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

Bhaskar DasGupta
Univ. of Illinois Chicago

João P. Hespanha
Univ. of California Santa Barbara

Eduardo Sontag
Rutgers University

Summary

Present a computationally hard (continuous) optimization problem

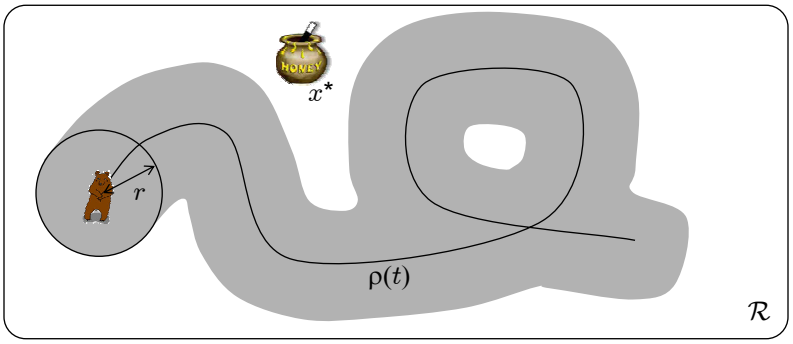
- honey-pot search
- multiple application areas: rescue, mining, land-mine clearance, defense, etc.
- maximize reward, subject to cost constraints

Apply general principles of aggregation/refinement to this problem

- low(er) computation
- solution guaranteed to satisfy constraints but, in general, suboptimal
- compute upper-bound on achievable reward to assess the penalty introduced by this approach

Honey-pot searching

UCSB



$\mathcal{R}$ $\equiv$ search region (bounded)
$x^* \in \mathcal{R}$ $\equiv$ target position (stationary)
$\rho(t)$ $\equiv$ searcher's path
r $\equiv$ radius of circular detection area ("cookie cutter" detection)

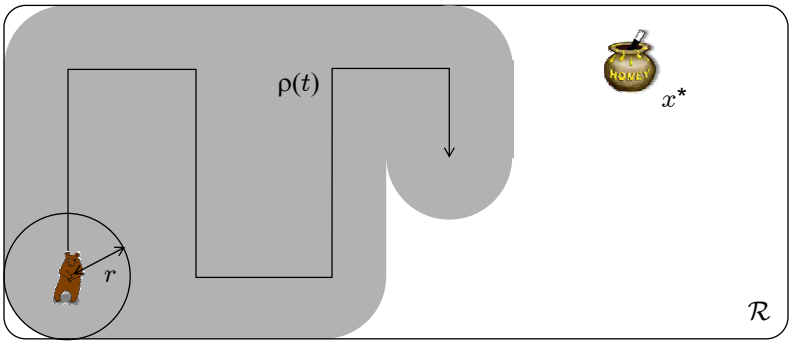
Assuming $x^* \sim$ unif($\mathcal{R}$)

$$\mathrm{P}(\text{find}) = \frac{r\text{-band area around } \{\rho(t) : t \in [0,T]\}}{\text{total area of } \mathcal{R}}$$

to be maximized by choice of $\rho(t)$, $t \in [0,T]$

Honey-pot searching

UCSB



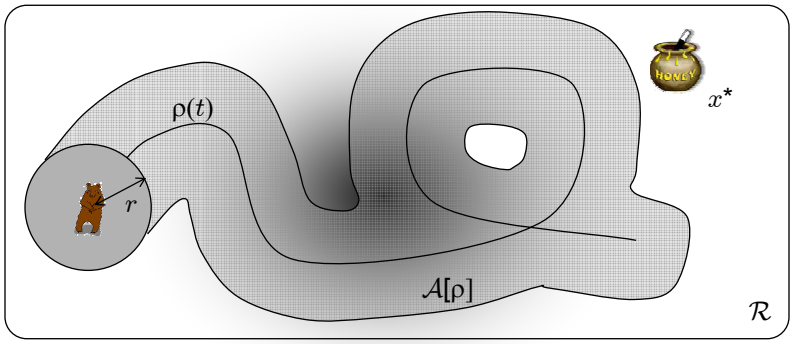
Assuming $x^* \sim$ unif($\mathcal{R}$), fixed time T, fixed velocity $\|\dot{\rho}\|$

$$\mathrm{P}(\text{find}) = \frac{r\text{-band area around } \{\rho(t) : t \in [0,T]\}}{\text{total area of } \mathcal{R}}$$

lawn mower paths are (essentially) optimal

Honey-pot searching

UCSB



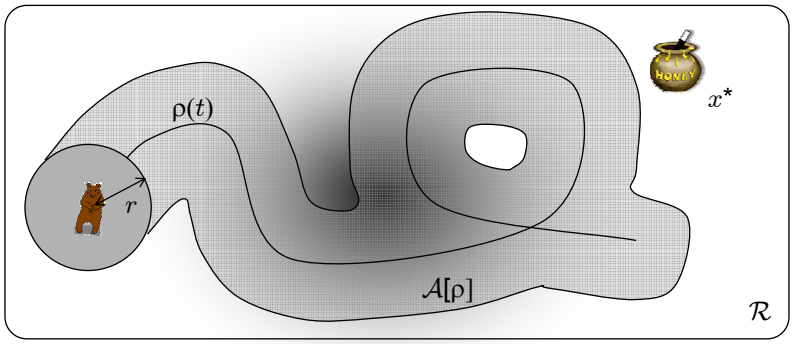
Assuming general pdf $f(x)$ for target position x^*

$$\mathrm{P}(\text{find}) = \int_{\mathcal{A}[\rho]} f(x)dx$$

$\mathcal{A}[\rho] \equiv r$-band set around $\{\rho(t)\colon t \in [0,T]\}$

Honey-pot searching

UCSB



Assuming general pdf $f(x)$ for x^*, unimodal

$$\mathrm{P}(\text{find}) = \int_{\mathcal{A}[\rho]} f(x)dx$$

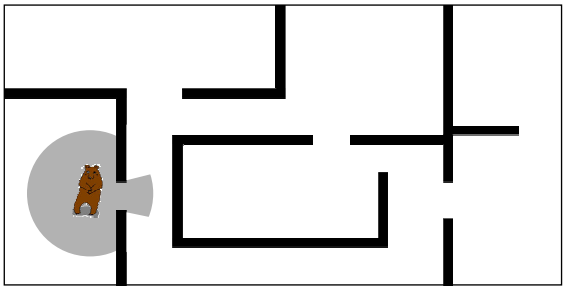
$\mathcal{A}[\rho] \equiv r$-band set around $\{\rho(t)\colon t \in [0,T]\}$

optimal paths "cover" region $\{\, x \in \mathcal{R} : f(x) \geq c \,\}$

Search Theory Koopman & Stone (WWII, 1940s–)

General constrained honey-pot searching

UCSB



1. Position-dependent detection area
   $\mathcal{S}[x] \equiv$ area "seen" by searcher at position x

$$R[\rho] := \mathrm{P}(\text{find}) = \int_{\mathcal{A}[\rho]} f(x)dx$$

$$\mathcal{A}[\rho] := \bigcup_{t\in[0,T]} \mathcal{S}[\rho(t)]$$

2. General constraint on the path

$$C[\rho] := \int_0^T c(\rho(t))dt \le L$$

constant

- time constraint: $c(x) = 1 \quad \forall\, x$
- fuel constraint: $c(x) =$ fuel consumption rate at position x

reward to be maximized by choice of $\rho:[0,T]\to\mathcal{R}$

Approaches

UCSB

- Mangel'89 – optimal control of searcher's velocity subject to PDE dynamics
- Spatial and temporal discretization [Stone'75]
  - Trummel&Weisinger'86, DasGupta,H.&Sontag'04 – NP-Hard
  - Stewart'79, Eagle&Yee'90 – reduction to integer programming
  - Eagle'84 – reduction to optimization of POMDP
  - H.,Kim&Sastry'99 – non-optimal greedy algorithm (POMDP)
  - DasGupta,H.&Sontag'04 - ϵ-optimal (polynomial-time, $\epsilon = 1/5$)

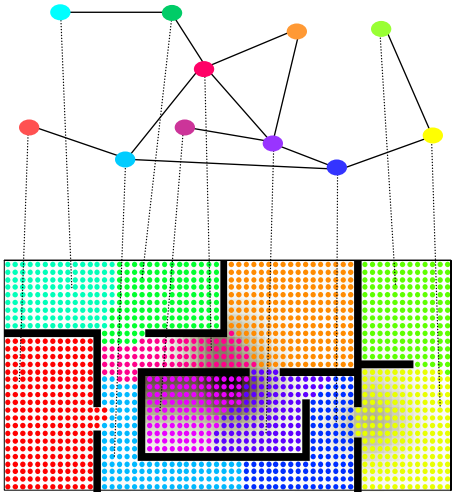
moving target

Most of this work was restricted to the discrete version of the problem and ignored the underlying continuous problem

Can we quantify the effect of discretization on the cost/reward of the final continuous path?

Aggregate/refinement approach

UCSB



Partition region into cells

$$\mathcal{R} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \cup \mathcal{C}_N$$

Construct graph

- *aggregate* each cell to single node
- cell-reward ≡ measure of prob. of finding target in cell
- edge-cost ≡ measure of cost incurred by moving between cells

Compute optimal path on the graph

Refine (optimal) discrete graph-path into (suboptimal) continuous path

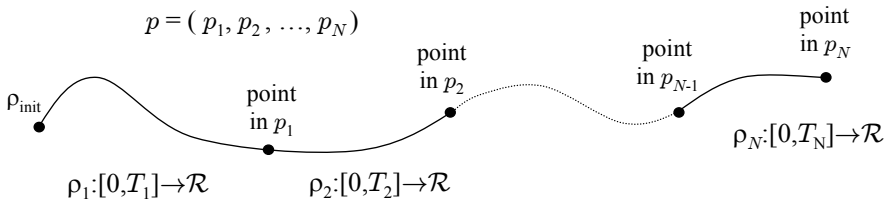
- satisfying cost-bound
- providing guaranteed reward

Refinement

UCSB

Given a graph-path (sequence of cells, possibly with repetitions)

$$p = (p_1, p_2, \ldots, p_N)$$



produce sequence of continuous paths $\rho_1, \rho_2, \ldots, \rho_N$ such that

$$R[\rho_k] \geq R[\rho_{k-1}] + r_{\text{worst}}(p_k)$$

worst-case reward gained by extending path to cell p_k

$$C[\rho_k] \leq C[\rho_{k-1}] + c_{\text{worst}}(p_{k-1}, p_k)$$

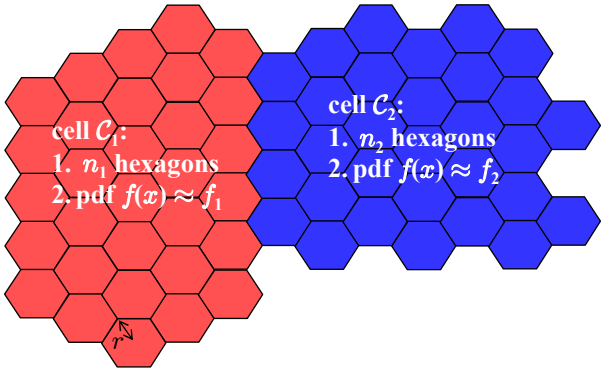
worst-case cost incurred by extending path from p_{k-1} to p_k

only needs to hold the first $|\, p_k \,|_{\text{worst}}$ times

the three functions $r_{\text{worst}}(\cdot)$, $|\cdot|_{\text{worst}}$, $c_{\text{worst}}(\cdot)$ characterize the refinement procedure

Example

UCSB



1. Target pdf is constant on each cell
2. Hexagonal detection area with "radius" r (area $\approx 2.6\ r^2$)
3. Each cell can be divided into a finite number of hexagons
4. Cost ≡ distance transversed

the first n_i times a path is extended to $\mathcal{C}_i$, the reward increases by $2.6\ r^2\ f_i$:

$$r_{\text{worst}}(\mathcal{C}_i) = 2.6\ r^2\ f_i \qquad |\mathcal{C}_i| = n_i$$

worst-case costs:

$c_{\text{worst}}(\mathcal{C}_i, \mathcal{C}_i) = 1.7\ r$ (distance to adjacent square)

$c_{\text{worst}}(\mathcal{C}_i, \mathcal{C}_j)$ = largest distance from a hexagon in $\mathcal{C}_i$ to another one in $\mathcal{C}_j$

Discrete Reward-Budget problem (dRB)

UCSB

Given:

- $G = (V, E)$ ≡ graph with vertex set V and edge set E
- $s \in V$ ≡ initial vertex
- $c : E \to [0,\infty)$ ≡ edge cost function
- $r : V \to [0,\infty)$ ≡ vertex reward function
- $|\cdot| : V \to \mathbb{N}$ ≡ vertex cardinality function

Find (possibly intersecting) path $p = (p_0 = s, p_1, p_2, \ldots, p_N)$ such that

$$\sum_{i=1}^{N} c(p_{i-1}, p_i) \le L$$

maximizing

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

sum over all vertices (without repetitions)

# of times vertex v appears in the path p

total reward collected on vertex v

NP-hard! (the fewer regions the better)

Suboptimal solution to continuous search

UCSB

Consider discrete Reward-Budget optimization with
- vertex reward $r_{\text{worst}}(\cdot)$,
- vertex cardinality $|\cdot|_{\text{worst}}$,
- edge cost $c_{\text{worst}}(\cdot)$

from path-refinement algorithm

1. dRB optimization provides *lower-bound* on achievable continuous-search reward

$$R_{\text{dRB}}[L] \le R^*[L]$$

best possible reward for dRB problem (with discrete cost $\le L$)

best possible reward for continuous problem (with continuous cost $\le L$)

2. dRB optimal path can be used to *construct admissible continuous path*

$p \equiv$ optimal path for dRB optimization

$\rho \equiv$ refinement of discrete path p

reward at least as good as that of dRB

$$C[\rho] := \int_0^T c\big(\rho(t)\big)dt \le L \qquad R[\rho] := \mathrm{P}(\text{find}) = \int_{\mathcal{A}[\rho]} f(x)dx \ge R_{\text{dRB}}[L]$$

satisfies reward constraint

Upper-bound on achievable reward

UCSB

For positive integer m, consider discrete Reward-Budget optimization with
- $r_{\text{best}}(p_k) \equiv$ upper-bound on instance of the continuous search problem starting in p_k with cost bounded by L/m
- $|p_k|_{\text{best}} \equiv$ upper-bound on $\int_S f(x)dx$ / $r_{\text{best}}(p_k)$, $S \equiv$ points that can be scanned from p_k, with cost not exceeding L/m
- $c_{\text{best}}(p_{k\text{-}1}, p_k) \equiv \max\{\, L/m,$ lower-bound on cost to go from $p_{k\text{-}1}$ to $p_k \,\}$

3. dRB optimization provides *upper-bound* on achievable continuous-search reward

$$R_{\text{dRB}}[L] \ge R^*[L]$$

best possible reward for dRB problem (with discrete cost $\le L$)

best possible reward for continuous problem (with continuous cost $\le L$)

when pdf is flat on each cell and m is small this bound can be fairly tight (but small m makes bound harder to compute)

Gap between bounds provide measure of optimality for the path obtained by refinement

Summary

UCSB

Present computationally hard (continuous) optimization problem

- maximize reward, subject to cost constraints
- multiple application areas: rescue, mining, land-mine clearance, defense, etc.

Apply general principles of aggregation/refinement to this problem

- solution guaranteed to satisfy constraints but, in general, suboptimal
- low(er) computation
- compute upper-bound on achievable reward to access the penalty introduced by this approach

Current/future work:

- compare these solutions with other suboptimal approaches (CDC'04)
- automated partition of search space to minimize the penalty introduced by the aggregation/refinement approach