Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 Keller Hall
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 15-021

Constrained Probabilistic Search for a One-Dimensional Random Walker

Narges Noori, Alessandro Renzaglia, Joshua Vander Hook, Volkan Isler

December 21, 2015

Constrained Probabilistic Search for a One-Dimensional Random Walker

Narges Noori, Alessandro Renzaglia, Joshua Vander Hook and Volkan Isler
University of Minnesota, Minneapolis, MN, USA

***Abstract*—This paper addresses a fundamental search problem in which a searcher subject to time and energy constraints tries to find a mobile target. The target's motion is modeled as a random walk on a discrete set of points on the line: at each time step the target chooses one of the adjacent nodes at random and moves there. We study two detection models. In the no-crossing model, the searcher detects the target if they are on the same node or if they take the same edge at the same time. In the crossing model, detection happens only if they land on the same node at the same time.**

For the no-crossing model, where move and stay actions may have different costs, we present an optimal search strategy under energy and time constraints. For the crossing model, we formulate the problem of designing an optimal strategy as a Partially Observable Markov Decision Process (POMDP) and solve it using methods which reduce the state space representation of the belief. The POMDP solution reveals structural properties of the optimal solution. We use this structure to design an efficient strategy and analytically study its performance.

Finally, we present preliminary experimental results to demonstrate the applicability of our model to our tracking system which is used for finding radio-tagged invasive fish.

I. Introduction

One-sided probabilistic search for a mobile target is a fundamental problem which has been studied by various research communities since 1940s [2], [3]. In robotics, monitoring natural environments for the purpose of biology research, finding intruders or locating lost hikers or runaway robots, and search-and-rescue missions are applications that can be modeled as one-sided search problems.

In one-sided probabilistic search problems, the goal is to design search strategies with provable performance guarantees to find a target. The target cannot observe the searcher and does not actively evade detection. Instead, the target's motion is modeled as a stochastic process. The simplest setting is when the target moves in discrete time steps in a discrete world consisting of only two cells according to a Markovian motion model [4]. It turns out that even this simple variant, which is referred to as the *two-cell* problem, is challenging. Pollock [4] uses dynamic programming to derive search strategies that minimize the expected number of looks to detect the target, or maximize the detection probability within a given number of looks for special cases of the two-cell problem. Wilson [5] provide a necessary and sufficient condition on the initial distribution of the target's position such that a search plan with finite expected capture time exists. Dobbie [6] studies the continuous time motion model and solves for optimal strategies that minimize the expected time to detection, or maximize the probability of detection in a given time in the hope to derive formulations that are easier to generalize to more than two cells. Kan [7] also attempted to generalize the problem to N cells by characterizing optimal strategies for special cases, for example when the cells form a clique and the target moves between all cells with equal probability. In fact, it has been shown that the problem of detecting a target, stationary or mobile, in a grid world within a fixed time horizon is NP-complete [8]. As a result, approximation methods have also been studied to tackle the problem. In this regard, branch-and-bound methods [9], [10] and POMDP formulations [11], [12] are popular. For example, Lau et al. [13] use a branch-and-bound framework to find an upper bound on detection probability for relatively small environments with around 20 nodes. Hollinger et al. [11] formulate the problem of maximizing the expected probability of finding the target at the earliest possible time by multiple searchers as a POMDP and use the sub-modularity of the joint discounted reward function to provide a constant factor approximation sequential allocation algorithm.

In this paper, we study the problem of finding a target which is moving according to a simple random walk on a linear graph. The target moves to neighboring nodes, i.e. to the left or right, with a given probability (Fig. 1(a)). The searcher can choose to stay on its current node, or move to the right or to the left. Furthermore, we consider the effect of imposing time or energy constraints for the searcher. Surprisingly, this one-dimensional search problem is open despite its practical applications and fundamental importance.

We study the problem under two detection criteria that we refer to as the *no-crossing* and *crossing* conditions. In the no-crossing model, the target is detected if it is on the same node or the same edge as the searcher. One immediate consideration for the no-crossing scenario is that in this model, the searcher can find the target simply by moving from one end to another. However, in real applications, such a solution is not always applicable due to limited battery life. As a result, it becomes crucial to design efficient search strategies by considering the energy consumption of the searcher. The crossing model where detection occurs only on a discrete set of nodes is more challenging. This is because if the players cross each other by taking the same edge in opposite directions at the same time, the target is not detected (Fig. 1(b)). Consequently, as the searcher sweeps the line the target might be on both sides of the searcher.

This paper is built upon our two earlier conference publications [14] and [1] where we study the crossing model. In [14] we proposed a class of randomized strategies. Within this class

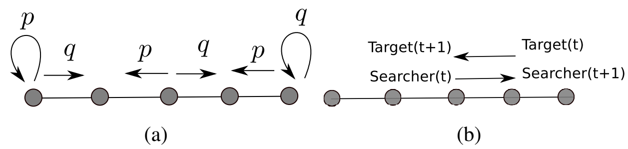


Fig. 1. (a) Target's motion model is a simple random walk: with probability p it moves one step to the left, and with probability $q = 1 - p$ it moves one step to the right. (b) A crossing event is illustrated: at time t, the searcher is at node i, the target is at node $i + 1$, and they move toward each other by taking the same edge in opposite directions.

we analytically solved for the optimal strategy that maximizes the capture probability subject to energy constraints. Here, the searcher incurs different energy costs for different actions. In [1] we studied deterministic strategies for the case where all actions have the same cost. We formulated the problem of computing the optimal search strategy as finding the optimal solution of a Partially Observable Markov Decision Process (POMDP) [15]. Since the state space of the proposed POMDP is large, we proposed a method based on binning the belief with non-uniform bins and then approximating the belief in each bin with a uniform distribution. We show that the solutions of this POMDP exhibit an interesting structure.

This paper includes and improves on these results as follows. We study both the crossing and the no-crossing models. For the no-crossing model, we present an analytical solution for finding the optimal search strategy subject to energy and time constraints, where different costs are associated to move and stay actions. For the crossing model, we present the formulation of the problem as a Mixed Observability Markov Decision Process (MOMDP) [12]. We show that the solutions follow the same structure as our approximate POMDP scheme in [1]. We provide a closed form solution for the optimal free parameters in the structure. Finally, we provide preliminary experimental results to show that our models are useful in an environmental monitoring application to search for invasive carp.

The paper is organized as follows. We present an overview of related work in Section II. The search problems we study are formalized in Section III. The capture strategies for the no-crossing model are studied in Section IV, and the crossing model is investigated in Section V. We discuss the applicability of our model in Section VI where we present our experimental results. Finally, we provide concluding remarks in Section VII.

II. Related Work

We already reviewed the one-sided search problem in Section I. In this section, we focus on the literature related to random walks. Random motions, both as discrete random walks and continuous diffusive motions, have been extensively studied as models of unknown animals motions or complex physical processes [16]. In particular they are widely used in the literature to simplify pursuit-evasion games and absorption (or search) processes. A large number of interesting properties closely related to searching missions are collected in [17] including: first passage probability (the probability for the random walker of visiting for the first time a given point at a given time), survival probability (the probability that the random walk has not been found at a given time) and mean capture time (the expected time to be found). Various characteristics of random walks in general graphs have been studied in [18]. Examples are hitting time, which is the expected number of steps before a node is visited, and cover time, which is the expected number of steps to visit every node at least once.

Although one-dimensional random walks might seem too simple, they present several interesting behaviors and properties and are still source of open problems. The survival probability of a particle that performs a random walk on a chain where traps are uniformly distributed with known concentration is studied in [19] and an asymptotically exact solution is provided. In [20], the authors study the survival probability of a prey on a line which is chased by more than one diffusive predator. The same problem but in a semi-infinite line where the boundary represents a haven for the prey is presented in [21]. Contrary to our paper, none of these works have addressed the capture problem restricted to constraints on the energy of the system, or on the maximum time for the chase.

In [23], the authors use random walks to tackle the coalescence problem, where the robots do not have any knowledge about the environment or positions of other robots. Each robot performs an independent random search and when two robots meet, they coalesce into a cluster which then moves as a single random walk. In [24] a group of patrolling robots are uniformly distributed around a circular fence in order to detect an adversary intruder. The robots are performing simple random walks and the goal is to optimize the motion probability that maximizes the detection probability.

It should also be noted that in our model, the searcher does not observe the target. If the target is adversarial and observes the searcher (a common assumption in game theoretic formulations), it is easy to see that capture can not be guaranteed in the crossing model and the entire line must be swept to guarantee capture in the no-crossing model. If the target can not observe the searcher, the resulting pursuit-evasion game is known as the Princess and Monster Game. In this game, the players are in a dark room (i.e. they can not observe each other unless the monster captures the princess.) This game was proposed by Isaacs [25] and solved by Gal [26] who presented a randomized strategy to find the evader in time proportional to the area of the environment. When the game takes place on a graph, it is known as the hunter and rabbit game. Adler et al. showed that the hunter can capture the rabbit in $O(n \log n)$ time in a graph with n vertices [27]. Isler et al. studied this game when the players have local visibility [28]. The pursuit strategy proposed in [29] ensures capturing a target performing a random walk on the plane by a pursuer who can observe it at all times. Finally, in [30] the expected times required for capturing an adversarial robber and a drunk one (performing a random walk) is compared. Upper and lower bounds are derived for the ratio between these two values when the search is done on special graph structures.

III. Problem Statement

In this section, we present the general formulation of the search problem. The environment is a set of discrete locations $\{0, 1, \ldots, N\}$ equally distanced along a line segment. The searcher and the target move in turns, in discrete time steps and with equal speed. The target starts from an unknown node, and afterwards, it performs a *simple random walk* as follows: from location $0 < i < N$, with probability q it moves one unit to the *right*, and with the remaining probability $p = 1 - q$ it moves one unit to the *left*[1]. In addition, the boundary points 0 and N are reflective (see Fig. 1(a)).

The searcher, on the other hand, starts from the left-most node $x(0) = 0$. At each time step, it can decide to move to the right, to the left or stay at its current node. Throughout the paper, we refer to these actions as R, L, S respectively. The searcher's strategy Γ is defined as a sequence of these actions $a \in \{R, L, S\}$. An example strategy can be $\Gamma = (R^i S^j L^l)^*$ which is move to the right for i steps, then stay for j steps, move to the left for l steps, and repeat forever.

We consider two different definitions for capture based on the searcher's sensing capabilities:

- no-crossing: the searcher can sense the target if they occupy the same node at the same time, and also if they cross each other by moving along the same edge in opposite directions. Note that in this case, if the searcher has energy/time sufficient to sweep the entire line, the target will be captured with probability one.
- crossing: the searcher is able to sense the target only on a node. As a result, crossing on edges, by which we mean taking the same edge in opposite directions, will not lead to capture (see Fig. 1(b)).

As we will see later, the searcher's belief about the target's location exhibit very different behavior for these two models.

The searcher starts with an initial energy budget E_0. For each action a, the searcher pays a cost which represents the energy required to perform the action. Let us denote the total energy change associated with action a by $cost(a)$. For the no-crossing model (Section IV) we allow different cost values for move and stay actions, and for the crossing model (Section V) we consider the special case where all actions have the same cost. In addition to the energy constraint, we consider also a maximum time T to complete the task. The objective of the searcher is to design a capture strategy that maximizes the probability of capture given the initial energy budget E_0, the costs for performing each action, and the maximum search time T:

$$\max_{\Gamma} P_c(\Gamma = a_1 a_2 \cdots a_k) \quad s.t. \tag{1}$$

$$0 \leq E(t) = E_0 + \sum_{i=1}^{k} cost(a_i) \tag{2}$$

$$k \leq T\,, \tag{3}$$

[1] Throughout the paper, we mainly focus on the case of a symmetric random walk, i.e. $q = p = 0.5$, but most of our results can be easily extended to other values of p and q as well as to the case of a non-zero probability of staying at the current node (i.e. when $1 - q - p \neq 0$).

where $\Gamma = a_1 a_2 \cdots a_k$ is the searcher's strategy and $P_c(\Gamma)$ denotes the corresponding probability of capture. In the following we first study the problem of designing Γ for no-crossing model in Section IV and then in Section V we investigate the crossing model.

IV. Search Strategies for the No-Crossing Model

In this section, we focus on the no-crossing case. Disregarding any constraint on time or energy, the most intuitive strategy would be to simply sweep the entire segment without stopping, i.e. the R^N strategy. The corresponding probability of capture would be trivially one and, as we will show, the mean capture time is $O(\frac{N}{2})$. However, the optimal strategy is not trivial when either the limited energy budget or restricted search time do not allow a complete sweep of the line segment. In fact, we show that by optimally combining the number of move and stay actions, the searcher can exploit the diffusive properties of the target and significantly increase the probability of capture. In what follows, we provide an analytical solution for this problem.

Let us start our analysis considering the energy constraint. In our model the searcher spends a certain amount of energy c_m to move, and a lower energy c_s to maintain its position, i.e. $cost(L) = cost(R) = -c_m$ and $cost(S) = -c_s$ on all nodes along the line, with $c_m \geq c_s > 0$. The problem we wish to solve is: what fraction of energy should be employed in waiting for the target to hit the searcher itself by staying in the same node. The second non-trivial challenge here is how to distribute these wait actions during the mission. We address the second challenge using the following proposition:

Proposition IV.1. *Let Γ_1, Γ_2 be two different searcher strategies and $x_1(t), x_2(t)$ the location of the searcher at time t when executing Γ_1 and Γ_2 respectively. Then:*

$$x_1(t) \geq x_2(t)\ \forall t \quad \Rightarrow \quad P_c(\Gamma_1) \geq P_c(\Gamma_2)\,,$$

where $P_c(\Gamma_i)$ is the capture probability executing the strategy Γ_i.

This proposition can be justified observing that the searcher captures the target if they cross each other. Therefore, any target captured by the strategy Γ_2 is captured also by the strategy Γ_1 with probability one.

Proposition IV.1 has two immediate consequences: 1) the left action is never useful and 2) given a set of N_R movements and N_S stay actions, the best strategy is to move N_R steps to the right and then spend N_S time steps waiting at $x = N_R$. In other words, given N_R and N_S the sequence of actions is fixed, i.e. $R^{N_R} S^{N_S}$. Therefore, the energy constraint (2) can be expressed as a function of the total number of movements N_R and stationary keeping actions N_S:

$$c_m N_R + c_s N_S = E_0\,. \tag{4}$$

Once that the structure of the strategy is fixed, the only optimization variable is the final searcher position N_R, since we can express N_S in terms of N_R and E_0 using (4). The next step is to obtain an analytical expression of the capture probability P_c as a function of N_R. The capture probability can be expressed as:

$$P_c = P_m + (1 - P_m)P_s\,, \tag{5}$$

where P_m is the probability that the target is captured while the searcher is moving to the right (at $t \le N_R$), and P_s is the capture probability as the searcher is waiting at the node N_R for the remaining N_S time steps. We analyze these two components separately: waiting at a given node and moving in one direction.

Static Searcher: We now focus on computing the capture probability while the searcher is staying at a given node (P_s). Instead of directly computing P_s, we consider the survival probability $\tilde{P}_s$ of a random walker moving in a bounded environment with an absorbing boundary at $x = 0$. The probability P_s is simply $P_s = 1 - \tilde{P}_s$. Even though the exact analytical expression is very complicated (see [17] for the result in a circular environment), it can be shown that the leading term in $\tilde{P}_s(t)$ decays exponentially in time, i.e. $\tilde{P}_s(t) = e^{-t/\tau}$. Moreover, the characteristic time of decay τ can be approximated with the expected capture time t_n, which is defined as the expected time for the random walk to reach $x = 0$ starting from the initial position $x = n$ ([17], Chapter 2). This quantity can be expressed recursively as follows:

$$t_n = p(t_{n-1}+1) + s(t_n+1) + q(t_{n+1}+1)\,,\ n = 1, ..., N-1\,,$$

where the time-step is one and $s = 1 - p - q$. The boundary conditions are $t_0 = 0$, and $\frac{dt_n}{dn}|_{n=N} = 0$ which correspond to absorption at $x = 0$ and reflection at $x = N$ respectively. For a symmetric random walk the previous equation becomes:

$$t_n = p(t_{n-1} + 1) + (1 - 2p)(t_n + 1) + p(t_{n+1} + 1)\,. \tag{6}$$

The solution of the previous recursive relation is:

$$t_n = A + Bn - \frac{1}{2p}n^2\,,$$

where n is the initial target position and A, B are constants imposed by the boundary conditions. In our case the solution becomes:

$$t_n = \frac{n}{2p}\left(2N - n\right)\,.$$

Since we do not know the initial target position, we compute the expected value $t_{\langle n\rangle}$ assuming a uniform probability distribution for the initial location of the target. The result is:

$$t_{\langle n\rangle} = \frac{N^2}{2p} - \frac{1}{2p}\frac{2N^2 + 3N + 1}{6} = \frac{4N^2 - 3N - 1}{12p}\,.$$

Substituting $p = \frac{1}{2}$, the leading term of the capture probability becomes

$$P_s(t) = 1 - \tilde{P}_s(t) \approx e^{-\frac{6t}{4N^2 - 3N - 1}}\,. \tag{7}$$

The approximation in (7) is compared with the average capture probabilities obtained from simulation[2] in Fig. 2(a). As shown in this figure, our expression closely approximates the capture probability P_s.

Moving Searcher: We now present our approximation for the probability of capturing the target while the searcher is moving to the right, i.e. P_m. In this case, the survival probability of the random walk strongly depends on the searcher's motion. Let $X_p(t) = t^\alpha$ be the searcher's equation of motion, whenever $\alpha > 1/2$ the target motion can be considered subdominant. In our case, $\alpha = 1$ and the target is almost static from the searcher point of view. A first consequence of this phenomenon is that the mean capture time is linear in the initial target position, instead of quadratic as in the case of a stationary searcher. In particular, similar to (6), it can be shown that:

$$t_n = p(t_{n-2} + 1) + (1 - 2p)(t_{n-1} + 1) + p(t_n + 1)\,,$$

which has the solution:

$$t_n = A + B\left(\frac{p}{p-1}\right)^n + n\,.$$

Therefore, the expected value $t_{\langle n\rangle}$ is linear in N. Another consequence is that the capture probability can be approximated with the probability that at time $t = 0$ the target is within the region that will be swept. As a result, we can approximate P_m by:

$$P_m \approx \frac{N_R}{N}\,. \tag{8}$$

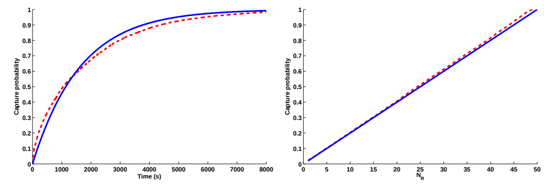
The comparison between the approximation in (8) and the capture probabilities obtained from simulation is shown in Fig. 2(b).

Now that we have approximations for P_s and P_m, we are ready to find the value of N_R that maximizes the total capture probability P_c (5). Using (7) and (8), we have:

$$\begin{aligned} P_c &\approx \frac{N_R}{N} + \left(1 - \frac{N_R}{N}\right)\left(1 - e^{\frac{N_S}{t_{\langle n\rangle}}}\right) \\ &= \frac{N_R}{N} + \left(1 - \frac{N_R}{N}\right)\left(1 - e^{-\frac{6(E_0 - c_m N_R)}{c_s\left[4(N-N_R)^2 - 3(N-N_R) - 1\right]}}\right) \end{aligned} \tag{9}$$

Note that $N_R \le N_R^{max} = \frac{E_0}{c_m}$. Also, P_c is continuous in the closed and bounded interval $[0, N_R^{max}]$. Therefore, P_c admits a maximum value.

The total capture probability P_c computed by our approximation in (9) is compared with the values obtained from simulation in Fig. 3(a)-3(c). In this scenario there are $N = 50$ nodes on the line, the initial energy budget is fixed to $E_0 = 50$, and the cost for moving is $c_m = 2$. Three different values of the cost c_s are considered. We see that with these values the solution of the optimization problem is not trivial and the best final searcher position $N_R{}^*$ is $0 < N_R{}^* < N_R^{max}$.



(a) (b)

Fig. 2. Comparison between the capture probabilities obtained from simulation (red dashed line) and our analytical approximation (blue line), for a segment composed of N=50 nodes. (a) for P_s using (7), and (b) for P_m using (8).

[2] All the results obtained by simulations are averaged on 10^4 trials.

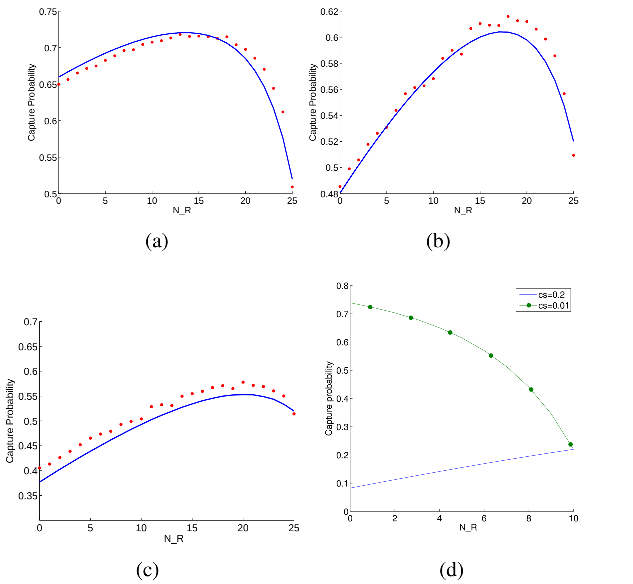


Fig. 3. Comparison between the approximations of the total capture probabilities (blue line) with the behaviors obtained in simulation (red dots) in function of the swept region N_R. The values considered are $N = 50$, $E_0 = 50, c_m = 2$ and $c_s = 0.03, 0.05, 0.07$ for (a), (b) and (c) respectively. (d) Behavior of the capture probability for the two limit cases: $c_s \to 0$ (in green, with dots) and $c_s \sim c_m$ (in blue line). The other parameters are fixed to: $N = 50$, $E_0 = 20$, $c_m = 2$.

Fig. 3(d), shows the optimal strategy obtained from (9) for the two limit cases. In the first case, moving is too expensive, i.e. $\frac{c_s}{c_m} \to 0$. Here, the best strategy is simply wait in the initial position, so ${N_R}^* = 0$. In the second case, moving and station keeping have the same cost, i.e. $c_s \sim c_m$. The best choice for this case is to keep moving without any stops, which corresponds to ${N_R}^* = N_R^{max}$. These two cases are in agreement with our intuition about the best strategy.

Time Constraint: So far, we studied the problem considering only a constraint on the total available energy. Let us now assume to have also the constraint (3) on the time to terminate the mission. Therefore, in addition to (4), the following constraint has to be satisfied:

$$N_R + N_S \leq T\,. \tag{10}$$

To find the optimal value of N_R under this new constraint we consider three different cases (recall that $c_m \geq c_s > 0$):

- $T > N_S^{max} = \frac{E_0}{c_s}$: The search time T is greater than the maximum duration of the task allowed by the energy constraint. In other words the time constraint (10) is not active and the solution is the same as the one presented for the energy constraint.
- $T < N_R^{max} = \frac{E_0}{c_m}$: The available search time T does not allow the searcher to consume all the energy budget. Therefore, waiting at a node will hurt the capture probability (this is a consequence of Proposition IV.1). Thus, the solution here is spending the search time only for moving to the right without any stop actions.
- $N_R^{max} < T < N_S^{max}$: This is the most interesting case and we show that this case can be translated to a constraint on N_R. Indeed, there exists a $\hat{N_R} < N_R^{max}$ such that

$$\begin{cases} \hat{N_R} + N_S = T \\ c_m \hat{N_R} + c_s N_S = E_0 \end{cases} \quad \Rightarrow \quad \hat{N_R} = \frac{E_0 - c_s T}{c_m - c_s}\,.$$

Therefore, for all values of N_R that are less than $\hat{N_R}$ the searcher will not be able to use all the available energy. In other words, each time that N_R is decreased by one unit, N_S can be increased only by one unit because of the time constraint. Thus, for $N_R < \hat{N_R}$ the actual cost of staying becomes equal to the cost of moving. However, we know that in this case employing energy to wait will decrease the capture probability (Proposition IV.1). As a result, we can state that:

$$P(N_R) < P(\hat{N_R})\,, \quad \forall N_R < \hat{N_R}\,.$$

In terms of the optimization problem, the previous result simply restricts the search domain from $[0, N_R^{max}]$ to $[\hat{N_R}, N_R^{max}]$.

V. Search Strategies for the Crossing Model

In this section, we study the crossing case, i.e. when capture is possible only if the searcher and the target are at the same node at the same time. Traveling along the same edge as the target does not count as capture in this model. As a result, at any time the target can be on both sides of the searcher. This makes the problem of analyzing the capture probability challenging even when the search strategy is given. We start by presenting a method for computing the capture probability of a given strategy in Section V-A. The calculation is later used to compare the performance of the proposed search strategies. To find the optimal search strategy, we formulate the problem as a POMDP in Section V-B. As it is well-known, the POMDP can be converted to a MDP by considering the searcher's belief, i.e. the probability distribution that the target is on each node, as the states. The resulting state space is large: it is exponential in the number of nodes. To deal with curse of dimensionality, we present two approximation methods. We refer to the first method as the belief-binning method. In this method, the intuition is that the shape of the belief at farther points is less important for the searcher in picking the best action at its current position. This is because by the time that the searcher reaches those points, the shape of the belief will change. In the second method, the problem is formulated as a Mixed Observability MDP (MOMDP) [12] where the fully observable variables of the state are separated from the partially observable ones. In both methods, the planning is done in a lower dimensional space. The two methods approach the problem differently and provide approximate solutions. However, both of them reveal a similar structure in their solutions (Section V-C). We analyze the performance of the strategies in this particular structure in closed form (Theorem V.1). We present the details of this analysis in the Appendix. We use this analysis to find the best set of parameters within the structure, and propose our final solution which is given in Table III.

Throughout this section, we focus on the special case in which the cost of station-keeping is equal to the cost of moving, allowing further reduction of the state space by avoiding energy level discretization. Essentially, both moving and station-keeping actions require a unitary cost both in time and energy, and so the initial energy budget E_0 translates into a time budget T. In summary, instead of the optimization problem (1)-(3), in this section, our goal is designing a capture strategy of length T that maximizes the probability of capture:

$$\max_{S} P_c(\Gamma = a_1 a_2 \cdots a_k) \quad s.t. \quad k = T\,. \tag{11}$$

A. Capture Probability of a Given Strategy

To compare different search strategies given the crossing model, we first show how the probability of capture for a given strategy can be computed. Let us denote the searcher's location at time t by $s(t)$ and the target's location by $e(t)$. The target is initially at $e(0)$. The searcher is performing the strategy Γ given as a sequence of actions R, L and S. Fig. 4 shows an illustrative example of the location of the searcher for a specific strategy, and the target for a specific random walk path as a function of time.

We are interested in computing the probability of capturing the target at time t denoted by $P_c(t)$. The capture events are those that the searcher and the target are at the same position at the same time. Thus, we must consider the target's paths that end up at $s(t)$ at time t, i.e. $e(t) = s(t)$. Counting the events that the target starts at $e(0)$ and reaches $s(t)$ at t is not too difficult. However, we cannot simply sum up the probability of these events to obtain P_c because they include overlapping capture events with $e(k) = s(k)$ for multiple values of $k \leq t$. First, we must only count the events such that the target has not been captured sooner than t. Second, since crossing is allowed, path interactions such as the one marked by the arrow in Fig. 4 do not count as capture. In order to tackle these two challenges, we look at the searcher's path $s(t)$ as a piecewise constant function. That is, $s(t)$ is composed of a set of time intervals $[t_k, t_{k+1})$ such that $s(t)$ is constant in $[t_k, t_{k+1})$ and has the value s_k (the searcher is staying at s_k in $[t_k, t_{k+1})$). In Fig. 4 the searcher is at s_{k-1} in the time interval $[t_{k-1}, t_k)$.

In order to compute $P_c(t)$, we take a divide and conquer approach to recursively compute the target paths that yield capture at time t but are *safe* before t. By *safe* we mean that the target has not been captured before time t. To do so, we consider the intervals before t. Each time-interval acts as the basis in our divide and conquer method, and we can employ them to overcome the two issues mentioned above as follows. Since in each interval $s(t)$ is constant we do not have the crossing events such as the one marked by the arrow in Fig. 4. We also can enumerate the target paths that co-locate the target and the searcher for the *first* time at t by counting the target paths that are completely to the left or to the right of s_i during $[t_i, t_{i+1})$ for $t_i < t$. This enables us to compute the capture events using recursive equations as follows. Let us introduce the following probabilities:

- $P_{safe}(x, t)$: the probability that the target *safely* arrives at location x at time t. Here *safe* refers to the fact that the target has not been captured before time t.
- $P_{safe}(x_1, t_k, x_2, t_{k+1} - 1)$: assuming that the target is at x_1 at time t_k, $P_{safe}(x_1, t_k, x_2, t_{k+1} - 1)$ is the probability that it *safely* reaches x_2 at time $t_{k+1} - 1$. Notice that the target has to stay either to the left, or to the right of s_k during $[t_k, t_{k+1})$ in order to avoid capture and remain safe. Referring to Fig. 4, the target paths must be either below or above s_k at $[t_k, t_{k+1})$. Note that this function is different from the previous one in the sense that it counts the safe events in a single interval $[t_k, t_{k+1})$.
- $F(x_1, x_2, t)$: the probability that for the first time the target reaches at x_2 after t time steps starting from x_1 (first passage probability).
- $G(x_1, x_2, t)$: the probability that the target starts at location x_1 and reaches at x_2 after t time steps (not necessarily for the first time).

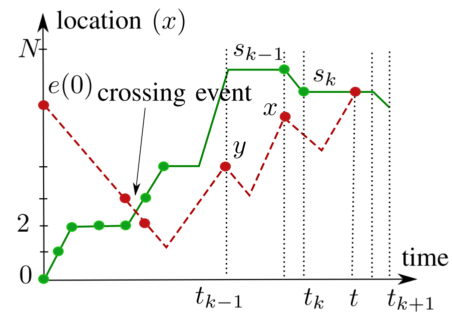


Fig. 4. The position of the players as a function of time. The target's path is shown in dashed lines. The time marked by the arrow is not capture because crossing is allowed.

Our goal is to compute $P_c(t)$, the probability of capturing the target at time t. First, we consider the time interval $[t_k, t_{k+1})$ that t belongs to. See Fig. 4. Let x be the target's position at time $t_k - 1$. The searcher is at s_{k-1} at $[t_{k-1}, t_k)$. The capture events can be described as follows: The target safely arrives at x at time $t_k - 1$, i.e. $P_{safe}(x, t_k - 1)$ and then, from x it reaches s_k for the first time after $t - t_k + 1$ time steps:

$$P_c(t) = \sum_{x} P_{safe}(x, t_k - 1) F(x, s_k, t - t_k + 1)\,.$$

The safe events $P_{safe}(x, t_k - 1)$ can be obtained from the following recursive equations:

$$P_{safe}(x, t_k - 1) = \sum_{y} P_{safe}(y, t_{k-1}, x, t_k - 1) P_{safe}(y, t_{k-1})$$

The probability function $P_{safe}(x_1, t_{k-1}, x_2, t_k - 1)$ is computed as follows:

$$P_{safe}(x_1, t_{k-1}, x_2, t_k - 1) = G(x_1, x_2, t_k - 1 - t_{k-1})$$
$$- \sum_{t=0}^{t_k - 1 - t_{k-1}} \left(F(x_1, s_{k-1}, t) G(s_{k-1}, x_2, t_k - 1 - t_{k-1} - t) \right)$$

In other words, the events that the target crosses s_{k-1} (second term in r.h.s. of the equation above) are excluded from the total number of events (first term in r.h.s. of the equation above). Notice that the searcher is at s_{k-1} in the time interval $[t_{k-1}, t_k)$.

Finally, computing $G(x_1, x_2, t)$ and $F(x_1, x_2, t)$ is straightforward and we refer the interested reader to [17] for the corresponding equations.

B. Partially Observable Markov Decision Process

In this section we present the formulation of the problem in (11) as a Markov Decision Process (MDP) [31]. We start by a brief overview of MDPs. An MDP is described by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R})$ where $\mathcal{S}$ is the set of possible states, $\mathcal{A}$ is the set of actions, $\mathcal{T}$ is the probability of transitioning between the states as a result of performing each action, and $\mathcal{R}$ is the reward collected for each transition. Here $\mathcal{T}$ and $\mathcal{R}$ are represented as matrices.

The states of the MDP could be defined as (s, e) where s is the position of the searcher and e is the position of the target. However, we cannot use this setup for the states of the MDP since the location of the target is not observable to the searcher. The searcher's only observation is that it has not captured the target yet. Therefore, our problem is in fact a Partially Observable Markov Decision Processes (POMDP). However, we can convert it to an MDP by defining the states as the searcher's belief about the target's location [15]. The goal is to find the policy that maximizes the reward collected by the searcher upon execution of the policy. The following are the sets which define our MDP (Fig. 5):

- The states are defined as (B, E, s) where B is the belief of the searcher about the position of the target, E is the current energy of the searcher and s is the current position of the searcher. Here B is represented as the probability vector $B = [p_0, p_1, ..., p_N]$ where p_i is the probability that the target is at position i given that it is not captured yet. A set of terminal states is given by $\{s_{capture}, s_{no-energy}\}$ where $s_{capture}$ denotes the capture state, and $s_{no-energy}$ denotes the state in which the robot runs out of energy.
- The set of actions that the searcher can perform in each state are: *stay* at its current position, move one step to the *left*, or one step to the *right*.
- The transition probability matrix with entries $P(s_i, s_j, a)$ that represent the probability that the searcher transitions to state s_j by performing action a in state s_i.
- The reward matrix which represents the transition reward from state s_i to state s_j after performing action a.

In the following, we describe the details of the state space and the proper definition of the reward function. We leave the calculation of the probability transition matrix since it is straightforward.

Method 1: The belief-binning approximation method Initially, the searcher starts from the location $x = 0$. The searcher begins its mission with no information about the position of the target (except that the target is not captured yet). Therefore, the initial belief vector is the uniform probability distribution over $[0, N]$.

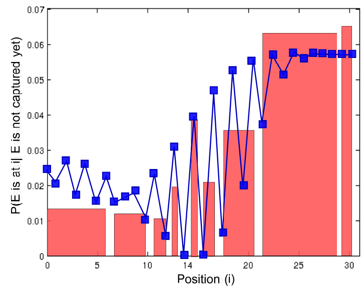


Fig. 6. Blue is the actual belief and red is its approximation by bins. The searcher is at $x = 14$.

Note that the number of states will be exponential in the number of discrete levels used for representing the probability vector $B = [p_0, p_1, ..., p_N]$ (with the exponent equal to N, the size of the environment). To deal with this problem, one method is to approximate the belief by a specific function which can be represented by a small number of parameters. However, this approach cannot be applied to the crossing case directly, because here the belief is not a smooth function. A sample of the searcher's belief is shown in Fig. 6.

Intuitively, the value of the belief at the nodes that are far away from the searcher's current position is not effecting the best action. Based on this observation, we represent the belief by bins with exponentially increasing width as follows. There are two bins with width 2^i that start at nodes $s+2^i$ and $s-2^i$ respectively $(0 \leq i \leq \log(N))$. The approximate belief in each bin is uniform. To compute its value we first compute the cumulative belief in each bin and then we take the average of this cumulative value in the corresponding bin. Finally, we assign the closest discretization level to the average value as the bin value. An example is shown in Fig. 6.

The reward matrix: As expressed in (1), we are looking for the strategy that maximizes the probability of capture. In order to associate the value of MDP states with the probability of capture, the reward function is defined as follows. The transition reward from all states (except $s_{capture}$ and $s_{no-energy}$) to the capture state $s_{capture}$ is one. All other transition rewards are zero. The state values of the aforementioned MDP is an approximation of the probability of capture. Therefore, the strategy that maximizes the state values, i.e. the solution to the MDP, is in fact the one that maximizes the probability of capturing the target.

The solution technique: Finally, we use finite-horizon MDP implementation available in INRA MDP MATLAB Toolbox [32] which uses backward induction algorithm. The number of stages is set to T, the time constraint. The terminal reward is set to zero for all states except $s_{capture}$ which is set to one.

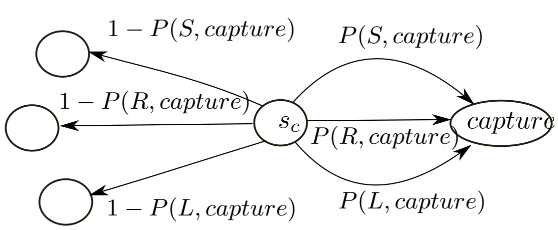


Fig. 5. MDP state transitions. In the current state s_c, by performing $a \in \{R, L, S\}$, with probability $P(a, capture)$ the searcher captures the target and with the remaining probability the next state will be s_n.

Method 2: The MOMDP approximation method An alternative approach to tackle the large state space of our problem is to formulate the problem as a Mixed Observability MDP [12]. In this formulation, the state components that are

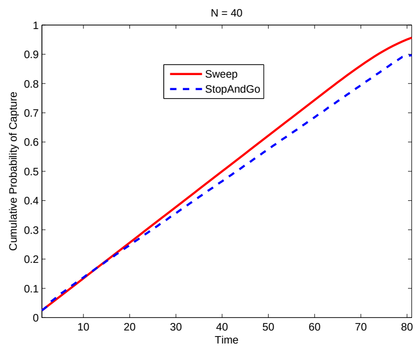


Fig. 7. Cumulative probability of capture obtained from Section V-A. Here $N = 40$.

fully observable are separated from the ones that are partially observable. As a result, the belief is maintained on a smaller set of variables, and the size of the state space can be reduced significantly.

More specifically, a MOMDP is specified as a tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{O}, \mathcal{T}_x, \mathcal{T}_y, \mathcal{Z}, \mathcal{R})$ where $\mathcal{X}$ represents the set of fully observable components, $\mathcal{Y}$ represents the set of partially observable components, and $\mathcal{A}, \mathcal{O}$ are the set of actions and observations respectively. The function $\mathcal{Z}(o, s, a)$, similar to POMDPs, gives the conditional probability of observing o after performing action o and moving to the state s. The function $\mathcal{T}_x(x, y, a, x')$ represents the probability that after taking action a in state (x, y) the fully observable state component x makes a transition to the new value x'. Similarly, the function $\mathcal{T}_y(x, y, a, y')$ gives the probability that the partially observable component has the new value y'. The belief is then represented as (x, b_y) where b_y is the belief defined only for the y component.

Adopting the MOMDP formulation to our problem, the searcher's position and also the time budget are fully observable while the target's location is partially observable. We use the Approximate POMDP Planning (APPL) toolkit which is available at [33]. The APPL toolkit combines MOMDP with SARSOP which is a point-based POMDP algorithm [12]. In this approach a set of points are sampled from the belief space and these samples are used as an approximate representation of belief. Exploiting these samples, a belief tree is maintained with the initial belief as the root node, and the subtrees that will never be visited by the optimal policy are pruned out. In order to estimate the optimal value function V^* a piecewise linear lower bound $\underline{V}$ and also an upper bound $\overline{V}$ are maintained. To improve these bounds, the algorithm applies Bellman backup operation until convergence is achieved.

C. Simulation Results

We are now ready to solve for the search strategies using the proposed formulations. We consider three cases: 1) T is enough for at least two sweeps of the line, i.e. $T > 2N$; 2) $T \leq 1.5N$; 3) $1.5N < T \leq 2N$. In the first case, we present intuitive strategies and show that they guarantee a high probability of capture. In the remaining cases, we provide the strategies obtained from the MDP formulations. The simulations for the MDP formulations are done on a Dell Poweredge 6950 machine with $14GB$ memory. In the belief-binning approach, we used 50 levels for discretizing each bin. Thus, if there are n bins, the number of possible states would be 50^n. In the MOMDP method, we let the solver run until a target precision[3] less than 0.001 is reached or a timeout happens after 3 hours of execution.

When Time is Enough for Two Sweeps
In this case, the searcher has enough time for at least two sweeps $T > 2N$. We start by considering the case that $T = 2N + 1$. One intuitive strategy is to sweep the whole line twice. However, if the searcher moves all the time, the parity of its distance to the target will never change, unless the target does a stay action at one of the boundary points (Fig. 1(a)). This is because, at other points, the target always moves one step to the right or to the left, and thus its distance to the searcher changes by two. Therefore, we add a wait step in order to increase the probability of capture in the event that the target starts from an odd node. We call this strategy the *Sweep* strategy: the searcher moves all the way to the right, waits for one step at the last point N and then moves back to the left toward the first node $(R^N SL^N ...)$.

We also analyze a second strategy, which we call *StopAndGo*: the searcher moves for one step, then waits for one step and so on $(RSRS...)$.

The probability of capture for these two strategies is computed using the analysis in Section V-A. Fig. 7 depicts the cumulative probability of capture for $N = 40$. As shown in this figure, these strategies achieve a very high probability of capture: 0.95 and 0.90 for Sweep and StopAndGo respectively. Since the highest possible performance for the probability of capture is one, we choose the Sweep strategy as our best strategy for this case. Notice that for larger values of $T > 2N$ we can repeat the proposed strategy until time exceeds T.

When Time is less than 1.5 sweeps
In this case $T \leq 1.5N$. The strategies found by the two approximation methods for $N = 20$ and $N = 31$ are shown in Table I and Table II respectively.

A first interesting result is that, as shown in these tables, the solutions have a common property: the stay actions are uniformly distributed among the right actions. In other words, the strategies are of the form $(R^k S)^m$. Let us refer to this class of strategies as the uniform strategies. Table I and Table II also present the best uniform strategy for the same values of N and T which are obtained by changing the number of rights k in each group of $(R^k S)$ and computing the probability of capture using the analysis in Section V-A. Observe that the uniform strategy is very close to the solutions found by the two MDP methods.

To compare the belief-binning and the MOMDP performance, observe that for $N = 20$ in Table I the solutions of belief-binning method are all better than the MOMDP in terms of capture probability. This is true also in Table II for for $N = 31$ and $T = 14, 24$. However, for $T = 35, 44$ the MOMDP solution outperforms the belief-binning strategies which is due to the error corresponding to the resolution in

[3]Target precision is a function of $|\overline{V} - \underline{V}|$ for the set of samples in the SARSOP method.

	$T = 9$	$T = 14$	$T = 19$	$T = 24$	$T = 29$
Belief-binning strategy	RSR^2SR^3S	$(RS)^2R^3S(R^2S)^2$	$(R^3S)^4R^2S$	R^2SR^3S $(R^2S)^3(R^3S)^2$	$R^2SR^3S(R^2S)^3$ $(RS)^2R^3SR^2SRS$
Belief-binning capture probability	0.2966	0.4201	0.5711	0.6991	0.8020
Uniform strategy	$(R^3S)^2R$	$(R^3S)^3R^2$	$(R^5S)^3R$	$(R^3S)^6$	$(R^2S)^9S^2$
Uniform capture probability	0.2971	0.4294	0.5642	0.7095	0.8031
MOMDP strategy	$(R^2S)^2RSL$	$R^2SR^3S(R^2S)^2L$	$R^2S(R^3S)^3R^2SL$	$R^2S(R^3S)^4R^2S^2L$	$(R^2S)^6(R^3S)^2S^2L$
MOMDP capture probability	0.2179	0.3787	0.5323	0.6633	0.7873

TABLE I
THE MDP SOLUTIONS FOR $N = 20$.

	$T = 14$	$T = 24$	$T = 35$	$T = 44$
Belief-binning strategy	$(R^2S)^3R^4S$	$R^3SR^5SR^3SR^6SR^2S$	$R^{30}S^5$	$RS^3R^{19}SR^3SR^2S(RS)^2$
Belief-binning capture probability	0.2830	0.4625	0.58	0.675
Uniform strategy	$(R^3S)^3R^2$	$(R^3S)^6$	$(R^4S)^7$	$(R^2S)^{14}R^2$
Uniform capture probability	0.2818	0.4656	0.6638	0.787
MOMDP strategy	$R^2SR^3S(R^2S)^2L$	$R^2S(R^3S)^4R^2S^2L$	$R^2S(R^3S)^2(R^4S)^3R^5S^3L$	$RS(R^2S)^{10}R^6SR^2S^2L$
MOMDP capture probability	0.2421	0.4219	0.6333	0.7651

TABLE II
THE MDP SOLUTIONS FOR $N = 31$.

binning levels. On the other hand, the MOMDP solutions exhibit the uniform structure in all instances which makes MOMDP a more suitable approach for larger values of N and T[4].

Next, we focus on the problem of finding a good uniform strategy. In fact, we would like to find the correct number of *right* actions before each *stay* action and optimize k. In order to find the best k, we take the following approach. We first derive closed form equations to approximate the capture probability of $(R^kS)^m$. Our approximation is applicable for $k \geq 3$. Therefore, we use it to find the best $k \geq 3$. We then compare this best solution with $k \in \{1, 2\}$ both in simulations and also using our analysis (Section V-A). From this comparison, we conclude our proposed optimal strategy of the form $(R^kS)^m$.

The following theorem, presents the aforementioned closed form. We present the details of the proof in the appendix.

Theorem V.1. *The probability of capture for the strategy* $(R^kS)^M$ *where* $T = M(k+1)$ *and* $3 \leq k$ *can be approximated as:*

$$P_c \approx \begin{cases} \frac{M(k+2)}{2N} = \frac{T+M}{2N} & \text{if } Mk < N \\ \frac{T+M-1}{2N} & \text{if } Mk = N \end{cases} \quad (12)$$

Therefore, for a given T the capture probability is maximum when M takes its largest possible value. That is the optimum number of rights in each group of R^kS is $k = 3$ when $k \geq 3$.

Next, let us compare the performance of $k = 3$ with $k \in \{1, 2\}$. Fig. 8 shows this comparison for the following strategies: $(RS)^{35}$ and $N = 35$, $(R^2S)^{17}$ and $N = 34$, $(R^3S)^{12}$. This comparison suggests that the performance of $(R^2S)^m$ is comparable to the other uniform strategies and sometimes it is the best. Thus, we pick $(R^2S)^m$ as our proposed strategy when $T < 1.5N$.

When Time is greater than 1.5 sweeps

Let us now investigate the case that $1.5N \leq T < 2N$. We

[4]Note that the APPL toolkit can handle instances where $N < 100$ and $T < 40$.

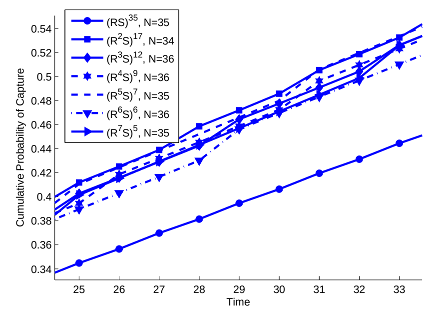


Fig. 8. Comparison of uniform strategies $(R^kS)^m$. Here $N = km$ to prevent extra R at the end of the strategy.

compare four strategies. The first one is the Sweep strategy we introduced earlier. Then, we define the following strategies:

- $(R^2S)^{\frac{N}{2}}(L^2S)^m$: repeat the pattern R^2S until the searcher is at node N, then repeat the pattern L^2S for the rest.
- $(R^3S)^kL^m$: repeat the pattern (R^3S) until the searcher is at node N, then move back to the left.
- `RightLeftRight`: move to the right for $\frac{3N}{4}$, then turn back and move to the left for $\frac{N}{2}$, and then move to the right for the rest of time-steps.

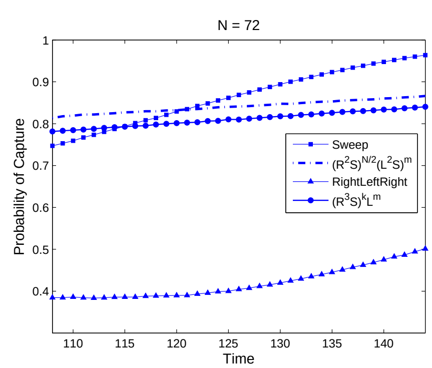


Fig. 9. Comparison of four strategies when $1.5N < T$ for $N = 72$ obtained from simulation.

As shown in Fig. 9 for $N = 72$, the performance of $(R^2S)^{\frac{N}{2}}(L^2S)^m$ is better for $T < 120$, and then afterwards the Sweep strategy becomes better. It is worth emphasizing that both are above 0.81 after time 120. We observed the same behavior for $N = 30$ and $N = 54$. Thus, we declare both of them as our candidate strategies for the case that $1.7N < T$. When $1.5N < T < 1.7N$, we consider $(R^2S)^{\frac{N}{2}}(L^2S)^m$ our best strategy.

A summary of the best strategies found for different cases is depicted in Table III.

	Best Strategy
$2N < T$	Sweep
$1.7N < T \leq 2N$	Sweep
$1.5N < T \leq 1.7N$	$(R^2S)^{\frac{N}{2}}(L^2S)^m$
$T \leq 1.5N$	$(R^2S)^m$

TABLE III
SUMMARY OF THE PROPOSED BEST STRATEGIES FOR DIFFERENT CASES.

VI. EXPERIMENTAL RESULTS

The proposed search strategies can be employed in many robotics applications e.g. border patrolling. Our motivating application is our ongoing project on an environmental monitoring problem: finding radio-tagged common carp. In the absence of any information about the movement of the fish, one can model the fish as an adversary target or a random moving target. We study the former model in [34] while in this paper we consider the latter model, i.e. the random walking motion model. We conduct our experiments in Minnesota lakes instead of rivers for the following reasons. In addition to safety reasons for our autonomous robots, we expect that considering a 1D search path is a reasonable assumption for this application because these fish tend to move near the shore (where there is more vegetation) most of the time [35]. See Fig. 11 for an illustration.

In this section, we present preliminary results which are mainly focused on modeling. First, we provide evidence that taking measurements on a discrete set of nodes produces reliable detection. In particular, as the robot stops and turns off its motor, the noise interference which is a main reason for false positives is reduced. It must be noted that an important feature of our system is the directional sensitivity of the antenna. That is, our sensor must be aligned with the target for maximum signal strength received from the target. As a first approach, we chose to rotate the antenna and take measurements at multiple orientations when the robot is stopped. This simplifies the modeling as we no longer need to model the antenna orientation. We present results from the summer months using Autonomous Surface Vehicles (ASV) and winter months using an Autonomous Ground Vehicle (AGV). In practice, the AGV will be used over frozen lakes, while the ASV is used in the summer. Finally, it must be noted that in the case of ASVs the ratio of movement cost to stationary keeping cost is $c_m/c_s = 5.7/0.2$ while in the case of AGVs the stationary keeping has zero cost.

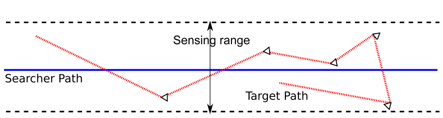


Fig. 11. The target is performing a random walk on a corridor of width equal to the sensing range (red dotted path). The searcher moves on the corresponding line segment (blue path).

A. ASV: System Description

Our first system, shown in Fig. 10(a), consists of two Autonomous Surface Vehicles (ASVs) carrying radio tracking equipment. The ASVs are built on boats manufactured by OceanScience [36] and were originally designed for remote operation. We added autonomous navigation with on-board GPS, digital compass and laptop, wireless communication via ad-hoc networking and remote override capabilities [37].

For our experiments, one of the ASVs was designated as the searcher robot and the other used as the random-walking target. The searcher robot (the red boat in Fig. 10(a)) was equipped with a directional antenna (Fig. 10(b)), real time spectrum analyzer, and a laptop to process and track signals. The target robot (yellow in Fig. 10(a)) had a radio tag attached to a tow line. The radio tags transmit a low-power, uncoded pulse on a unique frequency approximately once per second. The antenna is directionally sensitive. To detect nearby tags, the antenna must be aligned with the tag. Our experiments are performed in Lake Staring, Minnesota.

B. ASV: Experiment Details

In the following experiment, the searcher sweeps an L-shaped corridor of length $320m$ as shown in Fig. 10(c). For safety reasons, we use this short length in order to keep the ASVs within the communication range.

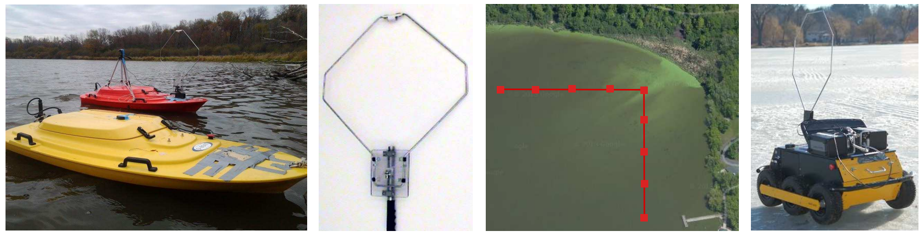
The searcher sweeps the path from the first node to the last node such that after uniform time intervals it turns off its motor and takes measurements by rotating the antenna. This will give us a comparison between the signal strength during motion versus waiting, and ultimately the effect of the noise interference from motor. In the following plots, we use the propeller speed as an indicator of the time intervals that the boat has turned off its motor.

Fourteen radio frequencies (corresponding to known tags which were not included in the trials) were monitored for transmissions, while the target transmitted only at frequency 49611 Mhz. Notice that in a realistic search for a target, the target frequency is unknown. Therefore, we need a comparison between the true and a noise frequency that we know it is not present in the trial.

To determine detection, we use the following method. For each frequency that the antenna is listening to, let m and σ be the corresponding mean and variance for the signal strength respectively. Also, let f be the current signal strength. Consider the following criterion for all frequencies:

$$\frac{f - m}{\sigma} \tag{13}$$

There will be four cases:



(a) (b) (c) (d)

Fig. 10. (a) Two ASVs used in field experiments. The tracking equipment are installed on the boat that acts as the searcher. (b) The directional antenna used for sensing the target. (c) The experiment area which includes 9 nodes with distance $40m$ (total length is $320m$) along an L-shaped path. (d) The Husky A100 as our AGV.

1) One particular frequency has a sharp rise in $\frac{f-m}{\sigma}$ while others do not. We declare detection for this frequency.
2) All frequencies have a sharp rise in $\frac{f-m}{\sigma}$. This case is a false positive because in a realistic situation we cannot differentiate between a possible detection from a nearby tag and the background noise.
3) No frequencies have a rise in $\frac{f-m}{\sigma}$. Then, that is a non-detection.
4) There is no rise in $\frac{f-m}{\sigma}$, but the tag is close-by. Then, this is a false negative. Since the antenna is directional, false negatives are caused by mis-aligned antenna.

Fig. 12 depicts the signal strength for the frequencies associated with target (tag number 49611) and tag number 49124 that is not present in the trial. First, observe that the signal strength for both frequencies drops considerably as the robot stops and turns off its motor (zero speed intervals in Fig. 12).

Second, notice the peak in the strength for 49124 KHz marked by t_6. Observe that according to criterion in (13) we would declare a detection which is clearly wrong as no transmitters for 49124 KHz were present. Also, this makes the target detection at time t_4 an uncertain detection. This instance is in favor of our crossing model. That is, we must turn off the motor on a discrete set of nodes in order to reduce the noise interference from motor and avoid false positives. Similar false positives and negatives were observed to occur on both the ASV and AGV platforms.

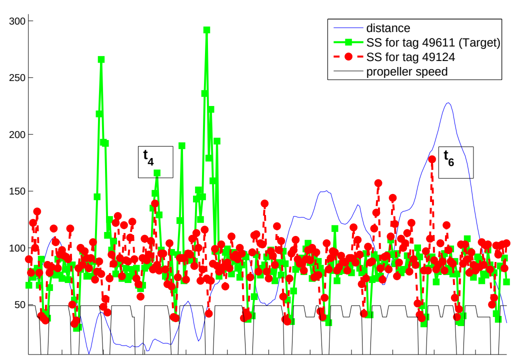


Fig. 12. The Signal Strength (SS) for the target tag (49611 KHz) and 49124 KHz that is not present in the trial. Also, the distance between the searcher and the target are shown. Notice that we have false negative at $[t_2, t_3]$ although the target is close to the searcher (around $15m$). One reason for a false negative could be mis-alignment between the antenna and the tag.

C. AGV: System Description

As our Automatic Ground Vehicle (AGV), we used the Husky A100 built by Clearpath Robotics [38] which is a six wheel, two motor, differential drive machine. Fig. 10(d) shows the Husky equipped with the antenna.

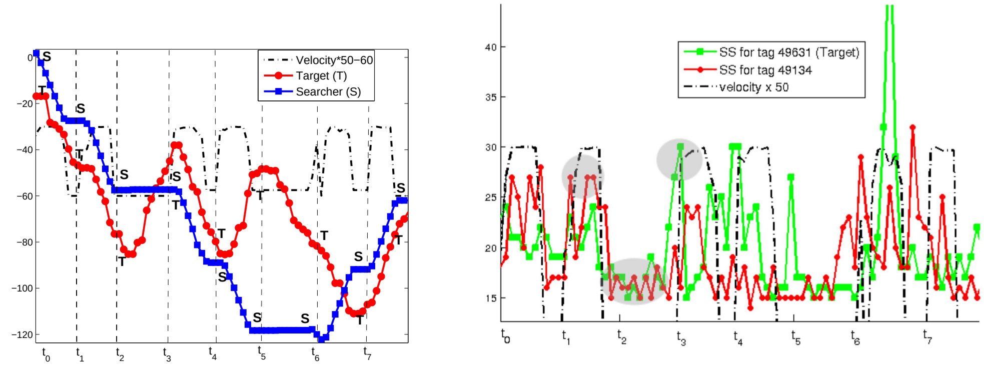
D. AGV: Experiment Details

In this experiment, the searcher and the target move along a corridor of length $120m$ while they have an offset of $20m$.

The target, carries the tag marked by 49631. Similar to the ASV case, the searcher sweeps the corridor from left to right. During this sweep, the searcher stops after uniform time intervals which are indicated by zero speed of the vehicle. Fig. 13(a) illustrates the location of the searcher and the target along the line as well as the waiting intervals of the searcher. Here, the searcher's strategy is an example of our proposed $R^2SR^2S...$ strategy.

Fig. 13(b) depicts the signal strength received by the searcher from the tag as well as the frequency associated with tag number 49134 which is not present in the trial. First, observe the peaks in the signal strength from the target tag. Here, according to the distances in Fig. 13, the peaks at t_3, t_4 and $[t_6, t_7]$ are true detections. Notice that the searcher does not miss the target at the particular crossing event at $[t_6, t_7]$. Also, the detection at t_5, despite the large distance ($73m$), is because of complete alignment between the antenna and the target.

Observe from Fig. 13(b) the two highlighted peaks in the signal strength at $[t_1, t_2]$ and t_3. Notice that the peaks for both frequencies are of similar maximum value and are both higher than their corresponding mean level (which is also highlighted). Therefore, according to our criterion $\frac{f-m}{\sigma}$ (13), we would declare detection for these frequencies at t_3 and $[t_1, t_2]$ respectively. However, the detection for 49134 would be a false detection since no transmissions on this frequency were used in the trial. Notice that this false detection is received while the vehicle is moving and its motors are on. On the other hand, the peak from the target frequency corresponds to a true detection since the searcher and the target are very close (see Fig. 13). Moreover, the measurement is taken while the searcher's motors are off. To summarize, this example supports our discrete crossing model. That is, the searcher has to stop in order to take reliable measurements.



(a) (b)

Fig. 13. (a) The target's and the searcher's location along the corridor versus time. Here, the searcher and the target are denoted by T and S respectively. (b) The Signal Strength (SS) for target tag (49631 KHz) and an example frequency (49134 KHz) that does not correspond to a nearby tag. The mean level (m) for both frequencies is highlighted by the middle ellipse. The highlighted peak for 49134 is a false positive detection of tag 49134.

VII. Conclusion

In this paper, we studied the problem of searching for a one-dimensional random walker on a discrete line segment. The goal is to design search strategies that maximize the probability of capturing the target subject to constraints on energy and time. We considered two different capture models and obtained corresponding optimal and near-optimal strategies. In the no-crossing case, when the target will be captured if it crosses the searcher, we present closed-form solutions for the optimal strategy. In the crossing case, using the structure of the MDP solutions we focus on a set of strategies which we call the *uniform* strategies characterized by groups of right actions interleaved with stay actions, i.e. $(R^kS)^m$. We derive the best strategy in this class and show that $(R^2S)^m$ is performing close to the strategies found by the MDP methods.

Finally we presented our preliminary experiments for application of our results to a practical problem, where the main objective is to find radio-tagged fish in a lake by using an autonomous surface/ground vehicle. After the description of the robotic system and its sensor model, we presented results from field experiments carried out in a Minnesota lake.

One immediate line of research for future work is to design search strategies for two-dimensional environments, where the target performs a random walk on a grid. Here, curse of dimensionality is a bigger challenge than the one-dimensional case. One interesting idea is to restrict the class of search strategies to those that sweep an entire column or row and thus the strategy can be specified as a set of columns and rows and the order to visit them. As a result the problem can be viewed as a one-dimensional search, as studied in this paper, by simply projecting the belief distribution in the grid along a given column (row). In this simplification, the connection between the one-dimensional and the two-dimensional belief distributions, search strategies and performance evaluation must be established.

Another interesting direction is to extend the experimental results for detecting radio-tagged fish. One approach is to define a more accurate model of the system. For example, the antenna might miss a tag (false negative) or it may report a tag mistakenly (false positive). In addition, the antenna is directional and so its detection region is more accurately modeled by an oval and not a circle. Another approach is designing strategies that are more robust to the conditions at the specific lake environment (such as wind) or the uncertainties in the platform (such as the sensor, the signal strength of the radio tags, and the underlying controller of ASVs).

VIII. Acknowledgment

The authors would like to thank Pratap Tokekar and Patrick Plonski for their help with system development and field experiments. This work has been supported by the National Science Foundation grants #1111638 and #0917676.

Appendix

Proof of Theorem V.1. We first show that (12) is valid for $7 \leq k$. For $3 \leq k \leq 7$, we compare the capture probability obtained from (12) with the one obtained from simulations. Since the approximation error is small, we conclude that (12) is valid for $3 \leq k$.

Our proof has two parts. First, we prove a structure on the belief. Second, using this structure we compute the capture probability.

Structure on the belief: Let s denote the current location of the searcher. By taking an inductive approach we show that after performing $(R^kS)^m$, the belief has the form:

$$Bel(s-2:N) = [p, 0, 0, p_0, 2p_0, 2p_0, ..., 2p_0] \tag{14}$$

Equation (14) can be derived in two steps. First, it can be shown that the belief after $(R^kS)^jR^i$ $(1 \leq i \leq k)$ is of the following form:

$$[\frac{p}{2^i(1-ip_0)}, \frac{2^{i+1}-2-i}{2^i}p_i, 0, \frac{2^{i+1}-1}{2^i}p_i, 2p_i, 2p_i, ..., 2p_i] \tag{15}$$

where $p_i = \frac{p_0}{1-i.p_0}$. For $6 \leq i$ the belief function in (15) converges to:

$$Bel(s-2:N) = [0, 2p_i, 0, 2p_i, 2p_i, 2p_i, ..., 2p_i]$$

Next, consider the last action in the sequence $(R^kS)^m$, which is the stay action. The belief function after this stay action will be: $Bel(s-2:N) = [p', 0, 0, p'_0, 2p'_0, 2p'_0, ..., 2p'_0]$ where:

$$p'_0 = \frac{p_k}{1-2p_k} = \frac{p_0}{1-(k+2)p_0} \tag{16}$$

Therefore, the belief function after performing $(R^kS)^{m+1}$ has the same form as in (14). Finally, for the base case of our inductive argument, it can be shown that after R^kS and for $k \geq 4$ we have:

$$Bel(s-2:N) \approx [b, 0, 0, b_{k+1}, 2b_{k+1}, 2b_{k+1}, ..., 2b_{k+1}]$$

which has the same format as in (14). Note that here:

$$b_{k+1} = \frac{b_0}{1-(k+1)b_0} \tag{17}$$

Capture probability: Next, let $P_c(i,m)$ denote the probability of capture after performing the i^{th} action in the m^{th} group of (R^kS) given that there was no capture beforehand. Also, let $\bar{P}_m$ denote the probability of *not* capturing the target after performing $(R^kS)^m$. From (14) and (15), it is easy to see that the capture probability is p_{i-1} for performing the i^{th} right action, and $2p_k$ for performing the stay action. Therefore:

$$P_c(i, m+1) = \bar{P}_m \times \begin{cases} \prod_{j=0}^{i-2}(1-p_j)p_{i-1}, & \text{if } i \leq k \\ \prod_{j=0}^{k-1}(1-p_j)2p_k, & \text{if } i = k+1 \end{cases}$$

Since $\prod_{j=0}^{i}(1-p_j) = 1-(i+1)p_0$ we get:

$$P_c(i, m+1) = \bar{P}_m \times \begin{cases} p_0^m, & \text{if } i \leq k \\ 2p_0^m, & \text{if } i = k+1 \end{cases} \tag{18}$$

where p_0^m is the belief function parameter after $(R^kS)^m$. Next, notice that: $\bar{P}_{m+1} = \bar{P}_m(1-(k+2)p_0^m)$ Also from (16) we have:

$$p_0^{m+1} = \frac{p_0^m}{1-(k+2)p_0^m} \tag{19}$$

As a result, we have:

$$\begin{aligned} P_{capture(i,m+2)} &= \bar{P}_{m+1} \times \begin{cases} p_0^{m+1}, & \text{if } i \leq k \\ 2p_0^{m+1}, & \text{if } i = k+1 \end{cases} \\ &= \bar{P}_m \times \begin{cases} p_0^m, & \text{if } i \leq k \\ 2p_0^m, & \text{if } i = k+1 \end{cases} \end{aligned} \tag{20}$$

which is the same as (18). In other words, the conditional capture probabilities for doing right actions is the same, and for the stay action is twice.

Next, the total capture probability is obtained as the sum of the probability gathered in the first group of (R^kS), and the probability gathered in the next ones. The former is $\frac{k+2}{2N}$, while the latter is $(M-1)(k+2)\bar{P}_1 \times p_0^1$ and p_0^1 is obtained from (17). This proves the value for the total probability of capture.

In order to assess the approximation error, we compared the estimated value with the actual one obtained from simulating the random walk and the strategy $(R^kS)^M$ for various values of M and k. The approximation error, i.e. $\frac{estimate-actual}{actual}$ is below 10% for $3 \leq k$ and below 2% for $7 \leq k$. $\square$

References

[1] N. Noori, A. Renzaglia, and V. Isler, "Searching for a one-dimensional random walker: Deterministic strategies with a time budget when crossing is allowed," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4811–4816, Nov 2013.

[2] S. J. Benkoski, M. G. Monticino, and J. R. Weisinger, "A survey of the search theory literature," *Naval Research Logistics (NRL)*, vol. 38, no. 4, pp. 469–494, 1991.

[3] T. Chung, G. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, 2011.

[4] S. M. Pollock, "A simple model of search for a moving target," *Operations Research*, vol. 18, no. 5, pp. 883–903, 1970.

[5] J. G. Wilson, "On optimal search plans to detect a target moving randomly on the real line," *Stochastic Processes and their Applications*, vol. 20, no. 2, pp. 315 – 321, 1985.

[6] J. M. Dobbie, "A two-cell model of search for a moving target," *Operations Research*, vol. 22, no. 1, pp. 79–92, 1974.

[7] Y. C. Kan, "Optimal search of a moving target," *Operations Research*, vol. 25, no. 5, pp. 864–870, 1977.

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

[9] T. Stweart, "Experience with a branch-and-bound algorithm for constrained searcher motion," *Search Theory and Applications*, vol. 8, pp. 247–253, 1980.

[10] A. R. Washburn, "Branch and bound methods for a search problem," *Naval Research Logistics (NRL)*, vol. 45, no. 3, pp. 243–257, 1998.

[11] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.

[12] S. C. W. Ong, W. P. Shao, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.

[13] H. Lau, S. Huang, and G. Dissanayake, "Probabilistic search for a moving target in an indoor environment," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3393–3398, Oct 2006.

[14] A. Renzaglia, N. Noori, and V. Isler, "Searching for a one-dimensional random walker: Randomized strategy with energy budget," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6019–6024, Nov 2013.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press Cambridge, MA, 2005.

[16] F. Bartumeus, M. G. E. da Luz, G. Viswanathan, and J. Catalan, "Animal search strategies: a quantitative random-walk analysis," *Ecology*, vol. 86, no. 11, pp. 3078–3087, 2005.

[17] S. Redner, *A Guide to First-Passage Processes*, 1st ed. Cambridge: University Press, 2001.

[18] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul Erdös is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[19] J. Anlauf, "Asymptotically exact solution of the one-dimensional trapping problem," *Physical review letters*, vol. 52, no. 21, pp. 1845–1848, 1984.

[20] S. Redner and P. Krapivsky, "Capture of the lamb: Diffusing predators seeking a diffusing prey," *American Journal of Physics*, vol. 67, pp. 1277–1283, 1999.

[21] A. Gabel, S. Majumdar, N. Panduranga, and S. Redner, "Can a lamb reach a haven before being eaten by diffusing lions?" *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 05, p. P05011, 2012.

[22] P. Krapivsky and S. Redner, "Life and death in an expanding cage and at the edge of a receding cliff," *American Journal of Physics*, vol. 64, no. 5, pp. 546–551, 1996.

[23] S. Poduri and G. S. Sukhatme, "Achieving connectivity through coalescence in mobile robot networks," *Proceedings of the 1st International Conference on Robot Communication and Coordination*, pp. 4:1–4:6, 2007.

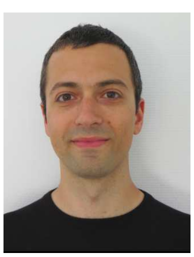
[24] N. Agmon, S. Kraus, and G. Kaminka, "Multi-robot perimeter patrol in adversarial settings," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2339–2345, May 2008.

[25] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Welfare and Pursuit, Control and Optimization*. John Wiley and Sons, 1965.

[26] S. Gal, "Search games with mobile and immobile hider," *SIAM Journal on Control and Optimization*, vol. 17, no. 1, pp. 99–122, 1979.

[27] M. Adler, H. Racke, N. Sivadasan, C. Sohler, and B. Vocking, "Randomized pursuit-evasion in graphs," *Combinatorics Probability and Computing*, vol. 12, no. 3, pp. 225–244, 2003.
[28] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion with local visibility," *SIAM Journal on Discrete Mathematics*, vol. 20, no. 1, pp. 26–41, 2006.
[29] S. Kopparty and C. V. Ravishankar, "A framework for pursuit evasion games in rn," *Information Processing Letters*, vol. 96, no. 3, pp. 114–122, 2005.
[30] A. Kehagias, D. Mitsche, and P. Pralat, "Cops and invisible robbers: The cost of drunkenness," *Theoretical Computer Science*, vol. 481, no. 0, pp. 100–120, 2013.
[31] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computation and Machine Learning Series. Mit Press, 1998.
[32] "Mdp matlab toolbox," http://www.inra.fr/mia/T/MDPtoolbox, accessed November, 2012.
[33] "Approximate pomdp planning (appl) toolkit," http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl, accessed March, 2015.
[34] A. Renzaglia, N. Noori, and V. Isler, "The role of target modeling in designing search strategies," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4260–4265, Sept 2014.
[35] P. Bajer and P. Sorensen, "Recruitment and abundance of an invasive fish, the common carp, is driven by its propensity to invade and reproduce in basins that experience winter-time hypoxia in interconnected lakes," *Biological Invasions*, vol. 12, no. 5, pp. 1101–1112, 2010.
[36] "Oceanscience," http://www.oceanscience.com/.
[37] P. Tokekar, E. Branson, J. Vander Hook, and V. Isler, "Coverage and active localization for monitoring invasive fish with an autonomous boat," *IEEE Robotics and Automation Magazine*, vol. 30, no. 3, pp. 33–41, 2013.
[38] "Clearpath robotics," http://www.clearpathrobotics.com/.

Narges Noori is a Ph.D student in the Department of Computer Science at the University of Minnesota, Twin Cities. She obtained her Master of Science degree from the School of Electrical & Computer Engineering at the University of Tehran in 2010. Her research interests are in pursuit-evasion games, motion planning, computational geometry and multi-robot systems. She is a recipient of 2014-2015 Doctoral Dissertation Fellowship from University of Minnesota, Twin Cities.



Alessandro Renzaglia received his Ph.D. in 2012 from the University of Grenoble, France. From to 2012 to 2014 he has been Postdoc at the University of Minnesota, Minneapolis, USA, in the Robotic Sensor Networks Lab, and he is currently part of the Robotics and Interactions group at LAAS-CNRS, Toulouse, France, as post-doctoral researcher. His research interests lie in the area of multi-robot systems and path planning.

Joshua Vander Hook is a Ph.D student in the Department of Computer Science at the University of Minnesota, Minneapolis. He obtained his Bachelor of Science degree in Computer Science from Minnesota State University, Mankato in 2010. His research interests are primarily in motion planning, tracking and estimation, and pursuit-evasion games. He has received a fellowship from the University of Minnesota, the ARCS Foundation and the Sigma Xi Research Society. After graduation, Josh will be joining the NASA Jet Propulsion Laboratory.

Volkan Isler is an Associate Professor at the University of Minnesota. His research interests are in robotics, sensing and geometric algorithms. He has served as an Associate Editor for IEEE Transactions on Robotics and IEEE Transactions on Automation Science and Engineering. He is the recipient of the CAREER award from the National Science Foundation.