Information Value-Driven Approach to Path Clearance with Multiple Scout Robots

Maxim Likhachev
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
maximl@seas.upenn.edu

Anthony Stentz
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
axs@rec.ri.cmu.edu

***Abstract*— In the path clearance problem the robot needs to reach its goal as quickly as possible without being detected by enemies. The robot does not know the precise locations of enemies, but has a list of their possible locations. These locations can be sensed, and the robot can go through them if no enemy is present or has to take a detour otherwise. We have previously developed an approach to the path clearance problem when the robot itself had to sense possible enemy locations. In this paper we investigate the problem of path clearance when the robot can use multiple scout robots to sense the possible enemy locations. This becomes a high-dimensional planning under uncertainty problem. We propose an efficient and scalable approach to it. While the approach requires centralized planning, it can scale to very large environments and to a large number of scouts and allows the scouts to be heterogenous. The experimental results show the benefits of using our approach when multiple scout robots are available.**

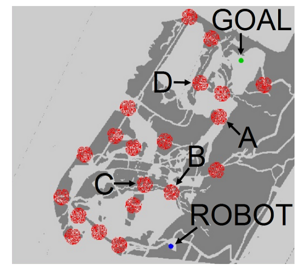
I. Introduction

In the path clearance without scout robots [5], the planner must come up with a plan for a robot whose task is to reach its goal as quickly as possible without being detected by an enemy. The robot (we will often refer to it as the main robot as opposed to the scout robots) does not know beforehand the precise locations of enemies, but has a list of their possible locations. When navigating, the robot can come to a possible enemy location, sense it using its long range sensor and go around the area if an enemy is detected or cut through this area otherwise.

The example in figure 1 demonstrates the path clearance problem without scout robots. Figure 1(b) shows the traversability map of the satellite image of a 3.5km by 3km area shown in figure 1(a). The traversability map is obtained by converting the image into a discretized 2D map where each cell is of size 5 by 5 meters and can either be traversable (shown in light grey color) or not (shown in dark grey color). Larger circles are possible enemy locations and their radii represent the sensor range of enemies (100 meters in this example). The radii can vary from one location to another. The locations can be specified either manually or automatically in places such as narrow passages. Each location also comes with a probability of containing an enemy (50% for each location in the example): the likelihood that the location contains an enemy. The probabilities can vary from one location to another.

The path the robot follows may change any time the robot senses a possible enemy locations (the sensor range of the

(a) 3.5 by 3.0 km satellite image



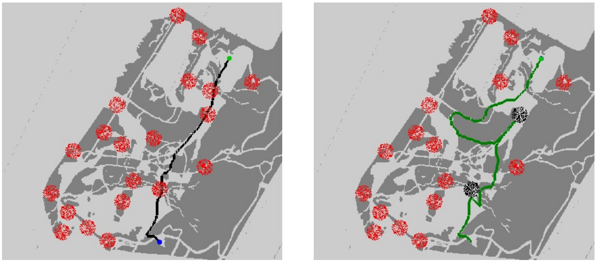
(b) corresponding traversability map

Fig. 1. Path clearance problem

robot is 105 meters in our example). A planner, therefore, needs to reason about possible outcomes of sensing and generate a plan (policy) that dictates which path the robot should take as a function of the outcome of each sensing. An example of such plan is shown in figure 3(a). This plan should ideally minimize the expected time for the robot to reach its goal. The planning problem, however, is hard since it requires planning under uncertainty about the environment.

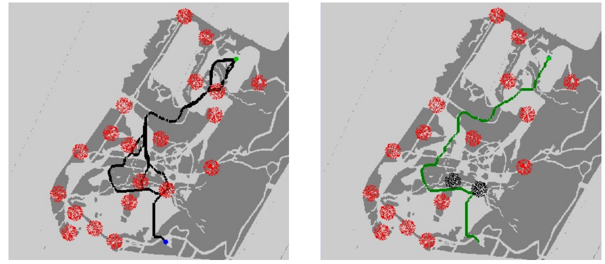
In [5], we have explained how we can use our previously developed PPCP algorithm [4] to solve very efficiently the problem of path clearance without scout robots. In this paper we investigate how we can address the problem when multiple scout robots are available. This problem is more complex because it requires the planner to generate the plans for scout robots: where they should go, what possible enemy locations they should sense.

Solving the problem optimally is intractable. In this paper we therefore propose an approach that solves the planning problem approximately. In brief, the approach can be summarized as the strategy of sending the scouts to such possible enemy locations as to maximize the expected decrease in the cost of the main robot plan, or in other words, the value of information - that is, by how much the plan that the robot follows can be improved if it was known whether a particular possible enemy location contains an enemy or not. We don't have the exact values of the cost decreases (these would be very expensive to compute), but we do have estimates for these values as a by-product of running PPCP when computing the plan for the main robot. We use these estimates. While the proposed approach finds an approximation of the optimal plan, it is computationally cheap, it scales very well to a large number of scouts, it can



(a) planned path (b) actual path of the robot

Fig. 2. Solving path clearance problem with freespace assumption



(a) generated plan (b) actual path of the robot

Fig. 3. Solving path clearance problem with PPCP

accommodate heterogeneous team of scouts and it shows a substantial decrease in the time it takes for the main robot to reach its goal.

II. Background

The path clearance problem falls into the general class of problems of planning under uncertainty about the environment. Such problems correspond to the problems of planning for Partially Observable Markov Decision Processes (POMDPs). Planning optimally for POMDPs is known to be hard (PSPACE-complete [7]), and various approximations techniques have been recently proposed instead [9], [8], [10]. The most relevant to our work on the path clearance is the work in [1]. This work develops a planning algorithm in the framework of robot navigation in unknown terrain and was shown to be capable of finding optimal policies several orders of magnitude faster than other optimal approaches. The motivation behind the development of our approach to the path clearance problem, however, is to avoid planning in the exponentially large belief state-spaces altogether as the above mentioned algorithms do. The algorithm that we use to plan under uncertainty is PPCP [4] that solves the high-dimensional planning problem via a series of low-dimensional (two-dimensional) searches. This makes it scalable to very large environments and usable in real-time. In the present work, we also show how to extend it to a higher dimensional problem of path clearance with multiple scouts.

The path clearance problem is closely related to the problem of robot navigation in a partially-known environment. In the latter, the robot senses the environment and also has to take a detour every time it senses an obstacle on its path. The difference is that in the path clearance problem, the area that is rendered untraversable if an enemy is detected is usually large. Therefore, the penalty for discovering an enemy can be very large.

To avoid the computational complexity, a robot operating in a partially-known environment often performs assumptive planning [6], [2], [11]. In particular, it often just follows a shortest path under the assumption that all unknown areas in the environment are free unless the robot has already sensed them otherwise. This is known as a freespace assumption [2]. The robot follows such path until it either reaches its goal or senses new information about the environment. In the latter case, the robot re-computes and starts following a new shortest path under the freespace assumption.

The freespace assumption is also applicable to the path clearance problem. The robot can always plan a path under the assumption that no enemy is present unless sensed otherwise. This makes the path clearance problem a deterministic planning problem and therefore can be solved efficiently. The fact that the robot ignores the uncertainty about the enemies, however, means that it risks having to take long detours, and the detours in the path clearance problem tend to be longer than in the problem of navigation in a partially-known environment as we have just explained.

For example, figure 2(a) shows the path computed by the robot that uses the freespace assumption. According to the path, the robot tries to go through the possible enemy location A (shown in figure 1(b)) as it is on the shortest route to the goal. As the robot senses the location A, however, it discovers that the enemy is present in there (the circle becomes black after sensing). As a result, the robot has to take a very long detour. Figure 2(b) shows the actual path traversed by the robot before it reaches its goal.

In [5], we have shown how the path clearance problem without scout robots can be solved efficiently using the PPCP algorithm [4]. As shown in [4], the plans PPCP returns are guaranteed to be optimal (minimize the expected cost) under certain conditions and are also often optimal in practice even when the conditions are not met. Figure 3(a) shows the plan returned by PPCP after it converged in about 30 seconds for our example. Every place where the plan branches out corresponds to where the robot senses a possible enemy location and chooses to go through it if no enemy is detected, or take a detour otherwise. In contrast to planning with freespace assumption, the plan produced by PPCP makes the robot go through the area on its left since there are a number of ways to get to the goal there and therefore there is a high chance that one of them will be available. The length of the actual path traversed by the robot (figure 3(b)) is 4,123 meters while the length of the path traversed by the robot that makes the freespace assumption (figure 2(b)) is 4,922 meters.

To the best of our knowledge, the work in [3] is the first to address the path clearance problem with multiple scouts. In it, after the plan for the main robot has been computed by PPCP, the scouts are sent to sense all the possible enemy locations that lie on any of the branches of the computed plan (in other words, all the locations through which the main robot may try to go through when following its plan).

This is a very greedy approach, however. For example, since the plan for the main robot, shown in figure 3(a), does not involve going through the location A (the location A is shown in figure 1), no scout is sent to sense it. This can be highly sub-optimal however. If a scout is near location A and the probability that location A contains an enemy is not too high, it is beneficial to send the scout to sense the location. If it turns out to be empty, then the main robot should go through this location as it will result in a much lower travel time. The present paper attempts to address this problem by reasoning whether to sent scout robots to the possible enemy locations that lie both on and off the plan for the main robot. At the same time, the approach continues to be efficient and scalable to large environments and large number of scout robots.

III. The Approach

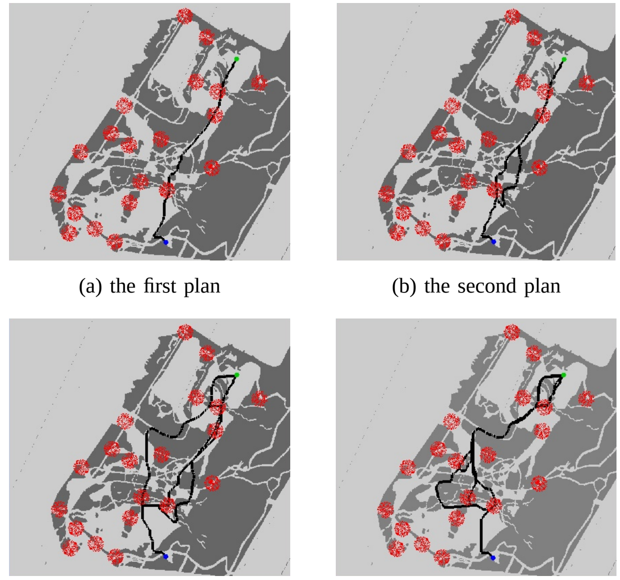
Our approach can be described as follows. We first compute a plan for the main robot using PPCP. If we have K scout robots available, we then decide what K possible enemy locations the scout robots should be sent to. We do this by picking the locations knowing whose status can result in the expected least-cost plan for the main robot. In other words, scouts are sent so as to maximize the value of the information they obtain. The main robot starts following its plan, while the scout robots are sent to sense the selected enemy locations (each scout is assigned to the nearest possible enemy location that was selected and was not yet assigned to any other scout).

In the next section we will briefly describe the PPCP algorithm and how it is used to generate a plan for the main robot in the path clearance example. A detailed description of the algorithm can be found in [4]. In the subsequent section we will describe the actual algorithm for selecting possible enemy locations for sensing by scout robots.

A. PPCP Algorithm

The PPCP algorithm is a general algorithm developed for planning under uncertainty about the environment. In the following, however, we will describe it specifically on the problem of path clearance. The algorithm assumes that the environment itself is fully deterministic and can be modeled as a graph. Thus, in the path clearance problem, if we knew the precise location of the enemies, then there would be no uncertainty about any robot actions: both sensing and move actions would be deterministic actions. In reality, there is, however, uncertainty about the actual location of enemies. As a result, there are two possible outcomes of a sensing action: an enemy *is* present at the possible enemy location that is being sensed and an enemy *is not* present.

The PPCP algorithm gains its efficiency by exploiting the fact that many planning under uncertainty problems including the path clearance problem have clear preferences on the state of variables that represent uncertainty. That is, for each action that has more than one possible outcome we can name beforehand what outcome we would prefer. Thus, in the path clearance problem, for each sensing action, it is clearly preferred to find out that an enemy is not present at



(a) the first plan (b) the second plan (c) after 5 secs (d) after 30 secs (PPCP converged)

Fig. 4. Operation of PPCP

the sensed location. This outcome would allow the robot to cut through the location on its way to the goal. The existence of clear preferences allows PPCP to solve the whole planning problem by running a series of two-dimensional A*-like searches instead of planning in the belief state-space which is an MDP whose size is exponential in the number of possible enemy locations.

Figure 4 shows the operation of PPCP. The algorithm first constructs an initial plan (figure 4(a)) by running an A*-like search from the robot position to the goal location assuming all enemy locations are free. Thus, the plan it generates is identical to the one generated by planning with freespace assumption as shown in figure 2(a). The plan, however, has not yet considered the alternative (non-preferred) outcomes of sensing. For example, it does not compute the detour the robot should take if the location B (see figure 1(b) for labels) does contain an enemy. At the next step, PPCP finds this detour by setting the location B to be untraversable and executing the A*-like search again but now to find a path from the position at which the robot senses the location B to the goal. Once the path is found, it is incorporated into the current plan. The new plan is shown in figure 4(b).

In the next iteration, PPCP tries to find a path from the robot position to the goal again but now the A*-like search it executes takes into account the length of the detour the robot has to take if the location B contains an enemy and the robot has to take detour. In this manner, PPCP explores alternative plans. Thus, after 5 seconds, its currently best plan is shown in figure 4(c). This plan already makes the robot try to avoid going through the location A by going through the area on its left first.

PPCP continues to run A*-like searches until convergence. At the time of convergence, the plan PPCP returns is guaranteed to minimize the expected execution cost under certain conditions (described in [4]). Figure 4(d) shows the plan PPCP returns after it converged in 30 seconds. The plan avoids going through the location A altogether. The robot can also stop PPCP earlier and begin following the plan PPCP currently has.

B. Incorporating Scouts

The algorithm for deciding what possible enemy locations the scouts should sense is driven by one objective: to minimize the expected cost (time to reach the goal) of the plan for the main robot. Suppose set S is the set of possible enemy locations that need to be sensed by scouts. Ideally, this set should be chosen such as to minimize the expected cost of the joint plan for the main robot and scout robots. In reality, the minimization of this plan is intractable because the size of the problem is exponential in the number of scout robots and in the number of possible enemy locations. We therefore make a number of approximations.

First, rather than considering all possible choices of S at the same time, we select possible enemy locations to S one by one. We also select at most K locations, where K is the number of scouts available. We do not try to solve the problem of coming up with a sequence of enemy locations that scouts have to sense. Instead, at any point of time we try to find possible enemy locations that available scouts have to sense next.

Given this approximation, the question is what possible enemy location should be sensed next and what scout should be assigned to it. In order to avoid planning in the joint state-space, the second approximation we make is that we ignore the fact that while a scout travels to sense a possible enemy location, the main robot also travels along its route. This is equivalent to assuming that once we choose a possible enemy location to be sensed, it is sensed instantaneously. We may still penalize the cost of sensing and travel by the scout though. Under these assumptions, the next location s to sense should be chosen such as to minimize the expected cost of executing an optimal plan for the main robot given that the status of s is known, plus the expected cost of sensing s. Mathematically, it can be expressed as:

$$s = \text{argmin}_{s'} E\{c(\pi^*(s' \text{ is known})) + costofsensing(s')\}$$

In this equation, $\pi^*(s' \text{ is known})$ stands for the optimal plan for the main robot that takes into account the fact that the status of s' is known, $c(\pi^*(s' \text{ is known}))$ is the cost of executing this plan, and the expectation is taken over all possible configurations of possible enemy locations including s'. The equation can also be re-written as:

$$s = \text{argmax}_{s'} E\{c(\pi^*) - c(\pi^*(s' \text{ is known})) - costofsensing(s')\} \quad (1)$$

In this equation, π^* stands for the optimal plan for the main robot that assumes that the status of all possible enemy locations including s' is unknown. In general, the expected values of the quantities $c(\pi^*)$ and $c(\pi^*(s' \text{ is known}))$ is hard to compute since they require finding optimal policies. However, as we have explained in the previous section, PPCP works by initially considering an optimistic plan (all possible enemy locations are free) and then using more and more informative estimates on the policies. We use this property of PPCP to estimate the quantity $E\{c(\pi^*) - c(\pi^*(s' \text{ is known}))\}$.

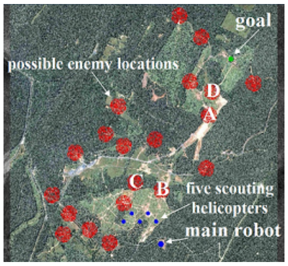
In particular, whenever PPCP produces a plan that makes the main robot to sense the location s' at some position x, PPCP itself computes an estimate of the expected cost of an optimal plan for the main robot starting at x for two scenarios: s' does not contain an enemy (denoted by $v(x, s' = \text{no enemy})$), and s' does contain an enemy (denoted by $v(x, s' = \text{enemy present})$). PPCP also computes the likelihood that the robot reaches position x when following its plan. We will denote this quantity as $P(x)$. The quantity $(v(x, s' = \text{enemy present}) - v(x, s' = \text{no enemy}))$ gives the cost of a detour due to having an enemy at s'. This detour will *have* to be taken if s' lies on the current plan of the main robot, the robot reaches s' and finds out that it contains an enemy. Therefore, we use $(v(x, s' = \text{enemy present}) - v(x, s' = \text{no enemy}))\, P(s' = \text{enemy present}) P(x)$ as an estimate of $E\{c(\pi^*) - c(\pi^*(s' \text{ is known}))\}$ if s' lies on the current plan of the main robot. The length of the same detour is by how much the expected cost of the robot's plan can at most decrease if s' does not lie on its current plan, but the robot switches to a plan that does contain s' and s' is known to be free of enemies. We therefore use $(v(x, s' = \text{enemy present}) - v(x, s' = \text{no enemy}))\, P(s' = \text{no enemy}) P(x)$ as an estimate of $E\{c(\pi^*) - c(\pi^*(s' \text{ is known}))\}$ whenever s' does not lie on the current plan of the main robot.

The term $E\{costofsensing(s')\}$ can be computed as the minimum expected cost of sensing s' across all available scouts. After we compute an estimate of $E\{c(\pi^*) - c(\pi^*(s' \text{ is known})) - costofsensing(s')\}$ for each s', we pick s' that maximizes it and assign it to the scout which minimized the term $E\{costofsensing(s')\}$. In the experiments, the scouts were helicopters and therefore the term $E\{costofsensing(s')\}$ was computed as the time it takes for a scout to reach the center of the location s' which was proportional to the Euclidean distance between the two.

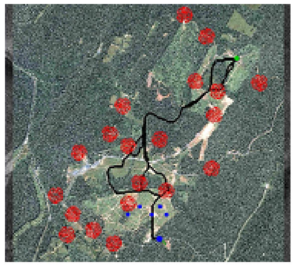
C. Example

Figure 5 shows the operation of our approach. Figure 5(a) shows the initial configuration. It is the same environment with the same set of possible enemy locations as in figure 1 with the only difference that now there are five scouting helicopters shown as smaller dots. (The main robot is shown as a larger dot.)

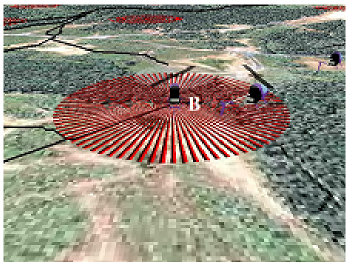
The PPCP planner starts planning a plan for the main robot and after 30 seconds it converges to the final plan shown in figure 5(b). Once the plan is ready, the main robot starts following it. At the same time, the scouting helicopters are assigned to the possible enemy locations (according to the algorithm described in section III-B). Figure 5(c) shows how some of the helicopters fly towards enemy locations to sense them. In particular, the first locations that are being sensed are locations B and C. Figure 5(d) shows that the location C turns out to be free of enemies, whereas a scout did detect an enemy in the location B. Once this information is passed



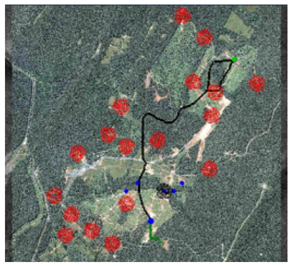
(a) initial configuration



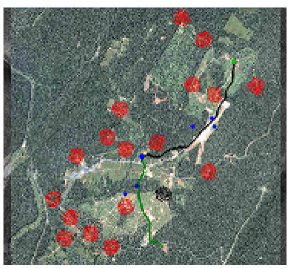
(b) plan after convergence (30 secs)



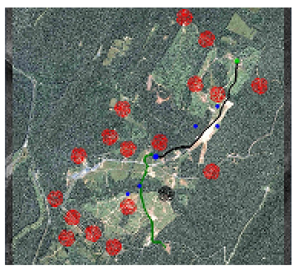
(c) scouting helicopters on mission



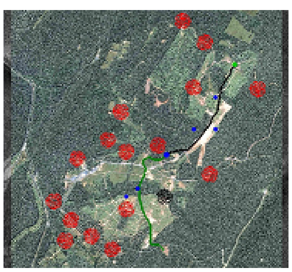
(d) locations B and C sensed (enemy detected in B)



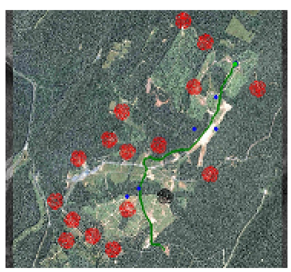
(e) location A sensed



(f) scout moves towards location D



(g) location D sensed



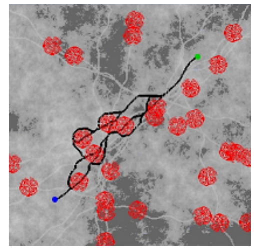
(h) main robot reaches its goal

Fig. 5. Planning with PPCP for the main robot and five scouts

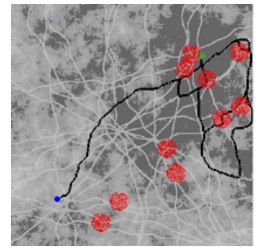
to the main robot, PPCP re-computes a plan. The new plan, also shown in figure 5(d), no longer directs the main robot towards the location B since it is already known to contain an enemy.

Note that even though the plan for the main robot in figure 5(d) does not involve going through the location A, one of the helicopters is still flown to detect this location. The reason for this is that after the helicopter detects that it is free, the main robot can now follow a much faster route towards the goal that goes through the location A that was just cleared. This is shown in figure 5(e): the location A was cleared and the new plan re-computed by PPCP makes the robot go through the location A.

In a similar fashion, one of the helicopters is sent to sense



(a) A typical group I environment



(b) A typical group II environment

Fig. 6. The example of environments used in testing and the plans generated by PPCP for each.

the location D even though it is also not on the plan that the main robot follows (figure 5(f)). After this location is cleared, a faster route for the main robot is re-computed by PPCP that cuts through location D (figure 5(g)). Figure 5(h) shows the final trajectory of the main robot.

IV. Experimental Analysis

This section reports the results of the evaluation of our approach in simulation. We have evaluated it on randomly generated fractal environments that are often used to model outdoor environments [12]. On top of these fractal environments we superimposed a number of randomly generated paths in between randomly generated pairs of points. The paths were meant to simulate roads through forests and valleys and that are usually present in outdoor terrains. Figures 6(a,b) show typical environments that were used in our experiments. The lighter colors represent more easily traversable areas. All environments were of size 500 by 500 cells, with the size of each cell being 5 by 5 meters.

The test environments were split into two groups. Each group contained 25 environments. For each environment in the group I we set up 30 possible enemy locations at randomly chosen coordinates but in the areas that were traversable. Figure 6(a) shows a typical environment from the group I together with the plan generated by PPCP. For each environment in the group II we set up 10 possible enemy locations. The coordinates of these locations, however, were chosen such as to maximize the length of detours. This was meant to simulate the fact that an enemy may often be set at a point that would make the robot take a long detour. In other words, an enemy is often set at a place that the robot is likely to traverse. Thus, the environments in group II are more challenging. Figure 6(b) shows a typical environment from the group II together with the plan generated by PPCP. For each possible enemy location the probability of containing an enemy was set at random to a value in between 0.1 and 0.9.

Tables I and II show the execution costs incurred by the main robot that uses different planning approaches. Table I is for the case when the speed of the scouts is the same as the speed of the main robot, whereas table II gives results for the case when the speed of scouts is four times faster than that of the main robot. In both scenarios, however, the scouts are assumed to be helicopters and therefore do not need to avoid obstacles on the ground. In all experiments there were 10 scouts.

	Group I		Group II	
	cost	overhead	cost	overhead
freespace, no scouts	5,602	8.9%(+/-2.6%)	4,595	14.2%(+/-3.8%)
PPCP, no scouts	5,351	4.1%(+/-1.8%)	4,405	10.9%(+/-3.6%)
PPCP, greedy scouts	5,168	1.6%(+/-1.3%)	4,055	3.0%(+/-1.5%)
PPCP, info. value scouts	5,076	0.0%(+/-0.0%)	3,931	0.0%(+/-0.0%)

TABLE I

EXPERIMENTS I. THE SPEED OF THE SCOUTS IS THE SAME AS THE SPEED OF THE MAIN ROBOT. NUMBERS IN PARENTHESES GIVE 95% CONFIDENCE INTERVALS.

	Group I		Group II	
	cost	overhead	cost	overhead
freespace, no scouts	5,601	12.1%(+/-2.8%)	4,595	16.8%(+/-3.9%)
PPCP, no scouts	5,349	7.6%(+/-2.6%)	4,405	13.5%(+/-3.8%)
PPCP, greedy scouts	4,988	1.6%(+/-0.7%)	3,927	2.9%(+/-1.6%)
PPCP, info. value scouts	4,902	0.0%(+/-0.0%)	3,819	0.0%(+/-0.0%)

TABLE II

EXPERIMENTS II. SCOUTS ARE FOUR TIMES FASTER THAN THE MAIN ROBOT. NUMBERS IN PARENTHESES GIVE 95% CONFIDENCE INTERVALS.

In each of the tables, the first row corresponds to planning with freespace assumption, as described in section II, and not utilizing scout robots. The main robot itself does sensing for enemies. The second row corresponds to planning with PPCP but again without utilizing scouts. The third row corresponds to planning with PPCP and using a greedy strategy for scouts that makes them always sense the possible enemy locations that are on one or more branches of the current plan for the main robot (see section II). Finally, the fourth row corresponds to planning with PPCP and utilizing scouts in the way we have described in section III-B (information value-driven approach).

In all of the experiments, the robot was given 5 seconds to plan while traversing 5 meter distance. This amount of time was always sufficient for planning with freespace assumption to generate a path. The PPCP planning, on the other hand, was interleaved with execution as we have explained in [5]. Thus, none of the approaches required the main robot to stop and wait for a plan to be generated.

The tables show execution costs, as well as the overhead in execution cost when planning with different approaches relative to the execution cost when planning with PPCP and using the information value-driven approach to scheduling scout robots (the last rows of the tables). Each entry is an average over 8 runs for each of the 25 environments in each group (200 samples total). For each run the true status of each enemy location was generated at random according to the probability having an enemy in there.

Table I shows that the overhead of *not* utilizing scouts while planning with freespace assumption can be up to 14.2%. This overhead goes even higher if the speed of the scouts is higher than the speed of the main robot. The overhead of *not* utilizing scouts while planning with PPCP is also substantial (up to 10.9% when the speeds are the same and up to 13.5% when the scouts move faster). The difference between the two approaches to utilizing scouts (the last two rows in each table) is smaller. The execution cost of the main robot utilizing scouts in a greedy fashion can on average be up to 3.0% worse. While this overhead may not seem to be very large, the overall behavior of scouts following the information value-driven approach is much more intelligent, and on the environments that were *not* randomly generated, such as the example in figure 1, the overhead of the greedy approach can be much higher. Unlike the greedy approach, the information value-driven approach is capable of taking advantage of the cases when sensing a possible enemy location that is not on the current plan of the main robot can result in much faster route for the main robot.

V. CONCLUSIONS

This paper presents an approach to solving the path clearance problem when multiple scout robots are available to sense possible enemy locations. While the approach requires centralized planning, its important advantages are that it is simple, efficient, scales to large environments and scales to a large number of heterogenous scout robots. The experimental results demonstrated the scalability of the approach and its benefits as compared to alternative approaches. It is future work to deploy the proposed approach on the team of real outdoor terrain vehicles.

VI. ACKNOWLEDGMENTS

This work was sponsored by the U.S. Army Research Laboratory, under contract Robotics Collaborative Technology Alliance (contract number DAAD19-01-2-0012). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

REFERENCES

[1] D. Ferguson, A. Stentz, and S. Thrun. PAO* for planning with hidden state. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004.

[2] S. Koenig and Y. Smirnov. Sensor-based planning with the freespace assumption. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1996.

[3] M. Likhachev and A. Stentz. Path clearance using multiple scout robots. In *Proceedings of the Army Science Conference (ASC)*, 2006.

[4] M. Likhachev and A. Stentz. PPCP: Efficient probabilistic planning with clear preferences in partially-known environments. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2006.

[5] M. Likhachev and A. Stentz. Goal directed navigation with uncertainty in adversary locations. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[6] I. Nourbakhsh and M. Genesereth. Assumptive planning and execution: a simple, working robot architecture. *Autonomous Robots Journal*, 3(1):49–67, 1996.

[7] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processses. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[8] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[9] N. Roy and G. Gordon. Exponential family pca for belief compression in pomdps. In *Advances in Neural Information Processing Systems*, 2002.

[10] M.T.J. Spaan and N. Vlassis. A point-based POMDP algorithm for robot planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2399–2404, 2004.

[11] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[12] A. Stentz. Map-based strategies for robot navigation in unknown environments. In *AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems*, 1996.