Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

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

Bhaskar DasGupta João P. Hespanha Eduardo Sontag

***Abstract*— We investigate the problem of searching for a hidden target in a bounded region by an autonomous agent that 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.**

I. Introduction

The problem addressed concerns searching for a hidden target by an autonomous agent. Suppose that a "honey-pot" is hidden in a bounded region $\mathcal{R}$ (typically a subset of the plane $\mathbb{R}^2$ or 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 *a-priori* probability density f. The goal is to find the honey-pot using an agent (called the *searcher*) that moves in $\mathcal{R}$ and is able to see only a "small region" around it. If the searcher get "sufficiently close," it will detect the honey-pot and the search is over. We assume that the target is stationary and that the probability density f does not change in time. Honey-pot searching is thus an (open-loop) path planning problem where one seeks a path that *maximizes the probability* of finding the honey-pot, given some constraint on the time or fuel spent by the searcher.

To formalize this problem, let $\mathcal{S}[x] \subset \mathcal{R}$ denote the set of points in $\mathcal{R}$ that the searcher can see from some position $x \in \mathcal{R}$. The so called *cookie cutter detection* corresponds to the special case where $\mathcal{S}[x]$ consists of a circle with fixed radius around x [9], but here we consider general detection regions.

Problem 1: Continuous Constrained Honey-pot Search (cCHS) 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 $\rho(0) = \rho_{\text{init}} \in \mathcal{R}$, that maximizes the probability of finding the honey-pot given by

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

subject to a constraint of the form

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

where L denotes a positive constant and $\mathcal{S}_{\text{path}}[\rho] := \big\{x \in \mathcal{R} : x \in \mathcal{S}[\rho(t)] \text{ for some } t \in [0,T]\big\}$ the set of all points that the searcher can scan along the path ρ. □

In the definition (1) of the reward, we have an integral over the area $\mathcal{S}_{\text{path}}[\rho]$ and *not a line integral along the path* ρ. The distinction may seems subtle but it is quite fundamental because if a searcher transverses the same location multiple times, a line integral would increase with each passage but the region $\mathcal{S}_{\text{path}}[\rho]$ that the searcher scans does not. This formulation does not prevent the path from returning to a point previously visited (which could be necessary) but does not reward the searcher for scanning the same location twice.

For bounded-time searches, $c(x) = 1$, $\forall x \in \mathcal{R}$ and L is the maximum time allowed for the search. For bounded-fuel searches, $c(\cdot)$ is the fuel-consumption rate and L is the total fuel available. The consumption rate can be position dependent when the terrain is not homogeneous. One can also "encode" obstacles in $c(\cdot)$ by making this function take large values in regions to be avoided.

The honey-pot search problem is inspired by the optimal search theory initiated by the pioneering work of Koopman [12] and later further developed by Stone [19] and others. A summary of this work can be found in the surveys [8, 20]. The development of search theory was motivated by U.S. Navy operations during the Second World War, which included the search for targets in transit, setting up sonar screens, and protection against submarine attacks [13]. In the context of Naval operations, search theory has been used more recently in search and rescue operations by the U.S. Coast Guard [8], as well as to detect lost objects such as the H-bomb lost in the Mediterranean coast of Spain in 1966, the wreck of the submarine USS Scorpion in 1968, or the unexploded ordnance in the Suez Canal following the 1973 Yom Kippur war. However, its application spans many other areas such as the clearing of land mines, locating parts in a warehouse, etc. The collection of papers [10] discusses several applications of search theory ranging from medicine to mining.

Until the 70s, most of the work in search theory decoupled the problems of finding the area that should be searched from that of finding a specific path for the searcher "covering" that area. This is sensible when (i) the cost-bound in

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

Dept. of Electrical and Computer Engineering, Univ. of California, Santa Barbara, CA 93106-9560. Email: hespanha@ece.ucsb.edu. Supported by NSF grants CCR-0311084, ECS-0242798.

Dept. of Mathematics, Rutgers Univ., New Brunswick, NJ 08903. Email: sontag@hilbert.rutgers.edu. Supported in part by NSF grant CCR-0206789.

0-7803-8335-4/04/$17.00 ©2004 AACC

(2) essentially poses a constraint on the total area that can be scanned and (ii) the optimal area turns out to be sufficiently regular so that one can find a continuous path ρ that sweeps it without overlaps. However, these assumptions generally only hold for time-constrained searches and Gaussian (or at least unimodal) *a-priori* target distributions. Complex distributions for f are likely to arise in many practical problems as discussed in [8].

More recently several researchers considered the so called *constrained search problem*, where it is explicitly taken into account it must be possible to "cover" the area to be scanned using one or more searchers moving along continuous paths. Mangel [15] considered continuous search problems where the goal is to determine an optimal path that either maximizes the probability of finding the target in a finite time interval or minimizes the infinite-horizon expected time needed to find the target. In Mangel's formulation this is reduced to an optimal control problem on the searcher's velocity $\dot{\rho}$, subject to a constraint in the form of a partial differential equation. In practice, this problem can only be solved for very simple *a-priori* target distributions.

An alternative approach that proved more fruitful consisted of discretizing time and partitioning the continuous space into a finite collection of cells. The search problem is then reduced to deciding which cell to visit at each time interval. Constraints on the searcher's motion can be expressed by only allowing it to move from one cell to adjacent ones [19]. At least when the time horizon is finite (and in some cases even when the time horizon is infinite [14]), the resulting optimal discrete search problem can be solved by finite enumeration of all possible solutions. However, this method scales poorly (exponentially!) with the number of cells. Eagle [6] noted that a discrete search can be formulated as an optimization on a partially observable Markov decision process (POMDP) and proposed a dynamic programing solutions to it. However, since the optimization of POMDPs is computationally very difficult, this approach is often not practical. Instead, Eagle and Yee [7], Stewart [17, 18] formulated the discrete search as a nonlinear integer programming problem and proposed branch-and-bound procedures to solve it, which in the case of [7] are optimal. Hespanha et al. [11] proposed a (non-optimal) but computationally efficient greedy strategy that leads to capture with probability one, but no claims of optimality are made. DasGupta et al. [5] proposed polynomial-time solutions to the discrete search problem that are also not guaranteed to find the optimal path but instead to find a feasible path with reward no smaller than $1/5$ of the optimal (in the worst case). The references [6, 7, 11, 17, 18] above considered the general case of a moving target, but as noted by Trummel and Weisinger [21] even the case of a stationary target is NP-Hard.

We pursue here an approximate solution to the cCHS Problem 1 that is also 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 to 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. In general, the solution obtained is not optimal for the original cCHS. A fundamental distinction between the work reported here and previous one on discrete search is that we provide bounds on how much performance degradation is introduced by the aggregation/refinement procedure. These bounds can in principle be used to determine partitions of $\mathcal{R}$ that minimize the performance degradation.

Our work is inspired by discrete abstraction of hybrid system, 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 (cf. survey [1]). 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 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 cCHS Problem 1 is outlined in Section II. This requires the definition of a discrete Aggregate Reward Budget (dARB) problem in Section II-A, which turns out to be NP-hard. In Section II-B we show how the solution to a particular instance of the dARB problem can be refined to provide a feasible solution to the cCHS problem with some guaranteed reward. In Section II-C we provide a different instance of the dARB problem that provides an upper-bound on the best achievable reward for the cCHS problem, which can be used to estimate the cost penalty incurred by the aggregation procedure. In Section III, we prove a few properties of the optimal solution to the dARB problem that can significantly decrease the search space and also reduce the conservativeness of our approach. Finally, Section IV contains a brief conclusion and directions for future research.

II. Aggregation

We pursue an aggregation/refinement-based approach to solve the cCHS Problem 1. The starting point 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.

A. 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 $\rho : [0,T] \to \mathcal{R}$, $T > 0$, $\|\dot{\rho}(t)\| \leq 1$, $\forall t \in [0,T]$. The algorithm operates recursively generating ρ as follows: It starts with the zero-length path

$$\rho_0 : [0,0] \to \mathcal{R}, \quad \rho_0(0) = \rho_{\text{init}},$$

and iteratively extends it according to

$$\rho_k = \mathfrak{E}(\rho_{k-1}, v_k), \qquad \forall k \in \{1, 2, \ldots, N\}, \tag{3}$$

where the operator $\mathfrak{E}$ "extends" the *partial-path* ρ_{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 the final 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 Requirements): 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 extended 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 at least $r_{\text{worst}}(v)$ is collected and (ii) each transition from region v to v' results in an added cost of at most $c_{\text{worst}}(v, v')$. 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 graph-optimization problem:

Problem 2: Discrete Aggregated Reward Budget (dARB)
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, $s \in V$ an initial vertex, $c : E \to [0,\infty)$ an edge cost function, $r : V \to [0,\infty)$ a vertex reward function, $|\cdot| : V \to \mathbb{N}$ 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 into $|v|_{\text{worst}}$ disjoint cells chosen so that the searcher 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 good reasons to want the number of regions in V to be small. In general it is computationally difficult to solve exactly the dARB 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 is a consequence of results in [5] and establishes the computational complexity of this problem.

Lemma 1: The dARB Problem 2 is NP-hard, even when $r(v) = 1$, $\forall v \in V$ and $c(e) = 1$, $\forall e \in E$. □

B. Suboptimal solution—lower bound on the 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 dARB Problem 2 by defining $G = (V, E)$ to be a fully connected graph whose vertices are the regions in V, s the region that contains ρ_{init}, and taking from Assumption 1 the edge cost, the vertex reward, and the vertex cardinality functions. The dARB 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 dARB problem to generate a path that is feasible for the original cCHS Problem 1 and exhibits some guaranteed reward:

Theorem 1: Consider an instance $\langle G, s, c_{\text{worst}}, r_{\text{worst}}, |\cdot|_{\text{worst}}, L\rangle$ of the worst-case dARB problem induced by a partition V and a path refinement algorithm $\mathfrak{R}$. Let $p = (v_1 = s, v_2, \dots, v_k)$ 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 dARB problem,

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

where $R^*[L]$ and $R^*_{\text{worst}}[L]$ denote the optimal rewards for the cCHS and the worst-case induced dARB problems, respectively. □

Proof: [Theorem 1] The feasibility of ρ 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 L. 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 $r_{\text{worst}}(v)\,\min\{|v|, \#(p, v)\}$. 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). ■

C. Upper bound on the reward

We formulate next another dARB problem that can be used to construct a bound to gauge how far from the optimal a path generated using the worst-case induced dARB 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 dARB 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_{init}; each edge cost $c_{\text{best}}(v, v')$, $v, v' \in V$ 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 cCHS Problem 1 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

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

where $S[v]$ denotes the set of all points that can be scanned from the region v with a cost not exceeding L/k. The dARB 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 dARB problems can be as hard as solving the original cCHS 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 optimal paths are straight lines.

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

Theorem 2: Given an instance $\langle G, s, c_{\text{best}}, r_{\text{best}}, |\cdot|_{\text{best}}, L\rangle$ of the best-case dARB problem induced by a partition V and the cost-bound L/k,

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

where $R^*[L]$ and $R^*_{\text{best}}[L]$ denote the optimal rewards for the cCHS and the best-case induced dARB problems, respectively. □

Proof: [Theorem 2] Let $\rho : [0, T] \to \mathcal{R}$ be a path for the cCHS problem that satisfies the cost constraint (2) and achieves a reward larger than or equal to $R^*[L] - \delta$ for some small $\delta \geq 0$ [1]. One can then pick a sequence of reals $t_0 := 0 < t_1 < \dots < t_k := T$ 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 dARB problem because from the definition of c_{best} we conclude that

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

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

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

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

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

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

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

$$\int_{\bar{\mathcal{S}}[v]} f(x)dx \leq \int_{\mathcal{S}[v]} f(x)dx \leq r_{\text{best}}(v)\,|v|_{\text{best}}. \quad (9)$$

From (8)–(9) we conclude that

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

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

$$\begin{aligned} R^*[L] - \delta &\leq R[\rho] \\ &\leq \sum_{v \in p} r_{\text{best}}(v)\, \min\left\{|v|_{\text{best}}, \#(p, v)\right\} \leq R^*_{\text{best}}[L]. \end{aligned}$$

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

III. Solution to the dARB problem

In Lemma 1 we saw that the dARB problem is computationally difficult. In this section we prove a few properties of the optimal solution to the dARB problem that can significantly decrease the search space. We consider instances $\langle G, s, c, r, |\cdot|, L\rangle$ of the dARB problems that are *subadditive*, meaning that the graph $G = (V, E)$ is fully connected and

$$c(v_1, v_2) + c(v_2, v_3) \geq c(v_1, v_3) + c(v_4, v_4),$$

$\forall v_1, v_2, v_3, v_4 \in V$. When $c(v, v) = 0$, $\forall v \in V$, this simply expresses a triangular inequality. In fact a similar assumption is made in [14]. Typically, the worst-case induced dARB problems introduced above are subadditive. In this case, the search space for optimal paths can be significantly reduced.

Theorem 3: The maximum achievable reward for a subadditive dARB problem does not increase if we restrict the 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: [Theorem 3] 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

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

(v_j, which is equal to v_i, was moved to right after v_i), p' has exactly the same reward as p and, because of subadditivity, its cost $C[p']$ satisfies

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

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

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

(v_i was removed), p' has exactly the same reward as p and, because of subadditivity, its cost $C[p']$ satisfies

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

Since p' has the same reward as p and no worse cost, it will not increase the maximum achievable reward for the dARB 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 dARB 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

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

(v_j was replaced by an extra v_i, right after the original one), 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

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

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

Theorem 3 allows one to reduce significantly the complexity of solving subadditive dARB problems. In fact, we simply have to determine in which order one needs to visit the different vertices (without repetitions). The problem still seems to have a combinatorial flavor but now only an enumeration of the ordering is needed, because the time spent on each vertex is uniquely determined once an order has been chosen.

Theorem 3 also simplifies considerably the construction of a refinement algorithm $\mathfrak{R}$ that satisfies Assumption 1, with tight bounds (4)–(5). Because of this theorem, the path extension $\mathfrak{E}$ will only be called with sequences p for which each v 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 in (5).

IV. Conclusions

We presented an aggregation-based approach to the cCHS Problem 1. 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 [16] on state aggregation in Markov chains. We are also working on approaches to compute suboptimal solutions to the dARB problem that are based on approximate solutions to the k-MST problem found in [2, 3]. Preliminary results are available in [5]. Another avenue for future research is the search for a mobile honey-pot, or more generally, the case where the probability distribution f is not constant.

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] D. V. Chudnovksy and G. V. Chudnovksy, editors. *Search Theory: Some Recent Developments*, volume 112 of *Lecture Notes in Pure and Applied Mathematics*. Marcel Dekker, New York, 1989.

[5] B. DasGupta, J. P. Hespanha, and E. Sontag. Computational complexity of honey-pot searching with local sensory information. In *Proc. of the 2004 Amer. Contr. Conf.*, June 2004.

[6] J. N. Eagle. Optimal search for a moving target when the search path is constrained. *Operations Research*, 32(5):1107–1115, 1984.

[7] J. N. Eagle and J. R. Yee. An optimal branch and bound procedure for the constrained path, moving target search problem. *Operations Research*, 38(1): 110–114, 1990.

[8] J. R. Frost and L. D. Stone. Review of search theory: Advances and applications to search and rescue decision support. Technical Report CG-D-15-01, U.S. Coast Guard Research and Development Center, Gronton, CT, Sept. 2001. URL `http://www.rdc.uscg.gov/reports/2001/cgd1501dpexsum.pdf`.

[9] S. Gal. Continuous search games. In Chudnovksy and Chudnovksy [4], pages 33–53.

[10] K. B. Haley and L. D. Stone, editors. *Search theory and Applications*, volume 8 of *NATO Conference Series*. Plenum Press, New York, 1980.

[11] 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.

[12] B. O. Koopman. Search and screening. Technical Report Operations Evaluations Group Report N0. 56, Center for Naval Analyses, Alesandria, VA, 1946.

[13] B. O. Koopman. *Search and Screening: General Principles and Historical Applications*. Pergamon Press, New York, 1980.

[14] U. Lössner and I. Wegener. Discrete sequential search with positive switch cost. *Mathematics of Operations Research*, 7(3):426–440, Aug. 1982.

[15] M. Mangel. Search theory: A differential equations approach. In Chudnovksy and Chudnovksy [4], pages 55–101.

[16] 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.

[17] T. J. Stewart. Search for a moving target when searcher motion is restricted. *Computers and Operations Research*, 6:129–140, 1979.

[18] T. J. Stewart. Experience with a branch and bound algorithm for constrained searcher motion. In Haley and Stone [10], pages 247–253.

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

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

[21] K. E. Trummel and J. R. Weisinger. The complexiy of the optimal searcher path problem. *Operations Research*, 34(2):324–327, Apr. 1986.