

PATH CLEARANCE USING MULTIPLE SCOUT ROBOTS

Maxim Likhachev* and Anthony Stentz

The Robotics Institute

Carnegie Mellon University

Pittsburgh, PA, 15213

maxim+@cs.cmu.edu, axs@rec.ri.cmu.edu

ABSTRACT

This paper presents the techniques that we have been developing recently to solve the problem of path clearance. In the path clearance problem the robot needs to reach its goal as quickly as possible without being detected by enemies. The problem is complicated by the fact that the robot does not know the precise locations of enemies, but has a list of their possible locations. Either the robot itself or scout robots can be used to sense these possible enemy locations before the robot traverses through them on the way to its goal. We have recently developed a general and efficient algorithm called PPCP (Probabilistic Planning with Clearance Preference) for planning under uncertainty in the environment. In this paper we first describe how it can be applied to the problem of path clearance when there are no scout robots available and show that there are significant benefits of planning with PPCP over commonly used alternatives. We then explain a strategy for how to use the PPCP algorithm in case multiple scout robots are available and show that this strategy reduces the time it takes for the robot to reach its goal even further.

1 INTRODUCTION

In the problem of path clearance, the task of the robot is to reach its goal location as quickly as possible without being detected by an enemy. We explain the problem on the example in figure 1. 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. Each cell in this map is of size 5 by 5 meters and can either be traversable (shown in light grey color) or not (shown in dark grey color). The robot's initial position is shown by the blue circle and its goal is shown by the green circle. Red circles are *possible* enemy locations and their radii represent their sensor range. In this example, the sensor range of enemies is

100 meters for every location. In general, though, we assume that it can vary from one location to another.

The possible enemy locations can be specified either manually or automatically in places such as narrow passages. Each location is associated with a probability of containing an enemy: the likelihood that the location does actually contain an enemy. In the given example, it is 50% for each possible enemy location. In general, though, we allow for it to vary from one location to another.

When navigating, the robot can come to a possible enemy location, sense it using its long range sensor (the range is assumed to be 105 meters in the example) and go around the area if an enemy is detected or cut through it otherwise. When planning for its route, though, the robot must take into account the possible enemy locations. For example, it may plan a path under the commonly used assumption that no enemy is present unless sensed otherwise (known as freespace assumption (Koenig & Smirnov, 1996)). With this assumption, however, it risks having to take long detours. For example, figure 1(c) shows the path planned by the robot that makes the freespace assumption. The path goes through the possible enemy location A, as it is on the shortest route to the goal. When the robot tries to execute it though, it discovers that an enemy is present at the location A (the red circle becomes black after sensing). Consequently, the robot has to take a very long detour. The actual robot's path is shown in figure 1(d).

A planner, therefore, needs to reason about possible outcomes of sensing and consider the consequences of these outcomes. In fact, such planner would generate more than a single path, it would generate a policy that dictates which path the robot should take as a function of the outcome of each sensing. Ideally, this policy should minimize the expected traversal distance of the robot.

Path clearance is just one example of many real-world problems where a planner needs to reason about the uncertainty in the environment. Examples of such

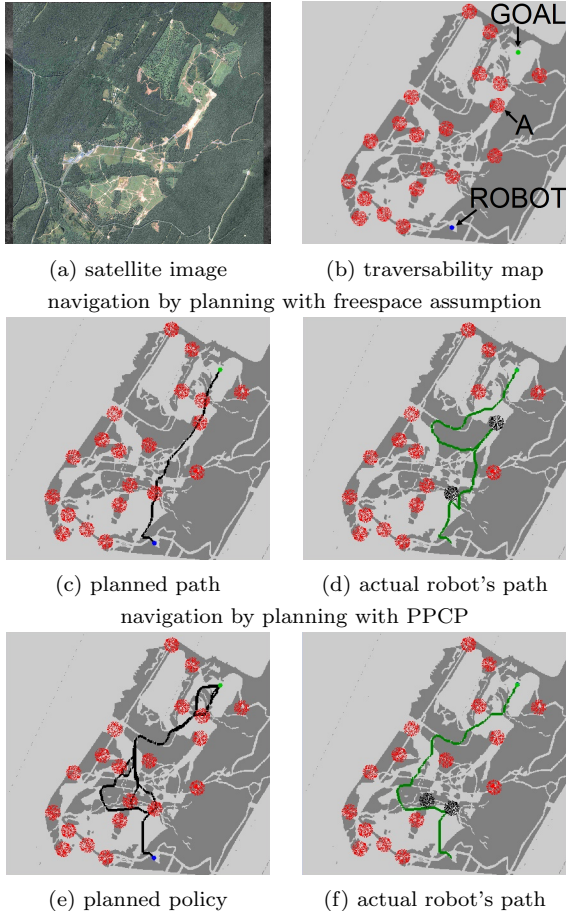


Figure 1: Path clearance problem. (c,d) show the initially planned path and the actual robot’s path when planning with freespace assumption. (e,f) show the policy and the actual robot’s path when planning with PPCP

problems include robot navigation in partially-known terrains, robot navigation in office-like environments where some doors can potentially be locked, route planning under partially-known traffic conditions, air traffic management with changing weather conditions and others. Ideally, in all of these situations, to produce a plan, a planner needs to reason over the probability distribution over all the possible instances of the environment. Unfortunately though, such planning in general corresponds to POMDP planning, which is known to be hard (PSPACE-complete (Papadimitriou & Tsitsiklis, 1987)). It turns out, however, that all of the above mentioned problems and many other ones exhibit a special property: one can clearly name beforehand the best (called *clearly preferred*) values for the variables that represent the unknowns in the environ-

ment. In the robot navigation problem in outdoor terrains, for example, it is always preferred to find out that an initially unknown location is traversable rather than not. In the robot navigation in office-like environments, it is always preferred to have a door unlocked rather than locked. In the problem of route planning, it is always preferred to have the lowest level of traffic on the road of interest. And in the air traffic management problem it is always preferred to have a good flying weather. PPCP (Probabilistic Planning with Clear Preferences) (Likhachev & Stentz, 2006) is a recently developed algorithm that scales up to large problems with a significant amount of uncertainty by exploiting this property. In this paper we show that the algorithm is also applicable to the path clearance problem since in this problem it is also always preferred to find out that an enemy is not present at the location of interest.

Figure 1(e) shows the policy produced by the PPCP algorithm (Likhachev & Stentz, 2006) applied to the path clearance problem. Each fork in the policy is where the robot tries to sense an enemy location and chooses the corresponding branch. In contrast to planning with freespace assumption, the policy 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 1(f)) is 4,123 meters while the length of the path traversed by the robot that makes the freespace assumption (figure 1(d)) is 4,922 meters.

In the following section we describe the PPCP algorithm. In section 3 we explain how we can apply the PPCP algorithm to solve the problem of path clearance when there are no scouts available. The experimental analysis in this section shows that planning with PPCP can result in substantial savings in execution cost in comparison to planning with freespace assumption. In section 4 we extend our method and present a strategy that takes advantage of available scout robots. We show that using scouts can decrease the time for the robot to reach its goal even further.

2 PPCP ALGORITHM

In this section we briefly describe the PPCP algorithm: what assumptions it makes and how it operates at a high level. A more detailed description of the algorithm can be found in (Likhachev & Stentz, 2006).

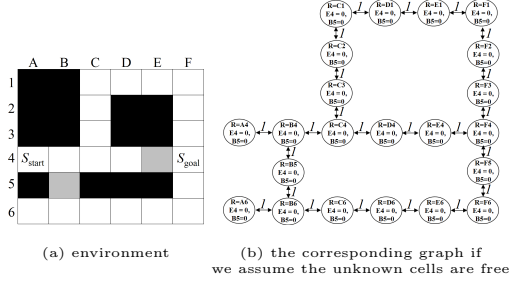


Figure 2: Robot navigation in a partially-known terrain. The numbers above graph edges represent the costs of the corresponding actions.

2.1 The assumptions

The PPCP algorithm is well-suited to planning in the environments that are only partially-known. 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. 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* detected at the possible enemy location that was sensed and an enemy *is not* detected. PPCP assumes perfect sensing. Once the robot sensed a particular location, it then knows with full certainty whether an enemy is present in there or not.

Figure 2 demonstrates these assumptions on a problem that is very similar to the path clearance problem: the problem of robot navigation in a partially-known terrain problem. Initially, the robot is in cell A4 and its goal is cell F4. There are two cells (shaded in grey) whose status is unknown to the robot: cell B5 and cell E4. For each, the probability of containing an obstacle is 0.5. In this example, we restrict the robot to move only in four compass directions. Whenever the robot attempts to enter an unknown cell, we assume that the robot moves towards the cell, senses it and enters it if it is free and returns back otherwise. The cost of each move is 1, the cost of moving towards an unknown cell, sensing it and then returning back is 2. Figure 2(b) shows the graph that corresponds to the problem when the status of cells B5 and E4 is known: both are free. In this graph each action has only one outcome. In reality, however, the robot does not know the status of these cells before it actually senses them. As a result, the action of moving east from cell D4 has two possible outcome states: (1) the robot enters cell E4 and now

knows that cell E4 is free; (2) the robot remains in cell D4 and now knows that cell E4 is obstructed. We use the notation $[R = E4; B5 = u, E4 = 0]$ to represent the former case and $[R = D4; B5 = u, E4 = 1]$ to represent the latter ($B5 = u$ denotes the fact that the status of cell B5 is unknown). Each outcome of the action is associated with the probability of seeing this outcome. In this example, the probability of each outcome is 0.5 since it is the probability of cell E4 being blocked.

The goal of the planner is to construct a policy that reaches any state at which $R = F4$ while minimizing the expected cost of execution. Figure 3(h) shows the policy generated by PPCP. The policy specifies the path the robot should follow after each outcome of sensing operation. All branches of the policy end at a state whose $R = F4$, in other words, the robot is at its goal location.

The main assumption that PPCP makes is that for each action that has more than one possible outcome we can name beforehand what outcome we would prefer. Thus, in the figure 2 example for each sensing action there exist two outcomes: a cell is blocked or unblocked. The latter is clearly the preferred outcome. The same property holds in the path clearance problem. For each sensing action, it is clearly preferred to find out that an enemy is not present at the sensed location. This outcome would allow the robot to cut through the location on its way to the goal.

2.2 The operation

The PPCP algorithm operates in anytime fashion. It quickly constructs an initial policy and then refines it until convergence. At the time of convergence, the policy it obtains is guaranteed to minimize the expected execution cost under certain conditions (described in (Likhachev & Stentz, 2006)). Figure 3 demonstrates how PPCP solves the problem of robot navigation in a partially-known terrain given in figure 2.

PPCP repeatedly executes deterministic searches that are very similar to the A* search (Nilsson, 1971), an efficient and widely-known search algorithm. During each iteration PPCP assumes some configuration of unknown cells and performs search in the corresponding deterministic graph. Thus, the first search in figure 3 assumes that both unknown cells are free and finds a path that goes straight to the goal (figure 3(a)). (The shown g - and h -values are equivalent to the g - and h -values maintained by the A* search.) PPCP takes this path and uses it as an initial policy for the robot (figure 3(b)). One of the actions on this policy, however, is move east from cell D4. The current policy

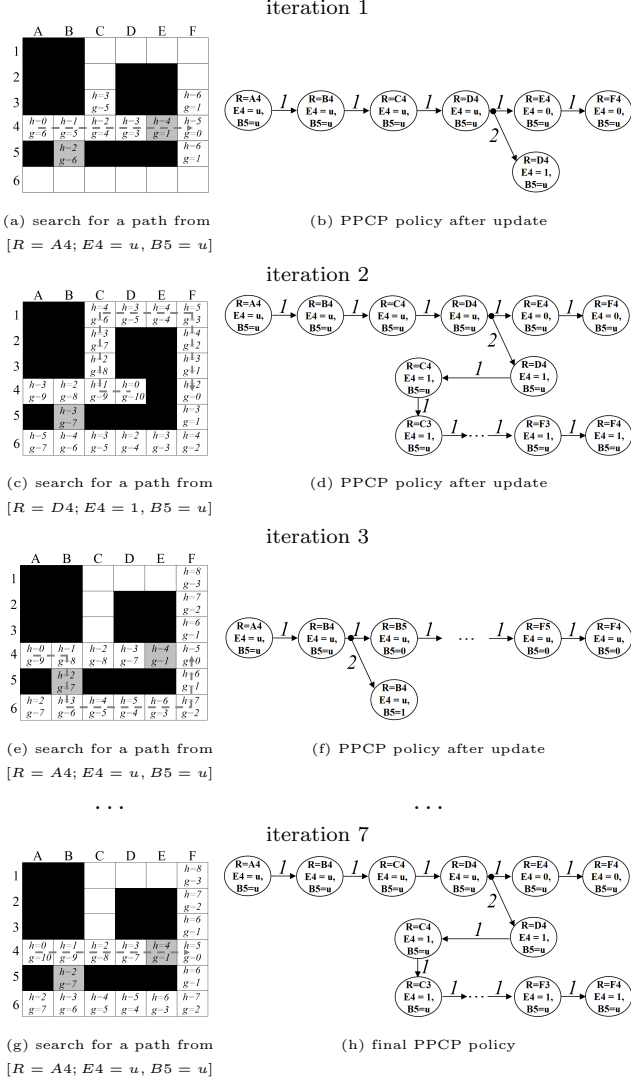


Figure 3: Example of how PPCP operates

has only computed a path from the preferred outcome state, the one that corresponds to cell D4 being free. The state $[R = D4; E4 = 1, B5 = u]$, on the other hand, has not been explored yet. The second search executed by PPCP explores this state by finding a path from it to the goal. This search is shown in figure 3(c). During this search cell E4 is assumed to be blocked, same as in the state $[R = D4; E4 = 1, B5 = u]$. The found path is incorporated into the policy maintained by PPCP (figure 3(d)).

In the third iteration, PPCP tries to find a path from the start state to the goal again (figure 3(e)). Now, however, it no longer generates the same path as initially (figure 3(a)). The reason for this is that it has learned that the cost of trying to traverse cell

E4 is higher than what it initially thought to be. The cost of the cheapest detour in case cell E4 is blocked (found in figure 3(c)) is rather high. Consequently, in the current iteration PPCP finds another alternative policy that goes through cell B5. This now becomes the new policy in PPCP (figure 3(f)). This policy, however, has an unexplored outcome state again, namely, state $[R = B4; E4 = u, B5 = 1]$. This would become the state to explore in the next iteration.

PPCP continues to iterate in this manner until convergence. In this example, it converges on the 7th iteration. The final policy is shown in figure 3(h). In this example it is optimal: it minimizes the expected cost of reaching the goal. In general, PPCP is guaranteed to converge to an optimal policy if it does not require remembering the status of any cell the robot has successfully entered (see (Likhachev & Stentz, 2006) for more details).

3 PATH CLEARANCE WITHOUT SCOUTS

PPCP can be used to solve the problem of path clearance. In this section we explain how it can be done and analyze the benefits of planning with PPCP. The discussion below assumes that we use the improved version of PPCP. This version as well as some additional details about its application to path clearance without scouts can be found in (Likhachev & Stentz, 2007).

3.1 The application of PPCP

As mentioned before, PPCP is guaranteed to find an optimal plan only if it does not require the robot to remember the outcomes of sensing if they were preferred. In the figure 3 example, for instance, the robot senses an unknown cell whenever it tries to enter it. If it is free then the robot enters it and does not really need to remember that it is free afterwards, its projected path is unlikely to involve entering the same cell again. It only needs to remember if some cells turn out to be blocked, and plans generated by PPCP do retain this information. This is different, however, when the robot has a long range sensor such as in the path clearance problem. The robot may sense whether an enemy is present before actually reaching the area covered by the enemy. It thus needs to remember the preferred outcomes if it wants to sense enemies from a long distance.

The solution to this problem is simple. Each state is augmented with the last k preferred outcomes of sensing. Thus, a state now consists of the location of the robot plus the last k locations that were sensed to be free of enemies. k can be set to a small number such

as two or three to keep the complexity of each search iteration in PPCP low. This adds a sufficient for our problem amount of memory about the preferred outcomes of sensing.

We also would like to be able to interleave planning with execution. Rather than wait until PPCP converges, the robot should start executing whatever current plan PPCP has and while executing it, PPCP should work on improving the plan. The interleaving in PPCP can be done pretty trivially. The robot first executes PPCP for few seconds. PPCP is then suspended and the robot starts following the policy currently found by PPCP. After each robot move, the state of the robot maintained by PPCP is updated. During each robot move, PPCP is resumed for the duration of the move. It automatically re-plans a policy so that its start corresponds to the current state of the robot. After it is suspended again, the policy that the robot currently follows is compared against the policy that PPCP currently has. The robot's policy is updated to the latter one only if it has a higher probability of reaching a goal location. If the robot's policy has a higher probability of reaching goal then the robot continues to follow its old policy. This way we avoid changing policies every time PPCP decides to explore a different policy and has not explored much of the outcomes on it. The probabilities of reaching a goal for a policy can be computed in a single pass over the states on the policy (starting with the robot's state and proceeding in the topological order) because the policies are acyclic. Once the algorithm converges to a final policy, the robot can follow the policy without re-executing PPCP again.

Figure 4 shows the application of PPCP to the path clearance example from figure 1. Before the robot starts executing any policy, PPCP plans for five seconds. Figures 4(a), (b) and (c) show the first, the second and the last policy generated by PPCP within these five seconds, respectively. The robot then starts executing the last policy. (The robot travels at the speed of 1 meter per second.) In the meantime, PPCP continues to refine its policy. After every five seconds the robot updates its state in PPCP and switches to executing the latest policy in PPCP if it has the same or higher probability of reaching the goal. Thus, figures 4(d) and (e) show the position of the robot and its current policy after ten and fifteen seconds of execution, respectively. After 30 seconds of execution, PPCP converges. Figure 4(f) shows the position of the robot at that time and the policy PPCP converged to.

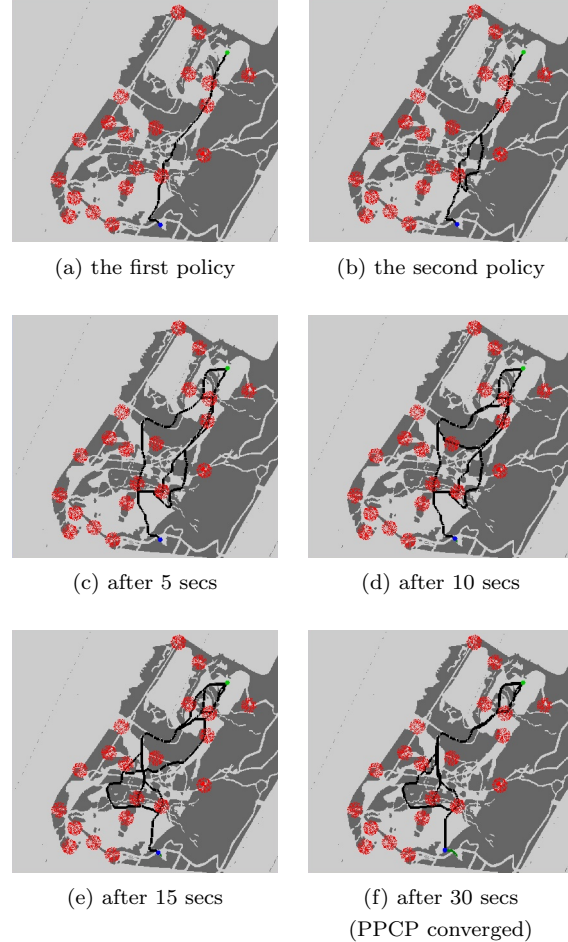


Figure 4: Path clearance using PPCP and no scouts

3.2 Experimental analysis

In this section we evaluate the benefits of solving the path clearance problem by planning with PPCP. In all of the experiments we used randomly generated fractal environments that are often used to model outdoor environments (Stentz, 1996). 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 5(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 ar-

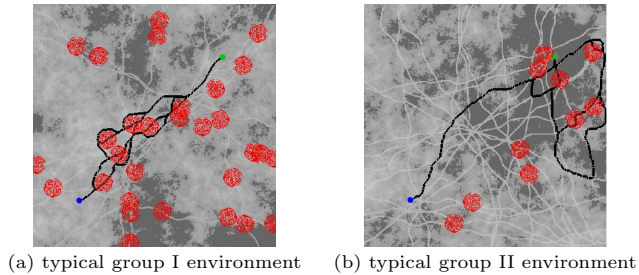


Figure 5: The example of environments used in testing and the plans generated by PPCP for each.

	Overhead in Execution Cost			
	Group I no penalty	Group II no penalty	Group I penalty	Group II penalty
PPCP	0%	0%	0%	0%
freespace	5.4%	5.2%	35.4%	21.6%
freespace2	0.5%	4.9%	4.8%	17.0%
freespace3	2.1%	4.3%	0.0%	12.7%

Figure 6: The overhead in execution cost when planning with freespace assumption over the execution cost when planning with PPCP

eas that were traversable. Figure 5(a) shows a plan the PPCP algorithm has generated after full convergence for one of the environments in group I. 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 the detours that avoid these locations. 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 in order to avoid taking a long detour. Thus, the environments in group II are more challenging. Figure 5(b) shows a typical environment from the group II together with the plan generated by PPCP. The shown plan has about 95% probability of reaching the goal (in other words, the policy has at most 5% chance of robot encountering an outcome for which the plan had not been generated yet). In contrast to the plan in figure 5(a), the plan for the environment in group II is more complex - the detours are much longer - and it is therefore harder to compute. 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.

Figure 6 compares the execution cost of the robot planning with PPCP versus the execution cost of the robot planning with freespace assumption (Koenig & Smirnov, 1996), a commonly used approach to planning under uncertainty in the environment. The plan-

ner operating under the freespace assumption always assumes that an enemy is not present unless it has already been detected. Under this assumption the planner plans a shortest route to the goal which the robot follows unless it detects an enemy on the way. In case it does, the planner re-plans its path. Figure 6 shows the overhead in the execution cost incurred by the robot that planned with the freespace assumption. The overhead is relative to the execution cost incurred by the robot that used PPCP for planning. (Hence, the first row shows zero percents.) The rows freespace2 and freespace3 correspond to making the cost of going through a cell that belongs to a possible enemy location twice and three times higher than what it really is, respectively. One may scale costs in this way in order to bias the paths generated by the planner with freespace assumption away from going through possible enemy locations. The results are averaged over 8 runs for each of the 25 environments in each group. For each run the true status of each enemy location was generated at random according to the probability having an enemy in there.

The figure shows that planning with PPCP results in considerable execution cost savings. The savings for group I environments were small only if biasing the freespace planner was set to 2. The problem, however, is that the biasing factor is dependent on the actual environment, the way the enemies are set up and the sensor range of an enemy. Thus, the overhead of planning with freespace for the group II environments is considerable across all bias factors. In the last two columns we have introduced penalty for discovering an enemy. It simulates the fact that the robot runs the risk of being detected by an enemy when it tries to sense it. In these experiments, the overhead of planning with freespace assumption becomes very large. Also, note that the best bias factor for freespace assumption has now shifted to 3 indicating that it does depend on the actual problem. Overall, the results indicate that planning with PPCP can have significant benefits, without requiring to tune any parameters.

4 PATH CLEARANCE WITH SCOUTS

In this section we present a simple strategy for employing scout robot when available to reduce the time it takes for the non-scout robot to reach its goal. We demonstrate the approach on an example and evaluate the effectiveness of the approach on a large number of runs.

4.1 Algorithm

Suppose there are K scout robots available. We first run PPCP to produce a policy for the non-scout robot as if there are no scout robots available, only the non-scout robot can detect enemies. Once we have the policy, we find K possible enemy locations that have the highest probability of being visited by one or more paths on the robot's policy. In other words, these are the locations that PPCP assumes the robot will sense on one of its branches in the policy. To select the ones that have the highest probabilities of being visited by the robot, we perform a single pass over all the states in the policy in topological order starting with the robot's state. During this pass we can propagate the probability of the robot visiting the state when following the policy according to the probability distribution of the policy action outcomes.

Once we compute these K possible enemy locations, we assign them to the nearest scout robots. Each scout robot starts traveling towards its assigned enemy location and performs sensing when it reaches the location. While each scout robot travels, the non-scout robot starts executing its policy. PPCP is also being executed to improve its policy as we have described in section 3.1. Every time the robot changes its policy onto the policy generated by PPCP, it re-computes the K possible enemy locations that scout robots need to sense and re-assigns them to the scout robots. Also, every time one of the scouts performs sensing, the robot updates the state of the robot that PPCP maintains with the outcome of sensing, so that all the subsequent planning done by PPCP takes this information into account. We then re-compute the K possible enemy locations that the scouts should sense.

Figure 7 shows the operation of our strategy. There are ten scouts available, shown by small blue dots in a two-row formation in figure 7(a). In this example the scouts are assumed to be aerial vehicles moving with the same speed as the robot. As the figures show, at any point in time at most four scout robots are used because this is the maximum number of enemy locations involved in the policy generated by PPCP. The total distance traversed by the robot is 3795 meters which is 328 meters shorter than in case of no scouts (figure 1(f)).

The presented strategy is greedy. For example, it was clearly worthwhile to send a scout to sense the location A (shown in figure 1(b)). If it turned out to be empty, then the path to the goal via this location would have been the shortest possible route for the non-scout robot. The advantages of the presented strategy, on the other hand, are that it is simple, very fast, scales well with a number of scouts and works well even if the

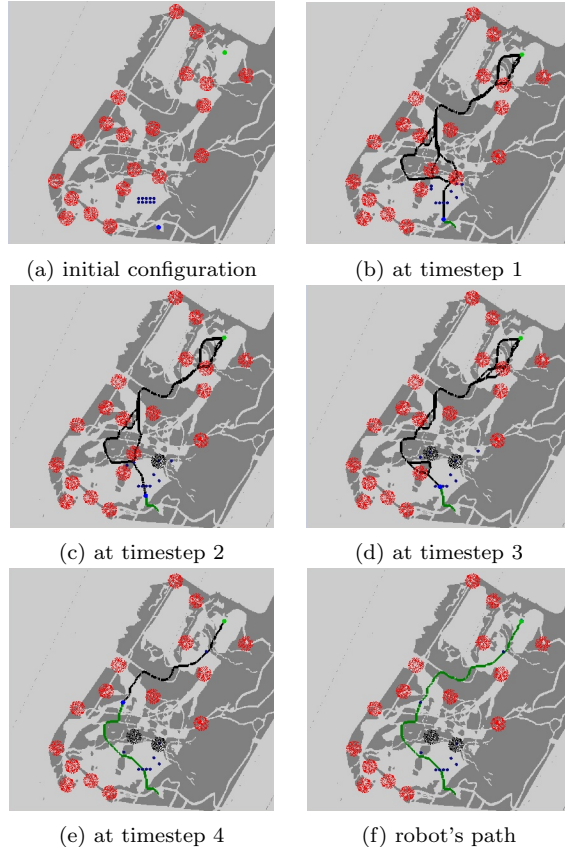


Figure 7: Path clearance using PPCP and 10 scouts

scouts are heterogeneous (for example, a mix of aerial and ground robots).

4.2 Experimental analysis

Figure 8 compares the execution cost of the robot that uses scouts according to the strategy we have just described over the robot that does not. In both cases the robot was planning with PPCP. Just as before, the shown results are averaged over 8 trials for each of the 25 environments for each group. Thus, each entry in the table is an average over 200 runs. The table shows the average of the actual execution cost incurred by the robot. In all the runs with scouts, the number of scouts was limited to 10. All scouts were assumed to be aerial vehicles moving at the same speed as the non-scout robot. According to the results, the robot that employs scouts can expect to incur a considerably smaller execution cost than the robot that does not.

	Execution Cost			
	Group I no penalty	Group II no penalty	Group I penalty	Group II penalty
no scouts	5,352	4,405	5,630	4,938
10 scouts	5,168	4,055	5,472	4,859

Figure 8: Average execution cost of using up to 10 scouts versus not using scouts. In both cases, PPCP was used for planning.

CONCLUSIONS

The goal of this paper was to explain at a high level the PPCP algorithm and to show its applicability to the problem of path clearance. We have shown that it can be quite beneficial to plan with PPCP when navigating in the environment with possible enemies. We have also presented a simple extension that allows to use scout robots, if available, to sense for enemies. This was shown to reduce the execution cost incurred by the non-scout robot even further. We believe that PPCP is a widely applicable algorithm that can be used to solve many planning problems that involve uncertainty in the environment. Its application to the path clearance problem is just one example.

ACKNOWLEDGEMENTS

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

- Koenig, S., & Smirnov, Y. 1996. Sensor-based planning with the freespace assumption. *In: Proceedings of the ieee international conference on robotics and automation (ICRA)*.
- Likhachev, M., & Stentz, A. 2006. PPCP: Efficient probabilistic planning with clear preferences in partially-known environments. *In: Proceedings of the national conference on artificial intelligence (AAAI)*.
- Likhachev, M., & Stentz, A. 2007. *Faster PPCP and its application to navigation with uncertainty in adversary locations*. In submission.
- Nilsson, N. 1971. *Problem-solving methods in artificial intelligence*. McGraw-Hill.
- Papadimitriou, C. H., & Tsitsiklis, J. N. 1987. The complexity of Markov decision processes. *Mathematics of operations research*, **12**(3), 441–450.
- Stentz, A. 1996. Map-based strategies for robot navigation in unknown environments. *In: AAAI spring symposium on planning with incomplete information for robot problems*.