Multi-Target Search Strategies

Malika Meghjani, Sandeep Manjanna and Gregory Dudek

***Abstract*— This paper addresses the problem of searching multiple non-adversarial targets using a mobile searcher in an obstacle-free environment. In practice, we are particularly interested in marine applications where the targets drift on the ocean surface. These targets can be surface sensors used for marine environmental monitoring, drifting debris, or lost divers in open water. Searching for a floating target requires prior knowledge about the search region and an estimate of the target's motion. This task becomes challenging when searching for multiple targets where persistent searching for one of the targets can result in the loss of other targets. Hence, the searcher needs to trade-off between guaranteed and fast searches. We propose three classes of search strategies for addressing the multi-target search problem. These include, data-independent, probabilistic and hybrid search. The data-independent search strategy follow a pre-defined search pattern and schedule. The probabilistic search strategy is guided by the estimated probability distribution of the search target. The hybrid strategy combines data-independent search patterns with a probabilistic search schedule. We evaluate these search strategies in simulation and compare their performance characteristics in the context of searching multiple drifting targets using an Autonomous Surface Vehicle (ASV).**

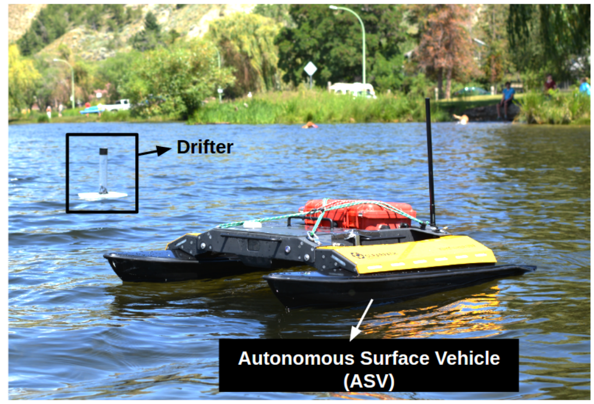


Fig. 1: Autonomous Surface Vehicle searching for passively floating drifter.

I. Introduction

This paper deals with the problem of search, specifically robotic search in a natural environment for a set of targets of interest whose position is represented only by an estimated search region or probability distribution. This problem has myriad applications, in fact search for compatible targets is one of the single most critical task to be accomplished by almost any living organism. In some applications the targets being searched for can either actively participate in the search process for a rendezvous, or can be actively seeking to evade from the persuader. A complementary example to active target search is search for passively drifting target where all we know at the outset is a probability for the target's location and a motion model of the target. It is this latter case that we refer to as passive non-adversarial search, which we address in this paper.

Searching for lost non-adversarial targets in natural environments is a challenging task, given the absence of landmarks to guide the search process. Some evocative applications include, searching for lost hikers in the forest and search for passengers, in a disaster scenario, who may have escaped in lifeboats from a sinking ship. Our work is aimed to a specific marine data collection application whereby a robotic boat must find a set of drifting data-collection beacons each of which has only a limited radio transmission range (as illustrated in Fig. 1). The searcher for these passively moving targets, can exploit a priori information such as the expected target motion (at the least a maximum speed) and an estimate of the initial probability distribution. These two parameters define a (growing) region within which the target can be found. This prior information can be combined to define the parameters of either a time-minimizing or a guaranteed search strategy (which are not necessarily the same thing). The search strategy can be tuned to account the importance of the target. For example, if the target is a human, the search strategy needs to be fast and guaranteed whereas searching for passively floating sensors a different optimization criteria may be paramount.

This search problem becomes more complex when there are multiple targets to be found. In such cases, different algorithms can include the desire to maximize the chance of finding at least one target, the chance of finding the maximum number of targets, or the desire to find as many targets as possible within a given time or distance budget. While some provable performance bounds are possible for several idealized variants of the multi-target search problem, we focus here on an experimental assessment of algorithms that can operate in the face of almost-arbitrary initial probability distributions for the targets, weak assumptions on the possible target motion, and reasonable (albeit idealized) model of short-range radio detection. We thus propose three classes of heuristic search strategies: data-independent, probabilistic and hybrid searches. Each of these strategies along with their algorithms are elaborated in Section IV. A performance comparison between these strategies is presented in Section V based on controlled simulations.

The authors are with the Centre for Intelligent Machines,
McGill University, Montréal, Québec, Canada.
email:{malika, msandeep, dudek}@cim.mcgill.ca

II. Related work

Search strategies for robotics applications are predominantly adapted from game theory where simple geometric environments such as straight lines and circles are studied [1]. The search strategies for natural environments without unique landmarks are inspired by biological process such as animals foraging food [2]. The most efficient search pattern reported in robotic search literature are spirals [3], [4] which minimize the time to find a single stationary "lost" target in several interesting idealized scenarios. In this paper, we extend the deterministic spiral strategy for multiple targets and compare it with probabilistic and hybrid strategies.

Gonzalez et al. [5] also proposed spiral patterns for efficient coverage of a search region. They suggested that structured patterns such as spiral are generally more efficient and robust to initial location of the searcher when compared to unstructured search patterns such as Brownian motion. Meghjani et al. [6] implemented spiral patterns on an Autonomous Underwater Vehicle (AUV) using the sawtooth motion to find passively drifting targets. They also provided a theoretical analysis for generating a spiral path for the AUV such that it guarantees the capture of the target under a given set of conditions. Bourgault et al. [7] and Furukawa et al. [8] probabilistically represented the search region and proposed an optimal controller for search. The result of their controller emerged to be a spiral search pattern.

Hollinger et al. [9] studied the problem of finding a non-adversarial target with multiple robots. They suggested a finite-horizon planner in which a searcher would plan for a finite number of cells and choose the best path to the horizon. This is an iterative online process and the searcher could incorporate the observations into the planner. For multi-robot coordination, the robots sequential choose their paths and centrally maintain a list of visited nodes. The nodes are allowed to be revisited and updated by other robots. The complexity of this sequential planning grows linearly with the number of robots.

There are other search strategies which update the search patterns based on online observations. One such example search problem is addressed by Saigol et al. [10]. In this work, the authors propose an automated planning algorithm to deal with uncertainty in searching for hydrothermal vents. They suggested an information lookahead and entropy change planners. The information lookahead is based on a Partially Observable Markov Decision Process (POMDP) formulation which was reported to be computationally expensive. Whereas, the entropy change maximization method requires a continuous feedback based on the probabilities of the observing chemical traces. In our work, we do not have such observations available to track and find the target.

Singh et al. in [11] have also looked at the problem of environmental sensing with multiple robots. They address the problem of maximizing the information gathered by all the agents while constraining on the amount of resources used. They use mutual information to analyze the quality of the robot's paths. A cost is associated with each path and is defined as the sum of sensing and traveling costs. Their goal is to find a set of paths (one for each robot), with minimum cost and maximum mutual information. Such a method is useful if the underlying target distribution is continuous and can help in analyzing the quality of the paths. In our problem formulation, the target distribution is discrete and we have a single searcher.

Blum et al. studied orienteering and discounted-reward *Traveling Salesman Problem* (TSP) graph problems in [12]. The graph in their work is represented by edges with lengths and nodes with rewards. In the orienteering problem, the goal is to find a path starting at a node that maximizes the reward collected subjected to constraints on the total path length. In the discounted reward TSP, the goal is to maximize the total discounted reward collected. The latter method is based on an approximation planning algorithm in Markov Decision Process (MDP) with infinite horizon discounted reward. The approximation algorithm is used for modeling one-time events and non-repeatable rewards which is similar to our Hybrid Search algorithm presented in this paper.

III. Problem Formulation

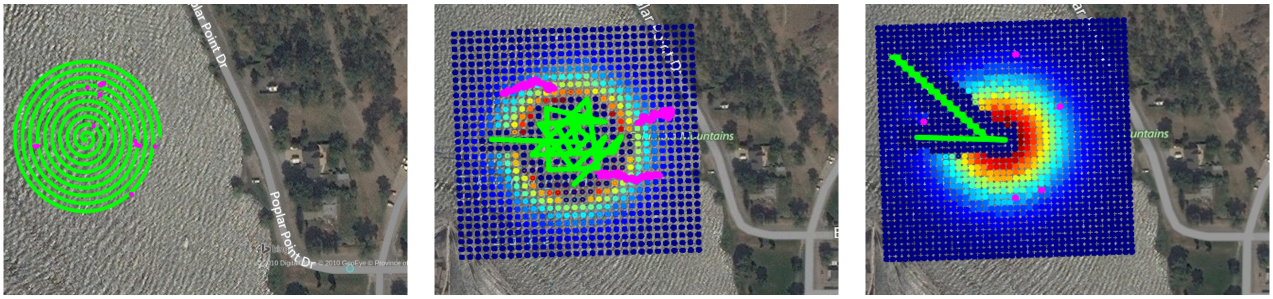
We propose search strategies that can be executed by a point-robot looking for arbitrary number of lost point-targets which move in an unpredictable manner in an obstacle-free environment and can be reliably detected if they are within the searcher's communication radius. The searcher's communication radius is hence, considered as the detection radius for the target. The searcher actively executes a search algorithm while the target is passively moving or stationary. In our problem formulation, we do not account for dynamically emerging search patterns which would require emergence of a helical search path from a spiral. Such a search strategy is discussed in our previous work [6], where an AUV is searching for a drifting target.

IV. Search Strategies

Our proposed strategies comprise, a search pattern and a search schedule. The search pattern provides the search path and the search schedule suggests the transitions between the targets for searching. The search patterns are derived from our previous work on search for single target [13], where we evaluated the search patterns using an Autonomous Surface Vehicle (ASV) (Fig. 1) to search for a single drifting target. In this paper, we present simulation results for performance comparison of the three proposed strategies in the context of searching multiple targets using an ASV.

A. Data-independent Search

We define data-independent search strategy as the strategy that have a pre-defined search pattern and search schedule to shuttle between multiple targets. The search pattern that we selected is spiral, as we have empirically and theoretically illustrated it to be an energy efficient and minimum failure rate search strategy for a single target search [13]. The spiral pattern is generated taking into account the communication radius and speed of the searcher and the target,



(a) Data-independent (b) Probabilistic: Global (c) Probabilistic: Local

Fig. 2: Search path of the robot (green) for single target. Multiple instances of a single target with Brownian like motion are illustrated by pink dots. The pink traces indicate the trajectories followed by the target over time. The probabilistic local search strategy is stuck in the local maxima Fig. 2(c).

as represented in Fig. 2(a) for a single target. The search schedule for multi-target, data-independent, search strategy has deterministic transition times between targets such that the searcher switches between the target search regions at regular intervals.

B. Probabilistic Search

The probabilistic search strategy is guided by the underlying estimate of the probability distribution of the target. We propose three algorithms for covering the target probability. These include global-maxima, local-maxima and greedy decision maker.

1) Global-Maxima: Given an initial target probability distribution, the global-maxima search strategy aims at clearing parts of the search region with highest probabilities. The search region is discretized into grids and each grid-cell is assigned with a value equal to the integral of the probability under that grid-cell. Global-maxima search strategy generates a trajectory that visits the grid-cell with highest value and clears that grid-cell once visited, then continues to visit the next maxima until the target is found or the search region is covered. This strategy provides a trajectory with multiple overlapping segments as the global-maxima shifts across the search region as shown in Fig. 2(b) for a single target. The target transitions happen naturally by following the global-maxima of the distribution. This strategy is prone to frequent transitions between the targets and hence results in longer capture times.

2) Local-Maxima: The local-maxima strategy, sequentially clears the grid-cells with maximum values within a local maxima-search radius. This search strategy is dependent on the initial location of the robot and is prone to getting stuck in the local maxima, not being able to transition between targets as illustrated in Fig. 2(c) for single target. As a consequence, this strategy does not allow transition between the targets. The success rate with this strategy is observed to be extremely low in our simulated trials for a single target.

3) Greedy Decision Maker: The aforementioned probabilistic search strategies have their respective shortcomings such as delayed capture times for global maxima and getting stuck in local maxima. We overcome these drawbacks by implementing a greedy decision maker similar to one-step MDP [12].

A MDP consists of a state space S, a set of actions A, a probabilistic transition function P, and a reward function R. In our formulation, we discretize the search region into $50\text{X}50$ grid cells representing the state space S and each of these cells is considered as a state ($s \subset S$). The action space A consists of four actions: North, South, West, East. We assume a truncated Gaussian prior distribution for the search region. The cumulative probability under each cell is considered as the reward for visiting that cell. The transition probability from state s to state s' given action $a \subset A$ is assumed to be $P(s'|s,a) = 1$ because of the use of Global Positioning System (GPS) to localize the agent and achieve a guaranteed state transition with the help of a precisely tuned controller. To avoid re-visiting the same cell, we clear the reward of the state once it is visited. The standard MDP formulation does not allow the reward function to change over time. Hence to accommodate these dynamic reward updates, we re-initialize the world at each iteration and model it as one-step MDP. In each iteration of our algorithm, we use Value Iteration algorithm to generate a policy. Once the searcher takes an action according to this policy, the reward in its current cell is cleared. The updated world is used as an initial state space for the next iteration. This is repeated until the target is found or the search probability goes below a pre-defined threshold.

We use the *Value Iteration* algorithm to compute the best action to be taken at a given state. Value iteration is a method of computing an optimal MDP policy. It computes the optimal value of a state $V^*(s)$, i.e., the expected discounted sum of rewards that the agent will achieve if it starts at that state and executes the optimal policy $\pi^*(s)$. $\forall s \in S$, the optimal value function $V^*(s)$ is defined by the following Bellman equation,

$$V^*(s) = \max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right), \quad (1)$$

where γ is a discount factor. Thus according to Eq. 1, the value of a state s is the sum of instantaneous reward and the expected discounted value of the next state, when the best available action is used. Optimal policy defines an action for every state that achieves the optimal value. Given the optimal value function for all states, optimal policy is defined by,

$$\pi^*(s) = arg\max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right) \tag{2}$$

In our algorithm we use a one step MDP approach, where we model every state transition of the agent as MDP in a new world and compute the value function over the updated rewards of the world. Thus the convergence of the value iteration technique still holds good for every state transition. The overview of our approach is presented in Algorithm 1. An extension of this algorithm was applied for creating a Bathymetric map of the underwater environment in [14].

Algorithm 1 Greedy Decision Maker

Input: Set of states S
Set of actions A
State transition probability $P(s'|s,a)$
$\forall (s,s') \in S$ and $\forall a \in A$
Reward $R(s,a)$ for each state $s \in S$
Discount factor γ
Starting state s_1
Convergence threshold R_{limit}
Convergence threshold ϵ
Output: Path $\vec{W} = (s_1, s_2, ...s_n)$, a sequence of states.

$\forall s \in S$,
  Initialize $V^*(s)$, $\pi^*(s)$, and current state $s_{cur} = s_1$
Repeat
  $\vec{W}$ = **Append**($\vec{W}, s_{cur}$)
  $\forall s \in S$,
    ($V^*(s)$, $\pi^*(s)$) = **ValueIteration**($S, A, P, R, \gamma, \epsilon$)
  Current Action, $a_{cur} = \pi^*(s_{cur})$
  ApplyAction a_{cur} on s_{cur} to obtain s_{next}
  $R(s_{cur}) = 0$, Clearing the reward at s_{cur}
  $s_{cur} = s_{next}$
until ($\sum_{s \in S} R(s) < R_{limit}$) **or** the region is fully covered.
Return $\vec{W}$

C. Hybrid Search

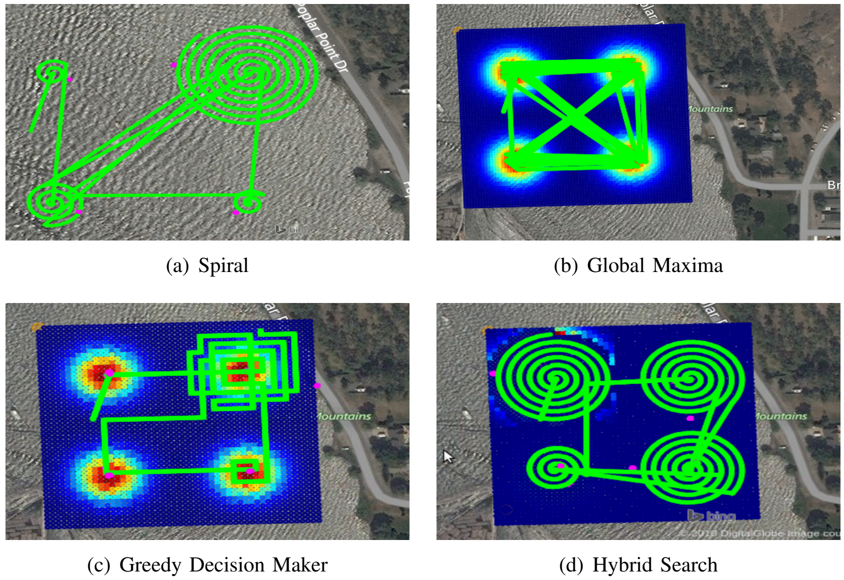
Though the data-independent search has fixed transition times between the targets it is computationally efficient and has fast execution time. The probabilistic greedy decision maker can be computationally expensive given the size of the state space however it has the natural capability of deciding on transitioning between the targets. Since the targets are drifting over time, there is a need to switch the searching between targets before all the targets escape the search region. Hence, we propose to combine the data-independent strategy with the greedy decision maker to get the best of two methods. We integrate the data-independent search pattern with the probabilistic greedy decision maker's search schedule to obtain a hybrid search strategy.

Each time data-independent search pattern completes one standard deviation of the target distribution it triggers the greedy decision maker. We adapt the one-step MDP from the greedy decision maker to decide on the next target to visit. In this strategy, each target represents a state. The transition between these states can be achieved using 9 actions: North, South, West, East, North-West, North-East, South-West, South-East and same-state. This 9-action space is generic and thus makes the model independent of the number of targets. The transition probability from state s to state s' given action $a \subset A$ is assumed to be $P(s|s', a) = 1$. We define the reward as the cumulative probability under the target distribution of the state divided by normalized distance to reach the state. This penalizes frequent hopping between multiple targets and thus minimizes the distance traveled by the searcher. We use Value Iteration method as described for Greedy Decision Maker to calculate the policy to switch between the targets.

V. CONTROLLED SIMULATION

We validated our analytical results using a real-time simulator that we developed to asses the search strategies. We preselected a search region of 100 meters radius, in Okanagan lake (Canada) for our simulations. The maximum speed of the searcher was set to $1.2\ m/s$ which is also the top speed of the boat used during the field trials for single target search. The drifter's speed was simulated to be $0.2\ m/s$ based on the actual drifter data collected during our field trials. The motion of the targets was modeled using beta distribution for the speed and uniform distribution for the direction, leading to a variant of Brownian motion. The maximum communication range between the robot and the drifters was considered to be 5 meters in radius to simulate degraded WiFi range.

The initial location of the drifter was simulated within the search region based on triangular distribution which samples target locations that are biased towards the center of the search region. In our previous work, we explored other representative target distributions such as $|x|$ and uniform for single target search [15]. The initial location of the searcher is preset to a fixed point within the search region for an unbiased comparison of all the search strategies. For the probabilistic search strategies, an underlying probability distribution is required to represent the searcher's belief about the target. In our experiments, a Gaussian distribution is chosen to represent the belief about each target, centered around their respective last known locations. Similarly, the deterministic spiral search path is also generated around the last known location of the target, based on the theoretical analysis from our previous work [6]. The spiral pattern is generated using the target speed, searcher speed and the communication radius between the searcher and the target. An instance of each of the four strategies using our real-time simulator is illustrated in Fig. 3.



(a) Spiral (b) Global Maxima

(c) Greedy Decision Maker (d) Hybrid Search

Fig. 3: Search paths of the robot (green) and the target (pink) for 4 targets.

VI. Experimental Results

The experiments presented in this section were run on the real-time simulator with a given initial target distribution and communication range. We present and discuss the results from four different search strategies. We compare the search strategies based on the performance factors such as failure rate, average distance traveled, and computation time. For each search strategy, 1000 simulations were run and the average over these trials is presented here. These simulations were executed on a computer using an Intel core i7 architecture with 6 cores (12 threads), 24 GB RAM operating at a clock speed at 3.20 GHz.

A. Failure Rate

In our experiments we consider four targets that need to be found. Finding all four targets is considered as success. The trials in which a target was missed is considered as a failed trial. The plot in Fig. 4 shows that the pure-probabilistic approaches, i.e., global-maxima search and greedy search, fail to find all the targets more frequently.

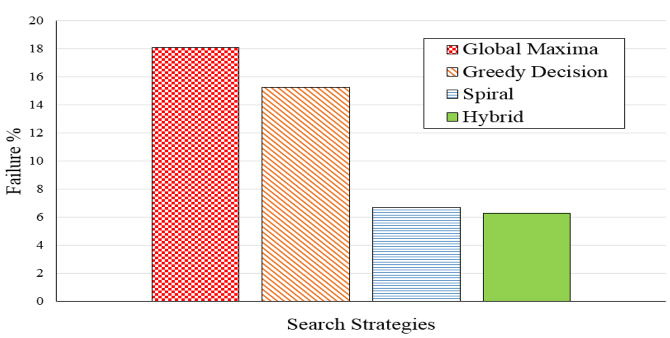


Fig. 4: Failure rate for different search strategies.

This behavior can be due to approximation of the prior distribution or due to the fact that these strategies do not cover the entire search region. They stop after their belief reaches a threshold. Whereas, the data-independent spiral search and the hybrid-spiral search strategies cover the search region fully before stopping their attempt.

B. Average Distance Traveled

If we have infinite resources and time, we can consider only the success rate as a performance measurement quality of a search strategy. However in practice, this is not true. Humans or robots, both have limited energy and time. Hence, we consider the distance traveled by the searcher using different search strategies as a performance measure for comparison. The plot in Fig. 5 illustrates the average distance traveled by all the four search strategies while searching for the targets.

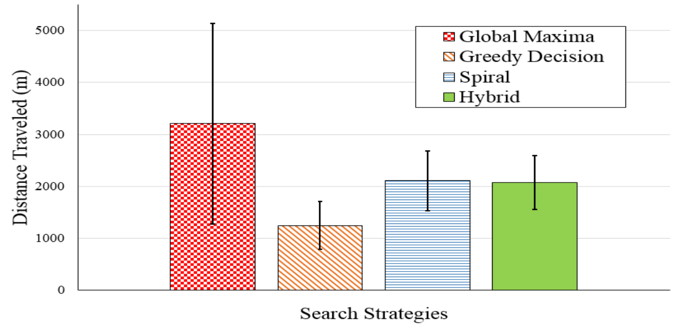


Fig. 5: Mean distance traveled to find the target using difference search strategies. The error bars represent two-sided standard deviation.

Global-maxima search on average travels the most to achieve its goal. This can be attributed to its nature of hopping between multiple targets to search the areas with highest cumulative probability. Whereas, MDP-based greedy decision maker travels the least as it weighs the local rewards higher, and jumps to next target only when local rewards are cleared. Both spiral search and hybrid search have similar behaviors as we selected the target transition times for the spiral search based on the hybrid search strategy. In absence of the target transition time information from the hybrid search strategy, the deterministic spiral search strategy would either have higher failure rate due to insufficient target transitions or larger distance traveled due to frequent transitions.

C. Computation Time and Complexity

The results of average distance traveled imply that the greedy decision search strategy is the best with the least distance traveled. However, the average computation times plotted in Fig. 6 shows that the greedy decision strategy is computationally very expensive and is not feasible to run on any real-time system for any real-time search operation. On the contrary, the hybrid-search strategy performs very effectively even though it has similar decision making mechanism as greedy decision strategy. This is due to the reduced size of the state space. The state space for greedy decision strategy is as big as the search region, but the state space for hybrid strategy is equal to the number of targets. In this plot, we truncate the bar plot for greedy strategy because of its large computation time compared to other strategies. Their numerical values are presented in Table I.

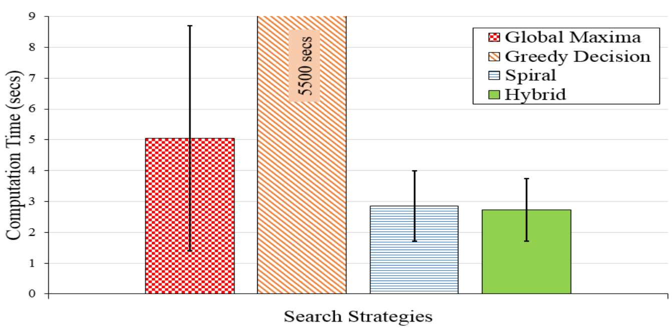


Fig. 6: Computation time for different search strategies. The error bars represent two-sided standard deviation.

Computational Complexity			
Global-Maxima	Greedy-Decision	Spiral	Hybrid
$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$
Computational Time (secs)			
Global-Maxima	Greedy-Decision	Spiral	Hybrid
5.1	5500	2.95	2.8

TABLE I: Computational complexity and computational time for search strategies.

Table I also presents the computational complexity for all the four search strategies, where n represents the size of the world. The Hybrid strategy has a constant computational complexity because it covers the whole search region in terms of standard deviations of the underlying Gaussian distribution. The proposed hybrid algorithm searches for only first three standard deviations of the Gaussian distribution as it covers 99% of the distribution and thus the search region.

VII. Conclusion

In this paper, we present three classes of search strategies for multiple targets, namely data-independent, probabilistic and hybrid strategies. We compared their performance in terms of their success rate, mean distance traveled and computation time. We found the data-independent and hybrid search strategies to have the best overall performance. The data-independent strategy however has the inherent drawback of fixed transition schedule to visit the search targets. Whereas, the hybrid strategy is capable of deciding on the transition schedule in real-time based on expected rewards.

For future work, we would like to test hybrid strategy with different priors on target distributions, drifter initial location distribution and target transition time conditions. In addition, we would like to study the emerging search patterns for dynamic targets which do not follow Brownian motion.

References

[1] Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous*, volume 55. Springer Science & Business Media, 2006.

[2] E. Gelenbe, N. Schmajuk, J. Staddon, and J. Reif. Autonomous search by robots and animals: A survey. *Robotics and Autonomous Systems*, 22(1):23–34, 1997.

[3] S. Burlington and G. Dudek. Spiral search as an efficient mobile robotic search technique. In *Proceedings of the 16th National Conf. on AI, Orlando Fl.* Citeseer, 1999.

[4] Ricardo A Baezayates, Joseph C Culberson, and Gregory JE Rawlins. Searching in the plane. *Information and computation*, 106(2):234–252, 1993.

[5] E González, P Aristizábal, and M Alarcón. Backtracking spiral algorithm: a mobile robot region filling strategy. In *Proceeding of the 2002 international symposium on robotics and automation, Toluca, Mexico*, pages 261–266, 2002.

[6] Malika Meghjani, Florian Shkurti, Juan Camilo Gamboa Higuera, Arnold Kalmbach, David Whitney, and Gregory Dudek. Asymmetric rendezvous search at sea. In *CRV '14: Proceedings of the 2014 Canadian Conference on Computer and Robot Vision*, pages 175–180. IEEE, 2014.

[7] Frédéric Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. Optimal search for a lost target in a bayesian world. In *Field and service robotics*, pages 209–222. Springer, 2006.

[8] Tomonari Furukawa, Frederic Bourgault, Benjamin Lavis, and Hugh F Durrant-Whyte. Recursive bayesian search-and-tracking using coordinated uavs for lost targets. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2521–2526. IEEE, 2006.

[9] Geoffrey Hollinger and Sanjiv Singh. Proofs and experiments in scalable, near-optimal search by multiple robots. *Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland*, 1, 2008.

[10] Zeyn A Saigol. *Automated planning for hydrothermal vent prospecting using AUVs*. PhD thesis, University of Birmingham, 2011.

[11] Amarjeet Singh, Andreas Krause, Carlos Guestrin, William J Kaiser, and Maxim A Batalin. Efficient planning of informative paths for multiple robots. In *IJCAI*, volume 7, pages 2204–2211, 2007.

[12] Avrim Blum, Shuchi Chawla, David R Karger, Terran Lane, Adam Meyerson, and Maria Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 46–55. IEEE, 2003.

[13] Malika Meghjani, Sandeep Manjanna, and Gregory Dudek. Multi-target rendezvous search. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[14] Sandeep Manjanna, Kakodkar Nikhil, M. Meghjani, and G. Dudek. Efficient terrain driven coral coverage using gaussian processes for mosaic synthesis. In *CRV '16: Proceedings of the 2016 Canadian Conference on Computer and Robot Vision*. IEEE, May 2016.

[15] Malika Meghjani and Gregory Dudek. Search for a rendezvous with lost target at sea. In *ICRA Workshop on Persistent Autonomy for Aquatic Robotics, 2015.* IEEE, 2015.