Complexity Issues
in
Probabilistic Mapping

João P. Hespanha

Center for Control Engineering
and Computation

University of California
at Santa Barbara, USA

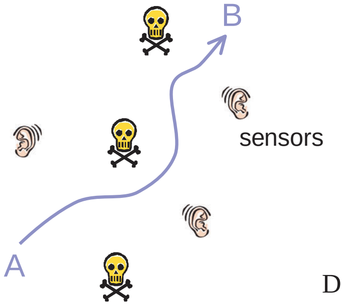
Motivating problems

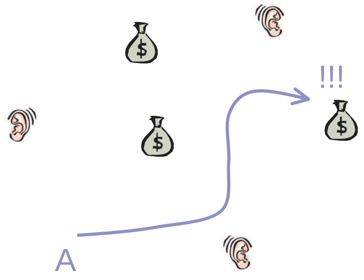
Risk avoidance:

How to go from A to B
avoiding the ☠ ?

Search:

How to find the 💰
as fast as possible ?

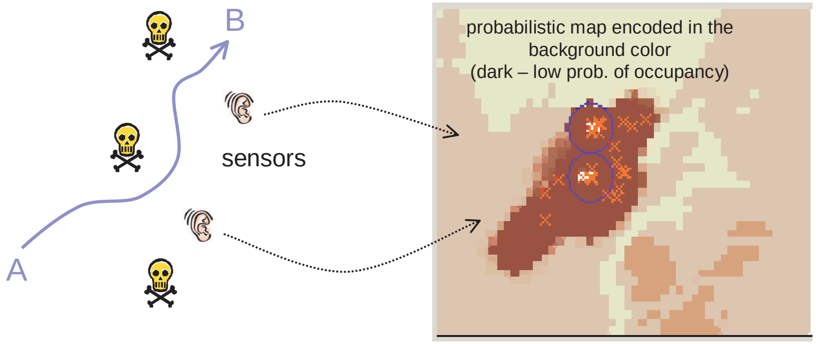




Decisions are made based on noisy data collected
by the 👂 regarding the position of the ☠ / 💰

sensors are not perfect → uncertainty in map of objects' positions constructed from sensor data

Probabilistic maps



probabilistic map encoded in the background color
(dark – low prob. of occupancy)

probabilistic map: conditional distribution of objects' positions given the available measurements

$$m_r(\, r \mid Y\,) := \mathrm{P}\,(\, \mathbf{r}(t_2) = r \mid \mathbf{Y}(t_1) = Y\,)$$

objects' position at time t_2

measurements collected up to time t_1

How to compute probabilistic maps of a large number of objects efficiently?

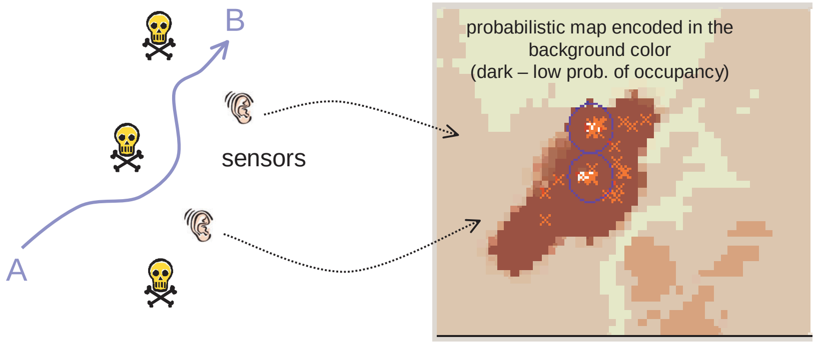
Outline

- Mathematical framework
- Sensor model

- Mapping static objects
- Mapping dynamic objects

- Application to minimum-risk path planning
- Application to optimal search

Computing probabilistic maps



$\mathcal{R}$ ≡ region being mapped (partitioned into cells)
$\mathbf{r} := \{\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_n\}$ ≡ positions of n objects of the same type. Each $\mathbf{r}_i \in \mathcal{R}$
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots$ ≡ sequence of measurements taken by sensors

Goal: Given a list of measurements $Y_k := \{ y_1, y_2, \ldots, y_k \}$ collected up to time t_k, compute the *a posteriori* conditional distribution of $\boldsymbol{r}$:

$$m_r(\, r|Y_k\,) := \mathrm{P}\,(\,\mathbf{r}(t) = r \mid Y_k\,), \qquad t \geq t_k$$

objects probabilistic maps

Sensor model

$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots$ ≡ sequence of measurements taken by sensors.
Each $\mathbf{y}_k = \{t_k, a_k, s_k\}$

time stamp

sensor position

$s_k = \emptyset \equiv$ no object seen
$s_k \in \mathcal{R} \equiv$ object detected at position s_k

objects are indistinguishable ($s_k \in \mathcal{R}$ provides no information regarding which object was seen)

Sensor parameters (not necessarily constant over time or space):

ρ sensor range
p_1 probability of not recognizing an object within range (false negative)
p_2 probability of recognizing a "distractor" instead of a real object when an object is within sensor range (false positive)
p_3 probability of recognizing a "distractor" instead of a real object when no object is within sensor range (false positive)
e maximum localization error, when the sensor detects a real object (typically $e \ll \rho$)

Sensor model

$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots \equiv$ sequence of measurements taken by sensors.

Each $\mathbf{y}_k = \{t_k, a_k, s_k\}$

time stamp

sensor position

$s_k = \emptyset \equiv$ no object seen

$s_k \in \mathcal{R} \equiv$ object detected at x

n-object sensor likelihood function

$$\ell_n(y,r) := \mathrm{P}(\mathbf{y}_k = y \mid \mathbf{r}(t) = r) \qquad y := \{t, a, s\}$$

each measurement $\mathbf{y}_k$ is conditionally independent of any other measurement given an object configuration $\mathbf{r}(t_k)$

Sensor model (single object)

$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots \equiv$ sequence of measurements taken by sensors.

Each $\mathbf{y}_k = \{t_k, a_k, s_k\}$

time stamp

sensor position

$s_k = \emptyset \equiv$ no object seen

$s_k \in \mathcal{R} \equiv$ object detected at x

n-object sensor likelihood function

$$\ell_n(y,r) := \mathrm{P}(\mathbf{y}_k = y \mid \mathbf{r}(t) = r) \qquad y := \{t, a, s\}$$

For the *single object* case ($n = 1$):

$$\ell_1(y,r) = \begin{cases} \frac{1-p_1-p_2}{B} & |r-a| \le \rho,\ |s-r| \le e \\ 0 & \text{otherwise} \end{cases} + \begin{cases} \frac{p_2}{A} & |r-a| \le \rho,\ |s-a| \le \rho \\ \frac{p_3}{A} & |r-a| > \rho,\ |s-a| \le \rho \\ 0 & \text{otherwise} \end{cases}$$

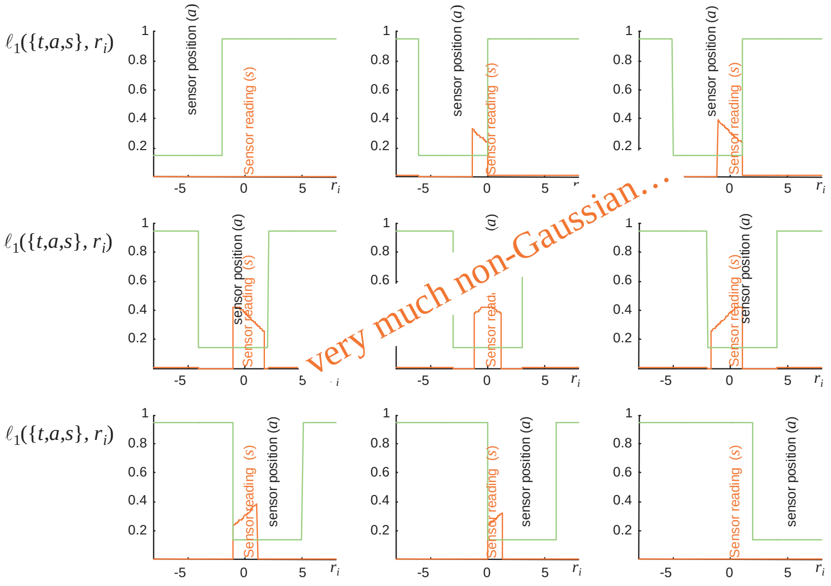
A and $B \equiv$ normalizing constants

Sensor parameters

ρ	sensor range
p_1	prob. not recognizing an object within range
p_2	prob. recognizing "distractor" instead of real object when an object is within range
p_3	prob. recognizing "distractor" instead of real object when no object is within range
e	max. localization error, when the sensor detects a real object (typically $e \ll \rho$)

Typical single-object sensor likelihood function

UCSB



very much non-Gaussian…

Sensor model (multiple objects)

UCSB

n-object sensor likelihood function

$$\ell_n(y, r) := \mathrm{P}(\mathbf{y}_k = y \mid \mathbf{r}(t) = r) \qquad y := \{t, a, s\}$$

Sparseness assumption

1 minimum distance between any two objects $d_{\min} \geq 2\,\rho$
(no two objects in sensor range)

or 1’ global sensor ($\rho = +\infty$) and $d_{\min} \geq 2\,e$
(no two objects within area of localization uncertainty)

or 1’’ sensor only produces positive readings and $d_{\min} \geq 2\,e$

Theorem: For every $r \in \mathcal{R}^n$ for which no two objects are closer then $d_{\min}$

normalizing constant

$$\ell_n(\mathbf{y}_k, r) = c \prod_{i=1}^{n} \ell_1(\mathbf{y}_k, r_i), \qquad \forall k$$

single measurement
$\{t_k, a_k, s_k\}$

possible configuration
for the *n* objects
$(r := \{r_1, \ldots, r_n\})$

single-target sensor
likelihood function

Mapping static objects

When $\mathbf{r}(t) = \mathbf{r}(0)\ \forall\, t \geq 0\ldots$

$$m_r(r|Y) := \mathrm{P}(\mathbf{r} = r \mid \mathbf{Y}_k = Y) \qquad r := \{r_1, r_2, \ldots, r_n\}$$

$$= c\left(\prod_{i=1}^{n} f(r_i|Y)\right) p_0(r) \qquad Y := \{\, y_1, y_2, \ldots, y_k\}$$

normalizing constant

a priori distribution

$$f(x|Y_k) := \prod_j \ell_1(y_j, x),\ x \in \mathcal{R}$$

aggregate measurement function

(encapsulates all the information contained in the sequence of measurements Y)

# of cells in $\mathcal{R}$

# of objects

1. $m_r(\,\cdot\,|Y)$ is defined on $\mathcal{R}^n$ thus the memory complexity to represent it is o(N^n)
2. $f(\,\cdot\,|Y)$ is defined on $\mathcal{R}$ thus the memory complexity to represent it is only o(N)
3. $f(\,\cdot\,|Y)$ can be computed efficiently recursively (one measurement at a time)

$$f(x, \{y_1, y_2, \ldots, y_{k+1}\}) = \ell_1(y_{k+1}, x)\, f(x, \{y_1, y_2, \ldots, y_k\})$$

high-dim. maps can be efficiently represented and computed using a low-dim. function

---

Moving objects

Markovian motion model

$$\Phi(r', r; t + dt, t) := \mathrm{P}(\mathbf{r}(t + dt) = r' \mid \mathbf{r}(t) = r)$$

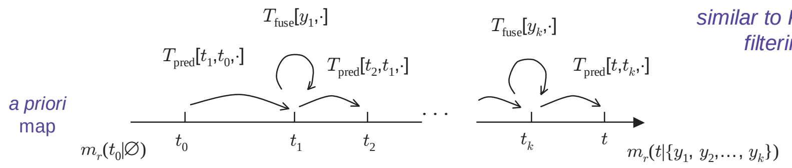
n-object transition probability function

Can take into account:

1. uncertainty in the motion
2. regions of high/low mobility (e.g., roads versus, rocky terrain)
3. preferential directions of motion

The n-object transition probability function Φ is such that no two objects are closer than $d_{\min}$ with probability one for all $t \geq t_0$

Recursive map building



similar to Kalman filtering

$T_{\text{pred}}[t,\tau,m_r(\cdot)] \equiv$ prediction operator takes a map at time τ and propagates it forward in time to obtain a map at time $t>\tau$

$$m_r(\,\cdot\,;t|Y) = T_{\text{pred}}[t,\tau,m_r(\,\cdot\,;\tau|Y)]$$

map at time $t > \tau$

map at time τ

$T_{\text{fuse}}[y,m_r(\cdot)] \equiv$ fusion operator takes a map at time t_i and, given a list of measurements Y_{i-1} taken before t_i and a new measurement y_i taken at t_i, produces a new map given all measurements

$$m_r(\,\cdot\,;t_i|Y_i) = T_{\text{fuse}}[y_i, m_r(\,\cdot\,;t_i|Y_{i-1})]$$

$\{y_i,Y_{i-1}\}$

new measurement taken at t_i

measurements taken before t_i

Map building operators

The fusion operator is defined by:

n-object sensor likelihood function (sensor model)

$$m_r(r;t_i|Y_i) = c\,\ell_n(y_i,r)\,m_r(r;t_i|Y_{i-1})$$

$c \equiv$ *normalizing constant*

$\{y_i,Y_{i-1}\}$

new measurement at t_i

measurements before/at t_i

The prediction operator is defined by:

measurements before/at t

$$m_r(r';t+dt|Y_i) = \sum_{r\in\mathcal{R}^n} \Phi(r',r;t+dt,t)\,m_r(r;t|Y_i)$$

n-object transition probability function (motion model)

to use these formulas directly one needs an extensive representation of the map – memory and computational complexity o(N^n)

\# of cells in $\mathcal{R}$

\# of objects (exp !!!)

Separable maps

Definition: A probabilistic map is called separable if it can be written as

$$m_r(r;t|Y) := c\Big(\prod_{j=1}^{n} f(r_j;t|Y)\Big)\delta(r;d_{\min}) \qquad \forall r \in \mathcal{R}^n$$

aggregate measurement function
(defined in $\mathcal{R}$)

$$\delta(r;d_{\min}) := \begin{cases} 0 & \exists i,j : |r_i - r_j| < d_{\min} \\ 1 & \text{otherwise} \end{cases}$$

zero probability of two objects closer than d_{min}

From previous result: for stationary objects and the class of sensors considered here, the probabilistic maps are separable

still approximately true for mobile objects

Fusion operator for separable maps

Assume map at time t_i, given measurements Y_{i-1} is separable

$$m_r(r;t_i|Y_{i-1}) := c\Big(\prod_{j=1}^{n} f(r_j;t_i|Y_{i-1})\Big)\delta(r;d_{\min}) \qquad \forall r \in \mathcal{R}^n$$

map at time t_i, given measurements Y_{i-1}

Theorem: *The fusion operator preserves separability*, i.e.,

$$m_r(r;t_i|Y_i) := c\Big(\prod_{j=1}^{n} f(r_j;t_i|Y_i)\Big)\delta(r;d_{\min}) \qquad \forall r \in \mathcal{R}^n$$

map at time t_i, given measurements Y_i

where

sensor likelihood function (sensor model)

$$f(r;t_i|Y_i) := \ell_1(y_i,r)\, f(r;t_i|Y_{i-1}) \quad \forall r \in \mathcal{R}$$

$\{y_i, Y_{i-1}\}$ new measurement at t_i measurements before/at t_i

fusion can be efficiently done for large number of objects

Prediction operator for separable maps

objects essentially move independently
while they are away from each other
(does not preclude specific global motion patterns)

Independent motion assumption

$$\Phi(r', r; t+dt, t) = c(r) \left(\prod_{k=1}^{n} \varphi_1(r'_k, r_k; t+dt, t) \right) \delta(r'; d_{\min})$$

normalizing constant

single-object transition probability function

$$\delta(r'; d_{\min}) := \begin{cases} 0 & \exists i, j : |r'_i - r'_j| < d_{\min} \\ 1 & \text{otherwise} \end{cases}$$

zero probability of two objects
coming closer than d_{min}

Bounded velocity assumption

$$\varphi_1(r', r; t+dt, t) := \begin{cases} 1 - o(dt) & r = r' \\ o(dt) & r \neq r' \end{cases} \qquad dt \to 0$$

Prediction operator for separable maps

Assume map at time t, given measurements Y_i is separable

$$m_r(r; t|Y_i) := c \left(\prod_{j=1}^{n} f(r_j; t|Y_i) \right) \delta(r; d_{\min}) \qquad \forall r \in \mathcal{R}^n$$

map at time t, given measurements Y_i

Theorem: *The prediction operator approximately preserves separability*, i.e.,

$$m_r(r; t+dt|Y_i) = c \left(\prod_{j=1}^{n} f(r_j; t+dt|Y_i) \right) \delta(r; d_{\min}) + \epsilon(r; dt) \quad \forall r \in \mathcal{R}^n$$

map at time t+dt, given measurements Y_i

where

transition probability function
(motion model)

$$f(r'; t+dt|Y_i) = \sum_{r \in \mathcal{R}} \varphi_1(r', r; t+dt, t) f(r; t|Y_i)$$

$\epsilon(r;dt) = o(dt^2)$ except for configuration with all objects further apart than $d_{\min}$ but for which this could be violated by the motion of a single object (in which case $\epsilon(r;dt) = o(dt)$)

prediction can be efficiently done for large number of objects

Summary so far…

Probabilistic mapping can be done efficiently for large number of objects with memory and computational complexity $o(N)$ [and not $o(N^n)$]

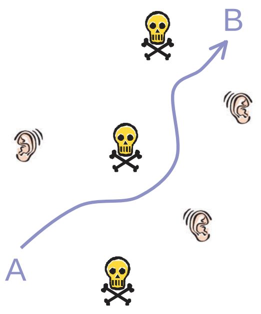
# of cells in $\mathcal{R}$

# of objects (exp !!!)

1. Under some form of **sparseness assumption**, which imposes a minimum distance between any two objects
2. Assuming a **Markovian motion** model, under which objects essentially move independently while they are away from each other
3. The map will exhibit an **error** essentially o(dt^2)

motion
"integration-step"

Application to minimum-risk path planning



Robot navigates over the region $\mathcal{R}$ populated by objects ☠

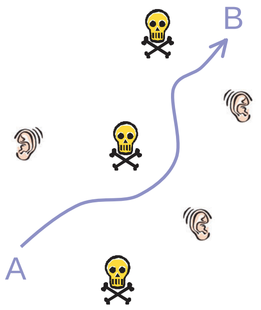
Each object ☠ induces danger if robot comes close to it

Risk parameters (not necessarily constant over time or space):

ρ_{danger}	range over which object is dangerous
p_{destroy}	probability that robot will be destroyed if it comes close to object

Goal: compute path from initial position A to final position B that maximizes probability of p_{survive} of not being damaged

Application to minimum-risk path planning



Risk parameters (not necessarily constant over time or space):

ρ_{danger} range over which object is dangerous

p_{destroy} probability that robot will be destroyed if it comes close to object

Goal: compute path from initial position A to final position B that maximizes probability of p_{survive} of not being damaged

Given a path: $x_1 = \text{A}, x_2, x_3, \ldots, x_m = \text{B} \in \mathcal{R}$

$$p_{\text{survive}} = \prod_{j=1}^{m} \left(1 - m_{\text{danger}}(x_j|Y)\right)$$

danger map

$$m_{\text{danger}}(x|Y) := \text{P}(\text{robot destroyed when passing through } x \mid \mathbf{Y}_k = Y)$$

$$x \in \mathcal{R}$$

Minimum-risk path planning

Given a path: $x_1 = \text{A}, x_2, x_3, \ldots, x_m = \text{B} \in \mathcal{R}$

$$p_{\text{survive}} = \prod_{j=1}^{m} \left(1 - m_{\text{danger}}(x_j|Y)\right) \quad \Rightarrow \quad \log p_{\text{survive}} = \sum_{j=1}^{m} \log_{\text{live}}(x_j|Y)$$

probability that robot will survive

$$\log\left(1 - m_{\text{danger}}(x_j|Y)\right)$$

Minimum risk path

$$\max_{\substack{x_1, x_2, \ldots, x_m: \\ x_1 = \mathbf{A},\ x_m = \mathbf{B}}} \sum_{j=1}^{m} \log_{\text{live}}(x_j|Y)$$

Minimum risk path, subject to length constraint: $m \leq M$

$$\max_{\substack{x_1, x_2, \ldots, x_m: \\ x_1 = \mathbf{A},\ x_m = \mathbf{B}, \\ m \leq M}} \sum_{j=1}^{m} \log_{\text{live}}(x_j|Y)$$

Minimum length path, subject to risk constraint: $p_{\text{survive}} \geq p^*$

$$\min_{\substack{x_1, x_2, \ldots, x_m: \\ x_1 = \mathbf{A},\ x_m = \mathbf{B}, \\ \sum_{j=1}^{m} \log_{\text{live}}(x_j|Y) \geq p^*}} m$$

optimizations
in graphs with
additive
cost/constraint

Computing danger maps

Danger map:

$$m_{\text{danger}}(x|Y) := \text{P}(\text{robot destroyed when passing through } x \mid \mathbf{Y}_k = Y)$$

$$= \sum_{||z-x|| \leq \rho_{\text{danger}}} p_{\text{destroy}} m_o(z,Y)$$

occupancy map

$\text{P}(\exists i : \mathbf{r}_i = z \mid \mathbf{Y}_k = Y)$

(conditional) probability that there exists one object at location x

Risk parameters (not necessarily constant over time or space):

ρ_{danger} range over which object is dangerous

p_{destroy} probability that robot will be destroyed if it comes close to object

Theorem:

$$m_o(x,Y) \approx \bar{c} f(x,Y) g_0(x) \left(n f_0(Y)^{n-1} + \sum_{k=1}^{n-1} \frac{n!\, f_0(Y)^{n-k-1}}{(n-1-k)!} \sum_{\sigma \in \mathcal{S}^{(k)}[x]} f_{\sigma_1}(Y) f_{\sigma_2}(Y) \dots f_{\sigma_k}(Y) \right)$$

where $f_i(Y) := \sum_{x \in \mathcal{R}_i} f(x,Y) g_0(x)$

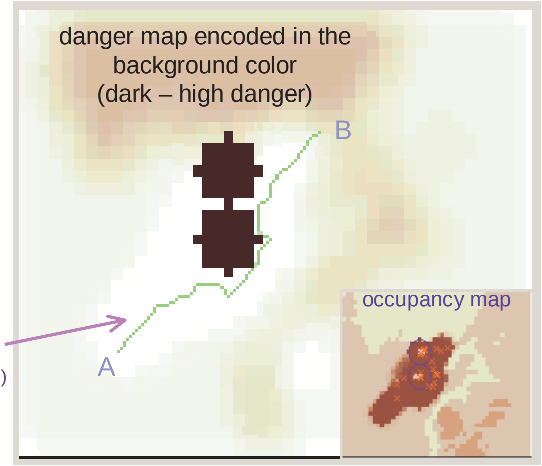
permutations of k elements from $\{i : x \in \mathcal{R}_i\}$

partition of $\mathcal{R}$, with peaks of $f(\cdot,Y)$ in $\mathcal{R}_1,\dots\mathcal{R}_m$

Danger maps can be computed directly from the a.m.f. $f(\cdot,Y)$

Minimum-risk path planning

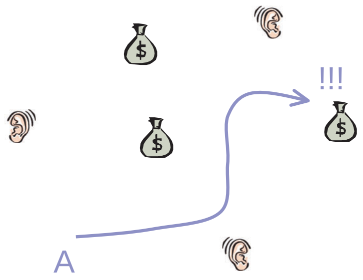
Goal: compute path from initial position A to final position B that maximizes probability of p_{survive} of not being damaged



minimum risk path
(subject to length constrain)

From a complexity view-point, minimum risk path planning is a well-behaved problem…

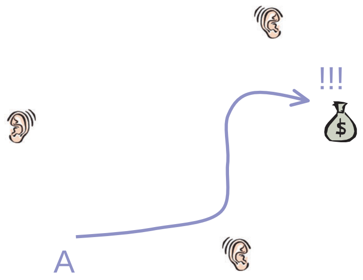
Application to search



Robot navigates over the region $\mathcal{R}$ populated by objects 💰

Goal: compute path starting in an initial position A that maximizes probability of p_{find} of finding an object in a given time interval.

Application to search



Robot navigates over the region $\mathcal{R}$ with a single static object 💰

Goal: compute path starting in an initial position A that maximizes probability of p_{find} of finding the object in a given time interval.

Given a path: x_1 = A, x_2, x_3, …, $x_m \in \mathcal{R}$

occupancy map
(probability the object is at x)

$$p_{\text{find}} = \sum_{x\in\{x_1,x_2,\ldots,x_m\}} m_o(x,Y) \le \sum_{i=1}^{m} m_o(x_i,Y)$$

summation over path *as a set*
(i.e., repeated cells are counted only once – seeing a position twice does not increase probability of finding the object)

Optimal search

Given a path: $x_1 = \text{A}, x_2, x_3, \ldots, x_m \in \mathcal{R}$

$$p_{\text{find}} = \sum_{x \in \{x_1, x_2, \ldots, x_m\}} m_o(x, Y)$$

probability that object will be found

Maximum probability path, subject to length constraint: $m \leq M$

$$\max_{\substack{x_1, x_2, \ldots, x_m: \\ x_1 = \mathbf{A}, m \leq M}} \sum_{x \in \{x_1, x_2, \ldots, x_m\}} m_o(x, Y)$$

Minimum length path, subject to probability constraint: $p_{find} \geq p^*$

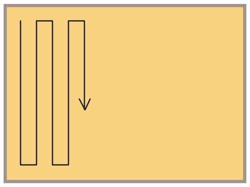
$$\min_{\substack{x_1, x_2, \ldots, x_m: \\ x_1 = \mathbf{A}, \\ \sum_{x \in \{x_1, x_2, \ldots, x_m\}} m_o(x, Y) \geq p^*}} m$$

optimizations
in graphs with
non-additive
cost/constraint

approximating p_{find} by the additive cost $\sum_i m_o(x_i, Y)$ will not work because it would lead the robot to the location with largest probability and keep it there…

A hierarchical algorithm…

Premise: If the map the distribution were uniform, sweeping would be optimal



Hierarchical algorithm:

1. Partition map into a (small) number of connected areas where the distribution is approximately uniform.
   one can use, e.g., state-aggregation algorithms for Markov chains
2. Solve the high-level optimal problem of deciding which areas to visit, in what order, and how long to stay in each one.
   still a combinatorial problem
3. Inside each area use a sweeping path (low-level)

overall sub-optimal but computationally much better

Conclusions

Probabilistic maps can be economically represented and computed with complexity independent of the number of objects under “reasonable” assumptions (sparseness, Markovian motion)

Probabilistic maps can be used to efficiently solve minimum risk path planning problems

Optimal search problems are computationally much worse. Efficient sub-optimal hierarchical solutions are possible