Computational Complexities of Honey-pot Searching with Local Sensory Information

Bhaskar DasGupta‡

Department of Computer Science

University of Illinois at Chicago

Chicago, IL 60607

Email: dasgupta@cs.uic.edu

João P. Hespanha†

Department of Electrical & Computer Engineering

University of California

Santa Barbara, CA 93106-9560

Email: hespanha@ece.ucsb.edu

Eduardo Sontag$^{\mathbb{P}}$

Department of Mathematics

Rutgers University

New Brunswick, NJ 08903

Email: sontag@hilbert.rutgers.edu.

‡ Supported by NSF Grants CCR-0296041, CCR-0206795, CCR-0208749 and IIS-0346973.

† Supported by NSF grant ECS-0242798.

$\mathbb{P}$ Supported by NSF grant CCR-0206789.

Problem addressed:

- a "honey-pot" is hidden in a bounded region $\mathcal{R}$ (typically, $\subset \mathbb{R}^2$ or $\subset \mathbb{R}^3$)
- the exact position $\mathbf{x}^*$ of the honey-pot is unknown but we do know the probability density f of $\mathbf{x}^*$.
- goal: 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, the honey-pot is detected and the search is over.
- Given a finite amount of time T, which translates into a finite-length path for the robot, find a path that maximizes the probability of finding the honey-pot.

A formalization/formulation of the problem

Denote by $\mathcal{S}[x] \subset \mathcal{R}$ the set of points in $\mathcal{R}$ that the robot can see from some position $x \in \mathcal{R}$

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 maximizes $P_c[\rho] = \int_{x \in \mathcal{S}_{\text{path}}[\rho]} f(x)dx$ 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.

Implicit assumptions:

- it is possible to "insert" the robot at an optimal starting point

    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.

- the region in which the search takes place is known via some *a priori* "map-learning" phase

Discrete Version

- Break $\mathcal{R}$ 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.
  - 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*

    As a result, restrict the search to paths that go from tile to tile, remaining on each tile for one unit of time.
- $p_k = \int_{\mathcal{R}_k} f(x)dx$ denotes the probability that the honey-pot is in the k^{th} tile
- 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

$$\mathbf{P_d}[\sigma] = \sum_{\mathbf{k} \in \boldsymbol{\Sigma}} \mathbf{p_k}$$

  where Σ is the set of distinct elements in the sequence σ.
- time needed to transverse the path is

$$\mathbf{T}[\sigma] := \sum_{\mathbf{i=1}}^{\mathbf{N-1}} \mathbf{t_{k_i, k_{i+1}}}$$

  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}.

Problem 2 (Discrete Honey-pot Search).
Find a sequence of tiles $\sigma := \{k_1, k_2, \ldots, k_N\}$ that maximizes

$$\mathbf{P_d}[\sigma] := \sum_{\mathbf{k} \in \sigma} \mathbf{p_k}$$

subject to the constraint that

$$\mathbf{T}[\sigma] := \sum_{\mathbf{i=1}}^{\mathbf{N-1}} \mathbf{t_{k_i, k_{i+1}}} \leq \mathbf{T}$$

Graph-theoretic Formulation of the Discrete Version

Problem 3 (Reward Budget (RB)).

Instance: $\langle G, c, r, L \rangle$, where

- L is an integer
- $G = (V, E)$ is a graph with
  - edge cost function $c : E \to [0, \infty)$ and
  - vertex reward function $r : V \to [0, \infty)$

Valid Solutions: A (possibly self-intersecting) path $p = (v_1, v_2, \ldots, v_k)$ in G with $v_i \in V$ such that $C[p] := \sum_{i=1}^{k-1} c(v_i, v_{i+1}) \leq L$

Objective: maximize the total reward $R[p] = \sum_{v \in P} r(v)$ where P denotes the set of vertices in p

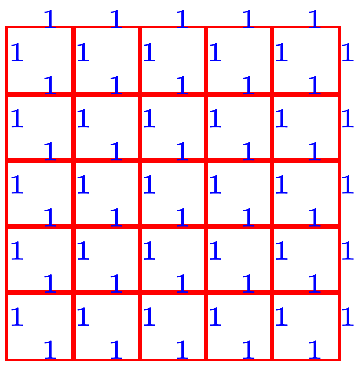
Some Definitions/Notations from Approximation Algorithms Community

Maximization Problem: **maximize** an objetive function

OPT: maximum (optimum) value of the objective function

ε-approximate solution or ε-approximation
a solution with an objective value of at least $\frac{1}{\varepsilon}$OPT

Unit Grid Graph (definition)



Summary of our hardness results

Lemma 1. *The RB problem is NP-hard even when*

- $\mathbf{r}(\mathbf{v}) = \mathbf{1}$ *for every vertex* v, $\mathbf{c}(\mathbf{e}) = \mathbf{1}$ *for every edge* e *and the graph* G *is planar bipartite with the maximum degree of any vertex being* 3, *or*

- G *is a unit grid graph and* $r(v)$ *is* 0 *or* 1 *for every vertex* v

Proof is straightforward via a reduction from the Hamiltonian path problem and using the following references:

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

+ 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.

Summary of our approximation results

Theorem 1. **(a)** *For any constant $\varepsilon > 0$, an r-approximate solution to the RB problem can be found in polynomial time where*

$$r = \begin{cases} 2+\varepsilon & \textit{if } c(e)=1 \textit{ for every edge } e \\ 5+\varepsilon & \textit{otherwise} \end{cases}$$

(b) *If $r(v)=1$ for every vertex v and $c(e)=1$ for every edge e, then a 2-approximate solution to the RB problem can be computed in $O(|V|+|E|)$ time.*

Proof ideas for Theorem 1(b)

Via depth-first-search (DFS) and Eulerian tours with doubled edges:

- do a DFS on G starting at some vertex s computing a DFS tree
- replace every edge in the DFS tree by two edges
- compute an Eulerian cycle
- output the path consisting of the first L edges starting at s in this Eulerian cycle

Proof ideas for Theorem 1(a)

General outline:

- consider a "dual" version of the RB problem (the RQ problem)
- show that a good approximate solution to the RQ problem translates to a corresponding good approximate solution of the RB problem via a binary search similar to that by Johnson *et al.*, path decompositions and Eulerian tours via doubling edges.

  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.

Proof ideas for Theorem 1(a) (continued)

General outline (continued):

- solve the RQ problem by using the $(2+\varepsilon)$-approximation results on the k-MST problem by Arora and Karakostas:

  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.

  that builds upon the 3-approximation results on the same problem by Garg:

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

Proof ideas for Theorem 1(a) (continued)

Problem 4 (Reward Quota (RQ)).
(dual of Reward Budget (RB))

Instance: $\langle G, s, c, r, R \rangle$, where

- $G = (V, E)$ is a graph
- $c : E \to [0, \infty)$ is an edge cost function,
- $r : V \to [0, \infty)$ is a vertex reward function
- R is a positive integer

Valid Solution: A (possibly self-intersecting) path $p = (v_1 = s, v_2, \ldots, v_k)$ such that $\sum_{v \in P} r(v) \geq R$ where P denotes the *set* of vertices in the path p.

Objective: **minimize the total cost** $\sum_{i=1}^{k-1} c(v_i, v_{i+1})$.

Proof ideas for Theorem 1(a) (continued)

$(2+\mu)$ approximation of RQ $\longrightarrow$ solution of RB of total reward $\geq \frac{1}{1+\rho}\mathrm{OPT}\left(\frac{L}{2+\mu}\right)$

$$\mathrm{OPT}\left(\frac{L}{2+\mu}\right) \geq \frac{1}{5}\mathrm{OPT}(L)$$

$$\Rightarrow$$

$$\frac{1}{1+\rho}\mathrm{OPT}\left(\frac{L}{2+\mu}\right) \geq \underbrace{\frac{1}{5+5\rho}}_{\parallel \atop \varepsilon}\mathrm{OPT}(L)$$

general case

$$\mathrm{OPT}\left(\frac{L}{2+\mu}\right) \geq \frac{1}{2+\mu}\mathrm{OPT}(L)$$

$$\Rightarrow$$

$$\frac{1}{1+\rho}\mathrm{OPT}\left(\frac{L}{2+\mu}\right) \geq \frac{1}{1+\underbrace{2\rho+\mu+\rho\mu}_{\parallel \atop \varepsilon}}\mathrm{OPT}(L)$$

$c(e) = 1$ **for all** $e \in E$

Thank you!!