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

Computational Complexities of Honey-pot Searching with Local Sensory Information

Bhaskar DasGupta João P. Hespanha Eduardo Sontag

Abstract

In this paper 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 formalize a discrete version of the problem as a "reward-collecting" path problem and provide efficient approximation algorithms for various cases.

I. Introduction and Motivation

The problem addressed in this paper concerns searching for a hidden evader 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 the probability density f of $\mathbf{x}^*$. 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 a finite amount of time T, which translates into a finite-length path for 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 is then as follows.

Problem 1 (Continuous Honey-pot Search). Find a continuously differentiable path $\rho : [0,T] \to \mathcal{R}$, with $\|\dot{\rho}(t)\| \leq 1$ for all $t \in [0,T]$, that minimizes $P_c[\rho] = \int_{x \in \mathcal{S}_{\text{path}}[\rho]} f(x)dx$, 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 points that the robot can scan along the path ρ.

In the above formulation of the problem, the implicit assumption is that it is possible to "insert" the robot at an optimal starting point. This formulation is appropriate for problems in which a fast movement (not in "search mode") to a desired location is possible, such as in land rescue missions where a team is deposited by air at a starting point. A variant of the problem would be that in which the initial point is fixed. Our hardness results still apply to this variant, but the approximation results need further elaboration and will be addressed in future research. In an even more general version of this problem, the region that the robot sees from the position x 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 origin of this 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, Kim and Sastry [3] considered probabilistic approaches to a more difficult type of searching problem where the agents are 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. In the above formulation of our problem, the assumption is that the region in which the search takes place is known via some a priori "map-learning" phase (e.g., see [4, 11, 12] for the general case and [2] for the simpler rectilinear case).

A. Discrete Search

Problem 1 can be discretized by breaking $\mathcal{R}$ into a finite number of tiles $\{\mathcal{R}_k \subset \mathcal{R} : k \in \mathcal{K}\}$, where $\mathcal{K}$ is a finite index set. Typically, the tiles are rectangular or hexagonal forming a regular lattice. We assume that the size of the tiles is chosen so that when the robot is located at the center of one tile it can scan the whole tile in one unit of time. One can then restrict the search to paths that go from tile to tile, remaining on each tile for one unit of time. Let $p_k = \int_{\mathcal{R}_k} f(x)dx$ denote the probability that the honey-pot is in the k^{th} tile. Then, the probability that the honey-pot will be found as the robot follows a path ρ, defined by a sequence of tiles $\sigma = \{k_1, k_2, \ldots, k_N\}$, is given by $P_d[\sigma] = \sum_{k \in \Sigma} p_k$ where Σ is the set of distinct elements in the sequence σ. The time needed to transverse the path is given by $T[\sigma] := \sum_{i=1}^{N-1} t_{k_i,k_{i+1}}$ where $t_{k_i,k_{i+1}}$ denotes the time it takes for the robot to move from tile k_i to tile k_{i+1}. The discretized version of the problem can then be

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.

formalized as follows:

Problem 2 (Discrete Honey-pot Search). Find a sequence of tiles $\sigma := \{k_1, k_2, \ldots, k_N\}$ that maximizes $P_d[\sigma] := \sum_{k\in\sigma} p_k$ subject to the constraint that $T[\sigma] := \sum_{i=1}^{N-1} t_{k_i,k_{i+1}} \le T$.

B. A Summary of Our Results

Motivated by Problem 2, we formulate the following Reward Budget (RB) problem on graphs:

Problem 3 (Reward Budget (RB)).

- Instance: $\langle G, c, r, L\rangle$, where L is an integer and $G = (V, E)$ is a graph with an edge cost function $c : E \to [0, \infty)$ and a vertex reward function $r : V \to [0, \infty)$.
- Valid Solution: A (possibly self-intersecting) path $p = (v_1, v_2, \ldots, v_k)$ in G with $v_i \in V$ such that $C[p] := \sum_{i=1}^{k-1} c(v_i, v_{i+1}) \le L$.
- Objective: maximize the total reward $R[p] = \sum_{v\in P} r(v)$ where P denotes the set of vertices in p.

For any reasonable application of the RB problem to the actual honey-pot search problem, the underlying graph must at least be planar; an interesting case of significance is when G is a unit grid, i.e. $V = \{(i, j) \in [0, m-1] \times [0, n-1]\}$, $E = \{\{(i, j), (k, \ell)\} : |i - j| + |k - \ell| = 1\}$ and $c(e) = 1$ for every $e \in E$. An ε-approximate solution (or simply an ε-approximation) of a maximization problem is defined to be a solution with an objective value no smaller than $1/\varepsilon$ times the value of the optimum. A list of our main results are as follows[1]:

Lemma 1. The RB problem is NP-hard even when (a) $r(v) = 1$ for every $v \in V$, $c(e) = 1$ for every $e \in E$ and the graph G is planar bipartite with the maximum degree of any vertex being 3, or (b) G is a unit grid graph and $r(v) \in \{0, 1\}$ for every vertex v.

Theorem 1. (a) For any constant $\varepsilon > 0$, an r-approximate solution to the RB problem can be found in polynomial time where $r = \begin{cases} 2+\varepsilon & \text{if } c(e)=1 \text{ for every } e\in E \\ 5+\varepsilon & \text{otherwise} \end{cases}$.
(b) If $r(v) = 1$ for every $v \in V$ and $c(e) = 1$ for every $e \in E$, then a 2-approximate solution to the RB problem can be computed in $O(|V| + |E|)$ time.

C. A Summary of Proof Techniques Used

For the NP-hardness results in Lemma 1 we use the results of Itai et al. [5] and Garey et al. [6].

[1]At the time of submission, we were informed that the $5+\varepsilon$-approximation result in Theorem 1(a) was independently obtained by several other researchers in a paper that will be published in another conference in the future.

To prove the results in Theorem 1(a), we first need to consider a dual version of the RB problem and show that a good approximate solution to the dual problem translates to a corresponding good approximate solution of the RB problem via a binary search similar to that by Johnson et al. [8], path decompositions and Eulerian tours via doubling edges. To solve this dual problem, we need to use the $(2+\varepsilon)$-approximation results on the k-MST problem by Arora and Karakostas [1] that builds upon the 3-approximation results on the same problem by Garg [7].

The result in Theorem 1(b) can be proved via DFS and Eulerian tours with doubled edges.

II. Proofs

In this section, we provide proofs of Lemma 1 and Theorem 1. For the purpose of investigating computational complexities of the RB problem, it will be convenient to consider the Reward Quota (RQ) problem, which can be intuitively thought of as a dual of the RB problem. The RQ problem is defined as follows:

Problem 4 (Reward Quota (RQ)).

- Instance: $\langle G, s, c, r, R\rangle$, 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, and R a positive integer.
- Valid Solution: A (possibly self-intersecting) path $p = (v_1 = s, v_2, \ldots, v_k)$ in G with $v_i \in V$ such that $R[p] := \sum_{v\in P} r(v) \ge R$ where P denotes the set of vertices in the path p.
- Objective: minimize the total cost $C[p] := \sum_{i=1}^{k-1} c(v_i, v_{i+1})$.

For computational complexity results we restrict cost and reward values in the above problems to take integer values. The decision problem for RB (as required for NP-hardness proofs) provides an additional real number R and asks if there is a valid solution of total reward at least R. An ε-approximate solution (or simply an ε-approximation) of a maximization problem is a solution with an objective value no smaller than $1/\varepsilon$ times the value of the optimum. Analogously, an ε-approximation of a minimization problem is a solution with an objective value no larger than ε times the value of the optimum. We will also use the following notations/conventions consistently throughout the rest of the paper (unless otherwise stated explicitly):

- $\mathrm{OPT}_{RB}(L)$ (or, simply $\mathrm{OPT}(L)$) denotes the optimum value of the objective function for a given instance $\langle G, s, c, r, L\rangle$ of the RB problem as a function of L.
- $\mathrm{OPT}_{RQ}(R)$ denotes the optimum value of the objective function for a given instance $\langle G, s, c, r, R\rangle$ of the RB problem as a function of R,

Although the RB and RQ problems defined above seem to be novel, there are a few related problems that will be useful to us:

The budget and quota problems in [8]:
: These are similar to the problems defined above, except that they only searched for a subtree (vertex induced subgraph with no cycles), which may not necessarily be a path[2]. Nonetheless many of the ideas there are also of use to us.

Metric k-TSP
: An input to the metric k-traveling salesman problem problem is a weighted graph in which the edge costs satisfy the triangle inequality, i.e., for every three vertices u, v and w the edges costs satisfy the inequality $c(u, v) + c(v, w) \geq c(u, w)$. The goal is to produce a simple (i.e., non-self-intersecting) cycle of minimum total edge cost visiting at least k vertices that includes a specified vertex s. The authors in [1] provide a $2 + \varepsilon$-approximate solution to this problem via a primal-dual schema.

A. Proof of Lemma 1 (NP-hardness Results)

We prove part (a) as follows. The Hamiltonian path problem for a graph G is NP-complete even if G is planar bipartite with maximum degree of any vertex being 3 [5]. One can see that setting $r(v) = 1$ for every $v \in V$, $c(e) = 1$ for every $e \in E$ and $L = n - 1$, G has a Hamiltonian path if and only if $\mathrm{OPT}(n-1) = n - 1$.

We can also prove part (b) as follows. It is known that the Hamiltonian path problem is NP-hard for graphs which are vertex-induced subgraphs of a unit grid [5]. Given an instance I of the Hamiltonian path problem on such graphs, we consider a corresponding instance I' of the RB problem in which the graph is a minimal unit grid graph $G' = (V', E')$ containing the vertex-induced subgraph $G = (V, E)$ of I,

$$r(v) = \begin{cases} 1 & \text{if } v \in V \\ 0 & \text{otherwise} \end{cases}$$

and $L = |V| - 1$. Obviously, if I has a Hamiltonian cycle then I' has a solution with a maximum total reward of $|V|$. Conversely, suppose that I' has a solution with a maximum total reward of $|V|$. Then, this solution must visit all the vertices in V using $|V| - 1$ edges and thus it is a Hamiltonian path of G.

B. Proof of Theorem 1 (Approximation Algorithms)

A proof of the theorem is provided in the next three subsections.

[2]By doubling every edge in a subtree and finding an Eulerian tour on the new graph, we do get a self-intersecting cycle; however, the total edge cost of such a path is twice that of the given subtree.

1) Relating Path and Cycle Versions of RB and RQ: For the purpose of designing efficient algorithms for the RB problem, it would be more convenient sometimes to consider this problem for a (possibly self-intersecting) cycle, i.e., a (possibly self-intersecting) path that starts and ends on the same vertex s. The following lemma states that these two versions of this problem are essentially similar in their approximability issues.

Lemma 2. Assume that we have a polynomial time r-approximation algorithm (for some constant $r \geq 1$) for the version of RB in which we are seeking a cycle instead of a path. Then, for any constant $\varepsilon > 0$, there is a polynomial time $(r + \varepsilon)$-approximation algorithm for RB.

Proof: [Proof of Lemma 2.] Let OPT and OPT′ denote the optimum solutions for the path and the cycle versions of the RB problem and let A' be the solution returned by the r-approximate algorithm for the cycle version of the RB problem. Hence, $A' \geq \frac{1}{r}\,\mathrm{OPT}'$. Set $x = 1 + \frac{r}{\varepsilon}$. Since x is a constant, we can solve the RB problem for paths involving at most x edges in polynomial time (e.g., by checking all $\binom{|E|}{x}$ subsets of x edges for a possible solution). Otherwise, an optimal solution for RB for paths (and hence for cycles as well) involves more than x edges. Thus, by removing the least cost edge from the solution for A', we get a solution $A > \left(1 - \frac{1}{x}\right) A'$ of the RB problem for paths. Since $\mathrm{OPT} \leq \mathrm{OPT}'$, we get

$$\frac{A}{\mathrm{OPT}} > \left(1 - \frac{1}{x}\right) \frac{A'}{\mathrm{OPT}'} \geq \left(1 - \frac{1}{x}\right) \frac{1}{r} = \frac{1}{r + \varepsilon} \quad \blacksquare$$

2) Relating the RQ Problem to the RB Problem: In this section, we show that an approximate solution to the RQ problem can be used to provide an approximate solution to the RB problem.

Lemma 3. Assume that we have a polynomial time r-approximation algorithm for RQ for some constant $r \geq 1$. Then, for any constant $\varepsilon > 0$, there is a polynomial time approximation algorithm for the RB problem that returns a solution with a total reward of at least $\frac{1}{1+\varepsilon}\,\mathrm{OPT}\left(\frac{1}{r} L\right)$.

Proof: [Proof of Lemma 3.] Our proof is similar in nature to a proof in [8] that related the budget and quota problems for connected subtrees of a graph. We can do a binary search over the range $[0, \sum_{i=1}^{n} r(v_i)]$ for the total reward in the algorithm for the RQ problem to provide us in polynomial time a total reward value A such that the solution obtained by the approximation algorithm has a cost of at most L if the total reward is at least A, but has a cost greater than L if the total reward is at least $A(1+\varepsilon)$. We then output the solution to the RQ problem with a total reward of at least A as our approximate solution to the RB problem. By choice of A, an optimal solution to the RQ problem with a total reward of at least $A(1+\varepsilon)$ must have a total cost greater than $\frac{1}{r}L$. Hence $\mathrm{OPT}\left(\frac{1}{r}L\right) \leq A(1+\varepsilon)$. ∎

Remark 1. Clearly Lemma 3 holds when we consider the cycle versions of both the RB and the RQ problems.

3) Completing a Proof of Theorem 1: The following proposition is needed in our proof of Theorem 1 to relate suboptimal solutions of the RB problem (for either the path or the cycle version).

Proposition 1.

(a) $\mathrm{OPT}(\frac{1}{2}L) \geq \frac{1}{3}\,\mathrm{OPT}(L)$.

(b) $\mathrm{OPT}(\frac{1}{2+\varepsilon}L) \geq \frac{1}{5}\,\mathrm{OPT}(L)$ for any $\varepsilon > 0$.

(c) Assume that $c(e) = 1$ for all $e \in E$. Then, $\mathrm{OPT}(\lceil \frac{L}{k} \rceil) \geq \frac{1}{k}\,\mathrm{OPT}(L)$ for any integer $1 \leq k \leq L$.

Proof: [Proof of Proposition 1.] To prove (a), assume that $p = (v_1, v_2, \ldots, v_k)$ be an optimal solution with a total reward of $\mathrm{OPT}(L)$. If $r(v_i) \geq \frac{1}{3}\,\mathrm{OPT}(L)$ then obviously our claim is true. Otherwise, starting with the first edge in p, partition p into two disjoint subpaths such that total reward of the first subpath is greater than $\frac{1}{3}\,\mathrm{OPT}(L)$ but the total reward of the first subpath excluding its last edge is at most $\frac{1}{3}\,\mathrm{OPT}(L)$. It follows then that the total reward of the first subpath is less than $\frac{2}{3}\,\mathrm{OPT}(L)$ and hence the total reward of the second subpath is at least $\frac{1}{3}\,\mathrm{OPT}(L)$. At least one the two subpaths must have a total cost of at most $\frac{1}{2}L$. The proof of (b) if similar to that of (a).
To prove (c), note that an optimal path of total cost L can be partitioned into k disjoint paths each of total cost at most $\lceil \frac{L}{k} \rceil$. At least one of these paths must have a total reward of at least $\frac{1}{k}\,\mathrm{OPT}(L)$. ∎

We prove Theorem 1(a) by using the results in [1] together with applications of Lemma 3 (via Remark 1) and Proposition 1. In the sequel, we refer to the cycle version of the RQ and RB problems. Implicitly replace every vertex v with $r(v) > 0$ by one original vertex v' of reward 0 (which will be connected to other vertices in the graph) and an additional $r(v)$ vertices, each of reward 1, connected to v' with edges of zero costs. Obviously, an optimal solution of the RQ problem in the original graph remains an optimal solution of the RQ problem in this new graph.

Now we run the polynomial time $(2+\mu)$-approximation algorithm for the metric R-TSP problem in [1] on our new graph (with s as the specified vertex). We observe the following:

- The new zero-cost edges connected to a vertex v is dealt implicitly in one step in the TREEGROW step in [1] by coalescing the vertices together. This ensures that the algorithm indeed runs in polynomial time (as opposed to pseudo-polynomial time).
- The metric property of the graph is necessary to avoid self-intersection of the solution path in the R-TSP problem. Since the RQ problem allows self-intersection, we do not need the metric property.
- The authors in [1] solve the problem by finding a solution of the R-MST problem, doubling every edge and then taking short-cuts (to avoid self-intersection) to get exactly R vertices. However, for the RQ problem we do not count the reward of the same vertex twice, hence taking a simple Eulerian tour on the solution of the R-TSP problem with every edge doubled does not change the total reward.

As a result, we are able to obtain a in polynomial time a $(2+\mu)$-approximate solution (for any constant $\mu > 0$) to the RQ problem. Applying Lemma 3 (via Remark 1) with a constant $\rho > 0$, we can obtain in polynomial time a solution to the RB problem that returns a total reward of at least $\frac{1}{1+\rho}\,\mathrm{OPT}\left(\frac{L}{2+\mu}\right)$. We can now apply Proposition 1 to get a $(5+5\rho)$-approximation for the general case and a $((1+\rho)(2+\mu))$-approximation for the case when $c(e) = 1$ for all e (for the cycle version of the RB problem). Finally, using Lemma 2 to relate path and cycle versions of the RB problem, we can obtain, for any constant $\nu > 0$, a $(5+5\mu+\nu)$-approximation for the general case and a $((1+\rho)(2+\mu)+\nu)$-approximation for the case when $c(e) = 1$ for all e for the RB problem. The result now follows by setting the constants ρ, μ and ν appropriately.

To prove Theorem 1(b) we do a depth-first-search (DFS) on G starting at s that computes a DFS tree. Now we replace every edge in the DFS tree by two edges, compute an Eulerian cycle and output the path consisting of the first L edges starting at s in this Eulerian cycle. Obviously, $\mathrm{OPT}(L) \leq L$. Since we replaced each undirected edge by two directed edges, we collect a total reward of at least $\frac{L}{2} + 1$. ∎

Acknowledgements. We would like to thank Robert Sloan and György Turán for pointing out reference [5] and for helpful discussions.

References

[1] S. Arora and G. Karakostas. A $2+\varepsilon$ approximation for the k-MST problem, 11^{th} ACM-SIAM Symposium on Discrete Algorithms, 754-759, 2000.

[2] X. Deng, T. Kameda and C. H. Papadimitriou. How to learn and unknown environment I: The rectilinear case, JACM, 45 (2), 215-245, March 1998.

[3] J. P. Hespanha, H. J. Kim and S. Sastry. Multiple-Agent Probabilistic Pursuit-Evasion Games, Proceedings of the 38th IEEE Conference of Decision and Control, 3, 2432-2437, 1999.

[4] J. P. Hespanha and H. Kizilocak. Efficient Computation of Dynamic Probabilistic Maps, 10th Mediterranean Conference on Control and Automata, July 2002.

[5] A. Itai, C. H. Papadimitriou and J. L. Szwarcfiter. Hamiltonian Paths in Grid Graphs, SIAM Journal of Computing, 11 (4), 676-686, November 1982.

[6] M. R. Garey, D. S. Johnson and R. E. Tarjan. The planar Hamiltonian circuit problem is NP-complete, SIAM Journal of Computing, 5, 704-714, 1976.

[7] N. Garg. A 3-approximation for the minimum tree spanning k vertices, 37^{th} Annual Symposium on Foundations of Computer Science, 302-309, 1996.

[8] D. S. Johnson, M. Minkoff and S. Phillips. The prize collecting Steiner tree problem: theory and practice, 11th ACM-SIAM Symposium on Discrete Algorithms, 760-769, 2000.

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

[10] L. D. Stone. The Process of Search Planning: Current Approaches and Continuing Problems, Operation Research, 31, 207-233, 1983.

[11] S. Thrun. Robotic Mapping: A Survey, in Exploring Artificial Intelligence in the New Millennium, G. Lakemeyer and B. Nebel (eds.), 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), 1-25, 1998.