

The International Journal of Robotics Research

<http://ijr.sagepub.com/>

Efficient Multi-robot Search for a Moving Target

Geoffrey Hollinger, Sanjiv Singh, Joseph Djugash and Athanasios Kehagias

The International Journal of Robotics Research 2009 28: 201

DOI: 10.1177/0278364908099853

The online version of this article can be found at:

<http://ijr.sagepub.com/content/28/2/201>

Published by:



<http://www.sagepublications.com>

On behalf of:



[Multimedia Archives](#)

Additional services and information for *The International Journal of Robotics Research* can be found at:

Email Alerts: <http://ijr.sagepub.com/cgi/alerts>

Subscriptions: <http://ijr.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>

Citations: <http://ijr.sagepub.com/content/28/2/201.refs.html>

Geoffrey Hollinger
Sanjiv Singh
Joseph Djughash

Robotics Institute,
Carnegie Mellon University,
Pittsburgh, PA 15217,
USA
{gholling, ssingh}@ri.cmu.edu, josephad@ri.cmu.edu

Athanasios Kehagias

Division of Mathematics,
Department of Mathematics, Physics,
and Computer Sciences
Aristotle University of Thessaloniki,
Thessaloniki GR54124,
Greece
kehagiat@gen.auth.gr

Efficient Multi-robot Search for a Moving Target

Abstract

This paper examines the problem of locating a mobile, non-adversarial target in an indoor environment using multiple robotic searchers. One way to formulate this problem is to assume a known environment and choose searcher paths most likely to intersect with the path taken by the target. We refer to this as the multi-robot efficient search path planning (MESPP) problem. Such path planning problems are NP-hard, and optimal solutions typically scale exponentially in the number of searchers. We present an approximation algorithm that utilizes finite-horizon planning and implicit coordination to achieve linear scalability in the number of searchers. We prove that solving the MESPP problem requires maximizing a non-decreasing, submodular objective function, which leads to theoretical bounds on the performance of our approximation algorithm. We extend our analysis by considering the scenario where searchers are given noisy non-line-of-sight ranging measurements to the target. For this scenario, we derive and integrate online Bayesian measurement updating into our framework. We demonstrate the performance of our framework in two large-scale simulated environments, and we further validate our results using data from a novel ultra-wideband ranging sensor. Finally, we provide an analysis that demonstrates the rela-

tionship between MESPP and the intuitive average capture time metric. Results show that our proposed linearly scalable approximation algorithm generates searcher paths that are competitive with those generated by exponential algorithms.

KEY WORDS—multi-robot coordination, autonomous search, approximation algorithms, range-only sensing, decentralized computation, POMDPs

1. Introduction

The problem of searching for a mobile target in an indoor environment is one that is relevant to many real-world scenarios. Military and first response teams often need to locate lost team members or survivors in disaster scenarios. The increasing use of search and rescue robots and mechanized infantry necessitates the development of algorithms for autonomously searching such environments. The major application that has motivated our work is that of locating a lost first responder in an indoor environment (Kumar et al. 2004). In this application, a moving first responder is lost during disaster response, and a team of robots must locate the first responder. Noisy non-line-of-sight measurements may or may not be available to assist in the search. A similar scenario arises if a group of ground vehicles must locate a target on a road network while an air vehicle provides surveillance.

The coordinated search problem can be divided into three main categories: efficient search, guaranteed search and constrained search. *Efficient search* seeks to locate (capture) a

non-adversarial target in the minimum expected time. A target can be considered non-adversarial if its motion model is independent of the state of the searchers. High-quality searcher paths in the efficient search domain are those that on average lead to quickly locating a non-adversarial target. A second subproblem of coordinated search is *guaranteed search*, where searchers seek to clear an environment to ensure that even an adversarial target cannot escape capture¹. Guaranteed search may additionally seek to minimize capture time, which leads to a multi-objective optimization problem. Finally, *constrained search* performs either efficient or guaranteed search while maintaining some constraints on the searchers (e.g. remaining within communication range). This paper presents a framework for solving the efficient search problem, and related work is discussed in other coordinated search domains.

If non-line-of-sight measurements to the target are available, the efficient search problem is one of both estimation and control. For instance, searchers may receive the range to the target from ultra-wideband ranging radios capable of providing the distance between nodes through walls. If this information is properly utilized, better estimation can lead to better search paths. However, perfect estimation is useless without good control strategies. Our algorithm integrates both estimation and control into a unified efficient search framework by using probabilistic planning alongside Bayesian measurement updating.

The efficient search problem can be formulated as a partially observable Markov decision process (POMDP). The full state of the world is defined by the union of the searcher positions and the target position. A solution to the POMDP provides a mapping from the estimated state of the target to searcher actions. The target's position is partially observed, and observations are provided by measurements received by the searchers. In the formulation without non-line-of-sight measurements, the knowledge that the target is not at the same location as the searchers provides information. In the POMDP formulation, the target's motion model is assumed to be Markovian (i.e. it depends solely on the target's current location in the environment and does not depend on the history of its paths). The Markov assumption allows for the modeling of a rich space of motion models for the target including stationary and random models. The searchers receive a reward if they move to the same location as the target, which is considered as capturing the target. A solution to the multi-searcher POMDP provides paths for the searchers that maximize the probability of capturing the target over a time interval. While this is not equivalent to minimizing expected capture time, maximizing probability of capture over a time interval practically relates to low average capture times.

Solving the POMDP formulation of efficient search quickly grows intractable for large environments and multiple searchers because it requires the joint space (or Cartesian product) of the searchers' paths to be considered². The size of this planning space grows exponentially in the number of searchers. This is an example of *explicit coordination* during which the searchers explicitly plan for their teammates. Alternatively, if each searcher plans individually without taking into account the future actions of its teammates, the size of the search space no longer increases. Since the searchers are no longer coordinating in any way, this is an instance of *no coordination*. Paths generated without any coordination often perform poorly because the searchers have no mechanism for reasoning about their teammates' actions. If the searchers share information about their paths, they provide valuable information that can improve path quality. In this case, the searchers are not explicitly planning for their teammates, but they are implicitly coordinating by sharing information about their paths. Our algorithm utilizes this *implicit coordination* along with finite-horizon search to simplify the POMDP formulation and achieve far better scalability. There is a tradeoff, however, in that implicit coordination does not consider all paths considered by explicit coordination. Thus, it is possible that implicit coordination might produce paths poorer than those generated by explicit coordination. In contrast, our results demonstrate that implicit coordination generates paths with capture times that are competitive with those generated by explicit coordination while using far fewer computational resources. This suggests that situations in which explicit coordination provides significantly lower capture times than implicit coordination are rare in the efficient search domain.

In this paper we present a bounded approximation algorithm using implicit coordination that solves the multi-robot efficient search path planning (MESPP) problem in indoor environments with a known floorplan. The proposed algorithm considers both mobile and stationary targets, and it operates both with and without noisy measurements to the target. Our algorithm is linearly scalable in the number of searchers and leverages the property of submodularity to provide a constant-factor approximation guarantee of the optimal solution. An objective function is submodular if it follows an intuitive property of diminishing returns. The more areas (nodes) in the environment that the searchers have visited, the less incremental reward is gained. The submodularity of the MESPP objective function directly leads to bounds on the performance of sequential allocation. We complement these theoretical results with empirical results both in simulation and using data from ultra-wideband ranging radio sensors. We originally introduced this algorithm in prior work (Hollinger et al. 2007b,a)

1. An adversarial target is one that actively evades capture. A target may also be considered *cooperative* if it actively seeks capture. These cases are not considered in this paper.

2. To formally define the joint path space, consider the possible locations $s_k(t) \in N$ of K searchers at times $t \in \{1, 2, \dots, T\}$. The searchers' configuration space at time t is $\Phi(t) = \{(s_1(t), \dots, s_K(t)) \mid s_1(t) \in N, \dots, s_K(t) \in N\}$. The searchers' joint path space is defined as the Cartesian product $\Psi_S = \Phi(1) \times \dots \times \Phi(T)$.

and later showed that it provides a bounded approximation of the optimal solution (Hollinger and Singh 2008). This paper provides a unified description of the experimental and theoretical results. In addition, we provide new analysis on the relationship between the MESPP reward metric and average capture time.

This paper is organized as follows. In Section 2 we discuss related work in the areas of pursuit-evasion, multi-robot coordination and probabilistic estimation. In Section 3 we formally define the MESPP problem and show that it optimizes a non-decreasing, submodular set function. In Section 4 we give a description of our coordination and estimation algorithms and utilize the result from the previous section to prove a bounded approximation of the optimal solution. In Section 7 we present both simulated results and results using ultra-wideband ranging radio data verifying the performance of our algorithm. Finally, in Section 8 we draw conclusions and discuss avenues for future work.

2. Related Work

Our proposed algorithm provides a solution to the non-adversarial efficient search problem for multiple searchers in large indoor environments. Related work in the area of coordinated search either does not directly deal with the efficient search problem or is not scalable to multiple searchers (or both). In addition, much early related work does not estimate the target's position, which does not allow for the use of noisy measurements.

Early research in coordinated search almost exclusively treated the target as an adversarial evader that actively avoids capture. Parsons (1976) developed some of the earliest methods for solving the classical adversarial pursuit-evasion problem on graphs. He considered the graph to be a system of tunnels represented by the edges of the graph in which an evader was hiding, and he defined the *search number* of a graph to be the minimum number of pursuers necessary to catch an adversarial evader with arbitrarily high speed. Determining the search number of a graph is known to be an NP-complete problem (Megiddo et al. 1988). In this early work in pursuit-evasion, the evader can only hide in the edges of the graph, and solutions deal with guaranteed or *worst-case* search. This does not directly address the efficient search problem for a non-adversarial target because it does not reason about target motion modeling or uncertainty in the target's position.

Guibas et al. (1999) extended classical pursuit-evasion techniques to guarantee capture in polygonal environments. Their algorithm discretizes polygonal environments into conservative visibility regions and then uses an information space approach to develop complete algorithms that guarantee capture in 1-searchable graphs. Gerkey et al. (2006) applied these ideas to searchers with limited sensing. For a single pursuer,

these algorithms are guaranteed to find a solution if one exists. When scaled to multiple pursuers, however, they lose this property. In addition, these algorithms are difficult to extend to complex environments because of the sheer number of (often very small) cells necessary in a conservative visibility discretization. These algorithms are also not applicable to complex environments in which capture cannot be guaranteed. This drawback becomes particularly prominent with limited pursuers and in any environment with a loop.

The classical pursuit-evasion methods above do not reason about capture time during search, and they do not account for partial knowledge of the target's location or its motion model. Our algorithm uses a probabilistic formulation with dispersion and capture matrices to reason about target motion and uncertainty. Our formulation takes advantage of the non-adversarial assumption on the target's motion model to reason about its expected behavior rather than its worst-case behavior.

Other researchers have taken different approaches to modeling the probabilistic coordinated search problem. Adler et al. (2003) introduced expected capture time to pursuit-evasion by examining the hunter and rabbit problem. They defined the *escape length* of a strategy as the worst-case expected number of rounds for the hunter to catch the rabbit, and they derive error bounds for various hunter strategies. Isler et al. (2005) advanced the concept of probabilistic pursuit-evasion to polygonal environments. They developed coordination strategies for one or two pursuers in simple polygonal environments based on the assumption that an adversarial evader does not have knowledge of some actions made by the pursuer. These algorithms still do not directly solve the efficient search problem because they do not fully relax the adversarial assumption on the target's motion.

Many probabilistic coordinated search problems, including efficient search, can be formulated as a Markov decision process (MDP) if the target's position is known or a POMDP if it is unknown. These formulations provide fully probabilistic representations of the problem, which can easily reason about uncertainty. Eaton and Zadeh (1962) discussed optimal solutions to the MDP guaranteed search problem. Roy et al. (2005) later discussed how belief compression can be used to make the POMDP guaranteed search problem tractable for a single pursuer. Smith (2007) developed a heuristic search value iteration algorithm for solving general POMDPs, which provides near-optimal solutions for problems with thousands of states. These solutions provide fully probabilistic solutions to coordinated search problems, but they suffer from poor scalability because they plan in the joint space of searcher paths. In the multi-robot efficient search domain, the number of states in the POMDP formulation can easily reach several billion due to exponential scalability in the number of searchers. For instance, if one wanted to plan paths for six searchers looking for one target in an environment with 100 cells, a POMDP with $100^{6+1} = 10^{14}$ states would need to be solved. Solving POMDPs of this size is far beyond the reach of even state-of-

the-art POMDP solvers. The (PO)MDP formulation of coordinated search problems provides a fully probabilistic solution, but it suffers from these scalability issues. In contrast, our algorithm utilizes implicit coordination to remain tractable even for large numbers of searchers.

In a similar manner to fully probabilistic formulations, researchers have also applied optimization techniques to the coordinated search domain. Lau et al. presented a dynamic programming approach for efficiently finding a single non-adversarial target (Lau et al. 2006) and a branch and bound approach for finding multiple targets (Lau et al. 2005). These techniques provide optimality bounds on their solutions, but they suffer from poor scalability because they plan in the joint space for multiple searchers. Calisi et al. (2007) also provided a solution to the single robot exploration and search problem using Petri nets. Their solution is domain-specific, and they only formulate it for the case of a single searcher.

The poor scalability of coordinated search solutions to large environments and multiple searchers has motivated some researchers to utilize either heuristic methods or auction-based methods. To the best of our knowledge, no scalable approximation algorithm has been proposed for the efficient search problem with a mobile, non-adversarial target. Sarmiento et al. (2004) presented a framework for finding stationary targets in polygonal environments with multiple robotic searchers using a one-step cost heuristic. Instead of planning in the joint space, their algorithm reduces reward along previously traveled paths. This can be seen as an instance of implicit coordination. They do not extend this work to mobile targets, and they do not present results in large-scale environments.

One popular technique to improve scalability in multi-agent domains is for robots to coordinate using synthetic “auctions”. The auction-based methods are specifically designed to explicitly coordinate when it is particularly beneficial and implicitly coordinate when it is not. Such methods are particularly well-suited for problem domains in which agents must occasionally perform a very tightly coordinated task. Researchers have applied auction methods to multi-agent coordinated search domains. Kalra (2006) showed an application of auction-based plan sharing in a constrained exploration domain. Her method allows searchers to actively coordinate by running auctions when they are presented with high-cost situations. Gerkey et al. (2005) also developed a parallel stochastic hill-climbing method for small teams that is closely related to auction-based methods. His algorithm dynamically forms teams of searchers that work together to solve tasks.

While auction-based algorithms are more scalable than joint planning approaches, they still rely on auctions and/or team formation, which can consume large amounts of communication bandwidth and planning time. In many domains, implicit coordination is sufficient to produce high-quality results, and the additional overhead of running auctions is unnecessary. Our results show that implicit coordination yields low capture times in the efficient search domain without the

overhead of auctions or team formation. We assert that this is the case because situations in which explicit coordination is beneficial are rare in the efficient search domain.

This paper shows that the MESPP problem requires the optimization of a submodular objective function, and this key insight provides optimality bounds on sequential allocation, an algorithm linearly scalable in the number of searchers. Submodularity has been utilized in related domains to provide theoretical guarantees on sequential allocation. Guestrin et al. used submodular set functions to develop algorithms for sensor placement problems in Gaussian Processes (Guestrin et al. 2005) and in more general domains (Krause and Guestrin 2007). They also extended their algorithms to robust observation selection against sensor failure (Krause et al. 2007). These applications deal primarily with placing sensors to monitor information in an environment (e.g. monitoring algae blooms in lakes and temperature in a building). These algorithms do not incorporate moving nodes (searchers) and thus are not suitable for MESPP. Singh et al. (2007) developed algorithms for solving the multi-robot informative path planning (MIPP) problem, which does allow for moving nodes. We extend their theoretical bounds to the MESPP problem. To the best of our knowledge, our work is the first approximation algorithm with theoretical guarantees in the MESPP domain.

Prior work in target estimation is also relevant to the MESPP problem. Ferris et al. (2006) used Gaussian processes to estimate the position of a moving target using wireless signal strength. Their algorithm discretizes the environment in a similar manner to ours and then uses a particle filter to estimate position. Our proposed method provides an alternative for position estimation that closely couples with planning and remains computationally tractable. Our algorithm also does not require prior training data, and it is completely online. As a tradeoff, our method is less appropriate than theirs for sensors with very complex or unknown noise models. They later extended this technique to allow for unknown access point locations using Gaussian process latent variable models (GPLVMs) (Ferris et al. 2007). The present authors showed that a similar technique can be used effectively with ranging radios for both known and unknown radio node locations (Hollinger et al. 2008). This suggests a possible extension of our efficient search algorithm to domains with unknown searcher locations.

3. Problem Setup

This section formally defines the problem of locating a mobile, non-adversarial target with multiple searchers (the MESPP problem). It also shows that the MESPP problem optimizes a submodular objective function.

To formulate the MESPP problem, the environment in which the searchers and target are located needs to be described. First, divide the environment into convex cells (see

the algorithm description below). Taking into account the cell adjacency in a discretized map yields an undirected graph that the searchers can traverse. Let $G(N, E)$ be the undirected environment graph with vertices N and edges E . At any time t , a searcher exists on vertex $s(t) \in N$. The searcher's movement is deterministically controlled, and it may travel to vertex $s(t+1)$ if there exists an edge between $s(t)$ and $s(t+1)$. A target also exists on this graph on vertex $e(t) \in N$. The target moves probabilistically between vertices. The searcher receives reward by moving onto the same vertex as the target, $s(t) = e(t)$, and no reward is gained after this occurs. Refer to this as a capture event. Reward is discounted by γ^t , where γ is a constant discount factor. Thus, the searchers receive more reward for finding the target at a lower t . This discount factor corresponds to the probability that the search will end at a given time. For instance, the target may leave the search area or expire. This necessitates locating the target in a short time.

The target's movement model is known to the searcher, and it is independent of the searcher's position on the graph. Further assume that the target's motion model is Markovian (i.e. it depends solely on its current cell). This assumption allows for a rich space of motion models including those followed by randomly moving and stationary targets. The searcher knows its own position and it has knowledge of the target's position at a time t in the form of a belief distribution over all vertices, $b(t)$. Since $b(t)$ can be an arbitrary distribution, this formulation allows multi-modal estimates of the target's position. Call the problem so far the efficient search path planning (ESPP) problem.

To extend to MESPP, place K searchers on the vertices; the location of the k th searcher at time t is $s_k(t) \in N$ (for $k = 1, 2, \dots, K$ and $t = 1, 2, \dots, T$). The searchers now gain reward if any of them are on the same vertex as the target. Incorporating additional searchers forces both the state and action space to grow exponentially. Refer to the combined action space of all searchers as the *joint* action space. The MESPP can be formulated as a POMDP with the following reward function:

$$J(U(1), \dots, U(T)) = \sum_{t=0}^T \gamma^t P(\exists k : s_k(t) = e(t)), \quad (1)$$

where $U(1), \dots, U(T)$ are the deterministic *actions*, with $U(t) = [U_1(t), \dots, U_K(t)]$ being a K -dimensional control vector specifying the location of each searcher (i.e. $s_k(t) = U_k(t)$). The goal is to choose $U(1), \dots, U(T)$ so as to maximize $J(U(1), \dots, U(T))$ as given by Equation (1).

The reward function J is both *non-decreasing* and *submodular* on the set of nodes of the *time-augmented search graph*. Intuitively, the meaning of these terms is the following (exact definitions are given in Appendix A).

1. A set function is non-decreasing if adding more nodes to the observed set always increases reward. This is clearly

the case in MESPP since visiting more places can only increase likelihood of capture.

2. A set function is submodular if it satisfies the notion of diminishing returns. In other words, the more places in the environment that have been visited, the less can be gained by visiting more.
3. The time-augmented search graph G' is a *directed graph* and is obtained from G as follows: if u is a node of G , then (u, t) is a node of G' , where $t = 1, 2, \dots, T$ is the *time stamp*; if uv is an edge of G , then $(u, t)(v, t+1)$ and $(v, t)(u, t+1)$ are *directed edges* of G' for every t . There is also a directed edge from (u, t) to $(u, t+1)$ for all u and t . In other words, G' is a "time evolving" version of G and every path in G' is a "time-unfolded" path in G .

Searchers choose a feasible set of paths that maximizes the expected probability of intersecting the target's path at the earliest possible time (before reward is heavily discounted). Any such set of paths involves a set of G' nodes, call it $A \subseteq G'$ (note that A encodes both which nodes of the original G have been visited and at which times). Hence, the reward $J(U(1), \dots, U(T))$ can also be written as $J(A)$. Now we write $J(A)$ in more detail as follows: let Y denote a path (in G') taken by the target, let $P(Y)$ be the probability of this path, let Ψ be the space of all possible target paths; finally, let $F_Y(A)$ be the discounted reward received by searcher paths A if the target chooses path Y . This gives us the following reward function:

$$J(A) = \sum_{Y \in \Psi} P(Y) F_Y(A) \quad (2)$$

We now have a theorem concerning the properties of $J(U(1), \dots, U(T)) = J(A)$.

Theorem 1. *The objective function optimized by the MESPP problem is $J(A)$, and this is a non-decreasing, submodular set function.*

The proof of Theorem 1 is given in the Appendix A along with more formal definitions of terms. This result is used in the next section to show bounds on the performance of sequential allocation in this domain. Since the Markov assumption is made on the target's motion model, calculating the expectation in Equation (2) can be done using dispersion matrices. This greatly simplifies the computation of $F(A)$ over the space of searcher paths. This is explained in more detail below.

4. Environment and Target Modeling

4.1. Map Discretization

The formulation of the efficient search problem requires the discretization of continuous environments into discrete cells.

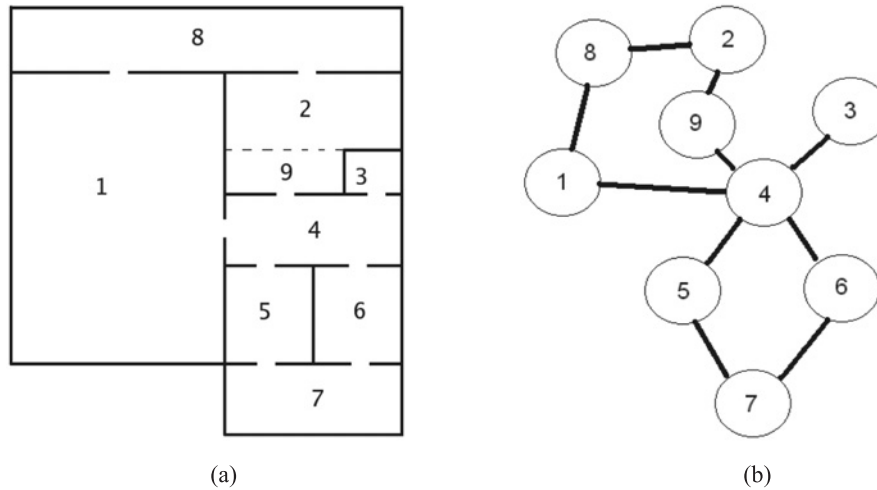


Fig. 1. Example discretization of house environment (a) and undirected graph resulting from the house discretization (b). Searchers plan paths on the undirected graph that maximize capture probability of a non-adversarial target.

Partitioning the environment into cells allows for planning on a finite graph and greatly reduces the complexity of planning and estimation. One method for discretization takes advantage of the inherent characteristics of indoor environments. To discretize an indoor map by hand, simply label convex hallways and rooms as cells and arbitrarily collapse overlapping sections. This method is simple enough that it can be performed by hand even for large maps. Alternatively, a suitable discretization can be found automatically using a convex region finding algorithm (such as Quine–McClusky (Singh and Wagh 1987)). Take into account that the cell adjacency in a discretized map yields an undirected graph that the searchers can traverse. Figure 1 shows an example discretization of a small house environment and the resulting undirected graph. Subsequent sections refer to this simple example for explanatory purposes.

This method for discretization has the advantage of ensuring that a searcher in a given convex cell has line-of-sight to a target in the same cell. This allows the capture event to be reduced to the attainment of line-of-sight to the target. This makes intuitive sense because gaining line-of-sight effectively collapses the unknown state of the target to a known state. Gaining line-of-sight is relevant to many sensors that a robotic searcher would possess including cameras and laser rangefinders.

Figure 2 shows example discretizations of the office building and museum environments used for simulated testing. Both of these environments are larger and more complex than those discussed in previous work in coordinated search (Guibas et al. 1999; Sarmiento et al. 2004; Lau et al. 2005). The museum environment is particularly challenging because it contains many cycles by which the target can avoid detection. In contrast, the

office environment has two major cycles that correspond to the hallways.

In comparison with the visibility-based discretization proposed by Guibas et al. (1999), this discretization technique yields far fewer cells making it more applicable to large, complex environments. The tradeoff is that this method does not provide a discretization suitable for use with LaValle and Guibas's visibility-based guaranteed search algorithms. The large number of cycles in the museum and office make these environments poorly suited for these guaranteed methods anyway.

4.2. Target Motion Modeling

To integrate a motion model of the target into the efficient search framework, “capture” and “dispersion” matrices can be applied to the target's state vector. This formulation greatly simplifies the calculation of Equation (2) and its optimization.

The location of the target is represented by a belief vector $b(t) = [b_0(t), \dots, b_N(t)]$ where $b_0(t)$ represents the probability the target has been captured by time t (the capture state), and $b_n(t)$ represents the probability that at time t the target is in the n th discretized cell. Mathematically represent a capture event on that state vector by defining a matrix that moves all probability from all cells visible from searcher k 's current cell $s_k(t)$ to the capture state. The capture matrix can also contain non-unity values if the probability of seeing a target when it is in a searcher's line-of-sight is less than one. The appropriate capture matrix $C_{s_k(t)}$ for cell $s_k(t)$ is applied at time t as in

$$b(t+1) = b(t)C_{s_k(t)}. \quad (3)$$

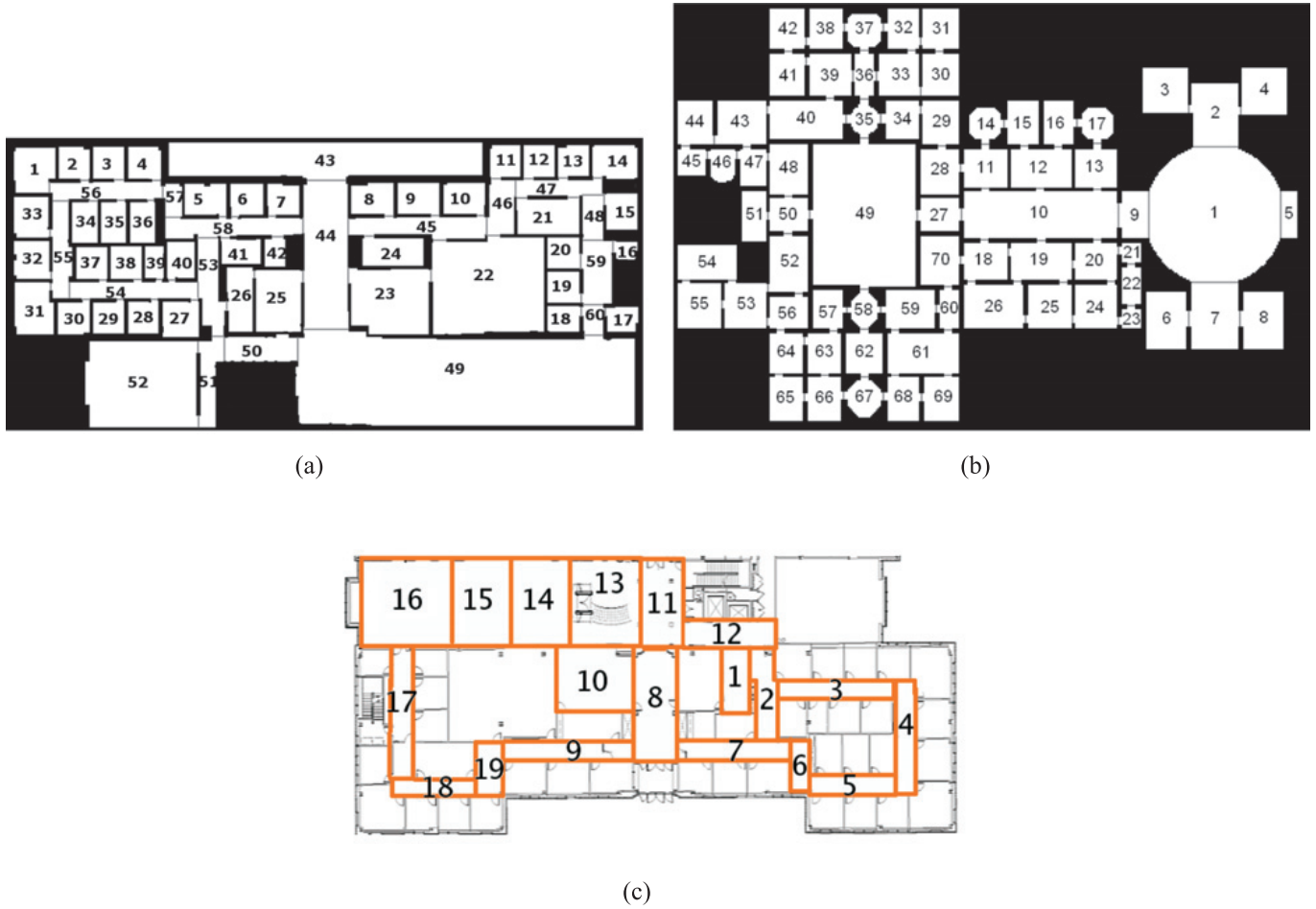


Fig. 2. Example floorplans of environments used for efficient search trials. The larger maps (a) and (b) were used for simulated testing. The smaller map (c) was used for “hybrid” trials in which simulated searchers found a Pioneer robot with measurements from an experiment.

For example, if it is assumed that the searcher cannot see through doorways, the capture matrix for a searcher in cell one would be the $(N + 1) \times (N + 1)$ identity matrix with the second row unity value shifted to the first column.

Similarly, define dispersion matrices to represent the expected motion of the target in the environment. The discretization of the environment yields an undirected graph of possible target movements between cells. Based on a motion model, assign probabilities to each of these movements and define a matrix that properly disperses the target’s probable location. This assumes that the target’s motion model obeys the Markov property. This is equivalent to saying that the probability of the target moving to a given cell is dependent solely on the target’s current cell. This assumption allows for the representation of many motion models including random and stationary. It also allows for the target’s motion model to be modified based on the size of the cells or the probability that the target will move through certain areas of the map.

The dispersion matrix formulation also has the advantage that the target’s probability distribution is fairly insensitive to changes in the motion model. Since the transition probabilities are solely dependent on the current state, changing the probability of moving between cells slightly will not cause large changes in the probability distribution.

The dispersion matrix D at time t can be applied to yield a new target state vector at time $t + 1$ as in Equation (4). If a stationary model for the target is preferred, D can be set to the identity:

$$b(t + 1) = b(t)D. \quad (4)$$

For instance, if it is assumed that the target will remain in its current cell or move to any adjacent cell with equal probability at the next time step, the dispersion matrix for the house environment in Figure 1 is (note that this is only one of the many motion models expressible by dispersion matrices)

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{\frac{1}{6}} & \mathbf{0} & \mathbf{\frac{1}{6}} & \mathbf{\frac{1}{6}} & \mathbf{\frac{1}{6}} & \mathbf{\frac{1}{6}} & \mathbf{\frac{1}{6}} & \mathbf{0} & \mathbf{0} & \mathbf{\frac{1}{6}} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} \end{pmatrix}$$

The top row of the dispersion matrix corresponds to the capture state. For reference, the fifth row (shown in bold) corresponds to the probabilities associated with cell 4. If the target's cell transitions were not equally likely, the row values would no longer be equal. It is important to note that while there are as many capture matrices as there are cells in an environment, there is a single dispersion matrix for the entire environment.

The capture matrix of a searcher k and the dispersion matrix can be multiplied yielding a new target state vector

$$b(t+1) = b(t)DC_{s_k(t)} \quad (5)$$

In larger environments, it is desirable to use multiple searchers. The capture matrices for all searchers can be multiplied to yield the new state

$$b(t+1) = b(t)D \left[\prod_{k=1}^K C_{s_k(t)} \right], \quad (6)$$

where K is the number of searchers. To perform this step, it is necessary for the searchers to communicate their states at each time step.

A state evolution equation can now be written for the system of searchers and target. The state vector at time t is $S(t) = [s_1(t), \dots, s_K(t), b_0(t), \dots, b_N(t)]$, and the state evolution equation has the form:

$$S(t+1) = f(S(t), U(t+1)), \quad (7)$$

where $U(t+1) = [U_1(t+1), \dots, U_K(t+1)]$ is a K -dimensional control vector specifying the next location of each searcher (as described in Section 3), and the evolution of

the probabilities $b_0(t+1), \dots, b_N(t+1)$ is determined by Equation (6).

Recall that the reward function is

$$J(U(1), \dots, U(T)) = \sum_{t=0}^T \gamma^t P(\exists i : s_k(t) = e(t)), \quad (8)$$

where T is some finite ending time.

Hence, our MESPP optimization problem is to maximize Equation (8) subject to the state evolution shown in Equation (7)³.

The computational complexity of the dispersion and capture matrix application is determined by the number of cells in the environment (the size of the matrices) and the number of searchers. It is $O(K|N|^3)$, where K is the number of searchers and $|N|$ is the number of cells in the environment. While $O(|N|^3)$ may grow intractable for very large numbers of cells, we stress that many competing methods scale exponentially in both $|N|$ and K (e.g. polygonal search using visibility graphs Guibas et al. (1999)). We also stress that the matrices are often quite sparse, which reduces the complexity of applying the dispersion and capture matrices.

4.3. Measurement Incorporation

If non-line-of-sight measurements of the target's location are available during ESPP, searchers can utilize them to assist in search. After receiving a new measurement, searchers can replan using the new belief distribution on the target's location. Since searchers replan after receiving measurements, online measurement incorporation heuristically improves path quality but does not affect theoretical guarantees. In the next section, we show how combining sequential allocation, finite-horizon path enumeration and Bayesian measurement updating yields a scalable and online algorithm for solving the MESPP problem.

5. Coordination Methods

5.1. Explicit Coordination

Having properly modeled the target's motion, the searchers can look for paths on the discretized floorplan with high reward. One method is to use a centralized planner to solve the full POMDP for the infinite horizon. This method grows intractable very quickly, so an alternative is to search all possible joint paths on the floorplan graph to a given depth. This outputs paths for all searchers that maximize reward. This algorithm can be decentralized by assigning identification numbers

3. Also recall that Equation (1) can be used as the reward function in a POMDP formulation of MESPP.

to the searchers and forcing each searcher to plan as if it were the centralized solver (the searchers must still communicate their state). We refer to this method as explicit coordination because each searcher explicitly plans for its teammates.

The advantage of explicit coordination is that each searcher takes the future positions of the other searchers into account during planning. However, the dimensionality of the search space grows such that explicit coordination scales exponentially with the number of searchers. The number of cells that must be searched is $O(b^{dK})$ where K is the number of searchers, d is the lookahead depth and b is the maximum branching factor of the cell graph. This does not lead to a tractable solution for more than a small number of searchers with a short lookahead.

5.2. No Coordination

To decouple planning during efficient search, each searcher can plan for itself while assuming that the states of the other searchers are fixed. This prevents the search space from growing in complexity as the number of searchers increases. With this assumption, each searcher must simply plan for its optimal path given the current state information of the other searchers. The complexity of this planning algorithm *on each searcher* is not affected by the number of searchers: $O(b^d)$. The total complexity is thus $O(Kb^d)$, linear in the number of searchers. Even though each searcher's planning is independent of its teammates' future actions, the current positions of other searchers provide information that the target is not in that cell. For this purpose, searchers must communicate their locations at each time step.

5.3. Implicit Coordination

Without coordination, the searchers cannot reason about the future actions of their teammates. To overcome this disadvantage, a coordination method can be introduced that is intermediate between explicit coordination and no coordination at all. During implicit coordination, the searchers share their current paths after planning. Other searchers then plan for their own paths while assuming that the transmitted paths will be followed. This strategy leads to higher quality solutions than the no coordination strategy with the same scalability on each searcher: $O(b^d)$.

The major cost of implicit coordination is that searchers must now communicate their entire paths rather than just their current locations. Since the environment is coarsely discretized, this does not lead to much increase in communication bandwidth. However, implicit coordination does lead to a broadcast communication requirement, which can delay communication in large networks. In this case searchers may need to plan with outdated information. This problem is alleviated

by the sparsity of the network because searchers will typically only communicate with teammates in close proximity. This gives them up-to-date information from searchers near them and somewhat outdated information from searchers far away. Since planning is up to a finite horizon, information from near searchers is more important. Thus, network delay will have a minimal effect on solution quality in sparse networks. In addition, communication complexity can be reduced using heuristics, such as broadcasting paths only to searchers within the horizon length.

Figure 3 shows how implicit coordination can lead to high-quality searcher paths. In this example, the searchers correctly branch into the two major cycles on the map. Searcher two communicates its intention to move left to searcher one. Searcher one then uses that information to choose the path to the right.

This leads to an implicit coordination algorithm utilizing sequential allocation. The searchers plan one at a time and share their paths with the other searchers. Each searcher takes into account information from their teammates but can only modify its own path.

6. Proposed Algorithm

This section describes our algorithm for non-adversarial search with multiple robots utilizing sequential allocation and finite-horizon planning. Theoretical performance bounds are shown using the non-decreasing submodularity of the MESPP objective function. This somewhat surprising result shows that sequential allocation, an algorithm linearly scalable in the number of searchers, generates near-optimal paths in the MESPP domain.

Algorithm 1 gives pseudocode for the MESPP sequential allocation algorithm. The algorithm maintains a list of nodes $V \subseteq N'$ that have been visited by the searchers. Note that N' is the time-augmented version of the nodes N in the environment, which allows for nodes in the original graph to be revisited at later times. The searchers choose paths A_k that maximize the objective function $F(V \cup A_k)$ and then add the nodes they have visited to V . Effectively, subsequent searchers treat the paths of previous searchers as "given", and they are not allowed to change them. Sharing nodes to update V is an instance of implicit coordination as described above. Since the search space does not grow with the number of searchers, the complexity of sequential allocation is linear in the number of searchers. It is important to note that while path planning occurs sequentially, the execution of paths is simultaneous.

Algorithm 1 requires the objective function to be maximized for the ESPP problem as a subroutine. Any algorithm for solving the ESPP problem can be inserted here. However, if the ESPP solver is bounded, the non-decreasing submodularity of the objective function leads to guarantees on the performance of sequential allocation. The following theorem (from

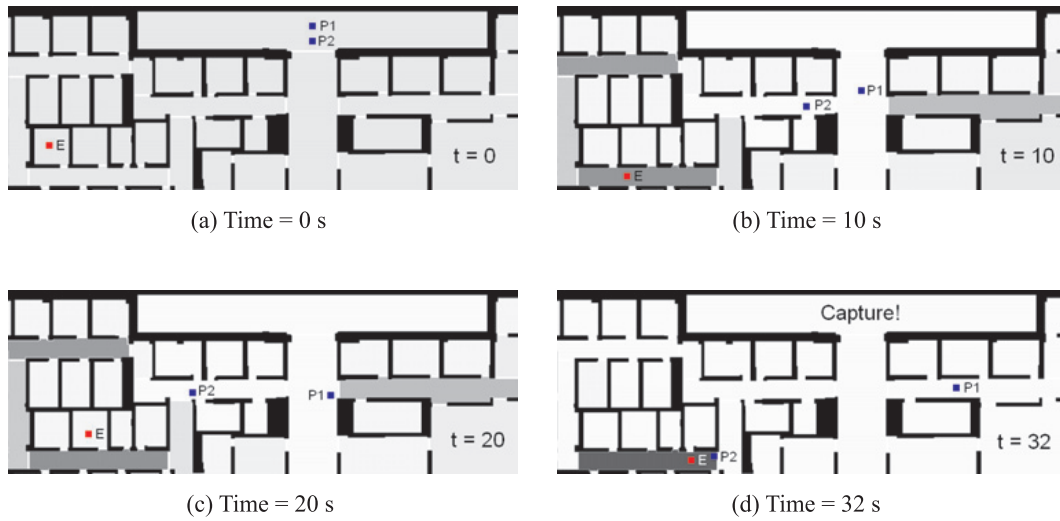


Fig. 3. Snapshots of implicit coordination during coordinated search at different time steps until capture event. The searchers (labeled P1 and P2) branch into the two major cycles on the map to search for the target (labeled E). Darker cells denote more probable target locations.

Algorithm 1 Sequential allocation MESPP algorithm

Input: Multi-agent efficient search problem
 % $V \subseteq N'$ is the set of nodes visited by searchers
 $V \leftarrow \emptyset$
for all searchers k **do**
 % $A_k \subset N'$ is a feasible path for searcher k
 % Finding this arg max solves the ESPP for searcher k
 $A_k \leftarrow \arg \max_{A_k} F(V \cup A_k)$
 $V \leftarrow V \cup A_k$
end for
 Return A_k for all searchers k

Singh et al. (2007)) shows that sequential allocation leads to theoretical guarantees in the informative path planning domain.

Theorem 2. (Singh et al. (2007)). *Let κ be the approximation guarantee for the single path instance of the informative path planning problem for any non-decreasing, submodular function. Then sequential allocation on the MESPP problem will yield $(1 + \kappa)$ for the multi-robot informative path planning problem.*

The findings in Theorem 1 can be leveraged to extend these results to MESPP. The following corollary states that if an ESPP solver has an approximation guarantee of κ , then sequential allocation on the MESPP will yield an approximation guarantee of $(1 + \kappa)$.

Corollary 1. *If a solver achieves an approximation guarantee of κ for the ESPP problem, sequential allocation yields an approximation guarantee of $(1 + \kappa)$ for the MESPP problem.*

Proof. This is immediate from Theorems 2 and 1. Theorem 2 states that sequential allocation achieves this bound for any single-agent path planning problem optimizing a non-decreasing, submodular function. Theorem 1 shows that the ESPP problem requires the optimization of such an objective function. \square

Here an approximation guarantee κ states that if the MESPP solver returns a set of nodes $A \subseteq N'$, then $F(A) \geq (1/\kappa)F(A^{\text{OPT}})$, where A^{OPT} is the set of nodes visited by the optimal paths. Clearly, $\kappa \geq 1$ since $F(A)$ cannot be greater than the optimal reward. The case where $\kappa = 1$ corresponds to solving the ESPP problem optimally. In this case, sequential allocation can achieve no worse than half the optimal reward. This theoretical result allows single-agent performance bounds to be extended to the multi-agent case using a linearly scalable algorithm, albeit with a loss in approximation quality. In the next section we present a bounded algorithm for solving the ESPP problem using finite-horizon path enumeration.

6.1. Finite-horizon Planning

In large environments, even the ESPP may be intractable to solve optimally (or even near-optimally) due to the computational overhead of considering many infinite-horizon paths. In

Algorithm 2 Finite-horizon path enumeration for ESPP

Input: Single-agent efficient search problem

for All feasible paths A to horizon d **do** Calculate $F(A)$ **end for**Return $A \leftarrow \arg \max_A F(A)$

these cases, one option is for the searchers to plan a finite number of cells ahead and choose the best path to that horizon. At any time while traversing this path, the searcher can plan again utilizing new information on a new horizon. This leads to an online solution to MESPP, and it allows for the incorporation of measurements of the target's position as they become available. Algorithm 2 gives pseudocode for solving the ESPP problem using finite-horizon path enumeration. As the finite-horizon method relies on path enumeration to solve ESPP, it scales exponentially with the search depth: $O(b^d)$, where b is the maximum branching factor of the search graph, and d is the search depth in cells.

The following lemma derives optimality bounds for finite-horizon path enumeration, and the result extends to the multi-robot case with sequential allocation as in Theorem 3.

Lemma 1. Finite-horizon path enumeration on the ESPP problem achieves a lower bound of:

$$F(A^{\text{FH}}) \geq F(A^{\text{OPT}}) - \epsilon, \quad (9)$$

where A^{FH} is the path returned by finite-horizon path enumeration, A^{OPT} is the optimal feasible path and $\epsilon = R\gamma^{d+1}$.

Proof. Finite-horizon path enumeration achieves the optimal reward inside the horizon depth for ESPP because it checks all paths. The maximum reward that could be gained outside the horizon depth is given by $\epsilon = R\gamma^{d+1}$, where R is the reward received for locating the target, γ is the discount factor and d is the search depth. The bound is immediate. \square

Theorem 3. Finite-horizon path enumeration with sequential allocation on the K -robot MESPP problem achieves a lower bound of:

$$F(A_1^{\text{FH}} \cup \dots \cup A_K^{\text{FH}}) \geq \frac{F(A_1^{\text{OPT}} \cup \dots \cup A_K^{\text{OPT}}) - \epsilon}{2}. \quad (10)$$

Proof. The maximum reward outside the horizon remains the same as in Lemma 1 (i.e. ϵ is unchanged). Since the single-robot case achieves the optimal reward within the finite-horizon (i.e. $\kappa = 1$), sequential allocation yields an approximation guarantee as in Corollary 1 as $\kappa + 1 = 2$. \square

Intuitively, as the search depth increases, the bound tightens. In addition, decreasing the discount factor tightens the

bound. This is because lower discount factors more heavily weight the reward gained earlier, which is more likely to be within the finite-horizon. It is important to note that the quality of the bound is independent of the number of searchers. This is a worst-case bound for arbitrary starting distributions and motion models. In practice, searchers can run finite-horizon path enumeration repeatedly on a receding horizon, which leads to performance that far exceeds this lower bound.

6.2. Bayesian Measurement Updating

In this section we present a discrete Bayesian method for modifying the target's state probability vector given new sensor data. Suppose at time t we receive a measurement z_t . We are interested in computing the *conditional* probability $p_t^i = P(e_t = i | z_1, \dots, z_t)$. (Note that we denote the target's location $e(t)$ as e_t in this section to simplify notation.)

In other words, p_t^i is the probability that the target is in node i at time t , *conditioned on measurements* z_1, \dots, z_t . Note that in previous sections we were computing $P(e_t = i)$, i.e. the probability of the target being at node i without any measurement conditioning. We are also given a known motion model, which provides $P(e_t = i | e_{t-1} = j)$ for all cells i and j . This is encoded in the dispersion matrix D .

Using standard recursive Bayesian updating (Thrun et al. 2005), we can write a recursive update

$$p_t^i = \eta P(z_t | e_t = i) \sum_j P(e_t = i | e_{t-1} = j) p_{t-1}^j, \quad (11)$$

where η is a normalizing constant. Note that if a measurement is not received at time t , we would simply apply the motion model at that time step.

Assuming a known motion model, this reduces the problem of calculating the posterior to that of calculating a likelihood term $P(z_t | e_t = i)$. Since each cell is represented as a continuous set of points in the map plane, this calculation is difficult. To reduce the complexity of the problem, further discretize each cell into small subcells and calculate a likelihood at the center of each subcell. We denote the M^i subcells of cell i as m^{ij} for all $j \in \{1, \dots, M^i\}$. The calculation of $P(z_t | e_t = i)$ is now one of calculating a likelihood at many points and then taking the sum of these likelihoods.

For range measurements, the problem of calculating $P(z_t | e_t = i)$ is simply that of determining the expected range value for the center of each subcell. Let q^{ij} be the Euclidean distance from the ranging sensor to subcell m^{ij} , and let r_t be the received range measurement with assumed Gaussian noise variance σ^2 . The likelihood is then calculated as

$$P(z_t | e_t = i) = \sum_{j=1}^{M^i} N(r_t; q^{ij}, \sigma^2). \quad (12)$$

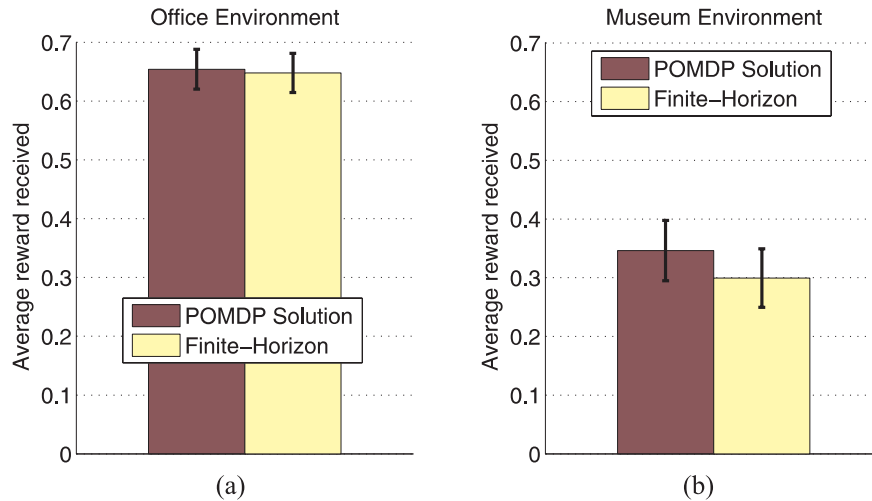


Fig. 4. Comparison of finite-horizon path enumeration (lookahead depth five) versus the POMDP solution for a single searcher in two complex simulated environments. The POMDP solution with two searchers was not tractable. Error bars are one standard error of the mean (SEM), and averages are over 200 trials. Target and searchers move at a maximum speed of 1 m s^{-1} .

To improve accuracy using the discrete Bayesian method, an element-based discretization method can also be incorporated (Furukawa et al. 2007). Element-based methods use irregular polygonal discretizations to interpolate the posterior between points. The proposed framework allows for this extension if greater accuracy is necessary than that provided by a regular grid.

We have extended this framework to scenarios in which the locations of the radio nodes are initially unknown (Hollinger et al. 2008). This technique uses probabilistic dimensionality reduction with GPLVMs to reconstruct a preliminary path of the target. The path is then used to map the locations of the radio nodes on an occupancy grid. The reconstructed node locations can be used for subsequent tracking. This provides a measurement incorporation framework for efficient search in environments without pre-installed infrastructure.

7. Results

7.1. Simulated Results

To test our proposed MESPP algorithm, we ran simulated trials using a multi-agent coordinated search simulation in C++ on a 3.2 GHz Pentium 4 processor. This simulation allows for multiple searchers and both stationary and moving targets. It was assumed that the average speed of the target is 1 m s^{-1} and that it moves holonomically between cell boundaries. The searchers also move with a maximum speed of 1 m s^{-1} , which would be a reasonable speed for state-of-the-art autonomous vehicles. The searchers start in the same location for all trials, and the location of the target is initialized at random

on the map. Simulated experiments were run in the museum ($150 \times 100 \text{ m}^2$) and office ($100 \times 50 \text{ m}^2$) environments shown in Figure 2. These environments are considerably larger than those searched by many authors using comparable methods (Guibas et al. 1999; Sarmiento et al. 2004; Gerkey et al. 2005; Roy et al. 2005).

In all tests, the performance metric is the average reward received over many trials. For a given trial, reward received is calculated as $R(t_c) = R\gamma^{t_c}$, where R is the reward for locating the target, γ is the discount factor and t_c is the time at which the target was found. The reward and discount factor were arbitrarily set to $R = 1$ and $\gamma = 0.95$ for all experimental trials. Extension 1 shows animations of these trials.

Figure 4 compares finite-horizon path enumeration (horizon depth five) to the infinite horizon POMDP solution for a single searcher. With a single searcher, the POMDP formulation of ESPP is still solvable using heuristic search value iteration (HSVI2) (Smith 2007). These results show that, for the single-searcher case, finite-horizon path enumeration yields average rewards competitive with those generated by the HSVI POMDP solution. Trials with two searchers were attempted with HSVI but were unsuccessful because the exponentially increased state-action space would not fit in memory. This demonstrates the poor scalability of the POMDP formulation of the MESPP problem.

Since solving the POMDP formulation is intractable for multiple searchers, trials were run using a finite-horizon explicit coordination algorithm that is identical to Algorithm 2 except that it enumerates paths for all searchers in the joint space. This scales as $O(b^{dK})$, where b is the branching factor, d is the search depth and K is the number of searchers. Figure 5 gives a comparison of the reward received by sequential

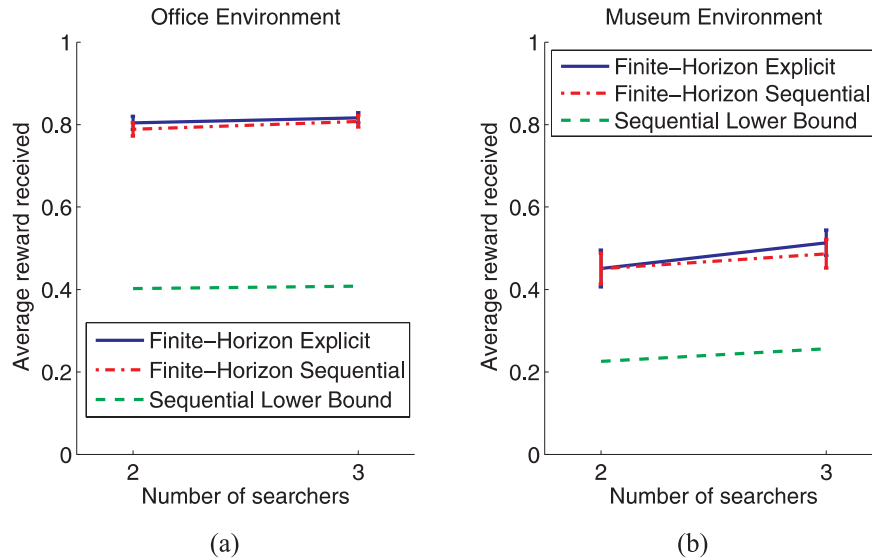


Fig. 5. Comparison of sequential allocation versus explicit coordination in two complex simulated environments. Finite-horizon path enumeration with lookahead depth two was used for both methods. Explicit coordination with more than two searchers was intractable. Error bars are one standard error of the mean (SEM), and averages are over 200 trials. Target and searchers move at a maximum speed of 1 m s^{-1} . Sequential allocation greatly outperforms its lower bound.

allocation and this explicit coordination algorithm. Since explicit coordination grows intractable at large lookahead depths, a depth of two was used for comparison. Figure 5 also shows a lower bound for sequential allocation calculated from Corollary 1 using the explicit coordination results. This bound is the lowest reward that sequential allocation could achieve if explicit coordination yielded the optimal reward. On both maps, sequential allocation greatly outperforms its lower bound.

These simulated experiments demonstrate that implicit coordination with sequential allocation yields results nearly equivalent to those achieved through explicit coordination. In sharp contrast with explicit coordination's exponential scalability, sequential allocation is linearly scalable in the number of searchers. Figure 6 demonstrates the scalability of sequential allocation by showing reward received with up to five searchers in the museum and office.

7.2. Ranging Radio Measurements

One major application of MESPP is that of finding lost first responders in disaster scenarios. To better model this scenario, an urban response test environment was set up using a Pioneer robot and five Multispectral ranging radio nodes (<http://www.multispectral.com/>). These sensors use time-of-flight of ultra-wideband signals to provide inter-node ranging measurements through walls. They have an effective operating distance of about 30 m indoors and provide ranging accuracy approximately within 1–2 m. In experiments, the Pioneer robot acted as a lost first responder and was teleoperated around

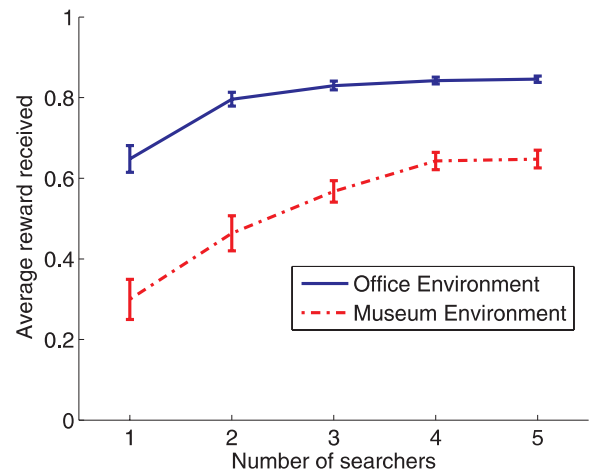


Fig. 6. Multiple searcher scalability trials for finite-horizon path enumeration (lookahead five) and sequential allocation in simulated environments. Error bars are one standard error of the mean (SEM), and averages are over 200 trials. Target and searchers move at a maximum speed of 1 m s^{-1} .

the environment carrying a ranging radio node. Four stationary nodes were placed in surveyed locations around the environment to provide range to the Pioneer. The Pioneer also carried a SICK laser rangefinder, and its location was found using laser AMCL-SLAM methods from the Carmen software package (Thrun et al. 2005). The Pioneer's laser localization was

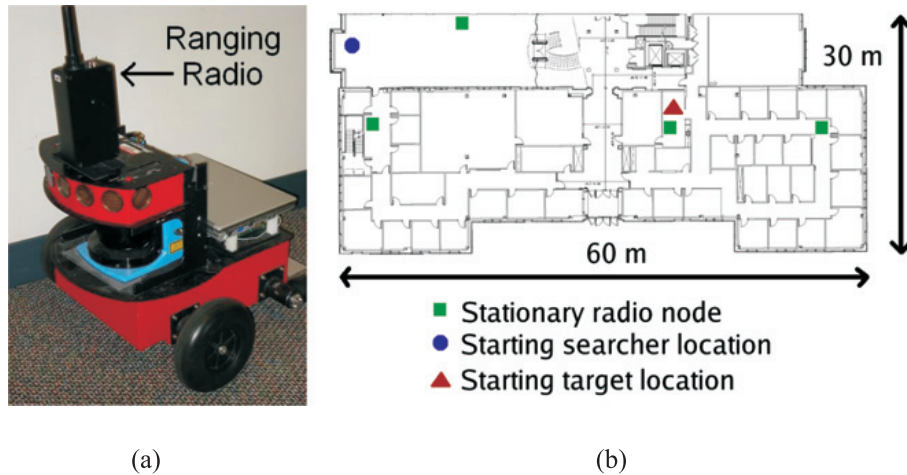


Fig. 7. Photograph of the Multispectral ultra-wideband ranging radio mounted on the Pioneer robot (a) and floorplan of the testing environment (b). The robot was teleoperated around the environment to act as the moving target.

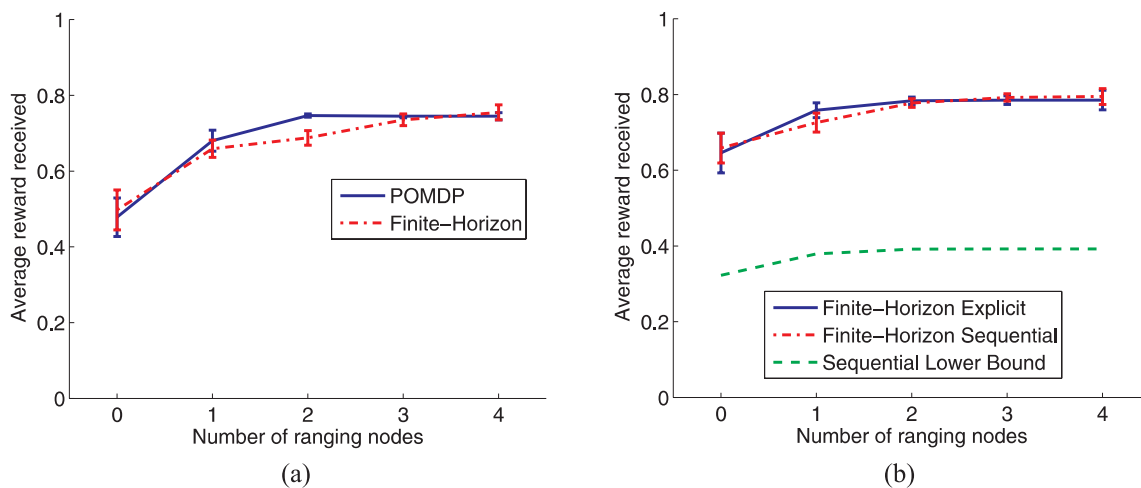


Fig. 8. (a) Comparison of finite-horizon path enumeration (lookahead depth five) versus the POMDP solution for a single searcher using ultra-wideband ranging radio measurements from experimental trials. (b) Comparison of sequential allocation versus explicit coordination with two searchers using ranging radio measurements. As in simulated trials, sequential allocation greatly outperforms its lower bound. Error bars in both parts are one standard error of the mean (SEM), and the searchers move at a maximum speed of 0.3 m s^{-1} .

used for ground truth but was not used to assist in search. The Pioneer's maximum speed was set to 0.3 m s^{-1} , the maximum that provided consistent laser localization. Figure 7 shows a photograph of the Pioneer robot as well as the office environment used for testing, and Extension 2 shows animations of these trials.

After gathering data from the ultra-wideband ranging sensors, simulated searchers were added to the environment. These searchers have access to the ranging measurements from

the stationary nodes in the environment, which allows them to utilize real range data from the experiment to find the target in the simulated world. The searchers were given a maximum speed of 0.3 m s^{-1} to match that of the Pioneer target. Our recursive Bayesian measurement incorporation method was used in these trials. Computational limits allowed for subcell sizes of approximately $10 \times 10 \text{ cm}^2$.

Figure 8 shows the results for one and two searchers in these "hybrid" trials. As in purely simulated trials, the finite-

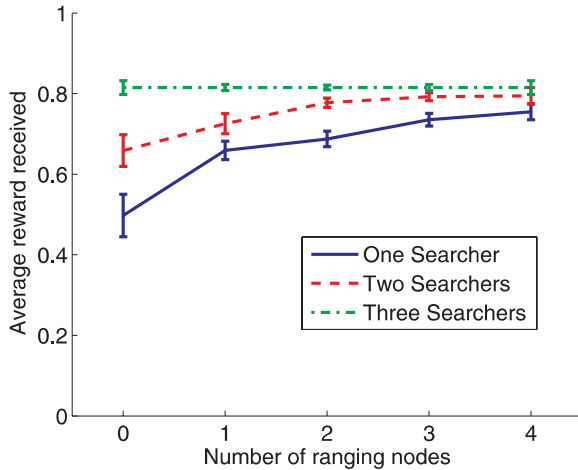


Fig. 9. Multiple-searcher scalability trials for finite-horizon path enumeration (lookahead five) and sequential allocation with ranging radio measurements. Error bars are one standard error of the mean (SEM). Target and searchers move at a maximum speed of 0.3 m s^{-1} . The zero-node case corresponds to the absence of radio measurements.

horizon path enumeration method provides nearly equivalent reward to the POMDP solution. Sequential allocation is competitive with explicit coordination in these results as well. These results show that our measurement incorporation framework is effective with real data from ultra-wideband ranging sensors.

Since the experiments are run in playback, we can vary the number of sensors used by turning off some sensors' data streams. Figure 9 shows average rewards using an increasing number of ranging radio nodes. The zero-node case corresponds to search without non-line-of-sight measurements. The results show that adding more searchers leads to decreasing capture times. Increasing the number of measurement beacons also leads to decreasing capture times. These results suggest that if a small number of searchers are available, this can be compensated for by using more measurement beacons, and vice versa. In this environment, adding more than three searchers and/or four measurement beacons no longer improves performance.

7.3. Relationship with Average Capture Time

The results presented above use the discounted reward metric. This is an important metric because it relates the efficient search problem to similar problems in probabilistic planning. In real scenarios, however, average capture time may be a more relevant metric. This section presents results showing that the proposed algorithm reduces average capture time as well as discounted reward.

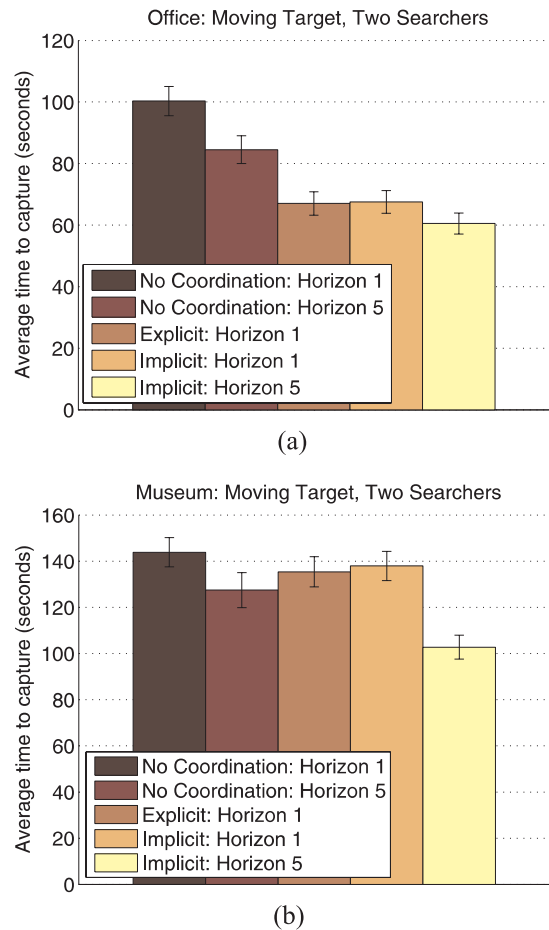


Fig. 10. Comparison of different coordination methods during efficient search. Results show that implicit coordination yields capture times competitive with explicit coordination. Two searchers were used in all trials. Error bars are one SEM and averages are over 200 trials. Target and searchers move at a maximum speed of 1 m s^{-1} .

Figure 10 shows a comparison of different coordination strategies and planning horizons in the museum and office using the average capture time metric. The results show that myopic planning (one-step lookahead) does not perform as well as using a five-cell lookahead. Further simulated testing (not shown), suggests little improvement in average capture times with lookahead greater than five cells. The results also show that implicit coordination provides low average capture times in these environments. Owing to its poor scalability, we could not run explicit coordination with a five-step lookahead. However, the improved scalability of implicit coordination allows for non-myopic planning and the lowest capture times.

For further comparison, Figure 11 shows average capture times for our proposed method (without non-line-of-sight measurements) against a random search strategy. In the ran-

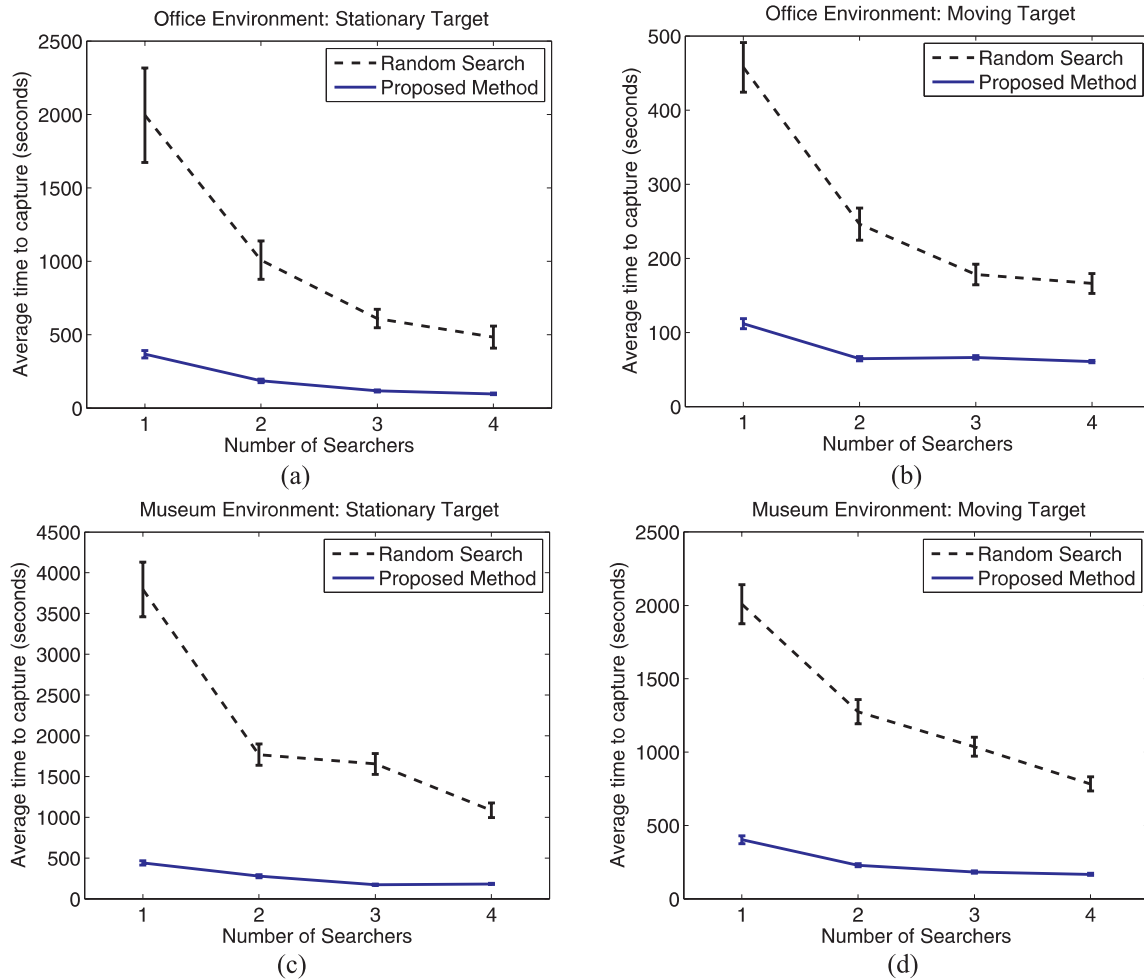


Fig. 11. Simulated coordinated search results comparing the proposed method to random search. The proposed method uses implicit coordination without non-line-of-sight measurements. Graphs show average capture times versus number of searchers for a stationary target (a) and a moving target (b). Error bars are one SEM, and averages are over 200 trials. Target and searchers move at a maximum speed of 1 m s^{-1} .

dom search strategy, the searchers randomly move between cells in the environments. The proposed method using implicit coordination and finite-horizon path enumeration yields nearly a factor of five improvement over the random strategy in many cases.

Figure 12 shows the average capture times from the “hybrid” trials with real data and simulated searchers. This graph shows average capture times using an increasing number of ranging radio nodes. The zero-node case corresponds to search without measurements. The results show that adding more searchers or increasing the number of measurement beacons leads to decreasing capture times. The graphs also show the point at which adding more searchers or measurements does not improve performance.

We also implemented the proposed framework on a single mobile robot searcher using the Player software (Gerkey

et al. 2003) in a laboratory environment. These results further confirm the trends above (Hollinger et al. 2007a).

8. Conclusions

In this paper we have presented a scalable algorithm for solving the MESPP problem of locating a non-adversarial target using multiple robotics searchers. We have defined the MESPP problem and shown how it can be modeled using a POMDP. We have also shown that current POMDP solvers are incapable of handling large instances of MESPP. Our proposed algorithm uses sequential allocation and finite-horizon path enumeration to remain computationally tractable for multiple searchers in large environments. We have given a rigorous theoretical analysis that shows the near-optimality of sequential

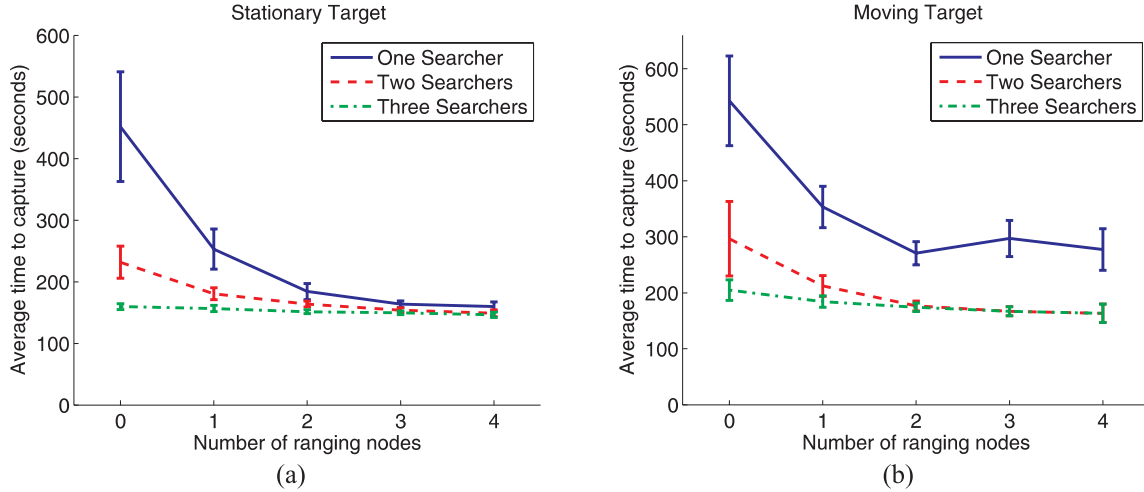


Fig. 12. Average capture times using ultra-wideband ranging radio measurements in an office environment. Simulated searchers find a real target using ultra-wideband ranging radio data. Searchers move at a maximum speed of 0.3 m s^{-1} . The x -axis shows the number of ranging nodes used for estimation. The zero-node case corresponds to search without range measurements. Results show that capture time decreases with more searchers and/or more measurement beacons. Error bars are one SEM.

allocation in this domain by exploiting the non-decreasing submodularity of the MESPP objective function. Sequential allocation is an instance of *implicit coordination* during which multiple robots share information rather than planning in the joint path space. Implicit coordination is linearly scalable, and it remains tractable in large problem instances when exponential methods using explicit coordination are far beyond computational limits. Our simulated and experimental results using ultra-wideband ranging radios show the performance of our algorithm in complex environments.

One extension is to apply our algorithm to the case where actions of searchers are no longer fully deterministic. For instance, a searcher may have a 50% chance of failing to move because of rubble blocking the way. The POMDP formulation of MESPP can easily express this scenario. The solution to this POMDP would no longer be a deterministic searcher path, but it would instead be a distribution over paths. Even though submodular set analysis does not apply directly, the resulting objective function on distributions over paths may still show qualities related to submodularity leading to theoretical guarantees for sequential allocation.

One important avenue for future work is a more comprehensive analysis of the communication requirement of implicitly coordinated solutions. For moderately sized teams, such as those examined in this paper, full broadcast communication is still tractable, but increasing team size can lead to a communication bottleneck. Distributed communication solutions have been proposed in the POMDP literature, which could potentially alleviate this problem for very large implicitly coordinated teams (Roth 2007).

Throughout this paper, we have made the assumptions that the target's motion model is both known and non-adversarial. We have shown that these assumptions lead to theoretical guarantees on the performance of sequential allocation during MESPP. If either of these assumptions are violated, the performance of our algorithm is no longer bounded. An extension would be to develop an algorithm robust to worst-case and unknown target motion models. This would require approximately solving the Multi-robot Guaranteed Search Path Planning (MGSP) problem. To the best of our knowledge, a scalable and bounded approximation algorithm for MGSP is still an open problem. However, methods utilizing implicit coordination show great promise in providing near-optimal solutions in the MGSP domain.

Acknowledgments

We thank Andreas Krause, Christopher Geyer, Benjamin Grocholsky and Bradley Hamner for their insightful comments. This work is supported by the National Science Foundation under Grant No. IIS-0426945.

Appendix A: Proof of Theorem 1

First, we give the formal definitions of “non-decreasing” and “submodular”. In what follows, we make use of the *time-augmented search graph* $G' = (N', E')$ as defined in Section 3. Here $\mathbf{P}(N')$ denotes the powerset of N' , i.e. the set of (time-stamped) node subsets.

Definition 1. A function $F : \mathbf{P}(N') \rightarrow \mathfrak{R}_0^+$ is called *non-decreasing* if and only if for all $A, B \in \mathbf{P}(N')$, we have

$$A \subseteq B \Rightarrow F(A) \leq F(B).$$

Definition 2. A function $F : \mathbf{P}(N') \rightarrow \mathfrak{R}_0^+$ is called *submodular* if and only if for all $A, B \in \mathbf{P}(N')$ and all singletons $C = \{(m, t)\} \in \mathbf{P}(N')$, we have

$$A \subseteq B \Rightarrow F(A \cup C) - F(A) \geq F(B \cup C) - F(B).$$

We are now ready to restate and prove Theorem 1.

Theorem 1. *The objective function optimized by the MESPP problem is $J(A)$, and this is a non-decreasing, submodular set function.*

Proof. It is immediate that $J(A)$ as defined in Equation (2) is simply another way to write $J(U(1), \dots, U(T))$ as defined in Equation (1). Let us now prove that it is non-decreasing and submodular.

In the following, for a given $Y \subseteq N'$ and any $A \subseteq N'$, we define $t_A = \min\{t : (m, t) \in A \cap Y\}$ and $F_Y(A) = \gamma^{t_A}$, with the understanding that $\gamma \in (0, 1)$, $\min \emptyset = \infty$, and $\gamma^\infty = 0$.

We first show that the function $F_Y(A)$ is non-decreasing for every $Y \subseteq N'$. Take an arbitrary Y , and fix it for the proof. Take any $A, B \subseteq N'$ and any $C = \{(m_0, t_0)\} \subseteq N'$. We have

$$t_A = \min\{t : (m, t) \in A \cap Y\},$$

$$t_B = \min\{t : (m, t) \in B \cap Y\},$$

$$t_C = \min\{t : (m, t) \in C \cap Y\},$$

and

$$A \subseteq B \Rightarrow \{t : (m, t) \in A \cap Y\} \subseteq \{t : (m, t) \in B \cap Y\}$$

$$\Rightarrow t_A \geq t_B \Rightarrow F_Y(A) = \gamma^{t_A} \leq \gamma^{t_B} = F_Y(B).$$

Hence, $F_Y(\cdot)$ is non-decreasing.

Regarding submodularity of $F_Y(A)$, note that, since C is a singleton, we have two cases: either $C \cap Y \neq \emptyset$ and so $t_C = t_0 < \infty$; or $C \cap Y = \emptyset$ and so $t_C = \infty$. We examine the two cases separately.

Case I: $t_C < \infty$. In this case we have three subcases.

1. We have $t_B \leq t_A \leq t_C$. Then $F_Y(B \cup C) = \gamma^{t_B}$, $F_Y(B) = \gamma^{t_B}$, $F_Y(A \cup C) = \gamma^{t_A}$, $F_Y(A) = \gamma^{t_A}$ and $F_Y(A \cup C) - F_Y(A) = \gamma^{t_A} - \gamma^{t_A} = 0 = \gamma^{t_B} - \gamma^{t_B} = F_Y(B \cup C) - F_Y(B)$.
2. We have $t_B \leq t_C \leq t_A$. Then $F_Y(B \cup C) = \gamma^{t_B}$, $F_Y(B) = \gamma^{t_B}$, $F_Y(A \cup C) = \gamma^{t_C}$, $F_Y(A) = \gamma^{t_A}$ and $F_Y(A \cup C) - F_Y(A) = \gamma^{t_C} - \gamma^{t_A} > 0 = \gamma^{t_B} - \gamma^{t_B} = F_Y(B \cup C) - F_Y(B)$.

3. We have $t_C \leq t_B \leq t_A$. Then $F_Y(B \cup C) = \gamma^{t_C}$, $F_Y(B) = \gamma^{t_B}$, $F_Y(A \cup C) = \gamma^{t_C}$, $F_Y(A) = \gamma^{t_A}$ and $F_Y(A \cup C) - F_Y(A) = \gamma^{t_C} - \gamma^{t_A} \geq \gamma^{t_C} - \gamma^{t_B} = F_Y(B \cup C) - F_Y(B)$, since $t_A \geq t_B \Rightarrow \gamma^{t_A} \leq \gamma^{t_B} \Rightarrow -\gamma^{t_A} \geq -\gamma^{t_B}$.

Case II: $t_C = \infty$. Then we have a single subcase: $t_B \leq t_A \leq t_C$, from which it follows that $F_Y(A \cup C) - F_Y(A) = 0 = F_Y(B \cup C) - F_Y(B)$ as already seen.

In every case the submodularity inequality holds.

For one searcher, the reward function $J(A)$ (where A is the searcher's path) is defined by

$$J(A) = \sum_{Y \in \Psi} P(Y) F_Y(A),$$

where the summation is over all possible target paths Ψ , and $P(Y)$ is the probability of the target taking path Y . The K -searcher reward function $J(A_1, \dots, A_K)$ (where A_k is the path of the k th searcher) is defined by

$$J(A_1 \cup \dots \cup A_K) = \sum_{Y \in \Psi} P(Y) F_Y(A_1 \cup \dots \cup A_K).$$

We now show that for any $K = 1, 2, \dots$ and every $Y \subseteq N'$, the function $J(A_1 \cup \dots \cup A_K)$ is non-decreasing and submodular. Non-decreasing submodularity is closed under non-negative linear combinations (and, hence, expectations). Here $F(A_1 \cup \dots \cup A_K)$ is the expected value of $F_Y(A)$, where $A = A_1 \cup \dots \cup A_K$, $F_Y(\cdot)$ is a non-decreasing submodular function, and the expectation is taken over all possible target paths. \square

Appendix B: Index to Multimedia Extensions

The multimedia extension page is found at <http://www.ijrr.org>

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Simulated coordinated search trials
2	Video	Coordinated search with ultra-wide-band ranging radio data

References

- Adler, M., Racke, H., Sivadasan, N. and Sohler, C. (2003). Randomized pursuit-evasion in graphs. *Combinatorics, Probability and Computing*, **12**(3): 225–244.
- Calisi, D., Farinelli, A., Locchi, L. and Nardi, D. (2007). Multi-objective exploration and search for autonomous rescue robots. *Journal of Field Robotics*, **24**(8–9): 763–777.

- Eaton, J. and Zadeh, L. (1962). Optimal pursuit strategies in discrete-state probabilistic systems. *Transactions of the ASME. Series D, Journal of Basic Engineering*, **62**: 23–28.
- Ferris, B., Fox, D. and Lawrence, N. (2007). WiFi-SLAM using Gaussian process latent variable models. *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2480–2485.
- Ferris, B., Hahnel, D. and Fox, D. (2006). Gaussian processes for signal strength-based location estimation. *Proceedings of the 2nd Robotics Science and Systems Conference*, pp. 303–310.
- Furukawa, T., Durrant-Whyte, H. and Lavis, B. (2007). The element-based method—theory and its application to Bayesian search and tracking. *Proceedings of the International Conference on Intelligent Robots and Systems*.
- Gerkey, B., Thrun, S. and Gordon, G. (2005). Parallel stochastic hill-climbing with small teams. *Proceedings of the 3rd International NRL Workshop on Multi-Robot Systems*.
- Gerkey, B., Thrun, S. and Gordon, G. (2006). Visibility-based pursuit-evasion with limited field of view. *The International Journal of Robotics Research*, **25**(4): 299–315.
- Gerkey, B., Vaughan, R. and Howard, A. (2003). The player/stage project: tools for multi-robot and distributed sensor systems. *Proceedings of the International Conference on Advanced Robotics*, pp. 317–323.
- Guestrin, C., Krause, A. and Singh, A. (2005). Near-optimal sensor placements in Gaussian processes. *Proceedings of the International Conference on Machine Learning*.
- Guibas, L., Latombe, J., LaValle, S., Lin, D. and Motwani, R. (1999). Visibility-based pursuit-evasion in a polygonal environment. *International Journal of Computational Geometry and Applications*, **9**(5): 471–494.
- Hollinger, G., Djughash, J. and Singh, S. (2007a). Coordinated search in cluttered environments using range from multiple robots. *Proceedings of the 6th International Conference on Field and Service Robotics*.
- Hollinger, G., Djughash, J. and Singh, S. (2008). Tracking a moving target in cluttered environments with ranging radios. *Proceedings of the International Conference on Robotics and Automation*.
- Hollinger, G., Kehagias, A. and Singh, S. (2007b). Probabilistic strategies for pursuit in cluttered environments with multiple robots. In *Proceedings of the International Conference on Robotics and Automation*.
- Hollinger, G. and Singh, S. (2008). Proofs and experiments in scalable, near-optimal search with multiple robots. *Proceedings of the Robotics: Science and Systems Conference*.
- Isler, V., Kannan, S. and Khanna, S. (2005). Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics*, **5**(21): 864–875.
- Kalra, N. (2006). A market-based framework for tightly-coupled planned coordination in multirobot teams. *PhD Thesis*, Carnegie Mellon University.
- Krause, A. and Guestrin, C. (2007). Near-optimal observation selection using submodular functions. *Proceedings of the 22nd Conference on Artificial Intelligence*.
- Krause, A., McMahan, B., Guestrin, C. and Gupta, A. (2007). Selecting observations against adversarial objectives. *Proceedings of Neural Information Processing Systems*.
- Kumar, V., Rus, D. and Singh, S. (2004). Robot and sensor networks for first responders. *IEEE Pervasive Computing*, **3**(4): 24–33.
- Lau, H., Huang, S. and Dissanayake, G. (2005). Optimal search for multiple targets in a built environment. *Proceedings of the International Conference on Intelligent Robots and Systems*.
- Lau, H., Huang, S. and Dissanayake, G. (2006). Probabilistic search for a moving target in an indoor environment. *Proceedings of the International Conference on Intelligent Robots and Systems*.
- Megiddo, N., Hakimi, S., Garey, M., Johnson, D. and Papadimitriou, C. (1988). The complexity of searching a graph. *Journal of the ACM*, **35**: 18–44.
- Parsons, T. (1976). Pursuit-evasion in a graph. *Theory and Applications of Graphs*, Alavi, Y. and Lick, D. (eds). Berlin, Springer, pp. 426–441.
- Roth, M. (2007). Execution-time communication decisions for coordination of multi-agent teams. *PhD Thesis*, Carnegie Mellon University.
- Roy, N., Gordon, G. and Thrun, S. (2005). Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research*, **23**: 1–40.
- Sarmiento, A., Murrieta-Cid, R. and Hutchinson, S. (2004). A multi-robot strategy for rapidly searching a polygonal environment. *Proceedings of the 9th Ibero-American Conference on Artificial Intelligence*.
- Singh, A., Krause, A., Guestrin, C., Kaiser, W. and Batalin, M. (2007). Efficient planning of informative paths for multiple robots. *Proceedings of the International Joint Conference on Artificial Intelligence*.
- Singh, J. S. and Wagh, M. D. (1987). Robot path planning using intersecting convex shapes: analysis and simulation. *IEEE Journal of Robotics and Automation* **3**(2): 101–108.
- Smith, T. (2007). Probabilistic planning for robotic exploration. *PhD Thesis*, Carnegie Mellon University.
- Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*. Cambridge, MA, The MIT Press.