuristic solutions to the OSP problem with specified rewards

Dell et al. [3] reviewed and compared a number of heuristics for multi-searcher OSP problems, including simple local search techniques, genetic algorithms and the use of branch and bound in a rolling horizon approach. In particular, two expected detection heuristics, titled H1 and H2 [8], were shown to be capable of obtaining good solutions in the problem sets attempted.

The H1 heuristic functions by repeatedly taking the first step of a plan that maximises the expected number of detections (ED) as the next step of the heuristic plan, updating the target probability distribution conditioned on non-detection at that step, and then computing another maximum ED plan again for the remaining time steps until an entire plan is constructed. In contrast to directly maximising PD, the heuristic takes advantage of the speed with which paths maximising ED can be computed.

Since simply maximising ED instead of PD can occasionally lead to clearly suboptimal actions, the H2 heuristic augments H1 by considering maximum ED paths starting from each possible state for the team to be next in. The candidate next state whose maximum ED path leads to the highest PD is then added as the next step of the plan. This approach however comes at the cost of examining many more paths. Further details for H1 and H2 can be found in [3].

With respect to the need in this paper to solve OSP problems with arbitrary rewards for detection by different agents at different times and locations, the H1 and H2 algorithms can be adapted to similarly function with paths maximising expected rewards instead of ED. Let H1r and H2r represent the suitably modified versions of the H1 and H2 heuristics, respectively. Similarly, let H1rd and H2rd be variants of H1r and H2r in which paths maximising discounted expected rewards in the sense of DMEAN_Reward are used. As suggested in [1], H1rd and H2rd would require similar computation effort as their non-discounted counterparts H1r and H2r. The details of these modified algorithms are omitted due to space limitations.

Replanning Heuristics (R1, R1d, R2, R2d) for Searcher/Scout problems

Using an appropriate heuristic for solving OSP problems with specified rewards, one can quickly generate a plan for the agents to follow and replan whenever a scout detects the target. The R1, R1d, R2 and R2d heuristics for the Searcher/Scout problem operate in this manner and are defined as follows:

Replanning Algorithm (R1d, R1, R2d, R2):

1. Solve $L_{\psi(0),p(\cdot,1),\Gamma,1}(\hat{d}^*,\hat{\psi}^*)$ using the H1rd, H1r, H2rd, or the H2r heuristic, respectively.
2. While T times steps have not been used and the target has not been serviced,
   a. In the absence of scout detections, follow plan $\hat{\psi}^*$.
   b. If scout s detects the target at time t, solve $L_{\hat{\psi}^*(t),Q_{c_s(i)},\Gamma,t+1}(\hat{d}^*,\hat{\psi}^*)$ using the chosen OSP heuristic and follow this new plan instead.

This approach has the advantage of quickly obtaining a useable plan. However, the optimality is reduced since one must estimate the rewards associated with future scout detections without the benefit of fully solving additional OSP problems. For the purpose of this paper, the approximate reward $\hat{d}^*(x,s,t), s>n$ is set to be 1 if at least one searcher can possibly intercept a target starting from cell $c_s(x)$ in the remaining time, and zero otherwise. Scout detections are therefore optimistically treated as searcher detections as long as a searcher can respond to this new information in time.

V. RESULTS

This section shows the computation results for a searcher/scout search problem in a 7×7 grid (Figure 4) using one searcher and one scout. The target is known to be in the centre cell at time step 1 ($p(25,1)=1$). At each subsequent time step it will either stay in the same cell (60% probability) or choose to move to a neighbouring cell (40% probability). The target probability mass thus spreads out gradually from the centre cell and first reaches cell 11 in time step 3. Both agents start in cell 1 and have a 60% glimpse probability. The search takes place over $T=10$ time steps.

Optimal Solutions

The optimal initial plan obtained by the *Algorithm for the Searcher/Scout Problem* described in Section III-A is shown in Figure 4. It calls for the searcher to head towards the centre of the grid (favouring the left side), with the scout travelling also towards the central area in an almost parallel track along the right. The plan thus corrals the bulk of the probability mass between the two agents. Should the target be found to be in cell 11 by the scout at time step 4, however, the plan then changes accordingly to contain the target. Figure 5 shows the revised search paths for this case. The optimal Probability of Detection (PD) when using one searcher and one scout (PD = 0.40630) is larger than that of a single searcher alone (Figure 6, PD = 0.33069) but is smaller than when two searchers are used (Figure 7, PD = 0.51715).

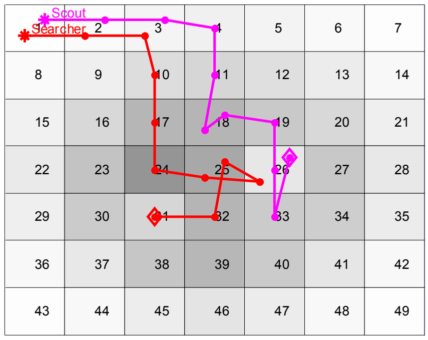

Figure 4 Initial optimal search paths for 1 searcher and 1 scout. Asterisks denote starting location and diamonds the final cell searched. PD = 0.40630. Shading of the cells corresponds to the final target location probability, which could be used to replan the actions of the team if the target was not found after the execution of the plan.

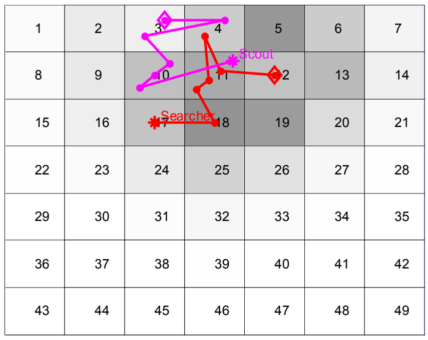

Figure 5 Revised optimal search paths if the scout detects the target in cell 11 at time step 4. Note that at step 4, the searcher is in cell 17 and the scout is in cell 11.

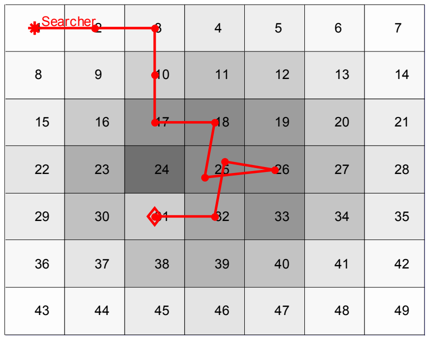

Figure 6 Optimal search path for 1 searcher only. PD = 0.33069.

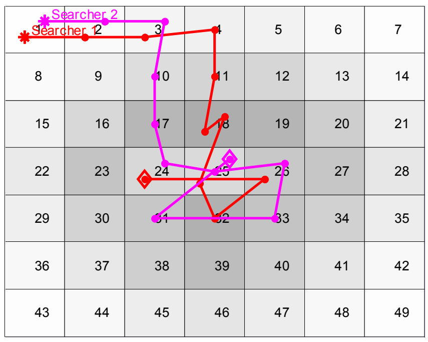

Figure 7 Optimal search paths for 2 searchers. PD = 0.51715.

The optimal set of plans for the example using one searcher and one scout was computed in 6765 seconds using a MATLAB implementation on a 2.6-Ghz AMD Opteron 152 processor. While much can be gained from a more efficient software implementation (for example, a C++ branch and bound implementation was shown to significantly reduce the time for solving OSP problems in [13]), faster heuristic methods would be necessary to address larger problems within reasonable time frames.

Heuristic Solutions

Figure 8 to Figure 11 show the respective initial paths generated by the R1d, R1, R2d and R2 heuristics for the 7×7 cell example problem with one searcher and one scout. The corresponding probabilities of detection for the approaches, calculated by similarly replanning for all possible scout detections, are also shown. While the heuristic plans are less efficient than the optimal solution, they can provide reasonable outcomes with a fraction of the computation time.

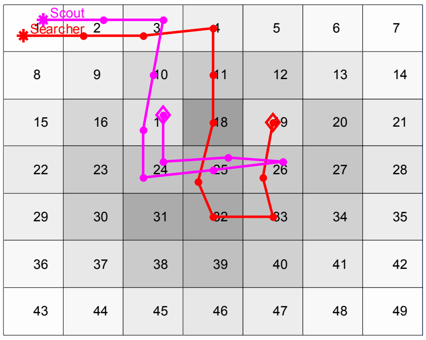

Figure 8 Initial paths obtained using R1d for example described in the first paragraph of Section IV. PD = 0.36487. MATLAB computation time: 0.81s.

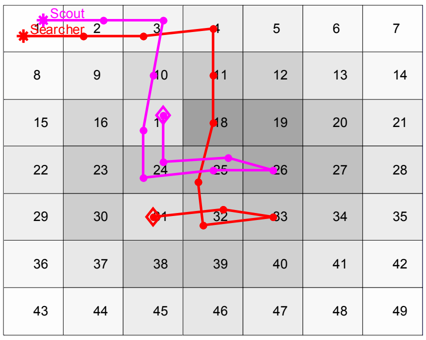

Figure 9 Initial paths obtained using R1. PD = 0.36413. MATLAB computation time: 0.67s.

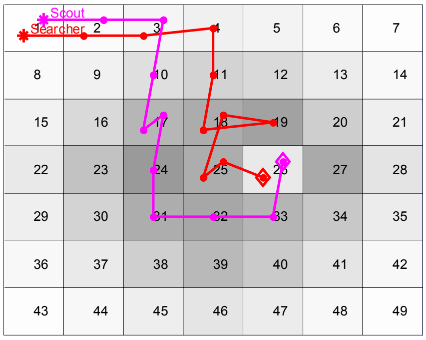

Figure 10 Initial paths obtained using R2d. PD = 0.31598. MATLAB computation time: 7.77s.

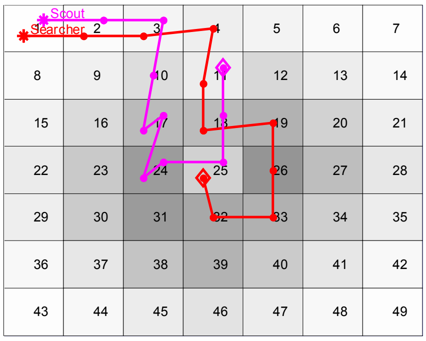

Figure 11 Initial paths obtained using R2. PD = 0.30758. MATLAB computation time: 6.39s.

Although it is difficult to compare search processes through just the initial plans alone, the fact that the scout's cell at time step $T-1$ is not reachable by the searcher at time T (Figure 10 and Figure 11) may partially explain the lower effectiveness of the R2d and R2 plans in this case. While none of the heuristics are superior to the others for all problem instances, experiments with a larger number of examples showed the R1d and R1 methods to generally produce more effective plans.

VI. Related Work

Except for cases involving false target detections [9], a searcher in most problems considered in search literature can only update its target location estimates based on failed attempts to detect the target. The closely related whereabouts [10], surveillance search, and cumulative reward problems [6] provide the main examples where a searcher can benefit from information gained through multiple target detections. The Generalised Surveillance Search Problem (GSSP) was formulated in [6] to incorporate these problems in addition to standard detection search problems. The main difference of [6] with this paper lies in its use of a single searcher and focus on infinitely divisible search effort. The work also did not consider constraints on how the search effort can be relocated from one time step to another, which renders each sub-problem much more complex. Stewart [11] proposed a problem in which the target leaves behind evidence in each cell it visits, such that the single searcher may also gain information from finding target trails. The emphasis was on exploiting clues deposited by the target in the environment, which differs from the aim of this paper.

The need for the agents to continue searching beyond the first target detection also brings the Searcher/Scout problem closer to the combined task of search and tracking. Furukawa et al. [12] examined the case where both helicopters and Unmanned Aerial Vehicles cooperate to search for drifting lifeboats using the Bayesian approach in [2], but only the former are capable of performing rescue. Once one or more targets have been found, the platforms then change to optimize their actions with respect to tracking an assigned target. The work is different from the Searcher/Scout problem in terms of representing the platforms' motion in the continuous domain, the consideration of multiple targets (although target assignment was not discussed) and the explicit switching between searching and tracking tasks. The solution method adopted was also not concerned with optimizing actions with respect to all future possibilities for replanning.

VII. CONCLUSIONS

This paper examined a problem of searching with heterogeneous agents, which augments the OSP problem in literature for cases when some agents can detect the target but cannot meaningfully engage it or perform rescue. The fact that the process does not necessarily terminate on target detection renders it more complicated to plan for and evaluate than a multi-searcher OSP problem. A solution approach was proposed that equivalently partitions the overall problem into a series of dependent, but importantly smaller, OSP problem instances. Optimal solutions consisting of the paths for the search team to initially follow as well as all the alternative paths should a scout detects the target were obtained. Optimal solutions, however, are not practical for realistically large scenarios due to the high complexity of the problem. To this end, a range of suboptimal methods based on heuristics adapted from literature were also developed and compared against the optimal with the aid of an example.

The proposed strategies for heuristically solving the Searcher/Scout problem involved only planning for the team's actions until the next scout detection and then replanning as necessary. Naturally, there are many other possible approaches that may strike a different balance between plan quality and speed. Research along this direction is currently under way.

The technique for optimally solving the Searcher/Scout problem can be extended to support the more general case where an agent may need to spend a number of time steps to move from one cell to another, as discussed in [13], provided that care is taken to synchronise the agents' actions. A complicating factor is that one must first define the expected behavior of an agent that is still transiting between cells when a scout detects the target. The development of optimal solutions for the Searcher/Scout problem with non-uniform travel times between cells is also part of future work.

ACKNOWLEDGMENT

This work is supported by the ARC Centre of Excellence Programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

REFERENCES

[1] Lau, H., Huang, S., and Dissanayake, G., 2006, 'Probabilistic search for a moving target in an indoor environment', In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06)*, Beijing, China, pp. 3393-3398.

[2] Bourgault, F., Furukawa, T., and Durrant-Whyte, H. F., 2003, 'Coordinated decentralized search for a lost target in a Bayesian world', In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, Las Vegas, Nevada, pp. 48-53.

[3] Dell, R. F., Eagle, J. N., Martins, G. H. A., and Santos, A. G., 1996, 'Using multiple searchers in constrained-path, moving-target search problems', *Naval Research Logistics*, vol. 43, pp. 463-480.

[4] Rybski, P. E., Papanikolopoulos, N. P., Stoeter, S. A., Krantz, D. G., Yesin, K. B., Gini, M., Voyles, R., Hougen, D. F., Nelson, B., Erickson, M. D., 2000, 'Enlisting rangers and scouts for reconnaissance and surveillance', *IEEE Robotics & Automation Magazine*, vol. 7, issue 4, pp. 14-24.

[5] Trummel, K. E. and Weisinger, J. R, 1986, 'The complexity of the optimal searcher path problem', *Operations Research*, vol. 34, no. 2, pp. 324-327.

[6] Tierney, L. and Kadane, J. B., 1983, 'Surveillance search for a moving target', *Operations Research*, vol. 31, no. 4, pp. 720-738.

[7] Washburn, A. R., 1998, 'Branch and bound methods for search problems', *Naval Research Logistics*, vol. 45, pp. 243-257.

[8] Martins, G. H. A., 1993, *A new branch-and-bound procedure for computing optimal search paths*, Master's Thesis, Naval Postgraduate School.

[9] Stone, L. D., 1989, *Theory of optimal search*, 2nd Edition, Academic Press.

[10] Stone, L., D., and Kadane, J. B., 1981, 'Optimal whereabouts search for a moving target', *Operations Research*, vol. 29, pp. 1154-1166.

[11] Stewart, T. J., 1985, 'Optimizing search with positive information feedback', *Naval Research Logistics*, vol. 32, no. 2, pp. 263-274.

[12] Furukawa, T., Bourgault, F., Lavis, B., and Durrant-Whyte, H. F., 2006, 'Recursive Bayesian Search-and-Tracking using coordinated UAVs for Lost Targets', In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, Florida, pp. 2521-2526.

[13] Lau, H., Huang, S., and Dissanayake, G., 2007, 'Discounted MEAN bound for the Optimal Searcher Path problem with non-uniform travel times', *European Journal of Operational Research*, accepted with minor revisions.