Aggregation-based Approaches to Honey-pot Searching with Local Sensory Information

Bhaskar DasGupta[*] João P. Hespanha[†] Eduardo Sontag[‡]

September 6, 2003

Abstract

We investigate the problem of searching for a hidden target in a bounded region of the plane, by an autonomous robot which is only able to use limited local sensory information. We propose an aggregation-based approach to solve this problem, in which the continuous search space is partitioned into a finite collection of regions on which we define a discrete search problem. A solution to the original problem is then obtained through a refinement procedure that lifts the discrete path into a continuous one. The resulting solution is in general not optimal but one can construct bounds to gauge the cost penalty incurred.

Keywords: Aggregation/refinement, Hybrid systems, Optimal Search.

1 Introduction

The problem addressed concerns searching for a hidden target by an autonomous robot. Suppose that a "honey-pot" is hidden in a bounded region $\mathcal{R}$ (typically a subset of the plane $\mathbb{R}^2$ of the 3-dimensional space $\mathbb{R}^3$). The exact position $\mathbf{x}^*$ of the honey-pot is not known but we do know its probability density f. The goal is to find the honey-pot using a point robot that moves in $\mathcal{R}$ and is able to see only a "small region" around it. If the robot get "sufficiently close," it will detect the honey-pot and the search is over. Given some constraint on the time or fuel spent by the robot, one would like to find a path that *maximizes the probability* of finding the honey-pot.

To formalize this problem, let us denote by $\mathcal{S}[x] \subset \mathcal{R}$ the set of points in $\mathcal{R}$ that the robot can see from some position $x \in \mathcal{R}$. Our search problem can then be defined as follows.

Problem 1 (Continuous Constrained Honey-pot Search (CHS)). Find a continuously differentiable path $\rho : [0, T] \to \mathcal{R}$, $T > 0$ with $\|\dot{\rho}(t)\| \leq 1$, $\forall t \in [0, T]$ starting at $\rho(0) = \rho_0 \in \mathcal{R}$, that maximizes the probability of finding the honey-pot

$$R[\rho] := \int_{x \in \mathcal{S}_{\text{path}}[\rho]} f(x)dx, \tag{1}$$

[*]Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7053. Email: dasgupta@cs.uic.edu. Supported by NSF grants CCR-0296041, CCR-0206795 and CCR-0208749.

[†]Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560. Email: hespanha@ece.ucsb.edu. Supported by NSF grant ECS-0242798.

[‡]Department of Mathematics, Rutgers University, New Brunswick, NJ 08903. Email: sontag@hilbert.rutgers.edu. Supported in part by NSF grant CCR-0206789.

where $\mathcal{S}_{\text{path}}[\rho] := \{x \in \mathcal{R} : x \in \mathcal{S}[\rho(t)] \text{ for some } t \in [0, T]\}$ denotes the set of all points that the robot can scan along the path ρ; subject to a constraint of the form

$$C[\rho] := \int_0^T c(\rho(t))dt \leq L, \tag{2}$$

where L denotes a positive constant. □

For bounded-time searches, the function $c : \mathcal{R} \to (0, T)$ in (2) is constant and equal to 1 over the maximum time allowed for the search. For bounded-fuel searches, c should be equal to the fuel-consumption rate, normalized so that the total fuel available is equal to L. This rate could be position dependent, e.g., if the terrain is not homogeneous. One can also "encode" obstacles in c by making this function take large values in regions to be avoided.

In a more general version of this problem, the region that the robot sees from a position x and the incremental cost c may depend on the robot's "orientation." In this case, we would define $\mathcal{S}[x, v] \subset \mathcal{R}$ to be the set of points in $\mathcal{R}$ that the robot sees when it is at position $x \in \mathcal{R}$ with orientation v, and define

$$\mathcal{S}_{\text{path}}[\rho] := \{x \in \mathcal{R} : x \in \mathcal{S}[\rho(t), \dot{\rho}(t)] \text{ for some } t \in [0, T]\}.$$

The constraint (2) could also be generalized to

$$C[\rho] := \int_0^T c(\rho(t), \dot{\rho}(t))dt \leq L.$$

This would be useful, e.g., in fuel-constrained searches on a non-flat 2-dimensional surface for which the fuel consumption depends on whether the robot is moving up or down.

The origin of the honey-pot search problem can be traced back to the pioneering work of Stone [9]; see also [10] and the references therein for a summary of part of this work with motivating applications in search operations by the U.S. Navy. Recently, Hespanha et al. [7] considered probabilistic approaches to a more difficult type of a searching problem, where the honey-pot is mobile (trying to avoid being captured) and proposes a greedy strategy that leads to capture with probability one. However, there is no claim of optimality or ϵ-optimality there. The above formulation assumes that the distribution f for the position of the honey-pot is know. Typically, this would be obtained via some *a priori* "map-learning" phase (e.g., see [6, 11, 12] for the general case and [5] for the simpler rectilinear case).

The continuous CHS Problem 1 does not fit in classical Calculus of Variations formulations because the integral in (1) is not computed along the path. In fact, this problem seems to be significantly more difficult than, e.g., shortest-path problems because its discretized version has been shown to be NP-hard [4]. We pursue here an approximate solution to CHS Problem that is based on a discretization of the continuous problem. We start by aggregating the continuous search space $\mathcal{R}$ into a finite collection of regions on which we define a discrete search problem. From the solution of this problem, we can then recover a solution of the original problem through a refinement procedure that lifts the discrete path into a continuous one. The solution obtained is in general not optimal but one can construct bounds to gauge the cost penalty incurred.

Our work is inspired by discrete abstraction of hybrid system (cf. survey [1]), where the behavior of a system with a state-space that has both discrete and continuous components is abstracted to a purely discrete system to reduce the complexity. In our problem, the original system has no discrete components but we still reduced it to a discrete system by an abstraction procedure. A key difference between the results here and, e.g., those summarized in [1] is that in general our abstraction procedure introduces some degradation in performance because the discretized system does not capture all the details of the original system. In particular, some information about the distribution of the honey-pot may be lost in the abstraction. However, by allowing some performance degradation we can significantly enlarge the class of problems for which the procedure is applicable.

The remaining of this paper is organized as follows. The aggregation-based approach to solve the continuous CHS Problem 1 is outlined in Section 2. This requires the definition of a discrete Aggregate Reward Budget(ARB) problem in Section 2.1, which turns out to be NP-hard. In Section 2.2 we show how the solution to a particular instance of the discrete ARB problem can be refined to provide a feasible solution to the continuous CHS problem with some guaranteed reward. In Section 2.3 we provide a different instance of the ARB problem that provides an upper-bound on the best achievable reward for the continuous CHS problem, which can be used to estimate the cost penalty incurred by the aggregation procedure. In Section 3, we prove a few properties of the optimal solution to the ARB problem that can significantly decrease the search space and also reduce the conservativeness of our approach. Finally, Section 4 contains a brief conclusion and directions for future research.

2 Aggregation

We pursue here a aggregation/refinement-based approach to solve the continuous CHS Problem 1. The starting point for this approach is a *partition* V of the region $\mathcal{R}$, i.e., V is a collection of subsets of $\mathcal{R}$ such that $\bigcup_{v \in V} v = V$, and $v \cap v' = \emptyset$ for every $v \neq v' \in V$. We use this partition to reduce the problem-space to a discrete set as follows:

1. We define a discrete constrained-search problem that seeks for a path consisting of a finite sequence of regions in V, satisfying an appropriate cost-constraint and maximizing an appropriately defined reward.

2. We refine the discrete path into a continuous one that is guaranteed to satisfy (2) and have a probability of finding the honey-pot at least as large as the reward obtained for the discrete problem.

To obtain the desired properties for the refinement, the selection of the cost and reward of the discrete problem must take into account the criteria (1), the constraint (2), and the refinement procedure. This is because the cost penalty introduced by the aggregation approach depends not only on the choice of the partition V but also on the cost and reward used for the discrete optimization.

In this paper we take the partition V as given. However, it will become clear that this partition should have a few desirable properties so as to make minimize the cost penalty introduced by the aggregation procedure.

2.1 Aggregate Reward Budget Problem

To define the discrete constrained-search problem we assume given a *path-refinement algorithm* $\mathfrak{R}$ that takes a finite sequence $(v_1, v_2, \ldots, v_N)$ of regions in V (possibly with the same region appearing multiple times) and produces a continuously differentiable path $\rho : [0, T] \to \mathcal{R}$, $T > 0$, $\|\dot{\rho}(t)\| \leq 1$, $\forall t \in [0, T]$. For simplicity we assume that the algorithm operates recursively generating ρ as follows:

$$\rho_0 : [0, 0] \to \mathcal{R}, \quad \rho_0(0) = \rho_0, \qquad\qquad \rho_k = \mathfrak{E}(\rho_{k-1}, v_k), \quad \forall k \in \{1, 2, \ldots, N\}, \tag{3}$$

where $\mathfrak{E}$ "extends" the *partial-path* $\rho_{k-1} : [0, T_{k-1}] \to \mathcal{R}$ to the next partial-path $\rho_k : [0, T_k] \to \mathcal{R}$, with $T_k \geq T_{k-1}$, ρ_{k-1} equal to ρ_k on $[0, T_{k-1}$, and $\rho_k(T_k) \in v_k$. The refined path ρ is then selected to be the final partial-path ρ_N.

To construct the cost/reward structure of the discrete problem we place the following requirement on the path-refinement algorithm.

Assumption 1 (Refinement Requirement). There are functions $c_{\text{worst}} : V \times V \to [0, \infty)$, $r_{\text{worst}} : V \to [0, \infty)$, $|\cdot|_{\text{worst}} : V \to \mathbb{N}$ such that, given a partial-path ρ_{k-1} that ends in a region v_{k-1} and a new region v_k, the next partial-path $\rho_k = \mathfrak{E}(\rho_{k-1}, v_k)$ satisfies

$$R[\rho_k] \geq R[\rho_{k-1}] + r_{\text{worst}}(v_k) \tag{4}$$
$$C[\rho_k] \leq C[\rho_{k-1}] + c_{\text{worst}}(v_{k-1}, v_k). \tag{5}$$

However, (4) only needs to hold for the first $|v_k|_{\text{worst}}$ times that the refinement algorithm is asked to extend the path to the region v_k. □

The requirement above can be informally expressed as: (i) the first $|v|_{\text{worst}}$ times that a region v appears in the discrete path a reward of $r_{\text{worst}}(v)$ is collected and (ii) each transition from region v to v' results in an extra cost of $c_{\text{worst}}(v, v')$. Note that in general $c_{\text{worst}}(v, v) > 0$. When these properties hold one can estimate worst-case lower and upper bounds on the reward and cost, respectively, that will be obtained for the refined continuous path. Moreover, one can optimize the discrete path to make these bounds as favorable as possible. This motivates the following Aggregated Reward Budget (ARB) graph-optimization problem:

Problem 2 (Aggregated Reward Budget (ARB)).

Instance: Given $\langle G, s, c, r, |\cdot|, L\rangle$, where $G = (V, E)$ denotes a graph with vertex set V and edge set E, $s \in V$ is a specified vertex, $c : E \to [0, \infty)$ is an edge cost function, $r : V \to [0, \infty)$ is a vertex reward function, $|\cdot| : V \to \mathbb{N}$ is a vertex cardinality function, and L a positive integer.

Valid Solution: A (possibly self-intersecting) path $p = (v_0 = s, v_2, \ldots, v_k)$ in G with $v_i \in V$ such that $\sum_{i=1}^{k} c(v_{i-1}, v_i) \leq L$.

Objective: *maximize* the *total reward*

$$\sum_{v \in p} r(v) \, \min\{|v|, \#(p, v)\}, \tag{6}$$

where $\#(p, v)$ denotes the number of times that the vertex v appears in the path p.

To make the aggregation/refinement procedure efficient, one would like the bounds in (4)–(5) to be tight. The construction of a "good" path refinement algorithms is simple when the probability density f used to define the reward and the function c used to define the cost are essentially constant within each region. In this case, one could simply break each region v in $|v|_{\text{worst}}$ disjoint cells chosen so that the robot could scan an whole cell from a single point (perhaps its center). Each time the path needs to be extended to v the continuous path would be taken to a cell not yet visited and one would collect a reward $r_{\text{worst}}(v)$ equal to the area of the cell times the (constant) probability density over the region. This reward would be collected until there are no more unvisited cells, i.e., at most $|v|_{\text{worst}}$ times. The costs $c_{\text{worst}}(v, v')$ could be obtained from shortest-path optimizations between the most unfavorable cells in the regions v and v'. The order in which the cells in a particular region v are selected could be chosen to approximately minimize $c_{\text{worst}}(v, v')$. This is straightforward when the regions in V have regular shapes and one can "sweep" the region. We will return to this issue later.

There are also good reasons to want the number of regions in V to be small. It turns out that it is still computationally difficult to solve exactly the ARB Problem 2. However, it can be solved efficiently when the number of regions is small or when one is willing to simply find an approximate solution to it. The following result establishes the computational complexity of this problem.

Lemma 1. *The ARB Problem 2 is NP-hard even when $r(v) = 1$ for every $v \in V$ and $c(e) = 1$ for every $e \in E$.*

Proof of Lemma 1. Given an instance $\langle G, s, c, r, L\rangle$ of the RB problem in [4], the instance $\langle G, s, c, r, |\cdot|, L\rangle$ of the ARB problem with $|v| = 1$, $\forall v \in V$ admits the same feasible paths and yields the same reward for each feasible path. We can therefore regard the RB problem as a special case of the ARB problem with constant vertex cardinality. Therefore the computational complexity of the ARB problem is at least as high as that of the RB problem.

On the other hand, given an instance $\langle G, s, c, r, |\cdot|, L\rangle$ of the ARB problem, we can construct an instance $\langle \bar{G}, \bar{s}, \bar{c}, \bar{r}, L\rangle$ of the RB problem, where the graph $\bar{G} = (\bar{V}, \bar{E})$ is obtained by expanding each vertex v in the graph $G = (V, E)$ into $|v|$ vertices with reward $r(v)$, internally connected by edges with cost $c(v, v)$. Moreover, for any edge in G between nodes v_1 and v_2, the graph $\bar{G}$ must have edges with cost $c(v_1, v_2)$ between all vertices that result from the expansion of v_1 and all nodes that results from the expansion of v_1. The start node $\bar{s}$ can be any node that resulted from the expansion of s. With this construction, for any feasible path of the RB problem we can construct a feasible path for the ARB problem with the same reward and vice versa. We can therefore also see the ARB problem as a special case of (a larger) the RB problem with $\sum_{v\in V} |v|$ nodes and $\sum_{v\in V} |v|\big(|v| - 1\big) + \sum_{(v_1,v_2)\in E} |v_1|\,|v_2|$ edges. Therefore the complexity of the RB problem at least as high as that of the ARB problem. The result then follows from Lemma 1 in [4]. ∎

2.2 Suboptimal solution–lower bound on the feasible reward

Given a partition V of the region $\mathcal{R}$ and a path-refinement algorithm $\mathfrak{R}$ satisfying Assumption 1, we can construct an instance $\langle G, s, c_{\text{worst}}, r_{\text{worst}}, |\cdot|_{\text{worst}}, L\rangle$ of the ARB Problem 2 by defining $G = (V, E)$ to be a fully connected graph whose vertices are the regions in V, s to be the region that contains ρ_0, and taking from Assumption 1 the edge cost, the vertex reward, and the vertex cardinality functions. The ARB problem just defined is said to be *worst-case induced by the partition V and the path refinement algorithm $\mathfrak{R}$*. As hinted above, we can use a solution to a worst-case induced ARB problem to generate a path that is feasible for the original continuous CHS Problem 1 and exhibits some guaranteed reward:

Lemma 2. *Consider an instance $\langle G, s, c_{\text{worst}}, r_{\text{worst}}, |\cdot|_{\text{worst}}, L\rangle$ of the worst-case ARB problem induced by a partition V and a path refinement algorithm $\mathfrak{R}$. Let $p = (v_1 = s, v_2, \ldots, v_k)$ be a feasible path for $\langle G, s, c_{\text{worst}}, r_{\text{worst}}, |\cdot|_{\text{worst}}, L\rangle$ and suppose one constructs a continuous path ρ*

using the path refinement algorithm $\mathfrak{R}$. The path ρ satisfies the constraint (2) *and its reward $R[\rho]$ is at least as large as that of p. Thus, if p is optimal for the ARB problem,*

$$R^*[\rho_0, L] \geq R[\rho] \geq R^*_{\text{worst}}[\rho_0, L],$$

where $R^*[\rho_0, L]$ and $R^*_{\text{worst}}[\rho_0, L]$ denote the optimal rewards for the continuous CHS and the worst-case induced ARB problems, respectively.

Proof of Lemma 2. The feasibility of p steams directly from the recursive construction in (3) together with the cost-bound provided by (5), from which one concludes that the cost of ρ does not exceed the cost of p, which is upper bounded by 1. As for the reward, take some $v \in V$ and let $\#(p, v)$ denote the number of times that v appears in the path p. The first $\min\{|v|, \#(p, v)\}$ times that the partial-paths are extended to the region v, the reward will increase by $r_{\text{worst}}(v)$. This will contribute to the total reward of ρ by at least $r_{\text{worst}}(v) \min\{|\mathbf{v}|, \#(p, v)\}$. Adding over all v in the path p, we conclude that the total reward of ρ must be no smaller than the total reward of p given by (6). ■

2.3 Upper bound on the feasible reward

We formulate next another ARB problem that can be used to construct a bound to gauge how far from the optimal a path generated using the worst-case induced ARB problem is.

Given a partition V of the region $\mathcal{R}$ and a positive integer k, we can construct an instance $\langle G, s, c_{\text{best}}, r_{\text{best}}, |\cdot|_{\text{best}}, L\rangle$ of the ARB problem by defining $G = (V, E)$ to be a fully connected graph whose vertices are the regions in V; s to be the region that contains ρ_0; each edge cost $c_{\text{best}}(v, v')$, $v, \mathbf{v}' \in V$ to be a either a lower-bound on the cost incurred in going from a point in v to a point in v' or L/k, whichever is greater; each vertex reward $r_{\text{best}}(v)$, $v \in V$ to be an upper bound on the maximum reward for an instance of the continuous CHS problem starting from any position in v with a cost bounded by L/k; and each vertex cardinality $|v|_{\text{best}}$ to be an upper-bound on

$$\frac{\int_{x \in S[v]} f(x) dx}{r_{\text{best}}(v)},$$

where $S[v]$ denotes the set of all points that can be reached from the region v with a cost not exceeding L/k. The ARB problem just defined is said to be *best-case induced by the partition V and the cost-bound L/k.*

Computing tight bound for the functions that define best-case induced ARB problems can be as hard as solving the original continuous CHS problem. However, also here when the probability density f used to define the reward and the function c used to define the cost are essentially constant within each region this task becomes much simpler because within a region optimal paths are straight lines.

We can use a solution to the best-case induced ARB problem to generate an upper bound on the achievable reward.

Lemma 3. *Given an instance $\langle G, s, c_{\text{best}}, r_{\text{best}}, |\cdot|_{\text{best}}, L\rangle$ of the best-case ARB problem induced by a partition V and the cost-bound L/k,*

$$R^*_{\text{best}}[\rho_0, L] \geq R^*[\rho_0, L], \tag{7}$$

where $R^*[\rho_0, L]$ *and* $R^*_{\text{best}}[\rho_0, L]$ *denote the optimal rewards for the continuous CHS and the best-case induced ARB problems, respectively.*

Proof of Lemma 3. Let $\rho : [0, T] \to \mathcal{R}$ be a path for the continuous CHS problem that satisfies the cost constraint (2) and achieves a reward larger or equal than $R^*[\rho_0, L] - \delta$ for some small $\delta \geq 0$ [1]. One can then pick a sequence of reals $t_0 := 0 < t_1 < \cdots < t_k := T$ such that

$$\int_{t_{i-1}}^{t_i} c\big(\rho(t)\big)dt = \frac{L}{k}, \qquad \forall i \in \{1, 2, \dots, k\}.$$

Suppose now that we define a path $p = (v_0 = s, v_2, \dots, v_k)$, where each v_i denotes the region on which $\rho(t_i)$ lies. This sequence is admissible for the best-case ARB problem because from the definition of c_{best} we conclude that

$$\sum_{i=1}^{k} c_{\text{best}}(v_{i-1}, v_i) \leq \sum_{i=1}^{k} \max\left\{\frac{L}{k}, \int_{t_{i-1}}^{t_i} c\big(\rho(t)\big)dt\right\} = L.$$

As for the reward, let $\bar{\mathcal{S}}[v]$ denote the set

$$\bar{\mathcal{S}}[v] := \big\{x \in \mathcal{R} : x \in \mathcal{S}[\rho(t)] \text{ for some } t \in [t_i, t_{i+1}),\ v_i = v\big\}.$$

Since $\mathcal{S}[\rho] = \cup_{v \in p} \bar{\mathcal{S}}[v]$, we have that

$$R[\rho] = \int_{x \in \cup_{v \in p} \bar{\mathcal{S}}[v]} f(x)dx \leq \sum_{v \in p} \int_{x \in \bar{\mathcal{S}}[v]} f(x)dx. \tag{8}$$

Let $\#(p, v)$ denote the number of times that v appears in the path p. From the fact that the points $\rho(t_i)$ are separated by path-segments with costs no larger than L/k and the definition of $r_{\text{best}}(v)$, we conclude that

$$\int_{x \in \bar{\mathcal{S}}[v]} f(x)dx \leq r_{\text{best}}(v)\, \#(p, v). \tag{9}$$

On the other hand, since $\bar{\mathcal{S}}[v] \subset \mathcal{S}[v]$ we also obtain from the definition of $|v|_{\text{best}}$ that

$$\int_{x \in \bar{\mathcal{S}}[v]} f(x)dx \leq \int_{x \in \mathcal{S}[v]} f(x)dx \leq r_{\text{best}}(v)\, |v|_{\text{best}}. \tag{10}$$

From (8)–(10) we conclude that

$$R[\rho] \leq \sum_{v \in p} r_{\text{best}}(v)\, \min\big\{|v|_{\text{best}}, \#(p, v)\big\}.$$

Since left hand side of the above inequality is larger or equal than $R^*[\rho_0, L] - \delta$ and the right-hand-side is the reward of an admissible path for the best-case ARB problem, we conclude that

$$R^*[\rho_0, L] - \delta \leq R[\rho] \leq \sum_{v \in p} r_{\text{best}}(v)\, \min\big\{|v|_{\text{best}}, \#(p, v)\big\} \leq R^*_{\text{best}}[\rho_0, L].$$

Inequality (7) follows since δ can be made arbitrarily close to zero. ■

[1] The need for $\delta > 0$ only arises when the optimal reward cannot be achieved for any admissible path.

3 Solution to the ARB problem

As shown above, the ARB problem is computationally very difficult. In this section we prove a few properties of the optimal solution to the ARB problem that can significantly decrease the search space. We consider instances $\langle G, s, c, r, |\cdot|, L\rangle$ of the ARB problems that are *subadditive*, meaning that the graph $G = (V, E)$ is fully connected and

$$c(v_1, v_2) + c(v_2, v_3) \geq c(v_1, v_3) + c(v_4, v_4), \qquad \forall v_1, v_2, v_3, v_4 \in V.$$

The worst-case induced ARB problems introduced above are often subadditive. As shown in the following Lemma, the search space for optimal paths can be significantly reduced for subadditive ARB problems.

Lemma 4. *The maximum achievable reward for the ARB problem does not increase, if we restrict valid paths $p := \{v_1, v_2, \ldots, v_N\}$ to satisfy:*

(a) If $v_i = v_j$, $i < j$ then $v_k = v_i$ for every $k \in \{i, i+1, \ldots, j\}$.

(b) The number of times $\#(p, v)$ that a vertex $v \in V$ appears in p never exceeds $|v|$.

(c) If a vertex $v \in V$ appears in p and

$$r(v) > \min_{v_i \in p} r(v_i),$$

then the number of times $\#(p, v)$ that v appears in p is exactly $|v|$.

Proof of Lemma 4. For (a), consider a path p with total cost $C[p]$ for which $v_i = v_j$, $i < j$ but $v_{j-1} \neq v_i$. If we then construct a path

$$p' := \{v_1, v_2, \ldots, v_{i-1}, v_i, v_j = v_i, v_{i+1}, \ldots, v_{j-1}, v_{j+1}, \ldots, v_N\},$$

it has exactly the same reward as p and, because of subadditivity, its cost $C[p']$ satisfies

$$C[p'] = C[p] - \big(c(v_{j-1}, v_j) + c(v_j, v_{j+1})\big) + c(v_i, v_i) + c(v_{j-1}, v_{j+1}) \leq C[p].$$

Since p' has the same reward as p and no worse cost, it will not increase the maximum achievable reward for the ARB problem. By induction, we conclude that any path for which (a) does not hold will also not improve the maximum achievable reward for the ARB problem.

For (b), consider a path p with total cost $C[p]$ for which v_i already appeared in p at least $|v_i|$ times before i. If we then construct a path

$$p' := \{v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_N\},$$

it has exactly the same reward as p and, because of subadditivity, its cost $C[p']$ satisfies

$$C[p'] = C[p] - \big(c(v_{i-1}, v_i) + c(v_i, v_{i+1})\big) + c(v_{i-1}, v_{i+1}) \leq C[p].$$

Since p' has the same reward as p and no worse cost, it will not increase the maximum achievable reward for the ARB problem. By induction, we conclude that any path in which v appears more than $|v|$ times will also not improve the maximum achievable reward for the ARB problem.

For (c), consider a path p with total reward $R[p]$ and total cost $C[p]$ for which the vertex v_i appears less than $|v_i|$ times and there is a vertex v_j such that $r(v_i) > r(v_j)$. If we then construct a path

$$p' := \{v_1, v_2, \dots, v_{i-1}, v_i, v_i, v_{i+1}, \dots, v_{j-1}, v_{j+1}, \dots, v_N\},$$

its reward $R[p']$ satisfies

$$R[p'] \geq R[p] - r(v_j) + r(v_i) > R[p].$$

and, again because of subadditivity, its cost $C[p']$ satisfies

$$C[p'] = C[p] - (c(v_{j-1}, v_j) + c(v_j, v_{j+1})) + c(v, v) + c(v_{j-1}, v_{j+1}) \leq C[p].$$

Since p' has better reward and no worse cost than p, it will not increase the maximum achievable reward for the ARB problem. By induction, we conclude that any path for which (c) does not hold will also not improve the maximum achievable reward for the ARB problem. ■

Lemma 4 allows one to reduce significantly the complexity of finding the optimal solution to subadditive ARB problems. In fact, we simply have to determine in which order one needs to visit the different vertices (without repetitions). Note that once an order has been chosen, the time spent on each vertex is uniquely determined. However, the problem still seems to have a combinatorial flavor.

Lemma 4 also simplifies considerably the construction of a refinement algorithm $\mathfrak{R}$ that satisfies Assumption 1, with bounds (4)–(5) not overly conservative. Note that in the iterative path construction, the path extension $\mathfrak{E}$ will only be called with sequences p for which each $\mathbf{n}_k$ only appears multiple times back to back (typically $|v|$ times). Assuming that the regions in the partition V have "regular" shapes one should be able to get very tight bounds at least a in (5).

4 Conclusions

We presented an aggregation-based approach to a continuous search algorithm using local sensory information. We start by aggregating the continuous search space $\mathcal{R}$ into a finite collection of regions on which we define a discrete search problem. A solution to the original problem is obtained through a refinement procedure that lifts the discrete path into a continuous one. The solution obtained is in general not optimal but one can construct bounds to gauge the cost penalty incurred.

We are currently working on algebraic algorithms that produce partitions of the continuous search space for which the procedure proposed in this paper results in a small cost penalty. These algorithms are inspired by the results in [8] on state aggregation in Markov chains. We are also working on approaches to compute suboptimal solutions to the continuous CHS problem that are based on approximate solutions to the k-MST problem found in [2, 3]. These are not aggregation-based and will provide a term of comparison.

References

[1] R. Alur, T. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proc. of IEEE*, 88(2):971–984, July 2000.

[2] S. Arora. Polynomial-time approximation schemes for euclidean traveling salesman and other geometric problems. *J. of the ACM*, 45(5):753–782, 1998.

[3] S. Arora and G. Karakostas. A $2 + \epsilon$ approximation for the k-MST problem. In *Proc. of the 11th ACM-SIAM Symp. on Discrete Algorithms*, pages 754–759, 2000.

[4] B. DasGupta, J. P. Hespanha, and E. Sontag. Honey-pot searching with local sensory information. Submitted to publication, July 2003.

[5] X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment I: The rectilinear case. *J. of the ACM*, 45(2):215–245, Mar. 1998.

[6] J. P. Hespanha and H. H. Kızılocak. Efficient computation of dynamic probabilistic maps. In *Proc. of the 10th Mediterranean Conf. on Contr. and Automat.*, July 2002.

[7] J. P. Hespanha, M. Prandini, and S. Sastry. Probabilistic pursuit-evasion games: A one-step Nash approach. In *Proc. of the 39th Conf. on Decision and Contr.*, volume 3, pages 2272–2277, Dec. 2000.

[8] R. Phillips and P. Kokotović. A singular perturbation approach to modeling and control of markov chains. *IEEE Trans. on Automat. Contr.*, 26(5):1087–1094, Oct. 1981.

[9] L. D. Stone. *Theory of Optimal Search*. Academic Press, 1975.

[10] L. D. Stone. The process of search planning: Current approaches and continuing problems. *Operations Research*, 31:207–233, 1983.

[11] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.

[12] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* and *Autonomous Robots* (joint issue), 31(5): 29–53, 1998.