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 13-001

Long-Term Search Through Energy Efficiency and Harvesting

Narges Noori, Patrick A. Plonski, Alessandro Renzaglia, Pratap Tokekar, Joshua Vander Hook, Volkan Isler

January 9, 2013

Long-Term Search Through Energy Efficiency and Harvesting

Narges Noori, Patrick Plonski, Alessandro Renzaglia,
Pratap Tokekar, Josh Vander Hook and Volkan Isler [*]

January 8, 2013

Abstract

We study a search problem motivated by our ongoing work on finding radio-tagged invasive fish with an Autonomous Surface Vehicle (ASV). We focus on settings where the fish tend to move along the boundary of a lake. This setting allows us to formulate the problem as a one-dimensional search problem in which the searcher chooses between station keeping and moving so as to maximize the probability of finding the target in a given amount of time without violating its energy-budget. We model the movement of the target as a random-walk and present a closed-form solution for this search problem. Next, we investigate how long-term autonomy can be enabled by energy harvesting. In this case, the search strategy should incorporate the amount of solar energy available at a particular location and particular time. We show how this quantity can be predicted by estimating the geometry of the tree line along the shore. We then obtain the optimal strategy which maximizes the probability of finding the target by formulating the problem as finding the optimal strategy for a Markov Decision Process. Data collected from field experiments validate our approach.

1 Introduction

One of the prerequisites for long-term autonomy is access to a long-term energy source. Unfortunately, most robotic systems which operate on the field are subject to severe energy limitations. In particular, robot developers face trade-offs between payload capacity and battery life since current batteries are heavy and last a short amount of time under full actuation. The lifetime of a robotic system can be further extended by energy-aware algorithms which use existing resources efficiently and harvest energy when available.

In this work, we focus on an energy-efficient search problem in which a robot with limited battery life is charged with finding a random-walker in a given amount of time. This problem arises in numerous settings such as surveillance, search and rescue, and reconnaissance. Our motivating application is monitoring radio-tagged invasive fish. In recent years, we have been working on building a system of Autonomous Surface Vehicles (ASVs) to locate radio-tagged carp in inland lakes [19]. The purpose of the system is to collect data which can be used for studying carp behavior. Recent experiments have shown that in Twin-Cities metro-area lakes, the fish tend to be near the shore most of the time. This observation allows us to restrict our attention to the boundary of the lake and reduce the problem to a one-dimensional (1d) setting.

Even though the properties of 1d random-walk processes have been extensively studied, it turns out that not much is known about designing algorithms for finding a 1d random-walker. Therefore, we start the paper with a fundamental search problem where the objective is design a search strategy that maximizes the probability of finding a random walker. The searcher is subject to a time constraint and an energy budget which is used for moving and station keeping. For this problem, we present the optimal solution in closed form.

Next, we investigate how the system life can be further extended with the aid of solar harvesting. In order to maximize system lifetime, the search strategy must take into account how much solar energy

[*] The authors are with the Department of Computer Science & Engineering, University of Minnesota. The corresponding author is Alessandro Renzaglia. Email: arenzagl@cs.umn.edu. This work was supported by National Science Foundation Awards #1111638, #0916209, #0917676, and #0936710.

will be harvested when following a particular strategy. This is non-trivial, as the amount of solar energy available along the path may vary depending on the shadows due to obstacles (like trees, buildings, etc.), the location of the sun, and the weather conditions. An accurate solar map is therefore of crucial importance in designing an optimal strategy. For this problem, we make two contributions: First, we show how the expected solar gains at a particular location and particular time can be estimated by using the environment model (lake boundary) and estimating environmental parameters (height of the tree-line along the boundary). Second, we show how the optimal search strategy which maximizes the system lifetime can be obtained by using this knowledge. Our approach is to formulate the problem using a Partially Observable Markov Decision Process (POMDP) in which the partially observable component is the location of the target. We show how the belief can be represented accurately using a fixed parameter distribution which allows us to model the problem using a Markov Decision Process (MDP). Throughout the paper, we use data from field experiments to extract system parameters and solar maps which are then used for simulations validating our algorithms.

The rest of the paper is organized as follows: related works are presented in the next section. Next, we present a detailed description of our system including energy-related parameters for actuation and harvesting (Section 3). In Section 4, we focus on the basic search problem (without harvesting) and present the optimal solution in closed form. Section 5 presents the details of building the solar map, especially for scenarios where the path is along the boundary of a lake. The MDP version of the considered problem is presented and solved in Section 6. Finally, in Section 7 the simulations results are provided.

2 Related Work

Random motions, both as discrete in time and space random walks and continuous diffusive motions, have been extensively considered and analyzed as models of unknown animals motions or complicated physical processes. A large numbers of results for problems related to searching tasks, such as: first passage problem, survival probability and mean capture time can be found in [15].

Different characteristics of random walks in general graphs have been studied in [12]. 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 simple processes, they present several interesting behaviors and properties. For this reason, they are largely studied in literature also as models for pursuit evasion games. The survival probability of a particle that performs a random walk on a chain when traps are uniformly distributed with known concentration is studied in [6] and an asymptotically exact solution is provided. In [16] the authors study the survival probability of a prey chased by N diffusive predators on a line. In this case the capture dynamics is exactly soluble by probabilistic techniques when the number of lions is very small but for three or more predators the exact solution is still not known. The same problem but in a semi-infinite line where the boundary represents a haven for the prey is presented in [9]. Krapivsky and Redner studied the behavior of random walk in a one dimensional bounded environment when absorption occurs whenever the random walk hits a boundary of the system [11]. In particular, the authors considered the case of an expanding *cage* and a receding *cliff*, i.e. with boundaries moving with a known motion law. Contrary to our paper, none of works include any kind of constraint neither on the energy of the system, limiting the predator's autonomy, nor a maximum time for the chase.

As already said, the considered problem can be included in the more general class of pursuit-evasion problems. An overview of recent results on pursuit evasion games can be found in [8]. Adler et al. [5] considered the limited sensing pursuit evasion game in general graphs where the players can sense each other only when both of them are on the same node. They propose a randomized pursuit strategy that achieves the expected capture time of $O(n \times \log diam(G))$ against *any unrestricted* evader strategy [1] where n is the number of nodes and $diam(G)$ denotes the graph's diameter. In this paper, we restrict the evader to move according to random walk similar to the pursuer to one of its neighboring nodes.

Information about solar energy collected from the environment is occasionally used to inform path planning, usually in a long-term mission planning sense. In open environments (e.g. in Antarctica [14] or

[1] The evader is unrestricted in the sense that it can jump to any node in the graph in contrast to the pursuer which is restricted to move to its neighboring nodes.

on the open ocean [17]) this energy depends on the path chosen only through the rotational asymmetry of the panel placement. Therefore energy can often be considered constant at a certain time of day if the solar panels are primarily oriented horizontally. Many environments of interest contain objects or terrain that casts shadows, so estimating these shadows and avoiding them can potentially increase system lifetime. The TEMPEST mission-level path planner [21] calculates the angle of the sun at a particular time and performs raytracing from the estimated shape of nearby terrain to build a shadowmap. Given that sensing the terrain in high detail is a difficult task, we examined estimating the solar map with Gaussian Process regression [13]. This approach did not have a good method to account for the movement of the sun between the time of the training measurements and the time of the planned path, so in this work we switch track and pursue a raytracing-based approach. We take advantage of the structure of the environment our robot operates in to determine the position of nearby occluding terrain using only the same minimal sensors that were required for our GP regression approach: a GPS receiver and a current sensor.

3 System Description

For long-term autonomy considerations, it is important to model and consider the energy consumption of the robot. In this section, we describe the setup of our system, and present results from system modeling experiments. We also describe our sensing system based on radio signal detection. The energy and sensing parameters obtained from these experiments were used for the simulations throughout the rest of the paper.

Our system, shown in Figure 1, consists of Autonomous Surface Vehicles (ASVs) carrying radio tracking equipment for detecting the target and solar panels for energy harvesting during operation. The ASVs are built on boats manufactured by OceanScience [3] 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 [20].

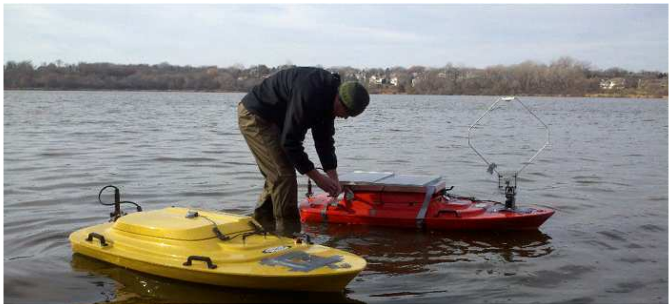


Figure 1: Two ASVs used in field experiments. The solar panels and tracking equipment are installed on the boat that acts as the searcher.

3.1 Energy for Motion

The ASVs use an outboard motor for propulsion and a servo controlled rudder for steering. A single battery is used for powering the propulsion and steering, and separate batteries power the on-board laptop and sensing equipment. The motor battery is capable of powering the robot for about one hour at nominal cruising speed, and forms the bottleneck in the system lifetime. The battery lifetime can be extended by optimizing the speed of operation.

The energy consumption of the motors consists of two components: power consumption while moving (c_m) and power consumption while stationary (c_s). The servo motor for steering and quiescent current of the motor controller circuits contribute to c_s. Next, we describe our modeling experiments to obtain these parameters.

We performed trials to log the power consumption of the ASV at various cruising speeds. The experiments were conducted on a day with typical wind conditions. A custom-built circuit was attached between the battery and motor control circuits to measure the current consumption during operation. The battery voltage remains roughly constant throughout, and hence only the current consumption is used to compute the energy. The robot was programmed to autonomously visit waypoints such that it moved a total of 150m each with and against the predominant wind. The waypoint navigation controller [19] applies a constant voltage to the propeller and controls the steering. As a result, the direct speed of the robot cannot be controlled, however, the actual speed can be recorded from the state estimates.

The average speed and the average current consumption of the propeller motor, motor controller circuit and steering servo is presented in Table 1. The data indicates that it is energy efficient to travel slower and consequently for a longer time (for same distance) than at higher speeds. Since the autonomous navigation at speeds lower than 1.28m/s is not reliable, we operated the robot such that it will move at average speed of 1.28m/s for the rest of the experiments and modeled the instantaneous power consumption as 5.72A (or coulombs/sec). The current consumption while stationary was directly measured and is about 0.2A. That is, the costs of moving for a distance d and being stationary for time t are given by,

$$c_m = 5.7\frac{d}{1.28} \text{ Coulombs}, \quad c_s = 0.2t \text{ Coulombs}. \tag{1}$$

3.2 Energy Harvesting

To improve the system lifetime, we equipped the search boat with two solar panels to harvest solar energy during execution. The solar panels used were SPM020Ps, manufactured by Solartech Power [4]. Each can provide up to 20 watts in standard test conditions of 1000 w/m^2 insolation and a temperature of 25°C. The panels were installed horizontally to allow equal harvesting under any robot orientation. Both panels were connected in parallel with the boat battery. A hall-effect sensor was used to measure the current flow from the panels to the battery. In this configuration, the solar intensity, current through the panels, and power into the battery are all proportional.

The solar energy harvested is a function of a number of factors: time of the day, time of the year, cloud conditions, environment of operation, amongst others. Section 5 presents the details of the solar energy modeling in our system. Typical energy harvesting parameters are shown in Table 1.

Avg. Speed (m/s)	Avg. Current (A)
1.28	5.72
1.50	16.30
1.58	22.99

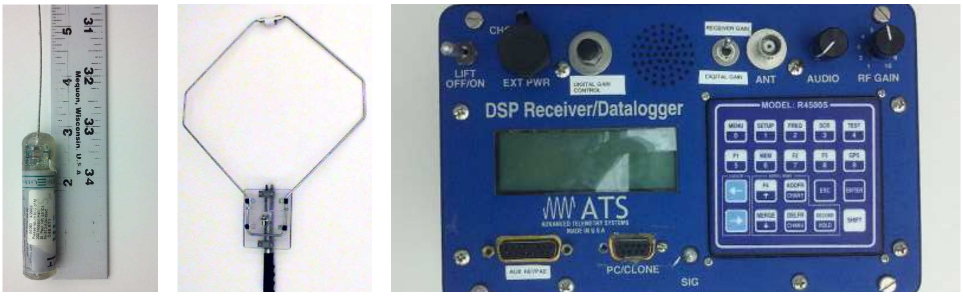
Table 1: Power to drive data for the boat under closed-loop control driving 150m each with and against the wind on a typical day.

Type of Position	13:45 CDT on June 9	11:15 CDT on September 16	12:00 CST on November 20
Unshaded	2.67 A	1.53 A	0.523 A
Shaded	0.279 A	0.302 A	0.257 A

Table 2: Measured solar current harvested at 12.5v on representative clear days. Clearly, a horizontal panel can collect more solar radiation in June than in November. Note however that power in the shade varies much less than power in the sun; this is because diffuse radiation is not very dependent on solar angle. See Section 5.1 for details.

3.3 Sensing for Detecting Target

For our experiments, one of the ASVs was designated the searcher robot and the other used as the random-walking target. We equipped the searcher robot with tracking equipment manufactured by Advanced Telemetry System (ATS) [1], shown in Figure 2-(b, c), and placed a radio transmitter on the target robot Figure 2(a). The tracking system consists of an antenna, digital signal processor, and radio-tags. Each tag transmits a low-power, uncoded pulse on a unique frequency approximately once



(a) (b) (c)

Figure 2: Tracking equipment: (a) Radio tags surgically implanted on fish, (b) Loop antenna used to detect radio tags within its sensing range, and (c) Radio receiver equipment providing a digital interface between antenna and on-board computer.

per second. The strength of the signal attenuates due to range, battery life, and being submerged in water.

The receiver is designed to be manually monitored. The incoming signal is converted to audio, which a human operator can use to detect the proximity of a radio tag by listening for the pulse from the tag. The receiver also attempts to isolate the radio pulses and estimate the signal strength relative to the background noise. The output of the this process is a radio signal strength indicator (RSSI). The RSSI values are integers from zero (when the tag is out of detection range) up to 255, which is used by the searcher robot to detect capture of the target.

We conducted field experiments to estimate the sensing range of the antenna and receivers. We found the receiver reliably reported a detection when the tag was less than 20 meters from the searcher. Beyond this range, detection was possible manually, but the reported signal strength was not reliable for automatic detections.

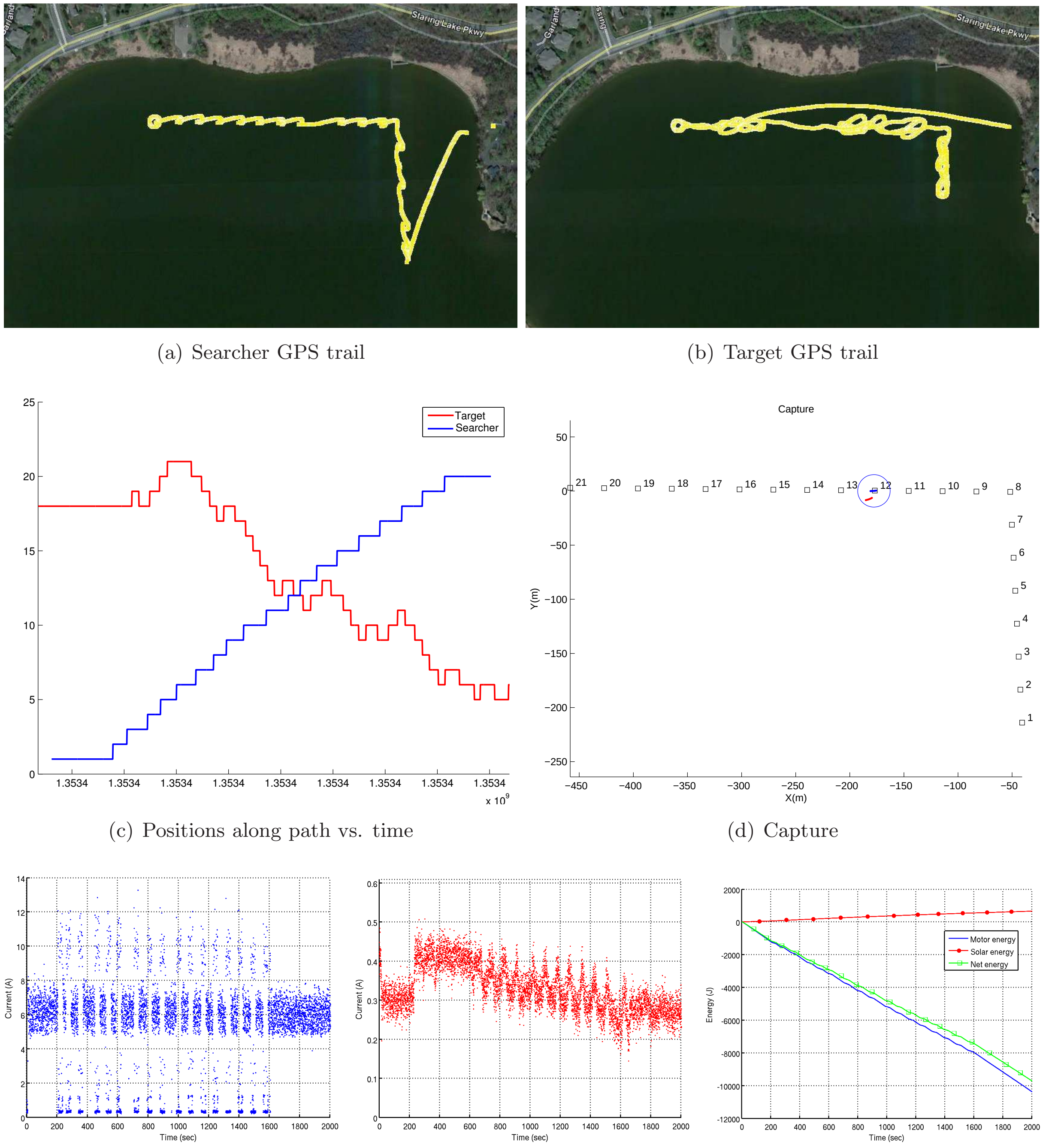
We also observed that the reliability of detection was considerably decreased when the boat was in motion. The antenna is sensitive to radio interference from the motors of the boat, which leads to false positive detections. To reduce such false positives, we require the boat to stop, and then take a measurement from the receiver. For added accuracy, we take more than one measurement after stopping to decide if a tag is present in the vicinity. Since the tag emits pulses roughly once per second, for field trials, we wait at least 10 seconds at each location. This sensing requirement naturally leads to discretizing the searcher path into nodes from which measurements can be obtained, as opposed to continuously taking measurements.

Figure 3 shows the results from a demo experiment performed, where the searcher successfully captured the target. The path had a total length of about 610 meters and was discretized into a total of 21 points. The experiment lasted for 33 minutes. The resulting strategy executed by the searcher is shown in Figure 3-(a) as the GPS trail obtained from the boat. The target executed a random walk strategy with equal left and right move probabilities set to 0.5. Figure 3-(b) shows the GPS trail of the target boat.

Figure 3-(c) shows the trajectories followed by the searcher and the target for this instance, with GPS time on the X-axis and the position of the searcher and target on the line on the Y-axis. The location where the target and searcher trajectory coincide is shown in Figure 3-(d). At this location, the radio antenna on the searcher boat successfully detected the signal from the emitter on the target boat, demonstrating capture.

Figures 3-(e, g) show the measured motor current, solar current and net energy consumption values. We can observe the motor current during motion was marginally greater than the modeled current of 5.7A, and the current while stationary was about the modeled value of 0.2A. The occasional high current values can be attributed to the spike in current consumption during transitions from stationary to motion. Figure 3-(g) shows the gains from installing solar panels.

The strategy executed by the robot for this trial was not the optimal strategy. Following sections describe how to find the optimal strategy: first considering the scenario without energy harvesting, next with energy harvesting given a solar map, and finally energy harvesting with changing solar maps during the day.



(a) Searcher GPS trail

(b) Target GPS trail

(c) Positions along path vs. time

(d) Capture

(e) Motor Current

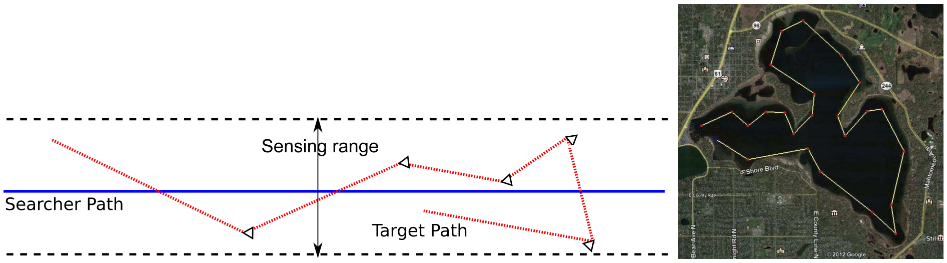
(f) Solar Current

(g) Net Energy Consumption

Figure 3: Demo experiment conducted using the setup shown in Figures 1 and 2. (a) and (b) shows the GPS trails followed by the boats acting as the searcher and the target, (c) shows their locations along the line. Capture occurs when their trajectories cross, shown in (d). The energy consumption, energy harvested and net energy are shown in (e), (f) and (g) respectively.

4 Limited Energy Searching

In this section, we study the search process subject to a limited energy budget, with the objective of maximizing the probability of sensing random walking targets. As mentioned previously, the invasive fish, in our application, are often found close to the boundary of the lake. We can assume that the fish movement is bounded in a corridor of width comparable to the sensing range. Hence, we can restrict the searcher motion to a 1d path around the boundary of the lake, as shown in Fig. 4-(a). Note that while the searcher moves along a path, the fish may move anywhere in a region of width less than the sensing radius of this path.



(a) (b)

Figure 4: (a) The target moves randomly in a region of width less than the sensing range. This allows the searcher to restrict its searching domain to a 1D path. In (b) it is shown a typical scenario in a lake where such path is next to the boundary.

In the absence of any reliable motion model for the targets (fish, in our application), we can reasonably assume the targets as performing a random motion. Since, as we explain in Section 3, our sensing system does not allow us to continuously take measurements, we can discretize the searcher's path as a sequence of nodes, which represent the sensing locations. With some conditions on the discretization step, the target's position can now be mapped on its closest node and its motion modeled by a random walk on this chain. In such a scenario, it is clear that the best strategy for the searcher is to sweep the path from one end to the other, for guaranteed capture. However, in real applications, such solution is not always achievable due to limited energy. Let us consider for example a typical setting in our motivating application, shown in Figure 4-(b). The path around the boundary of the lake is about 16km long. Our robotic boat (described in Section 3) is capable of only operating at 1-2m/s which would require more than 2 hours for completing the path, where as the on-board battery is capable of only up to an hour. As a result, for such long-term operations, it becomes crucial to consider more efficient strategies and to take into account the energy consumption of the robots.

Motivated by this, we focus our interest on scenarios where the robot does not have enough energy to traverse the entire path, and the main purpose of this paper is to optimize the motion, seen as a sequence of actions (move right, left or remain stationary), such that the probability of capturing a random-walking target is maximized. In this section, we only consider optimizing energy expenditure through motion or station-keeping, and in the following sections additionally consider energy harvesting.

The random-walking target motion can be modeled by the following Master equation:

$$P(i,t+1) = pP(i+1,t) + sP(i,t) + qP(i-1,t)\,. \tag{2}$$

where p, q, s are the probabilities of the target moving to the left node, right node, and staying at the same node. Because we deploy our system in inland lakes, we ignore the effects of water currents on the transition probabilities p and q, and assume them to be equal. The distance between two adjacent nodes $d(i,i+1)$ is chosen as a function of the sensing range. In our model this corresponds to the distance between two successive measurements. We wish to prevent the boat from passing over the fish while moving from a node to an adjacent node without sensing the target. To ensure this, the following condition must be satisfied:

$$d(i,i+1) < R_s\,. \tag{3}$$

Moreover, field studies have revealed that the fish motion is not spread along the entire boundary of the lake. On the contrary, there are some bounded, disconnected areas where they loiter: in either their individual "home regions," near food sources, or in spawning areas [7]. Formally, we can incorporate this behavior by imposing reflecting boundary conditions to our searching chain i.e., if the fish is on N at time t, where N is the number of nodes, at time $t+1$ it will be on $N-1$ with probability 1.

The objective is to find the best searcher strategy $\mathcal{S} = \{a_1, a_2, ..., a_k\}$ in order to maximize the probability of capture, where each action a_i can be: *go right*, *go left* and *stay.* The maximum number of actions k for a given strategy may be constrained by two factors:

- a limited amount of initial energy E_0 for the system;
- a maximum time T to complete the task.

Formally, the optimization problem we want to solve is:

$$\max_{\mathcal{S}} P_c(\mathcal{S} = \{a_1, a_2, ..., a_k\}) \quad s.t. \tag{4}$$

$$\sum_{i=1}^{k} cost(a_i) \leq E_0, \tag{5}$$

$$k \leq T\,. \tag{6}$$

If we disregard the cost constraint (Eq. (5)), the most intuitive, and at the same time most efficient, strategy would be to simply sweep the entire segment without stopping. In this case the probability of capturing the target is trivially 1 and, as we will show, the mean capture time is order $N/2$. A more interesting scenario, and one of crucial importance for many real applications, is when the initial energy budget does not allow the searcher to sweep the whole segment. As explained in the previous section, 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. Hence, the energy constraint (5) can be expressed as a function of the total number of movements L and stationary steps S:

$$c_m L + c_s S = E_0\,. \tag{7}$$

The problem we wish to solve is: what fraction of energy should be employed in moving and what fraction in waiting for the target by staying in the same node. Also interesting is how to distribute these actions in the area of interest. To analyze this problem we firstly consider the following Proposition:

Proposition 4.1. *Let $\mathcal{S}_1, \mathcal{S}_2$ be two different searcher strategies and $x_1(t), x_2(t)$ the location of the searcher at time t when executing $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. Then it holds:*

$$x_1(t) \geq x_2(t)\ \forall t \quad \Rightarrow \quad P_c(\mathcal{S}_1) \geq P_c(\mathcal{S}_2)\,, \tag{8}$$

where $P_c(\mathcal{S}_i)$ is the capture probability executing the strategy $\mathcal{S}_i$.

This Proposition can be justified observing that, imposing condition (3), so without considering crossing between searcher and target, any target captured by the strategy $\mathcal{S}_2$ is captured also by the strategy $\mathcal{S}_1$ with probability 1. On the other hand, the opposite is not always true.

Assuming that the searcher starts its mission from the left bound, i.e. $x = 0$, Proposition 4.1 has two immediate consequences: *i*) the left action is never useful and *ii*) given a set of L movements and S stay actions, the best possible strategy is to move L steps to the right and then employ S stay actions waiting at $x = L$.

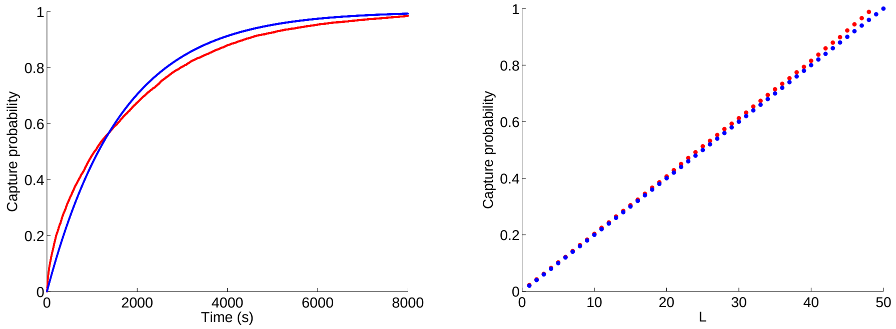
Once the sequence of actions has been fixed, and since we can express $S = S(L; E_0)$ in terms of L and E_0 using (7), the only optimization variable for the proposed problem is the final searcher position L. The next step is to obtain an analytical expression of the capture probability P_c as a function of L. Such probability can be expressed as:

$$P_c = P_m + (1 - P_m)P_s\,, \tag{9}$$

where P_m is the probability that a capture has occurred when the searcher reaches L and P_s is the capture probability for remaining S steps in the same node. To compute these probabilities we separately analyze the two components: waiting in a given node and moving in one direction.

4.1 Static Searcher

Instead of computing P_s directly, we consider the survival probability $\tilde{P}_s$ of a random walk moving in a bounded environment with an absorbing bound in $x = 0$ (then, P_s is simply $P_s = 1 - \tilde{P}_s$). Even if the exact analytical expression is very complicated (see [15] for the result in a circular environment), it can be shown that the leading term of such probability decays exponentially in time, i.e. it assumes the form: $\tilde{P}_s(t) = e^{-t/\tau}$. Moreover, as a first approximation, the characteristic time of decay can be identified with the mean capture time or, in other words, the expected time for the random walk to



(a) (b)

Figure 5: Comparison between the approximations of the capture probabilities for waiting and moving strategies (in blue) with the behaviors obtained in simulation (in red).

reach $x = 0$ starting from an initial position $x = n$ ([15], Chapter 2). This quantity can be expressed in a recursive way as follows:

$$t_n = p(t_{n-1} + 1) + s(t_n + 1) + q(t_{n+1} + 1)\,, \quad n = 1, 2, ..., N - 1\,, \tag{10}$$

where the time-step is 1 and the boundary conditions which correspond to absorption in $i = 0$ and reflection in $i = N$ are $t_0 = 0, t'_N = 0$. For a symmetric random walk the previous equation becomes:

$$t_n = p(t_{n-1} + 1) + (1 - 2p)(t_n + 1) + p(t_{n+1} + 1)\,, \quad n = 1, 2, ..., N - 1\,. \tag{11}$$

The solution of the previous recursive relation is:

$$t_n = A + Bn - \frac{1}{2p}n^2\,, \tag{12}$$

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

$$t_n = \frac{n}{2p}\left(2N - n\right)\,. \tag{13}$$

Since we do not know the initial target position, we compute the expected value $t_{<n>}$, assuming an initial uniform distribution. The result is:

$$t_{<n>} = \frac{N^2}{2p} - \frac{1}{2p}\frac{2N^2 + 3N + 1}{6} = \frac{4N^2 - 3N - 1}{12p}\,. \tag{14}$$

Let us consider for simplicity $p = 1/2$. The leading term of the capture probability becomes

$$P_s(t) = 1 - \tilde{P}_s(t) \approx e^{-\frac{6t}{4N^2 - 3N - 1}}\,. \tag{15}$$

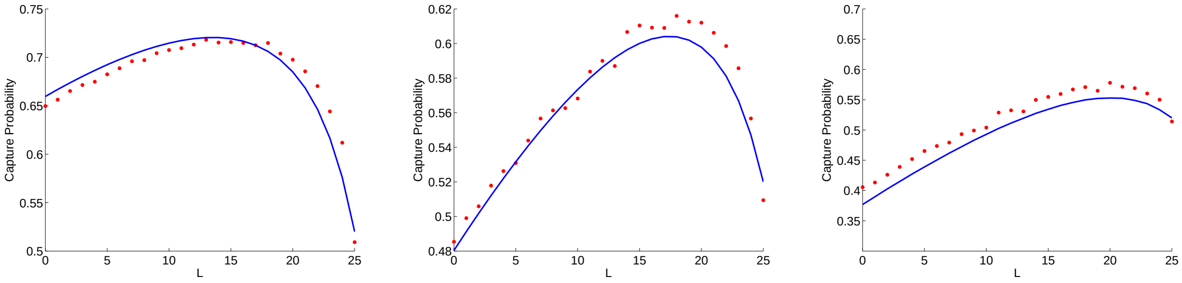
In Figure 5-(b) the previous approximation is compared with the average capture probabilities obtained in simulation[2].

4.2 Moving Searcher

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 sub-dominant. 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 in the case of a stationary searcher. This result can be proven in the same way of (11), which now assumes the form:

$$t_n = p(t_{n-2} + 1) + (1 - 2p)(t_{n-1} + 1) + p(t_n + 1)\,, \quad n = 1, 2, ..., N - 1\,. \tag{16}$$

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



(a) (b) (c)

Figure 6: 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 L. 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.

whose solution is:

$$t_n = A + B\left(\frac{p}{p-1}\right)^n + n\,. \tag{17}$$

In terms of capture probability, the previous behavior allows us to make the approximation $P_m = L/N$, which means that the capture probability is equal to the probability that at time $t = 0$ the target is within the region that will be swept. In Figure 5-(a) the previous approximation is compared with the real capture probabilities obtained in simulation.

Coming back to the main optimization problem, we are now able to write the approximation of the total capture probability (9) we want to maximize. Assuming for simplicity $s = 0$, we have:

$$\begin{aligned} P_c &\approx \frac{L+1}{N} + \left(1 - \frac{L+1}{N}\right)\left(1 - e^{S(L;E_0)/t_{<n>}}\right) \\ &= \frac{L+1}{N} + \left(1 - \frac{L+1}{N}\right)\left(1 - e^{-\frac{6(E-c_mL)}{c_s\left[4(N-L-1)^2-3(N-L-1)-1\right]}}\right) \end{aligned} \tag{18}$$

Note that, since the considered function is continuous in the closed and bounded interval $[0, L_{max}]$, where $L_{max} = E_0/c_m$, it always admits a maximum value.

In Figure 6 the comparison between the expected capture probability and the results obtained in simulation is shown. In particular, we considered a segment composed by $N = 50$ nodes. The initial energy budget is $E_0 = 50$ and the cost for moving $c_m = 2$. The three plots presented in Figure 6 correspond to different values of the cost c_s. We see that with these values the solution of the optimization problem is not trivial and the best final searcher position L^* is $0 < L^* < L_{max}$. On the other hand, using the previous result, as it is shown in Figure 7, we can find the intuitive optimal strategy for the two limit cases: $c_s \to 0$ and $c_s \sim \mathcal{O}(c_m)$. In the first case, moving is too expensive and the best strategy is simply wait in the initial position, so $L^* = 0$; in the second one the best choice is keep moving without any stops during the searching process, which corresponds to $L^* = L_{max}$.

4.3 Time Constraint

So far, we studied the problem considering only a constraint on the total available energy. Let us now assume to have also a constraint on the time to terminate the mission. So, in addition to (7), the following constraint has to be satisfied:

$$L + S(L;E_0) \leq T\,. \tag{19}$$

To find the optimal value of L under this new constraint we separately consider three different cases:

- $T > S_{max} = S(0;E_0)$: the time is greater than the maximum duration of the task allowed by the energy constraint. In other words the time constraint is never active and the solution is the same previously presented.

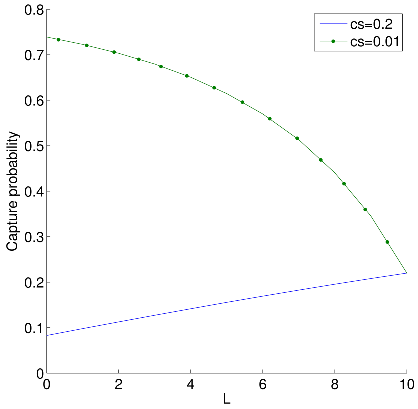


Figure 7: Behavior of the capture probability for the two limit cases: $c_s \to 0$ (in green, with dots) and $c_s \sim \mathcal{O}(c_m)$ (in blue). The other parameters are fixed to: $N = 50$, $E_0 = 20$, $c_m = 2$.

- $T < L_{max}$: the available time does not allow the searcher to consume all the energy budget, so the constraint (7) is not active. However, the new constraint can be seen as an energy constraint where $c_s = c_m$ and the solution to this problem is employing all the energy in moving without any stop.
- $S_{max} < T < L_{max}$: this is the most interesting case and we want to show that this case can be translated in a constraint on L. Indeed, if the previous relation holds, there exists an $\hat{L} < L_{max}$ such that
$$\hat{L} + S(\hat{L}; E_0) = T \quad \Rightarrow \quad \hat{L} = \frac{E - c_s T}{c_m - c_s}\,. \tag{20}$$
This means that $\forall L < \hat{L}$ the searcher will be not able to use all the available energy and every time L is decreased by an unit, S is increased by only 1 unit. Thus, for value of L lower than $\hat{L}$ the actual cost of staying becomes equal to the cost of moving. But we already know that in this case employing energy to wait is not the right strategy to increase the capture probability. As a result, we can state that:
$$P(L) < P(\hat{L})\,, \quad \forall L < \hat{L}\,. \tag{21}$$
In terms of the optimization problem, the previous result simply restricts the searching domain from $[0, L_{max}]$ to $[\hat{L}, L_{max}]$.

Thus, we can compute the optimal searching strategy given an energy and time budget. In this section, we only considered the energy expenditure through motion and station keeping. In general, by adding solar panels, energy can also be harvested leading to a more challenging optimization problem for complex solar maps. Before we address this optimization problem, we first present how to construct such solar maps, especially for regions near the boundary of a lake.

5 Solar Estimation

In this section we will present our method for estimating how much power the robot will get from the sun in a given position at a given time. We will see later how this information can be used to devise an optimal movement strategy in a realistic environment.

5.1 Background

First we will provide a brief overview of the relevant factors in solar photovoltaic power generation for our system. For a more in depth look at solar photovoltaics as they relate to mobile robotics, see [13].

There are well-known equations to determine the solar zenith and azimuth angles from GPS position and time of day, and there are also well-known equations to compute the intensity of direct and diffuse insolation, where direct is radiation directly from the solar disk and diffuse is radiation from the parts

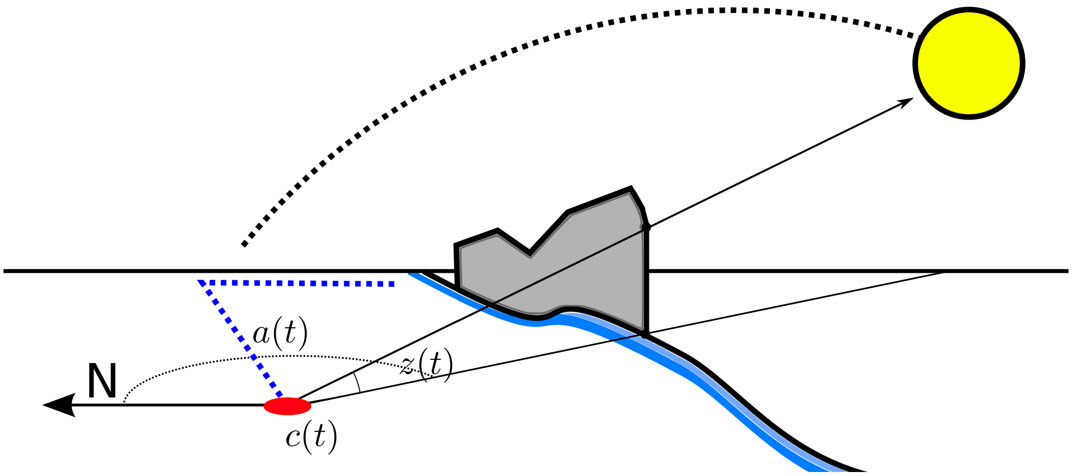


Figure 8: The solar modeling approach. For any time t, we know the robot's position as $c(t)$. Then the azimuth angle ($a(t)$) in degrees from north, and elevation angle ($z(t)$), can be easily calculated. We can construct an approximation of the obstacles which block the sun by projecting the points directly to the known shoreline.

of the sky other than the solar disk [10]. Our robot has two horizontal panels, so given an optical depth k and scaling factors C_r and C_f we compute a direct current I_r and diffuse current I_f for any zenith angle z as follows:

$$I_r = C_r \times e^{\frac{-k}{cos(z)}} \tag{22}$$

$$I_f = C_f \times cos(z) \times e^{\frac{-k}{cos(z)}} \tag{23}$$

If the entirety of both panels can see the sun, the total solar current $I_s = I_r + I_f$. If at least one cell in each panel is fully shaded, $I_s = I_f$. This is because each panel consists of cells in series with no bypass diodes, so output is limited by the weakest cell. Almost all positions in our lake environment can see the majority of the sky (even though the solar disk itself may be occluded), so it is reasonable to approximate diffuse insolation as equal throughout the domain. Also, we can expect there to only be a few positions where the panels are partially shaded but there isn't an entire cell in the shade, so for path planning purposes in our environment it makes sense to consider only the two options of whether or not the panel as a whole can see the sun and collect direct insolation.

Previously energy-aware path planning for a solar powered robot has been accomplished using this classification of space into sun and shade. This classification has come from raytracing on sensed objects that have the potential to occlude the sun. This usually requires the ability to sense the environment in high detail, a capability we do not have. However, we can use a similar raytracing method by exploiting the structure of our lake environment.

5.2 Our Strategy

In our previous work we sidestepped the problem of environment sensing and reconstruction by estimating solar power at a given position using only Gaussian Process regression from previously recorded solar power measurements at positions. This approach worked reasonably well over short time scales but GP regression is expensive when there are many sample points and it is difficult to extend this approach to account for the changing sun position and intensity during the day.

For this work we take a different approach, more in line with the raytracing strategy. Our solution to the sensing problem is to approximate all occluding objects as occurring along the boundary of the lake environment. This is a good approximation for the lakes we consider near the Twin Cities area in Minnesota because there is no dramatic elevation change; the predominant occluding objects are in fact the ring of trees on the edge of the lake.

In our domain we approximate that all occluding objects occur on the edge of the lake, and we also go one step further and represent the lake boundary as a heightmap function of distance along the boundary. This provides a simple way to express the intuition that if we detect shade we can expect that moving closer to the shadow casting object will provide only more shade. Now, most deciduous trees are at their broadest at a height other than the bottom of their trunk, so there are many specific counterexamples where moving towards the shore will actually result in switching the correct classification from sunny to shaded. Nevertheless, the heightmap approximation is useful enough at representing the relevant aspects of our environment that we use it for planning despite its shortcomings.

5.3 Implementation Details

The input to our solar map construction algorithm is a long series of time-stamped and position-stamped solar current measurements, and a lake boundary (which we manually extract from Google maps). The output is an estimate of solar current at a set of points at specified times.

Our algorithm is three part: first determine the calibration parameters k, C_r, and C_f (calibration), then construct the estimated heights (construction), and then use those estimated heights with the scale to estimate the requested currents (estimation).

5.3.1 Calibration

To estimate the solar power at any time of day we first need to estimate the solar scaling parameters for the current system conditions and weather conditions. We did this using the following method:

Calculate z_t for each measurement time in the training set, and classify each training measurement as sunny or shaded depending on the magnitude of all measurements given time of day (we assumed the reasonable $k = 0.2$ for this step). Then perform a numeric optimization to find the C_r, C_f, and k that minimize the error between theoretic and measured solar current, both in the sun and in the shade. When there is not a large measurement time window it can be difficult to determine the best k without overfitting, so the representative default of 0.2 may be used.

5.3.2 Construction

Once we have the calibration parameters, we need to estimate a height map for the shoreline. We did this using the following method:

Use calculated solar zenith and azimuth angles to project each solar measurement towards the sun until it intersects with a lake boundary. Associate this measurement and its height at the intersection with the intersection's position on the boundary. If it intersects the lake boundary multiple times, choose the lowest height (i.e. the intersection closest to the measurement point). Once all measurements are associated with positions on the lake boundary, discretize the boundary into bins. We chose bins of 3m in width because 3m is a bit larger than our usual GPS error but still small enough to ensure that each bin had sufficient measurements in it given the rate we were logging GPS positions.

Each bin has some number of measurements classified sunny and some measurements classified shaded. We can usually say that the height is greater than the highest shaded measurement height and less than the lowest sunny measurement height. In the occasional case where the highest shaded measurement height is *higher* than the lowest sunny measurement height, we simply swap the two bounds. If a bin does not have any of one of the classifications, set the min height to 0 or the max height to some defined max height (we used 40 meters).

5.3.3 Estimation

Given the solar scales and the heightmap estimate, predicting the solar current at a certain position at a certain time becomes straightforward. First calculate I_r and I_f for the desired time of day using equations 22 and 23. Then project the queried position towards the sun until it intersects a boundary and observe the intersection height compared with the upper bound and lower bound of the adjacent bins. If it is above the upper bound line, we expect to collect $I_r + I_f$. If it is below the lower bound we expect to collect only I_f. If it is somewhere in between there are many possible methods of estimation. We examined using the weighted average and this strategy did not work as well as simply approximating that every prospective measurement in the unknown region is equally likely to be sunny as shaded.

This is because the distance along the path between the lower bound line and the upper bound line is in fact a poor predictor of solar gain.

5.4 Demonstration

On September 16th 2012 we densely sampled the solar power collected on the eastern edge of Lake Como in St. Paul at around 0845, 1000, and 1115 CDT (so about 0745, 0900, and 1015 solar time). Lake Como is an urban lake that is not in a densely forested area, however like most lakes in the area it has a ring of trees around it. We cross validated our solar estimation approach by leaving out each data set in turn and estimating the solar current we would collect at each of the measurement positions in the left-out set. See Figure 9 for the constructed height maps of the shoreline, Figure 10 for the solar intensity calibration results, and Figure 11 for the comparison of predicted classification and actual solar current. See Figure 12 for the estimated solar map as it is expected to change during the day. The height maps were consistent across different times of day, suggesting that our approximation of obstacle positions is correct. The predicted solar intensity curves have quite different k values depending on which data set is left out. This is a problem that will be difficult to fix because k can be expected to vary significantly with air moisture and dust even on a clear day – that is, the system won't have a good idea of the correct k until data has been collected over a long time period in the present weather conditions. Therefore, error bounds are high when predicting solar conditions for a sun zenith angle far from any sampled zenith angle. The histograms compared with classification are very close to what would be expected from a good classifier. These results all combine to suggest that our obstacle and intensity assumptions were correct and therefore our solar estimation algorithm should work well in environments similar to Lake Como.

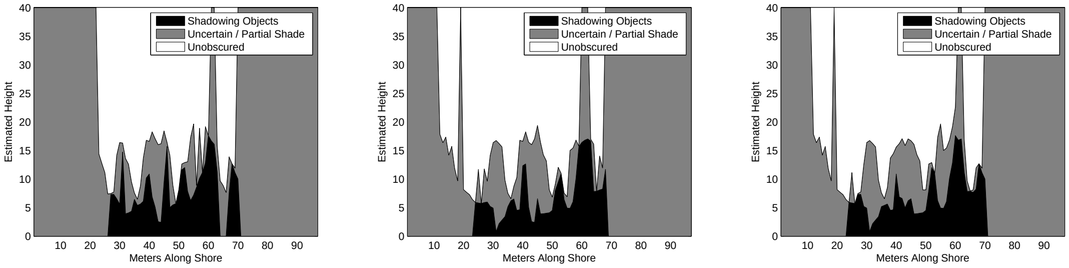


Figure 9: Constructed height maps when leaving out sets 1, 2, and 3 respectively.

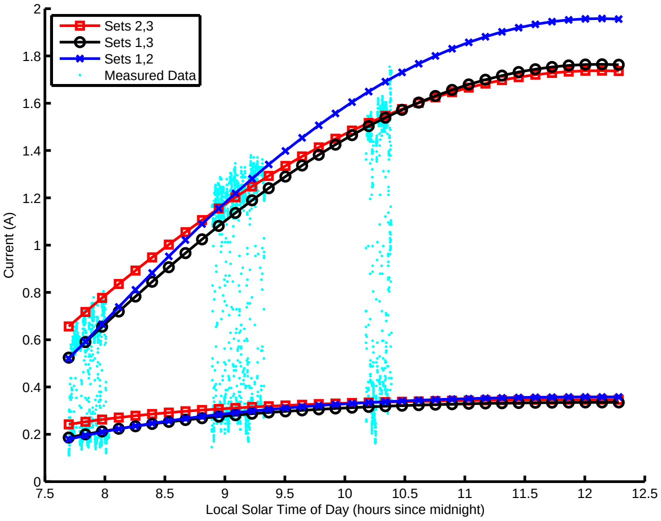


Figure 10: Expected solar panel current in the sun and the shade at different times of day, plotted against actual measured solar panel current.

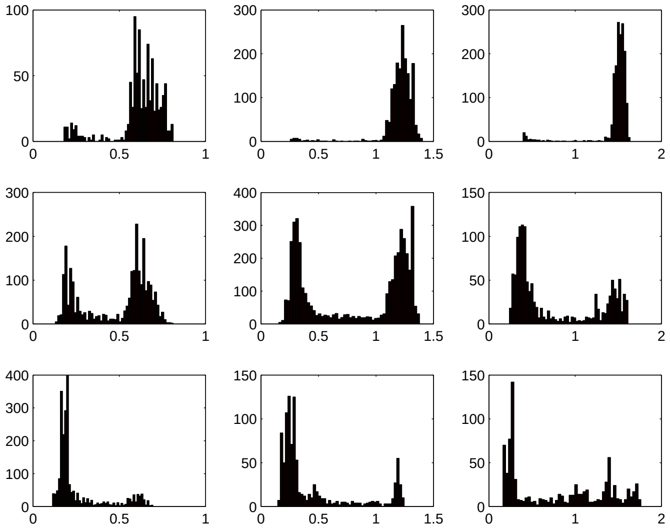


Figure 11: Histograms of actual solar power given classification as sunny (top), shaded (bottom), or uncertain (middle) for sets 1, 2, and 3 respectively from the left. Measurements were classified based on their position and time, with no knowledge of any true measurements in that set.

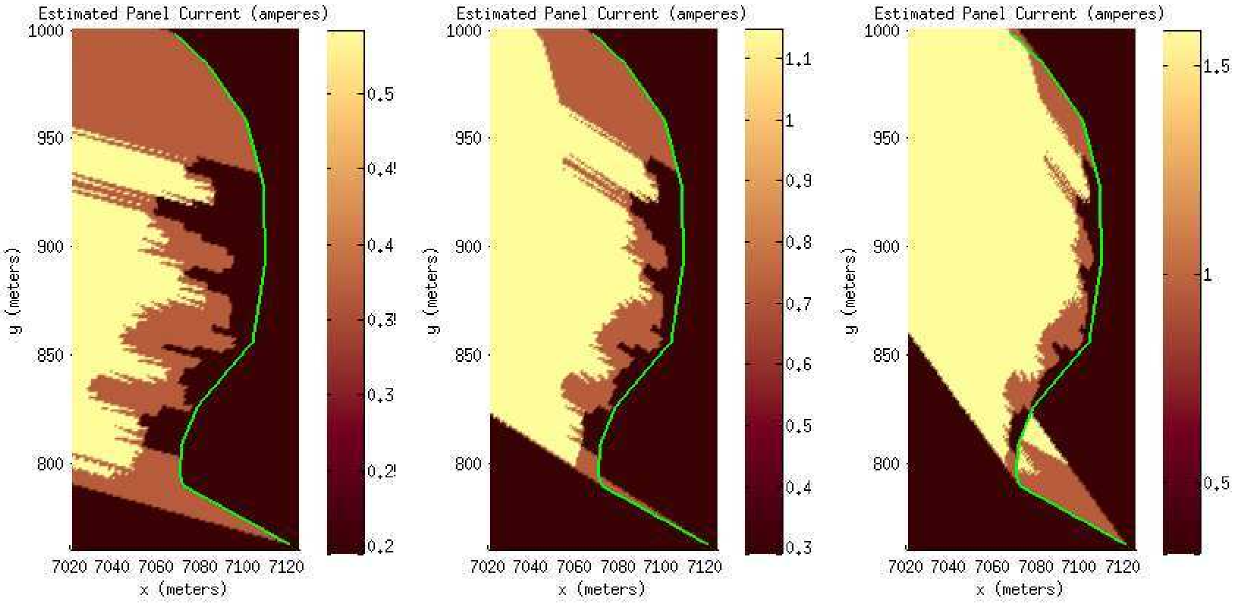
6 Searching with Harvesting

The optimal strategy obtained in Section 4 is not necessarily optimal when the searcher has the possibility to harvest solar energy along the path and increase the system lifetime. In particular, when a complex solar map is available for the searching area, the optimal strategy may stop at more than a single node to harvest solar energy. However, in such cases, it is not evident which nodes are selected for harvesting, and how much time the robot waits at each of the nodes.

To solve this new optimization problem, in this section, we present the formulation of our problem as a *Markov Decision Process (MDP)* [18]. An MDP is described by a four tuple (S, A, P, R), where S refers to the set of possible states, A refers to the set of actions, P is the probability of transitioning between the states for each action, and R is the reward collected for each transition. In our formulation, these quantities are defined as follows:

- $S = \{s_i | i = 1, ..., n\} \cup \{s_{capture}\} \cup \{s_{no-energy}\}$ is the finite set of possible states for the searcher. The states $s_{capture}$ and $s_{no-energy}$ are the terminal states, where $s_{capture}$ denotes the capture state, and $s_{no-energy}$ denotes the state in which the robot runs out of energy. We define the remaining states as $s_i = (B, E, c)$ where B is the belief of the searcher about the position of the target, E is the current energy of the searcher and c 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. Note that $\sum_{i=0}^{N} p_i = 1$, and $p_c = 0$ and $p_{c+1} = 0$ (otherwise the target would have been captured, given the searcher is at c).
- $A = \{stay, left, right\}$ is the set of actions that the searcher can perform in each state: *stay* at its current position, move one step to the *left*, or one step to the *right*.
- P is the $(n + 2) \times (n + 2) \times 3$ transition probability matrix with entries $P(s_i, s_j, a)$, i.e., the probability that the searcher transitions to state s_j by performing action a in state s_i. Note that since S includes the belief of the target, state transitions are not deterministic and instead depend on the unknown location of the target.
- R is the $(n+2) \times (n+2) \times 3$ reward matrix: the transition reward from state s_i to state s_j after performing action a.

The states: Initially, the searcher starts from the location $c = 0$. The searcher begins its mission with no information about the position of the target (except that the target cannot be at 0 or 1,



(a) 08:41 Solar Map

(b) 10:02 Solar Map

(c) 11:23 Solar Map

Figure 12: Three solar maps constructed for Lake Como on September 16, 2012

otherwise it would have been captured). Therefore, the initial state is $s = ([0, 0, \frac{1}{N-1}, ..., \frac{1}{N-1}], E_0, 0)$ where E_0 is the initial energy of the robot.

Note that the number of states $(n+2)$ will be exponential in the number of discrete levels used for the belief probability vector B (with the exponent equal to N, the size of the environment). In order to compactly represent the state, we approximate the belief vector $B = [p_0, p_1, ..., p_N]$ by a function of the form $f(x; x_c, \alpha) = k(1-\alpha^{(x-c-1)}), c < x \leq x_c$ where c is the current position of the searcher, α and x_c are two free parameters used for the approximation, and k is the normalization factor. Specifically, we approximate the searcher's belief B as:

$$f(x; x_c, \alpha) = \begin{cases} 0, & \text{if } 0 \leq x \leq c \\ k\left(1 - \alpha^{(x-c-1)}\right), & \text{if } c < x \leq x_c \\ k\left(1 - \alpha^{(x_c-c-1)}\right), & \text{if } x_c < x \leq N \end{cases} \tag{24}$$

Fig. 13-(a) demonstrates the function f for $c = 6$, $x_c = 20$ and $\alpha = 0.707$. The parameters x_c and α are free to be set: we find them empirically by minimizing the mean error between the actual belief and its approximation function $f(x; x_c, \alpha)$. In order to examine the approximation error of the function $f(x; x_c, \alpha))$, we generated 100 random searcher strategies such that each strategy has 9 right actions, 4 left actions and 10 stay actions in a random order. For each strategy, we computed the searcher's true belief, given that after executing these strategies the target is not captured. For given x_c and α, we compute the mean approximation error between the approximate and true belief as shown in Fig. 13-(b).

Using this approximation, we can reduce the number of parameters required to represent B from N entries to only two i.e. α and x_c.

The transition probability matrix: In the following, we will show how to compute the state transition probability matrix with entries of the form $P(s_i, s_j, a)$. Suppose that the current state is $s_i = ([p_0, p_1, ..., p_N], E, c)$. If by performing action a the robot runs out of energy, the next state will be $s_{no-energy}$ with probability one. That is, $P(s_i, s_{no-energy}, a) = 1$.

Next, consider transitions to the capture state. Suppose that the energy E is enough for action $a \in \{\text{stay, left, right}\}$. Also assume that $2 \leq c \leq N-2$ (the other cases are similar), then we have

$$P(s_i, s_{capture},\ \text{left}) = 0 \tag{25}$$
$$P(s_i, s_{capture},\ \text{stay}) = p \cdot p_{c+2} \tag{26}$$
$$P(s_i, s_{capture},\ \text{right}) = (s+p)p_{c+2} + p \cdot p_{c+3}. \tag{27}$$

Note that $p_c = 0$ and $p_{c\pm1} = 0$. The searcher's state will be $s' \neq s_{capture}$ with probability $(1 - P(s, s_{capture}, a))$ where $s' = ([p'_0, p'_1, ..., p'_N], E', c')$, c' is $c-1$, c or $c+1$ for the actions left, stay and right respectively. The belief $B' = [p'_0, p'_1, ..., p'_N]$ is obtained from:

$$B' = \begin{bmatrix} s+p & p & 0 & 0 & \dots & 0 \\ q & s & p & 0 & \dots & 0 \\ 0 & q & s & p & \dots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & q & s & p \\ 0 & 0 & \dots & \dots & q & s+q \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p_N \end{bmatrix} \tag{28}$$

Recall p, s, q are probabilities of the target moving left, staying in place, and moving right respectively. Since the target will be captured if it crosses the searcher or if they become neighbors, $p'_{c'}$ and $p'_{c'\pm1}$ must be set to zero. Then the next state's belief B' is obtained after normalizing B'.

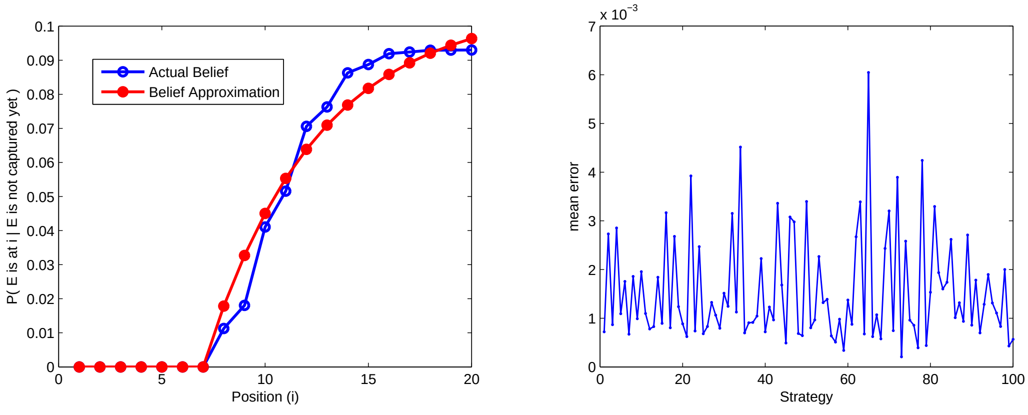
We observe that $f(x; x_c, \alpha)$ closely approximates not only the current belief, but also the belief obtained in successive states after applying the state transition probabilities.

Observation 6.1. *Suppose that the searcher is currently sitting at location i and the current belief vector is represented by a function $f(x; x_c, \alpha)$ of the form given by Eqn. 24 (Fig. 13-(a)). Then, the belief vector in the next state can be closely approximated by another function of the same form $f(x; x_c, \alpha)$.*

The reward matrix:

As expressed in Eqn. 4, we are looking for the strategy that maximizes the probability of capture within a limited time T. In order to associate the value of MDP states to 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.

In order to incorporate the time constraint (equation 19) we use finite horizon MDP for finding the optimal action at time step $k \leq T$.



(a)

(b)

Figure 13: The approximation of the belief using functions of the form $f(x; x_c, \alpha)$ given in Eqn. 24. (a) The actual belief and its approximation after performing searcher strategy $R^2SRL^2S^5R^2LRS^2RSLSR^2$. (b) The mean error of belief approximation for 100 random strategies (9 right, 4 left, and 10 stay actions in random order). Here $N = 20$, $p = q = 0.5$.

7 Simulations

In this section we present the results of the MDP formulation, studying three different setups: All Sunny, All Shadow, Arbitrary Solar Map. In the first two setups, the power gain is constant everywhere and an optimal strategy can be still obtained by means of theoretical considerations. For this reason we use them as test cases to check the MDP solution. At the same time, these case are also useful to better understand the searching process in the presence of harvested energy. In the third setup, the most challenging and interesting from a practical point of view, the solar gain can vary along the line. Before presenting the strategies, let us denote the optimal strategies with and without energy harvesting by OPT^* and OPT respectively.

For simplicity we set movement speed for this section as $1m/s$, which is a bit slower than the $1.28m/s$ our robot can reliably navigate at. However the cost per second is still the same.

In the following simulations, we used the finite horizon MDP implementation from INRA MDP MATLAB Toolbox [2]. The simulations are done on Dell Poweredge 6950 with $14GB$ memory and 8 processors. The current implementation is limited by the memory available to the simulator in order to store the probability transition matrix. The required memory space is determined by the total number of MDP states. The number of states in the following setups is around $4e10$ which is the maximum number of states our current memory resources allow for storage. The maximum running time of these simulations was 312 minutes.

	OPT	OPT^*
All Sunny	0.33	0.95
All Shadow	0.33	0.54
Realistic Solar Map	0.46	0.96

Table 3: Probability of capture with and without harvesting.

7.1 All Sunny

In this setup, the current gain is assumed to be constant and equal to $1.526A$ at all nodes. This is as much solar current as was received in the sun in the third data set at Lake Como (see Section5.4). The current draw for action stay is $c_s = 0.2A$ and $c_m = 5.7A$ for actions left or right. The number of points is $N = 20$ where the distance between points is $20m$ modeling an environment of length $400m$. The initial energy E_0 is enough for 1.33 minutes of movement with speed $1m/s$ while the time horizon T is set to 23.33 minutes. Since each time step is 20 seconds, the time horizon T can be expressed as 70 time steps.

As previously stated, in this case the optimal solution can be obtained from theoretical considerations. Indeed, thanks to the solar energy, the task duration is limited only by the time constraint and not by the selected strategy. This means that now, the stay action has the advantage only to harvest more energy and not, contrary to what happened in Section 4, to extend the searching. So, to get the optimal strategy we can use the result obtained in Proposition 4.1. As a result, in our case, the searcher has to stop only when it has no more energy to move and then it waits until the harvested energy is enough to allow it to move another step. This process continues until the maximum searching time is reached.

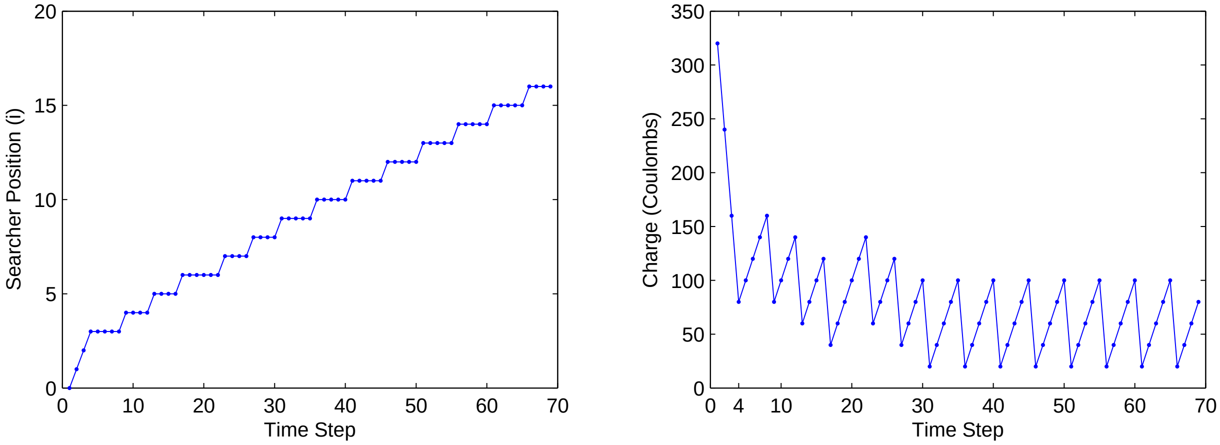
Fig. 14 shows the searcher's position and charge [3] as it performs the OPT^* strategy. We approximated the charge required for one right action as 80 Coulombs and the charge gain from one stay action as 20 Coulombs, because these were convenient numbers for discretization and close to the true (estimated) values of 83.48 Coulombs and 26.52 Coulombs. From Fig. 14-(b), it is possible to see that the initial charge is enough for 3 right actions. The OPT^* strategy confirms our previous intuition: the searcher must move to the right until it has no energy for more right actions, then stay in order to gain enough energy for only one right action and so on.

Without harvesting energy from sun, the OPT strategy is to stay until energy is depleted entirely i.e. no right action. The reason is that moving is significantly costlier that station keeping, and thus

[3] Note that charge and energy are directly proportional, related by the battery voltage which is around 12.5 volts and assumed constant. It is the same proportional relationship between current and power.

the optimal strategy without harvesting turns out to be to wait at the initial location.

The two strategies OPT and OPT^* are compared in terms of probability of capture in Table 3. Intuitively, by energy harvesting the searcher can move more and hence increase the chance of capture.



(a)

(b)

Figure 14: All Sunny Map. (a) The searcher's position as it performs the OPT^* strategy. (b) The searcher's charge as it performs the OPT^* strategy.

7.2 All Shadow

Similar to the All Sunny case, the current gains are the same everywhere. However in this case the gain is lower: $0.3023A$ (the same gain as in the shade in the third Lake Como set). The remaining parameters are the same as the All Sunny case. In particular, the charge required for one right action is 108 Coulombs while the charge gain from one stay action is only 2 Coulombs (these are both the nearest integer number of Coulombs to the true estimates). Also the initial charge E_0 is enough for 3 right actions. The searcher's position and charge as it follows the OPT^* strategy are shown in Fig. 15. Similar to the All Sunny case, in OPT^* strategy the searcher moves to the right until it has no more charge for another right action. Then it waits until enough charge for only one right action is gained, uses that charge for one right action and so on.

Without harvesting energy the optimal strategy OPT is to stay at the first point for the entire time horizon. As shown in Table 3 the probability of capture is higher in OPT^*.

7.3 Realistic Solar Map

Finally we examined the case where power gain varies at different points along the line. We used varying solar current values derived from the solar maps constructed for Lake Como, see Figure 12. For this planning we obtained the expected solar current at a point by sampling around a circle of 5m radius to account for the navigation and localization error of a real robot executing our strategy. These current gains obtained during moving and station keeping are shown in Fig. 16. Here $N = 30$, where distance between points is $10m$. The initial energy E_0 is enough for 1.67 minutes of movement (equivalent to 10 right actions) with speed $1m/s$ while the time horizon T is set to 16.7 minutes. Note that each time step is 10 seconds and so the time horizon can be stated as 100 time steps. Also recall that costs of moving and stationary keeping are $c_m = 5.7A$ and $c_s = 0.2A$ respectively.

The searcher's position and charge as it performs the OPT^* strategy obtained from MDP approach are shown in Fig. 17. The searcher first moves to the right for 10 steps i.e. until there is not enough energy for another right action. Then it stays at location $x = 11$ from time step $t = 12$ to $t = 30$. Note that in contrast to what we saw in All Sunny and All Shadow cases, here the gained energy is for more than one right actions (in All Sunny/All Shadow the gained energy was enough for only one more right action). This is because the power gain at location $x = 11$ is higher than the points ahead ($x > 11$). See Fig. 16-(a). For the same reason, the energy that the searcher gains is enough for more than one

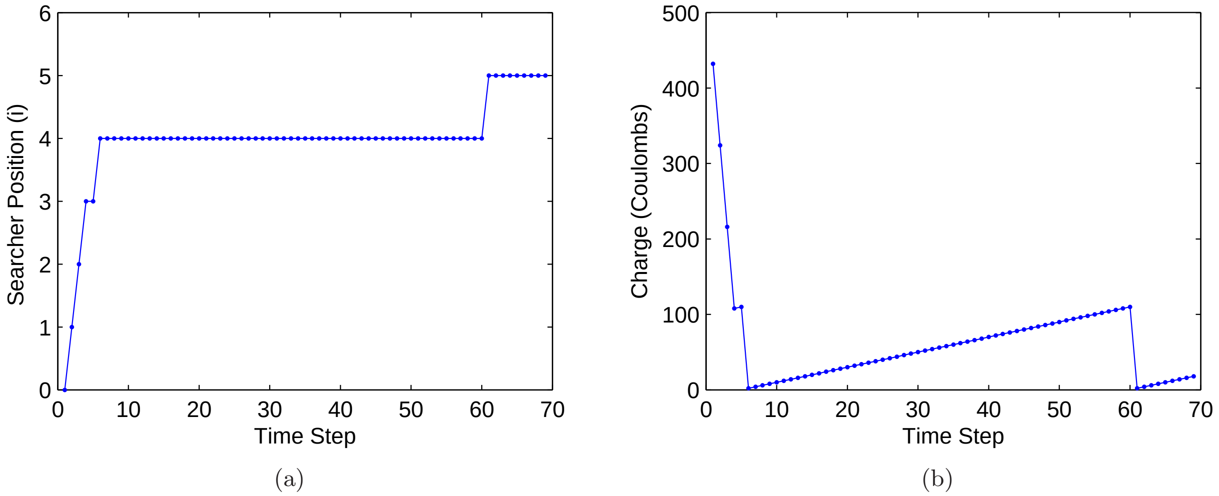


Figure 15: All Shadow Map. (a) The searcher's position as it performs the OPT^* strategy. (b) The searcher's charge as it performs the OPT^* strategy.

right action at time steps $t = 34$, $t = 62$, and $t = 74$ corresponding to locations $x = 15$, $x = 20$ and $x = 22$ respectively.

Without energy harvesting, the optimal strategy OPT is to move to the right for 6 steps and then stay. Intuitively the probability of capture is much lower than the OPT^* strategy (Table 3).

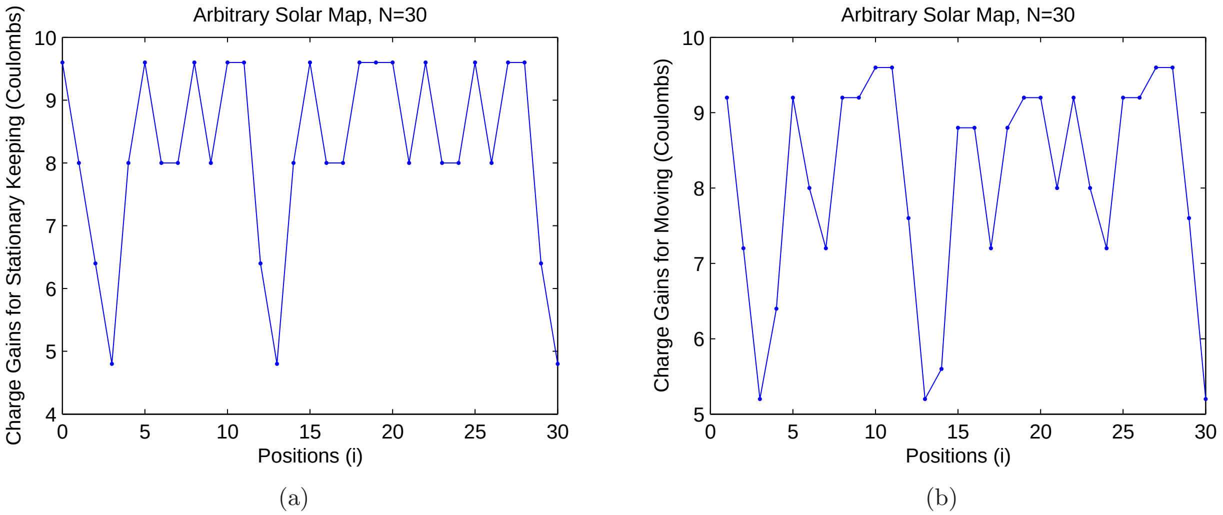
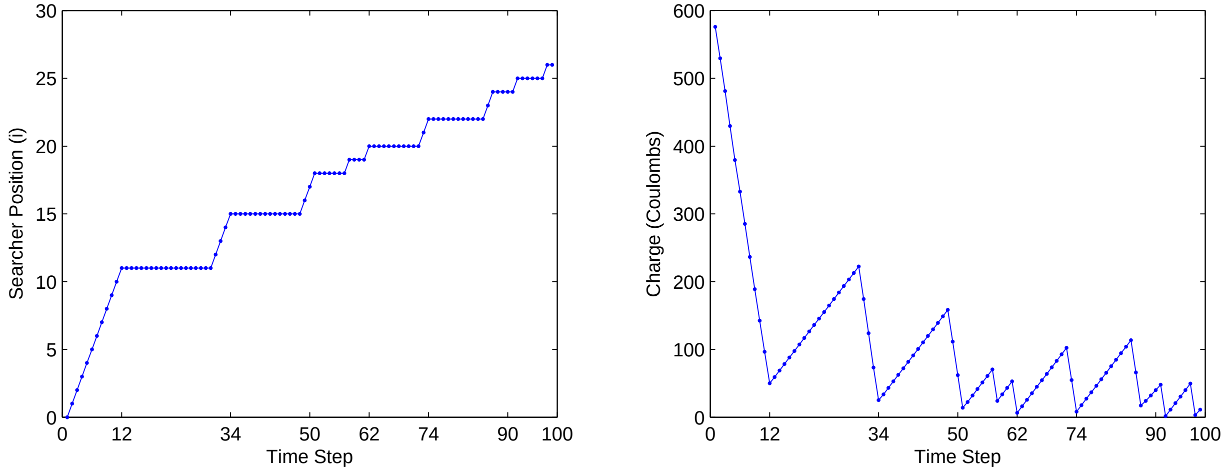


Figure 16: Arbitrary Solar Map. (a) The charge gains during stationary keeping (action stay). (b) The charge gains during movement (actions left or right).

8 Conclusion

In this paper, we studied a novel search problem where the goal is to maximize the probability of finding a target in a given amount of time. Our formulation addresses energy constraints to enable long-term autonomy: there is a cost associated with station keeping or moving and the robot has a fixed energy budget. We studied this problem for one dimensional environments (inspired by the application of finding radio tagged fish along the shore of a lake with an autonomous surface vehicle.) We presented a closed form solution for finding the optimal strategy for finding a random walker. Next we investigated how the robot's operation time and, hence, the probability of finding the target can be improved by



(a)

(b)

Figure 17: (a) The searcher's position as it performs the OPT^* strategy. (b) The searcher's charge as it performs the OPT^* strategy.

harvesting solar energy. We presented a method for estimating the expected solar gains at an arbitrary point on a lake, and used it for computing the optimal search strategy using a Markov Decision Process (MDP) formulation.

We have two immediate venues for future work. While the MDP framework yields optimal solutions, it is computationally very intensive. More efficient algorithms for computing optimal or near-optimal solutions is desirable. We are also interested in solving the search problem in more general environments such as two dimensional environments with obstacles.

References

[1] Advanced telemetry systems. `http://www.atstrack.com/index.aspx`. Accessed November, 2012.

[2] Mdp matlab toolbox. `http://www.inra.fr/mia/T/MDPtoolbox/`. Accessed November, 2012.

[3] Oceanscience. `http://www.oceanscience.com/`. Accessed November, 2012.

[4] Solartech power. `http://www.solartechpower.com`. Accessed November, 2012.

[5] M. Adler, H. Racke, N. Sivadasan, C. Sohler, and B. Vocking. Randomized pursuit-evasion in graphs. *Combinatorics Probability and Computing*, 12(3):225–244, 2003.

[6] J. Anlauf. Asymptotically exact solution of the one-dimensional trapping problem. *Physical review letters*, 52(21):1845–1848, 1984.

[7] P. G. Bajer, H. Lim, M. J. Travaline, B. D. Miller, and P. W. Sorensen. Cognitive aspects of food searching behavior in free-ranging wild Common Carp. *Environmental Biology of Fishes*, 88(3):295–300, mar 2010.

[8] T. Chung, G. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics. *Autonomous Robots*, 31(4):299–316, 2011.

[9] 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*, 2012(05):P05011, 2012.

[10] D. Y. Goswami, F. Kreith, and J. F. Kreider. *Principles of Solar Engineering*. Taylor & Francis, 2nd edition edition, 1999.

[11] P. Krapivsky and S. Redner. Life and death in an expanding cage and at the edge of a receding cliff. *American Journal of Physics*, 64(5):546–551, 1996.

[12] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46, 1993.

[13] P. Plonski, P. Tokekar, and V. Isler. Energy-Efficient Path Planning for Solar-Powered Mobile Robots. *Journal of Field Robotics (in review)*, 2012.

[14] L. Ray, J. Lever, A. Streeter, and A. Price. Design and Power Management of a Solar-Powered Cool Robot for Polar Instrument Networks. *Journal of Field Robotics*, 24(7):581–599, 2007.

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

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

[17] C. Sauze and M. Neal. Long term power management in sailing robots. In *OCEANS, 2011 IEEE - Spain*, pages 1 –8, june 2011.

[18] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series. Mit Press, 1998.

[19] P. Tokekar, D. Bhadauria, A. Studenski, and V. Isler. A robotic system for monitoring carp in minnesota lakes. *Journal of Field Robotics*, 27(6):779–789, 2010.

[20] 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*, 2012. to appear.

[21] P. Tompkins, A. Stentz, and D. Wettergreen. Mission-level path planning and re-planning for rover exploration. *Robotics and Autonomous Systems*, 54(2):174–183, 2006.