FL seminar 2008-10-20

Cooperative Optimal Search Control

Centralized Control and Distributed Control

FL08-15-2

Mamoru Saito

Search Problem

Search Problem

To locate a target to be found,
deploying agents with the available resources.

- search and rescue operations
- detecting lost objects

Objective

- formulate the Optimal Search Control Problem
  - maximize the probability of finding the target
  - reduce the control energy consumption
- analyze the agent's behavior

Outline

- review (single agent)
- Cooperative Search
  - Centralized Control
  - Distributed Control

Types of Search Problems

	One-sided Search the searcher chooses his strategy the target neither chooses a strategy nor reacts to the search	***Two-sided Search*** the searcher and the target can choose their strategies
Stationary Target	ex. ❑ search for lost car keys ❑ search for natural resources	ex. ❑ hide-and-seek
Moving Target	ex. ❑ search for a life raft in the ocean ❑ the target moves randomly	***Cooperative Search*** ex. the target attempts to make itself as detectable as possible ***Non-Cooperative Search*** the target tries to remain undetected ex. Pursuit-Evasion Game

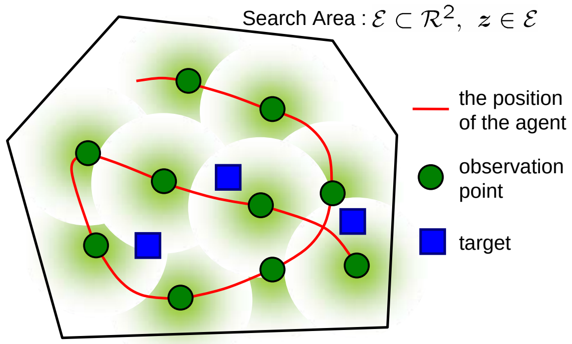
---

Outline

- review (single agent)
- Cooperative Search
  - Centralized Control
  - Distributed Control

Problem Setting

Search Area : $\mathcal{E} \subset \mathcal{R}^2, \ z \in \mathcal{E}$



Problem Setting

agent: **continuous-time linear system**

$\dot{x}(t) = Ax(t) + Bu(t)$

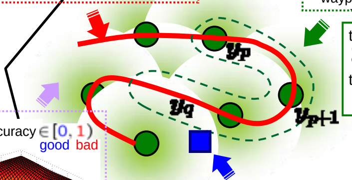
$x(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}$ position: $y(t) \in \mathcal{R}^2$

observation time (obs. time) :

$t_k = kh,\ k = 0,1,2,\ldots$

obs. point $y_k := y(t_k)$

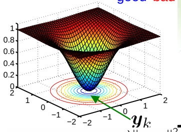
waypoint $x_k := x(t_k)$



the obs. points set $\mathcal{Y}_{p:q} := \{y_k\}_{k=p,p+1,\ldots,q}$

the obs. points row $\tilde{\mathcal{Y}}_{p:q} := (y_k)_{k=p,p+1,\ldots,q}$

sensing accuracy $\in [0, 1)$ good bad



$p(||z - y_k||) = 1 - e^{-\lambda||z-y_k||^2}$

A target appears randomly and stays for gh[s] $(g \in \mathbb{Z}_+)$

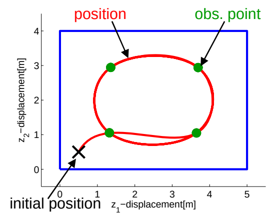
The target appears at time $t = t_t (t_j \le t_t < t_{j+1})$

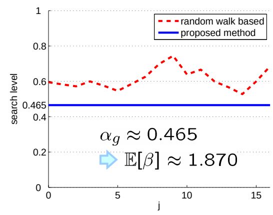
search level (small → good)

$S(\mathcal{Y}_{j+1:j+g}) := \int_{\mathcal{E}} \phi(z) \prod_{y_k \in \mathcal{Y}_{j+1:j+g}} p(||z - y_k||) dz$

Main Result

- formulate Optimal Search Control Problem
  - approximate solution
- the analysis of the agent's behavior
  - the expectation value of how many times the targets appears without being found
  - a necessary and sufficient condition for the agent's state and input to converge respectively to a periodic trajectory





Outline

- review (single agent)
- Cooperative Search
  - Centralized Control
  - Distributed Control

Cooperative Search

Cooperative Search is useful for more efficient search.

- Centralized Control
- Distributed Control

n_a : the number of the agent

remark :

The superscript (l) means the state of lth agent.

Centralized Cooperative Search

Centralized Cooperative Optimal Search Control Problem

$$\min_{u^{(l)}(t), t \in [t_k, t_{k+f}], l=1,2,\ldots,n_a} \sum_{l=1}^{n_a} \sum_{i=k}^{k+f-1} \int_{t_i}^{t_{i+1}} u^{(l)T}(t) R u^{(l)}(t) dt$$

$$\text{s.t.} \quad \min_{\tilde{\mathcal{Y}}^{(l)}_{k+1:k+f}, l=1,2,\ldots,n_a} \sum_{j=0}^{k+f-g} S\left(\bigcup_{l=1}^{n_a} \mathcal{Y}^{(l)}_{j+1:j+g}\right)$$

$$S(\mathcal{Y}_{j+1:j+g}) := \int_{\mathcal{E}} \phi(z) \prod_{y_k \in \mathcal{Y}_{j+1:j+g}} p(||z - y_k||) dz$$

solution:

$\mathcal{Y}^*_{1:gn_a} := \underset{\mathcal{Y}_{1:gn_a}}{\text{argmin}}\ S(\mathcal{Y}_{1:gn_a})$

reduce a combinatorial optimization

⇨ sort $\mathcal{Y}^*_{1:gn_a}$ to minimize the control energy consumption

Simulation

system

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u(t)$$

$h = 1$

$\mathcal{E} = [0, 5] \times [0, 4]$

$p(||z - y_k||) = 1 - e^{-\lambda||z-y_k||^2},\ \lambda > 0$

$\phi(z) = \frac{1}{20}$

$R = diag(1, 1)$

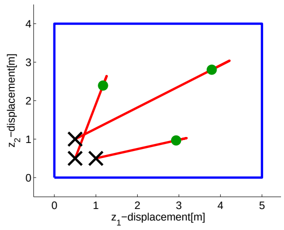
Simulation (g=1)

$\boldsymbol{x}_0^{(1)} = [0.5\ \ 0.5\ \ 0\ \ 0]^T$ $\lambda = 0.5$

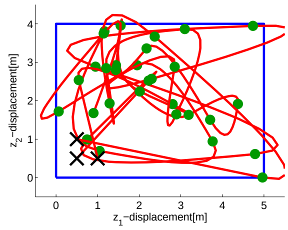
$\boldsymbol{x}_0^{(2)} = [1\ \ 0.5\ \ 0\ \ 0]^T$ $g = 1$

$\boldsymbol{x}_0^{(3)} = [0.5\ \ 1\ \ 0\ \ 0]^T$ $f = 5$

10 steps

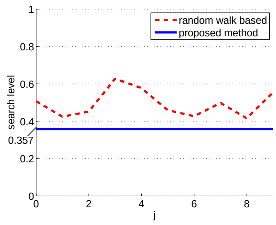


proposed method

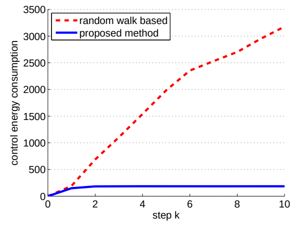


random walk based method

Simulation (g=1)



search level $S(\mathcal{Y}_{j+1:j+g})$



control energy consumption

$\alpha_g \approx 0.3573 \Rightarrow \mathbb{E}[\beta] \approx 1.556$

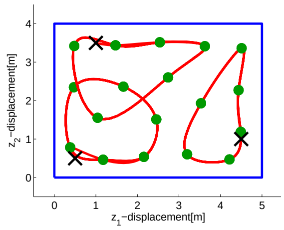
Simulation (g=6)

$\boldsymbol{x}_0^{(1)} = [0.5\ \ 0.5\ \ 0\ \ 0]^T$ $\lambda = 2$

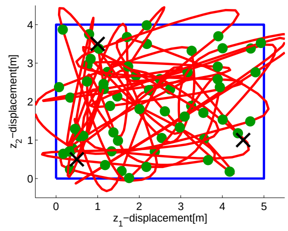
$\boldsymbol{x}_0^{(2)} = [4.5\ \ 1\ \ 0\ \ 0]^T$ $g = 6$

$\boldsymbol{x}_0^{(3)} = [1\ \ 3.5\ \ 0\ \ 0]^T$ $f = 10$

20 steps

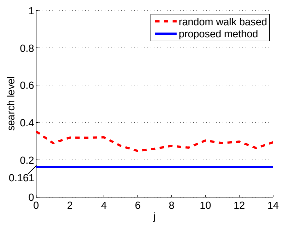


proposed method

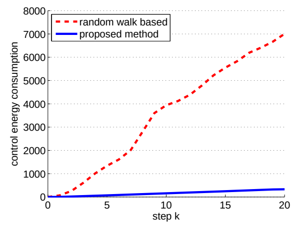


random walk based method

Simulation (g=6)



search level $S(\mathcal{Y}_{j+1:j+g})$



control energy consumption

$\alpha_g \approx 0.1612 \Rightarrow \mathbb{E}[\beta] \approx 1.192$

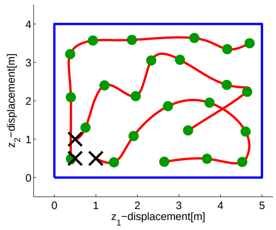
Simulation (g=k+f)

$\boldsymbol{x}_0^{(1)} = [0.5\ \ 0.5\ \ 0\ \ 0]^T$ $\lambda = 2$

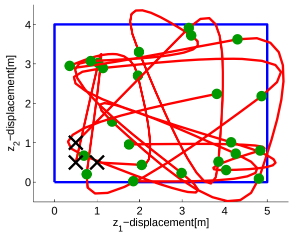
$\boldsymbol{x}_0^{(2)} = [1\ \ 0.5\ \ 0\ \ 0]^T$ $g = k + f$ (for stationary target)

$\boldsymbol{x}_0^{(3)} = [0.5\ \ 1\ \ 0\ \ 0]^T$ $f = 8$

8 steps

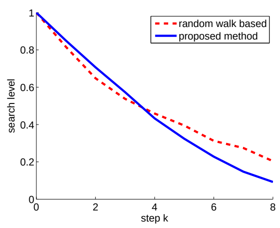


proposed method

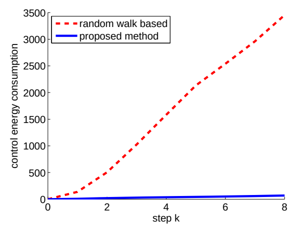


random walk based method

Simulation (g=k+f)



search level $S(\mathcal{Y}_{1:k})$



control energy consumption

Outline

- review (single agent)
- Cooperative Search
  - Centralized Control
  - Distributed Control

Distributed Cooperative Search

proposed method

1. divide the search area at time $t_k, \quad k = g, 2g, \ldots$
   ⇨ shared search area
2. Each agent computes the optimal control input.

Objective

1. share the search area optimally
2. maximize the probability of finding the target
3. minimize the control energy consumption

⇨ Multi-Objective Optimization

Divide the Search Area

$\boldsymbol{y}_k^{(1:n_a)} := \boldsymbol{y}_k^{(1)}, \boldsymbol{y}_k^{(2)}, \ldots, \boldsymbol{y}_k^{(n_a)}$

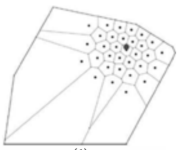
Voronoi Partition $\mathcal{V}(\boldsymbol{y}_k^{(1:n_a)}) := \left\{ \mathcal{V}_1(\boldsymbol{y}_k^{(1:n_a)}), \mathcal{V}_2(\boldsymbol{y}_k^{(1:n_a)}), \ldots, \mathcal{V}_{n_a}(\boldsymbol{y}_k^{(1:n_a)}) \right\}$

$\mathcal{V}_l(\boldsymbol{y}_k^{(1:n_a)}) := \left\{ \boldsymbol{z} \in \mathcal{E} \middle| \|\boldsymbol{z} - \boldsymbol{y}_k^{(l)}\| \le \|\boldsymbol{z} - \boldsymbol{y}_k^{(l')}\| \ \forall l' \ne l \right\}, l = 1, 2, \ldots, n_a$

Lemma

$M_{\mathcal{V}_l} := \int_{\mathcal{V}_l(\boldsymbol{y}_k^{(1:n_a)})} \phi(\boldsymbol{z}) d\boldsymbol{z}$

$C_{\mathcal{V}_l} := \frac{1}{M_{\mathcal{V}_l}} \int_{\mathcal{V}_l(\boldsymbol{y}_k^{(1:n_a)})} \boldsymbol{z} \phi(\boldsymbol{z}) d\boldsymbol{z}$ (19)



$\boldsymbol{y}_k^{(l)} = C_{\mathcal{V}_l}, \ l = 1, 2, \ldots, n_a$

$\Rightarrow \boldsymbol{y}_k^{(1:n_a)}$ (locally) minimizes $\sum_{l=1}^{n_a} \int_{\mathcal{V}_l(\boldsymbol{y}_k^{(1:n_a)})} \|\boldsymbol{z} - \boldsymbol{y}_k^{(l)}\|^2 \phi(\boldsymbol{z}) d\boldsymbol{z}$

proof: omitted. □

Discrete-Time Lloyd Algorithm

Algorithm 3 *Discrete-Time Lloyd Algorithm*

1: $k \leftarrow 0$
2: **while** not convergence
3: Compute $\mathcal{V}(\boldsymbol{y}_k^{(1:n_a)})$
4: Compute $C_{\mathcal{V}_l}$, $l = 1, 2, \ldots, n_a$ from (19)
5: $\boldsymbol{y}_{k+1}^{(l)} = C_{\mathcal{V}_l}$, $l = 1, 2, \ldots, n_a$
6: $k \leftarrow k + 1$
7: **end while**

ref. J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage Control for Mobile Sensing Networks. *IEEE Transactions on Robotics and Automation*, 20(2), pp.243-255, 2004.

Distributed Control Algorithm (Outline)

- at time $t_k, \quad k = g, 2g, \ldots \ (\Leftrightarrow mod(k, g) = 0)$
  $\boldsymbol{y}_k^{(l)} = C_{\mathcal{V}_l}$ and renew the Voronoi Partition $\mathcal{V}(\boldsymbol{y}_k^{(1:n_a)})$.
- at the other time
  optimal search control in $\mathcal{V}_l(\boldsymbol{y}_k^{(1:n_a)})$

remark :

Voronoi Partition may not be optimal division in this approach to cooperative search.

This is a synchronous method.

Distributed Cooperative Search

Distributed Cooperative Optimal Search Control Problem (DCOSCP)

$$\min_{\boldsymbol{u}^{(l)}(t), t \in [t_k, t_{k+f}]} \sum_{i=k}^{k+f-1} \int_{t_i}^{t_{i+1}} \boldsymbol{u}^{(l)T}(t) R \boldsymbol{u}^{(l)}(t) dt$$

s.t. $\min_{\tilde{\mathcal{Y}}_{k+1:k+f}^{(l)}} \sum_{j=k}^{k+f-g} S_{\mathcal{V}_l}(\mathcal{Y}_{j+1:j+g}^{(l)})$

$S_{\mathcal{V}_l}(\mathcal{Y}_{j+1:j+g}^{(l)}) := \int_{\mathcal{V}_l(\boldsymbol{y}_k^{(1:n_a)})} \phi(\boldsymbol{z}) \prod_{\boldsymbol{y}_k^{(l)} \in \mathcal{Y}_{j+1:j+g}^{(l)}} p(\|\boldsymbol{z} - \boldsymbol{y}_k^{(l)}\|) d\boldsymbol{z}$

s.t. $\boldsymbol{y}_i^{(l)} = C_{\mathcal{V}_l}$ if $mod(i, g) = 0, \ i = k+1, k+2, \ldots, k+f$

Distributed Control Algorithm

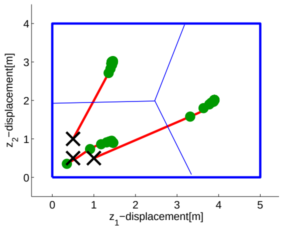
Algorithm 4 *Optimal Search Control Algorithm (Distributed Control)*

1: $k \leftarrow 0$
2: **while** 1
3: **if** $mod(k, g) = 0$
4: Compute $\mathcal{V}(\boldsymbol{y}_k^{(1:n_a)})$
5: Compute $C_{\mathcal{V}_l}$ from (19)
6: **end if**
7: Compute $\tilde{\mathcal{Y}}_{k+1:k+f}^{(l)}$ by solving DCOSCP
8: Compute $\boldsymbol{u}^{(l)}(t), t \in [t_k, t_{k+f}]$ from $\tilde{\mathcal{Y}}_{k+1:k+f}^{(l)}$,(14),(15)
9: Input $\boldsymbol{u}^{(l)}(t), t \in [t_k, t_{k+1})$
10: $k \leftarrow k+1$
11: **end while**

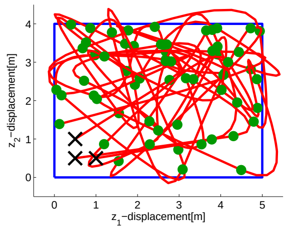
Simulation (g=1)

$x_0^{(1)} = [0.5 \;\; 0.5 \;\; 0 \;\; 0]^T$ $\lambda = 0.5$
$x_0^{(2)} = [1 \;\; 0.5 \;\; 0 \;\; 0]^T$ $g = 1$
$x_0^{(3)} = [0.5 \;\; 1 \;\; 0 \;\; 0]^T$ $f = 5$

20 steps

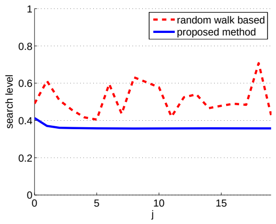


proposed method

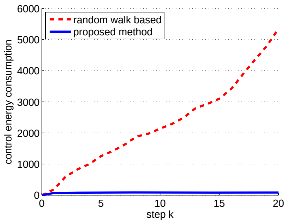


random walk based method

Simulation (g=1)



search level $S(\mathcal{Y}_{j+1:j+g})$



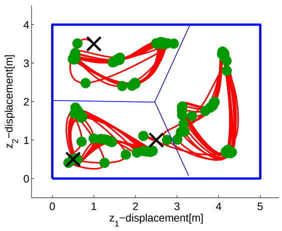
control energy consumption

$\alpha_g \approx 0.3572$ ⇨ $\mathbb{E}[\beta] \approx 1.556$

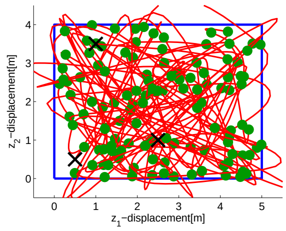
Simulation (g=4)

$x_0^{(1)} = [0.5 \;\; 0.5 \;\; 0 \;\; 0]^T$ $\lambda = 1.5$
$x_0^{(2)} = [2.5 \;\; 1 \;\; 0 \;\; 0]^T$ $g = 4$
$x_0^{(3)} = [1 \;\; 3.5 \;\; 0 \;\; 0]^T$ $f = 10$

40 steps

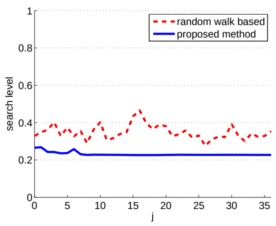


proposed method

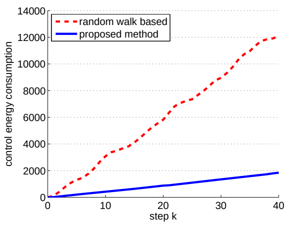


random walk based method

Simulation (g=4)



search level $S(\mathcal{Y}_{j+1:j+g})$



control energy consumption

$\alpha_g \approx 0.2269$ ⇨ $\mathbb{E}[\beta] \approx 1.294$

Conclusion and Future Works

Conclusion

- Centralized Cooperative Optimal Search Control
- Distributed Cooperative Optimal Search Control

Future Works

- Another Distributed Search method
  - how to divide the search area
  - asynchronous
- Collision Avoidance