Optimal Search Control

Convergence to Periodic Trajectory and Cooperative Search (Introduction)

FL08-09-2

Mamoru Saito

Outline

- Search Problem
- "One-sided search" – "Moving Target"
- Cooperative Search
- Conclusion, Future Works

Search Problem

Search Problem

To locate the target
deploying agents with the available resources.

- protection against submarine attacks
- search and rescue operations
- detecting lost objects
- clearing of land mines
- location parts in a warehouse
- medicines, mining, …etc.

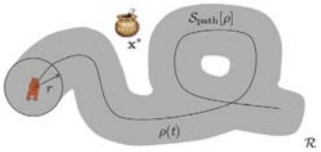
Types of Search Problems

	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

Continuous and Discrete

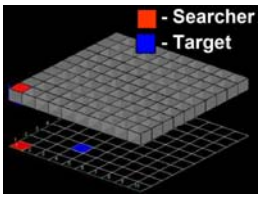
Continuous Search

continuous time space
and
continuous state space



Discrete Search

discrete time space
and(or)
discrete state space



Objective of Search

- ***Maximize the probability of locating the target.***
- ***Minimize the time to find the target.***
- ***Cover the search area.***
- ***Decide whether the target is present in the search area.***

In Previous Works

The dynamics is largely ignored.

agent

Unmanned Aerial Vehicles (UAVs)
(non-holonomic system)

agent

The argument about the available resources is not enough.

Approach to Search Problem

Approach
take account of agent system explicitly
limit control energy consumption

continuous-time linear system

discrete search
discrete observation time (but not discretize state space)

maximize the probability of locating the target

	One-sided Search	***Two-sided Search***
Stationary Target	M. Saito, et al., CCS, 2008.	
Moving Target	in this presentation	***Cooperative Search*** / ***Non-Cooperative Search***

Objective of Study

Optimal Search Problem
maximize the probability of locating the target

+

Optimal Control Problem
limit control energy consumption

formulate ***Optimal Search Control Problem***

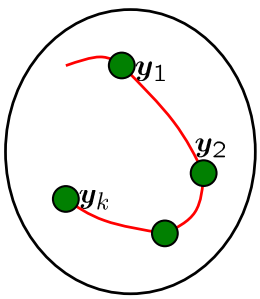
provides its approximate solution algorithm

The effectiveness of the proposed method is demonstrated through a numerical simulation

Outline

- Search Problem
- "One-sided search" – "Moving Target"
  - Problem Setting
  - formulation
  - approximate solution algorithm
  - converge to periodic trajectory
  - simulation
- Cooperative Search
- Conclusion, Future Works

Problem Setting



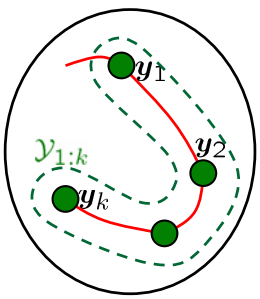
agent system

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$x(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} \quad \text{position}: y(t) \in \mathcal{R}^{n_y} \ (n_y \in \{1,2,3\})$$

observation time (obs. time) :
$t_k, k = 0, 1, 2, \cdots$

obs. point $y_k := y(t_k)$
waypoint $x_k := x(t_k)$

Problem Setting

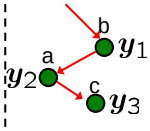


the obs. points set from t_p to t_q
$\mathcal{Y}_{p:q} := \{y_p, y_{p+1}, \cdots, y_q\}$

the obs. points row from t_p to t_q
$\tilde{\mathcal{Y}}_{p:q} := (y_p, y_{p+1}, \cdots, y_q)$

$sort(\mathcal{Y}_{p:q})$
→ sort of elements of $\mathcal{Y}_{p:q}$

ex.

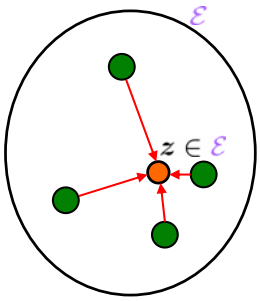


$\tilde{\mathcal{Y}}_{1:3} := (b, a, c)$
$\mathcal{Y}_{1:3} := \{a, b, c\}$
$sort(\mathcal{Y}_{1:3})$
$= \{(a,b,c),(a,c,b),(b,a,c),$
$(b,c,a),(c,a,b),(c,b,a)\}$

Problem Setting

search area: $\mathcal{E} \subset \mathcal{R}^2, \quad z \in \mathcal{E}$

the importance of search:
$\phi(z) > 0$ (large → important)



sensing accuracy $\in$ [0, 1) (good bad)
$p_y(||z - y_i||) = 1 - e^{-\lambda||z-y_i||^2}$

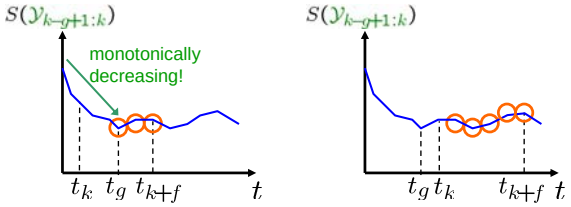
obs. level $\in$ [0, 1] (good bad)
$p(z, \mathcal{Y}_{p:q}) := \prod_{y_i \in \mathcal{Y}_{p:q}} p_y(||z - y_i||)$

search level at time t_k (small → good)
$S(\mathcal{Y}_{k-g+1:k}) := \int_{\mathcal{E}} \phi(z) p(z, \mathcal{Y}_{k-g+1:k}) dz$
$g \in \{1, 2, \ldots\}, \ f \geq g + 1$

Search and Control

optimal search minimize search level S ⇔ **optimal control** minimize control energy

tradeoff



criterion function for optimal search

$\tilde{S}(\tilde{\mathcal{Y}}_{1:k+f}) = \sum_{i=\max\{g,k+1\}}^{k+f} S(\mathcal{Y}_{i-g+1:i})$

Optimal Search Control Problem

Problem A1

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

s.t. $(y_i)_{i=k+1,k+2,\ldots,k+f} = \tilde{\mathcal{Y}}^*_{k+1:k+f} := \underset{\tilde{\mathcal{Y}}_{k+1:k+f}}{\operatorname{argmin}} \tilde{S}(\tilde{\mathcal{Y}}_{1:k+f})$

It's difficult to get **global optimal solution**.

relaxation of the condition

get **local optimal solution**

$\tilde{\mathcal{Y}}^*_{k+1:k+f} \in \tilde{\Phi}(\tilde{\mathcal{Y}}_{1:k}, f) := \left\{ \tilde{\mathcal{Y}}_{k+1:k+f} \middle| \frac{\partial \tilde{S}(\tilde{\mathcal{Y}}_{1:k+f})}{\partial y_i} = 0 \ \forall y_i \in \mathcal{Y}_{k+1:k+f} \right\}$

Reduce Problem A1

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

$$= \min_{X} \begin{bmatrix} x_k \\ X \end{bmatrix}^T N \begin{bmatrix} x_k \\ X \end{bmatrix}, \ X = \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+f} \end{bmatrix} \ (\because u^*(t) = \psi(x_i, x_{i+1}), \ t \in [t_i, t_{i+1}])$$

$$= \min_{Y, \dot{Y}} \begin{bmatrix} y_k \\ Y \\ \dot{y}_k \\ \dot{Y} \end{bmatrix}^T \left[\begin{array}{c|c} H_1[k] & H_2[k] \\ \hline H_2^T[k] & H_3[k] \end{array} \right] \begin{bmatrix} y_k \\ Y \\ \dot{y}_k \\ \dot{Y} \end{bmatrix}, \ Y = \begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+f} \end{bmatrix}$$

$$= \min_{Y} \underbrace{\begin{bmatrix} y_k \\ Y \\ \dot{y}_k \end{bmatrix}^T (H_1[k] - H_2[k] H_3^{-T}[k] H_2^T[k]) \begin{bmatrix} y_k \\ Y \\ \dot{y}_k \end{bmatrix}}_{J^*_{k:k+f}(\tilde{\mathcal{Y}}_{k+1:k+f})}, \ \dot{Y}^* = \begin{bmatrix} \dot{y}^*_{k+1} \\ \dot{y}^*_{k+2} \\ \vdots \\ \dot{y}^*_{k+f} \end{bmatrix} = -H_3^{-1}[k] H_2^T[k] \begin{bmatrix} y_k \\ Y \\ \dot{y}_k \end{bmatrix}$$

Problem A1 ⇨ **Problem A2**

$$\min_{\tilde{\mathcal{Y}}_{k+1:k+f}} J^*_{k:k+f}(\tilde{\mathcal{Y}}_{k+1:k+f})$$
s.t. $\tilde{\mathcal{Y}}_{k+1:k+f} = \tilde{\mathcal{Y}}^*_{k+1:k+f} \in \tilde{\Phi}(\tilde{\mathcal{Y}}_{1:k}, f)$

Approximate Solution Algorithm

Algorithm 1

if $k = 0$

1. $\mathcal{Y}^*_{1:g} \in \Phi(\emptyset, g) := \left\{ \mathcal{Y}_{1:g} \middle| \frac{\partial S(\mathcal{Y}_{1:g})}{\partial y_i} = 0 \ \forall y_i \in \mathcal{Y}_{1:g} \right\}$

   → gradient method $y_i^{j+1} = y_i^j + \alpha_j \frac{\partial S(\mathcal{Y}^j_{1:g})}{\partial y_i^j}$ j :index $i \in \{1, \cdots, g\}$

2. $\tilde{\mathcal{Y}}^*_{1:1+g} = \underset{\tilde{\mathcal{Y}}_{1:g} \in sort(\mathcal{Y}^*_{1:g}), y_{1+g} = y_1}{\operatorname{argmin}} J^*_{0:1+g}(\tilde{\mathcal{Y}}_{1:1+g})$

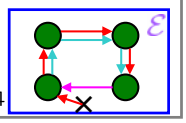
   get a round path to minimize control energy

   → enumeration method $O(g!)$

   → approximate solution : Ant Colony Optimization $O(g^2)$

end if

3. $y_{i+g} = y_i, \ i = 1, 2, \ldots$



Lemma 1

Lemma 1

Algorithm 1 leads below. $(k = g, g+1, \ldots)$

$\mathcal{Y}_{k-g+1:k} = \mathcal{Y}_{1:g}$ --- (1)

$S(\mathcal{Y}_{k-g+1:k}) = S(\mathcal{Y}_{1:g})$ --- (2)

$\mathcal{Y}_{k-g+1:k} \in \Phi(\emptyset, g) = \left\{ \mathcal{Y}_{k-g+1:k} \middle| \frac{\partial S(\mathcal{Y}_{k-g+1:k})}{\partial y_i} = 0 \ \forall y_i \in \mathcal{Y}_{k-g+1:k} \right\}$ --- (3)

$\frac{\partial S(\mathcal{Y}_{k-g+1:k})}{\partial y_j} = 0 \ \forall j \in \{1, 2, \ldots\}$ --- (4)

proof:(1) → $\mathcal{Y}_{1:g} = \{y_1\} \cup \mathcal{Y}_{2:g} = \mathcal{Y}_{2:g} \cup \{y_{1+g}\} = \mathcal{Y}_{2:1+g} (\because y_{1+g} = y_1)$
Repeat above.

(2) → obviously by (1)

(3) → obviously by $\mathcal{Y}_{1:1+a} \in \Phi(\emptyset, g)$ and (1)

(4) → $\frac{\partial S(\mathcal{Y}_{k-g+1:k})}{\partial y_j} = 0 \ \forall j \in \{1, 2, \ldots\} / \{k-g+1, \ldots, k\}$
because $S(\mathcal{Y}_{k-g+1:k})$ is not depend on y_j.
$\frac{\partial S(\mathcal{Y}_{k-g+1:k})}{\partial y_j} = 0 \ \forall j \in \{k-g+1, \ldots, k\}$ by (3). □

Theorem 1

Theorem 1

$\bar{\mathcal{Y}}_{k+1:k+f}$ gotten by using Algorithm 1
is one local optimal obs. points row of Problem A2,
so that, $\bar{\mathcal{Y}}_{k+1:k+f} = \bar{\mathcal{Y}}^*_{k+1:k+f} \in \bar{\Phi}(\bar{\mathcal{Y}}_{1:k}, f)$

proof:

$\forall i \in \{k+1, \cdots, k+f\},$

$$\frac{\partial \bar{S}(\bar{\mathcal{Y}}_{1:k+f})}{\partial \boldsymbol{y}_i} = \sum_{i'=\max\{g,k+1\}}^{k+f} \frac{\partial S(\mathcal{Y}_{i'-g+1:i'})}{\partial \boldsymbol{y}_i}$$

$$= 0 \quad \left(\because \frac{\partial S(\mathcal{Y}_{i'-g+1:i'})}{\partial \boldsymbol{y}_j} = 0 \ \forall j \in \{1,2,\ldots\} \text{ by Lemma 1 (4)} \right)$$

So $\bar{\mathcal{Y}}_{k+1:k+f} = \bar{\mathcal{Y}}^*_{k+1:k+f} \in \bar{\Phi}(\bar{\mathcal{Y}}_{1:k}, f)$

.

□

Receding Horizon Control

at time t_k horizon : $t \in [t_k, t_{k+f}]$

Optimal Search Control Problem

$\bar{\mathcal{Y}}_{k+1:k+f}$ (Algorithm 1)

$$\begin{bmatrix} \dot{\boldsymbol{y}}_{k+1} \\ \dot{\boldsymbol{y}}_{k+2} \\ \vdots \\ \dot{\boldsymbol{y}}_{k+f} \end{bmatrix} = -H_3^{-1}[k] H_2^T[k] \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} \\ \vdots \\ \boldsymbol{y}_{k+f} \\ \dot{\boldsymbol{y}}_k \end{bmatrix}$$

$\boldsymbol{u}^*(t) = \psi(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}), \ t \in [t_i, t_{i+1}]$

$\Rightarrow$ $\boldsymbol{u}^*(t), t \in [t_k, t_{k+f}]$

$\Downarrow$

$\boldsymbol{u}^*(t), t \in [t_k, t_{k+1})$

$\Downarrow$

at time t_{k+1} horizon : $t \in [t_{k+1}, t_{k+1+f}]$
same above …

$\Downarrow$

⋮

Converge to Periodic Trajectory

Lemma 2

Suppose $t_{i+1} - t_i = t_{i+1+g} - t_{i+g}, \ i = 1, 2, \ldots$ and $T = t_{i+g} - t_i$.

Algorithm 1 + $\dot{\boldsymbol{y}}(t_k + T) - \dot{\boldsymbol{y}}(t_k) \to 0$ (as $k \to \infty$)

$\Rightarrow$ The state (input) trajectory converges to a periodic trajectory. (period T)

proof: Algorithm 1 $\Rightarrow$ $\boldsymbol{y}(t_i + T) = \boldsymbol{y}(t_i), \ \boldsymbol{y}(t_{i+1} + T) = \boldsymbol{y}(t_{i+1})$

If $\dot{\boldsymbol{y}}(t_i + T) = \dot{\boldsymbol{y}}(t_i), \ \dot{\boldsymbol{y}}(t_{i+1} + T) = \dot{\boldsymbol{y}}(t_{i+1}),$
then $\boldsymbol{x}(t+T) = \boldsymbol{x}(t), \ \boldsymbol{u}(t+T) = \boldsymbol{u}(t), \ t \in [t_i, t_{i+1}]$
$\left(\because \boldsymbol{u}^*(t) = \psi_1(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}), \ \boldsymbol{x}^*(t) = \psi_2(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}), \ t \in [t_i, t_{i+1}] \right)$

So if $\dot{\boldsymbol{y}}(t_k + T) - \dot{\boldsymbol{y}}(t_k) \to 0$ (as $k \to \infty$),
then, the state (input) trajectory
converges to a periodic trajectory. □

Theorem 2

Theorem 2

Suppose $t_{i+1} - t_i = t_{i+1+g} - t_{i+g}, \ i = 1, 2, \ldots$ and $T = t_{i+g} - t_i$.

$$G[k] := -\begin{bmatrix} I_{n_y \times n_y} & 0 & \cdots & 0 \end{bmatrix} H_3^{-1}[k] H_2^T[k] \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_{n_y \times n_y} \end{bmatrix} \quad (G[k+g] = G[k])$$

$\boldsymbol{e}[k] := \dot{\boldsymbol{y}}(t_k + T) - \dot{\boldsymbol{y}}(t_k)$

If $\boldsymbol{e}[k+1] = G[k]\boldsymbol{e}[k], \ G[k+g] = G[k]$ is asymptotically stable,
then the state (input) trajectory converges to a periodic trajectory.

proof:

$$\dot{\boldsymbol{y}}(t_{k+1}) = -\begin{bmatrix} I_{n_y \times n_y} & 0 & \cdots & 0 \end{bmatrix} H_3^{-1}[k] H_2^T[k] \begin{bmatrix} \boldsymbol{y}_k \\ \vdots \\ \boldsymbol{y}_{k+f} \\ \dot{\boldsymbol{y}}(t_k) \end{bmatrix} \quad \left(\because \begin{bmatrix} \dot{\boldsymbol{y}}_{k+1} \\ \dot{\boldsymbol{y}}_{k+2} \\ \vdots \\ \dot{\boldsymbol{y}}_{k+f} \end{bmatrix} = -H_3^{-1}[k] H_2^T[k] \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} \\ \vdots \\ \boldsymbol{y}_{k+f} \\ \dot{\boldsymbol{y}}_k \end{bmatrix} \right)$$

$$\| \quad \text{(same } H_3^{-1}[k] H_2^T[k] \text{ as above)}$$

$$\dot{\boldsymbol{y}}(t_{k+1} + T) = -\begin{bmatrix} I_{n_y \times n_y} & 0 & \cdots & 0 \end{bmatrix} H_3^{-1}[k+g] H_2^T[k+g] \begin{bmatrix} \boldsymbol{y}_{k+g} \\ \vdots \\ \boldsymbol{y}_{k+f+g} \\ \dot{\boldsymbol{y}}(t_{k+g}) \end{bmatrix}$$

Theorem 2

$$\boldsymbol{e}[k+1] = \dot{\boldsymbol{y}}(t_{k+1} + T) - \dot{\boldsymbol{y}}(t_{k+1})$$

$$= -\begin{bmatrix} I_{n_y \times n_y} & 0 & \cdots & 0 \end{bmatrix} H_3^{-1}[k] H_2^T[k] \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \dot{\boldsymbol{y}}(t_{k+g}) - \dot{\boldsymbol{y}}(t_k) \end{bmatrix}$$

$$= -\begin{bmatrix} I_{n_y \times n_y} & 0 & \cdots & 0 \end{bmatrix} H_3^{-1}[k] H_2^T[k] \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{e}[k] \end{bmatrix}$$

$$= -\begin{bmatrix} I_{n_y \times n_y} & 0 & \cdots & 0 \end{bmatrix} H_3^{-1}[k] H_2^T[k] \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_{n_y \times n_y} \end{bmatrix} \boldsymbol{e}[k]$$

$$= G[k]\boldsymbol{e}[k]$$

If $\boldsymbol{e}[k+1] = G[k]\boldsymbol{e}[k], \ G[k+g] = G[k]$ is asymptotically stable,
then $\boldsymbol{e}[k] = \dot{\boldsymbol{y}}(t_k + T) - \dot{\boldsymbol{y}}(t_k) \to 0$ (as $k \to \infty$).

And by Lemma 2,
the state (input) trajectory converges to a periodic trajectory.

□

Corollary 1

Suppose $t_{i+1} - t_i = h, \ i = 1, 2, \ldots$ and $T = gh$,

$\Rightarrow$ $G[k+1] = G[k] =: G, \ k = 1, 2, \ldots$.

$\Rightarrow$ $\boldsymbol{e}[k+1] = G\boldsymbol{e}[k]$ is time-invariant system.

Corollary 1

If $|\lambda_i| < 1, \ i \in \{1, \ldots, n_y\}$ (λ_i is an eigenvalue of G),
then the state (input) trajectory converges to a periodic trajectory.

proof:

If $|\lambda_i| < 1, \ i \in \{1, \ldots, n_y\}$
then $\boldsymbol{e}[k+1] = G\boldsymbol{e}[k]$ is asymptotically stable.
and $\boldsymbol{e}[k] = \dot{\boldsymbol{y}}(t_k + T) - \dot{\boldsymbol{y}}(t_k) \to 0$ (as $k \to \infty$).

And by Lemma 2,
the state (input) trajectory converges to a periodic trajectory.

□

Simulation

agent system

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

$$x_0 = \begin{bmatrix} 12 & 12 & 0 & 0 \end{bmatrix}^T$$

$\mathcal{E} = [0, 40] \times [0, 40]$

$\phi(z) = 1$

$t_i = ih, \ h = 5$

$R = diag(1, 1)$

$\lambda = 0.02$

$f = 5, \ g = 4$

- eigenvalues of $G = -0.1250, -0.1250$
- converge to periodic trajectory (period $T = 20$)

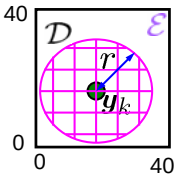
random walk based method

$$\mathcal{D} := \left\{ z \in \mathcal{E}_d \middle| \|z - y_k\| < r \right\}$$

$\mathcal{E}_d$: discretize $\mathcal{E}$

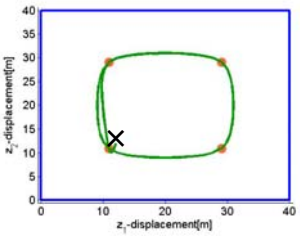
select $y_{i+1} \in \mathcal{D}$ randomly

$r = 60$

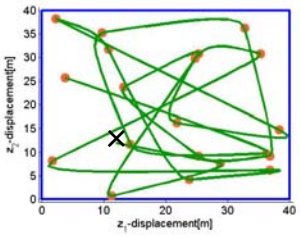


Simulation

20 steps

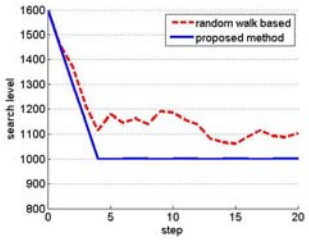


proposed method

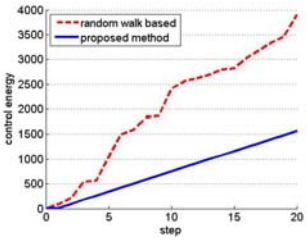


random walk based method

Simulation



search level $S(\mathcal{Y}_{k-g+1:k})$ at t_k



the sum of control energy

Outline

- Search Problem
- "One-sided search" – "Moving Target"
- Cooperative Search
- Conclusion, Future Works

Cooperative Search

Cooperative Search

"Two-sided Search" – "Moving Target"

ex. ***Rendezvous Search***

Each party behaves in a manner
that maximizes the chances of one party finding the other.
Each party attempts to make itself as detectable as possible.

ex. Two people have become separated in a crowd
and wish to find one another again.

ex. Searching for an intelligent person lost in the woods
who understands how the searchers will operate.

That person may try to move to a place
where he can be found more easily.

Rendezvous Search

Rendezvous Search

This is the problem that arises, for example,
when two people shopping together in a department store look around
and see that they are no longer together.
How should they proceed to find each other?
Should one stay still and hope that the other is searching the whole store?
What should they agree in advance to help them if this situation arises?

- quoted from [5]

- ***symmetric rendezvous search***
  Agents use same strategies.
- ***asymmetric rendezvous search***
  Agents can use distinct strategies.
  ex. "Wait for Mommy"
  One player stays still while the others searches the entire space.

Rendezvous Search Control

"Wait for Mommy" strategy

One agent stays still while the others searches the entire space.

1. Decide the role of each agent.
   (one target)
2. The target agent stays at initial position.
   The searcher searches "the stationary target".
   ⇨ ([27] and last presentation)
   Each searcher can detect the target agent with probability one (as $t \to \infty$).

When the searcher detects the target, he moves to the position of the target.

⇩

The rendezvous will be achieved!

Conclusion and Future Works

Conclusion

"One-sided search" – "Moving Target"

- formulation of Optimal Search Control Problem
- its approximate solution
- the condition to converge to a periodic trajectory
- The effectiveness of the proposed method was demonstrated through a numerical simulation

The introduction of "Cooperative Search"

Future Works

Cooperative Search

reduce computation time

search control under non-convex state constraints