A combinatorial model of two-sided search

Harout Aydinian, Ferdinando Cicalese, Christian Deppe, and Vladimir Lebedev

Lehrstuhl für Theoretische Informationstechnik
Technische Universität München
h.aydinyan@tum.de

SOFSEM 2016
Harrachov, January 23-28

Outline

Outline

Outline

Outline

Outline

Outline

Introduction

Search theory deals with the problem faced by a searcher:

Finding a hidden object, in a given search space, in minimum time

In most of early developments it is assumed that an object to be searched is stationary and hidden according to a known distribution or it is moving and its motion is determined, by some known rules. This model of search is called one-sided search.

Introduction

Search theory deals with the problem faced by a searcher:

Finding a hidden object, in a given search space, in minimum time

In most of early developments it is assumed that an object to be searched is stationary and hidden according to a known distribution or it is moving and its motion is determined, by some known rules. This model of search is called one-sided search.

In case the object can attempt to contrast the searcher's activity and react in some intelligent way in order not to be found, the problem is called two-sided search.

History

The first developments in search theory were made by Bernard Koopman and his colleagues in the Anti-Submarine Warfare Operations Research Group of the U.S. Navy during World War II.

Their purpose was to provide efficient ways to search for enemy submarines. Their work which was only published later, also mentioned two-sided search.

History

The first developments in search theory were made by Bernard Koopman and his colleagues in the Anti-Submarine Warfare Operations Research Group of the U.S. Navy during World War II.

Their purpose was to provide efficient ways to search for enemy submarines. Their work which was only published later, also mentioned two-sided search.

Here, we consider a *combinatorial* model of two sided search which was proposed by the late Rudolf Ahlswede during the Workshop "Search Methodologies II" (2010).

In a *combinatorial search problem* the object(s) to be found live in a discrete space and the tests to be asked satisfy certain specified requirements.

Problem statement

- $G = (\mathcal{N}, \mathcal{E})$: undirected graph

Problem statement

- $G = (\mathcal{N}, \mathcal{E})$: undirected graph
- Search space $\mathcal{N} = \{1, 2, \ldots, N\}$: the vertices of G

Problem statement

- $G = (\mathcal{N}, \mathcal{E})$: undirected graph
- Search space $\mathcal{N} = \{1, 2, \ldots, N\}$: the vertices of G
- The object to be found (or *target*) occupies some vertex of $\mathcal{N}$, unknown to the searcher.

Problem statement

- $G = (\mathcal{N}, \mathcal{E})$: undirected graph
- Search space $\mathcal{N} = \{1, 2, \ldots, N\}$: the vertices of G
- The object to be found (or *target*) occupies some vertex of $\mathcal{N}$, unknown to the searcher.
- At any time instance and for any $\mathcal{T} \subset \mathcal{N}$, called a *test set* the searcher can learn whether the target is located at some node in $\mathcal{T}$ or not.

Problem statement

- $G = (\mathcal{N}, \mathcal{E})$: undirected graph
- Search space $\mathcal{N} = \{1, 2, \ldots, N\}$: the vertices of G
- The object to be found (or *target*) occupies some vertex of $\mathcal{N}$, unknown to the searcher.
- At any time instance and for any $\mathcal{T} \subset \mathcal{N}$, called a *test set* the searcher can learn whether the target is located at some node in $\mathcal{T}$ or not.
- After each test, the target can move to any vertex adjacent to its current location or stay at the same place.

Problem statement

- $G = (\mathcal{N}, \mathcal{E})$: undirected graph
- Search space $\mathcal{N} = \{1, 2, \ldots, N\}$: the vertices of G
- The object to be found (or *target*) occupies some vertex of $\mathcal{N}$, unknown to the searcher.
- At any time instance and for any $\mathcal{T} \subset \mathcal{N}$, called a *test set* the searcher can learn whether the target is located at some node in $\mathcal{T}$ or not.
- After each test, the target can move to any vertex adjacent to its current location or stay at the same place.
- The search finishes when the searcher is able to locate the object with some predifined accuracy s

Problem statement

- $G = (\mathcal{N}, \mathcal{E})$: undirected graph
- Search space $\mathcal{N} = \{1, 2, \ldots, N\}$: the vertices of G
- The object to be found (or *target*) occupies some vertex of $\mathcal{N}$, unknown to the searcher.
- At any time instance and for any $\mathcal{T} \subset \mathcal{N}$, called a *test set* the searcher can learn whether the target is located at some node in $\mathcal{T}$ or not.
- After each test, the target can move to any vertex adjacent to its current location or stay at the same place.
- The search finishes when the searcher is able to locate the object with some predifined accuracy s

Problem

Find the location of the target, with a certain accuracy, in minimum time

Formal description of the adaptive search model

d_j: location of the target at time j
The sequence of target positions until time n is given by the vector $(d_1, \ldots, d_n) \in \mathcal{N}^n$ which defines a walk in the graph G, i.e., $(d_i, d_{i+1}) \in \mathcal{E}$ $(i = 1, \ldots, n-1)$.

$$f_{\mathcal{T}}(d) = \begin{cases} 0 \;\; \text{(No)} & , \text{ if } d \notin \mathcal{T} \\ 1 \;\; \text{(Yes)} & , \text{ if } d \in \mathcal{T} \end{cases}$$

$\mathcal{T}_j$: the test performed at time j.
The sequence $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n)$ is called a sequential or adaptive strategy of length n.
$\mathcal{D}_i$: the set of possible positions of the target after the ith test
Thus $\mathcal{D}_0 = \mathcal{N}$ and for $i \geq 1$

$$\mathcal{D}_i = \begin{cases} \Gamma(\mathcal{T}_i \cap \mathcal{D}_{i-1}) & , \text{ if } f_{\mathcal{T}_i}(d_i) = 1 \\ \Gamma(\mathcal{D}_{i-1} \setminus \mathcal{T}_i) & , \text{ if } f_{\mathcal{T}_i}(d_i) = 0, \end{cases}$$

Definition

Given a $G = (\mathcal{N}, \mathcal{E})$, a sequential strategy of length n, $\mathcal{T}_1, \ldots \mathcal{T}_n$, is called (G, s)–successful if for any possible sequence of the target's movements $(d_1, \ldots, d_n)$, we have that $|\mathcal{D}_i| \leq s$ for some $i \leq n$.

- $s^*(G)$: the minimum number s^* such that there exists a (G, s^*)–successful strategy
- $n(G, s)$: the minimum number n such that there exists a (G, s)–successful strategy of length n
- Such a strategy is called optimal (G, s) strategy.

Motivation

The model has application to the area of node selection for target tracking in sensor networks. When the task of the network is the tracking of objects, a major initial task is to determine an area where the object to be tracked is surely initially located, and from which the actual tracking procedure can start. This localization together with the minimization of the area of localization is one of the most critical and expensive part of the tracking procedure, as it is typically done by an exhaustive search. For the sake of reducing the bandwidth consumption, sensors networks are also hierarchically organized in graph and more specifically tree structures.

Therefore, our results can be used to support the localization phase while trying to reduce the area of localization and reducing the number of activations of sensors.

Relation to other works

Group testing in graph has been considered both in terms of searching for an edge and for a vertex, and for different models of the test allowed. However, in all these works the basic assumption is that the target is still which makes the problem significantly different.

Another area of research related to our problem is graph searching. Graph searching a wide variety of combinatorial problems related to the problem of capturing a fugitive residing in a graph using the minimum number of searchers. Although there are many different models of graph searching, none appear to cover the type of two side combinatorial search we are considering here.

Models of search related to ours appear to be the so called cop and robber game and domination search games.

Optimal Strategies for Cycles and Paths

C_N: a cycle on N vertices
P_N: a path graph on N vertices
We consider the dual of the parameters $n(C_N, s)$ and $n(P_N, s)$

Given integers $n, s \geq 1$, we denote by $N_c(n, s)$ (resp. $N_p(n, s)$) the maximum N, such that there exists an (C_N, s)–successful (resp. (P_N, s)–successful) strategy of length n.

Clearly,

$$n(C_N, s) = \min\{i : N_c(i, s) \geq N\}$$

$$n(P_N, s) = \min\{i : N_p(i, s) \geq N\}$$

Cycles

$C_N = (\mathcal{N}, \mathcal{E})$: a cycle of length N with a loop in each node.
$\mathcal{N} = \{1, \ldots, N\}$
$\mathcal{E} = \{\{i, i+1\} : 1 \leq i \leq N-1\} \cup \{N, 1\} \cup \{\{i, i\} : i \in \mathcal{N}\}$.

Proposition

For $N \geq 5$ there does not exist a (C_N, s)–successful strategy with $s \leq 4$, that is $s^*(C_N) \geq 5$.

For $s \geq 5$, we can characterize the size of optimal (C_N, s) strategies.

Theorem

For any $s \geq 5$ and any $n \geq 0$ we have $N_c(n, s) = 2^n(s-4)+4$.

Proof

Induction on n: The case $n = 0$ is trivial.
$n - 1 \to n$: Suppose $N_c(n, s) > 2^n(s - 4) + 4$ with $(\mathcal{T}_1, \ldots, \mathcal{T}_n)$.
Note that
$|\mathcal{D}_1| \geq |\mathcal{T}_1| + 2$ if $f_{\mathcal{T}_1}(d_1) = 1$ and $|\mathcal{D}_1| \geq |\mathcal{N} \setminus \mathcal{T}_1| + 2$ if $f_{\mathcal{T}_1}(d_1) = 0$,
with equality in both cases iff $\mathcal{T}_1$ is a path in $\mathcal{C}_N$.

This implies that
$|\mathcal{D}_1| \geq \lceil N_c(n, s)/2 \rceil + 2 > (2^n(s - 4) + 4)/2 + 2 = 2^{n-1}(s - 4) + 4$,
a contradiction with the induction hypothesis
$|\mathcal{D}_1| \leq N_c(n - 1, s) = 2^{n-1}(s - 4) + 4$.

Hence $\qquad N_c(n, s) \leq 2^n(s - 4) + 4$.

In case $N = 2^n(s - 4) + 4$ we take as $\mathcal{T}_1$ a path on $\frac{N}{2}$ vertices, which is sufficient (and necessary) to get an optimal strategy, in view of the induction hypothesis.

Paths

Let P_N be a path graph on N vertices, thus
$\mathcal{E} = \{\{i, i+1\} : 1 \leq i < N\} \cup \{\{i, i\} : 1 \leq i \leq N\}$.

Proposition

For $N \geq 5$ there does not exist a (P_N, s)-successful strategy with $s \leq 3$, that is $s^*(P_N) \geq 4$.

Theorem

For $N \geq 4$ we have $n(N, 4) = \left\lceil \frac{N}{2} \right\rceil - 2$

Corollary

For $n \geq 0$ we have $N_p(n, 4) = 2n + 4$.

Optimal strategy for $s \geq 4$

We need much less tests, if we search for a final target set of size $s \geq 5$

Theorem

For $n \geq 0$ and $s \geq 4$ we have

$$N_p(n, s) = (s - 4)2^n + 2n + 4$$

Complete q-ary trees

$B_k^q = (\mathcal{N}, \mathcal{E})$: a complete q-ary tree of depth k, where $\mathcal{N} = \{1, 2, \ldots, q^{k+1} - 1\}$ and $|\mathcal{E}| = |\mathcal{N}| - 1$.
B_k^q has one vertex of degree q, called the root,
$q^k(q-1) - 2$ vertices of degree $q+1$, called the inner vertices,
and q^k vertices of degree 1, called the leaves.

Theorem

For $q \geq 2$ we have
(i) $s^*(B_1^q) = q + 1$
(ii) $s^*(B_2^q) = 2q + 1$
(iii) $s^*(B_3^q) = \lceil \frac{q+1}{2} \rceil q + q + j$, *where* $j = 0$ *if* $q = 2$ *and* $j = 1$ *if* $q \geq 3$
(iv) $s^*(B_4^q) = \lceil \frac{q+1}{2} \rceil q + q + i$,
where $i = 1$ *if* q *is even and* $i = 2$ *if* q *is odd*
(v) $s^*(B_k^q) = \lceil \frac{q+1}{2} \rceil q + q + 2$ *for all* $k \geq 5$.

General case

$\Delta(T)$: the *maximum degree* of a tree T
$r(T)$: the *radius* of T, i.e., $r(T) = \min_{v \in T} \max_{u \in T} d(u, v)$where $d(u, v)$ denote the length of the unique path between u and v in T.

Theorem

Let T be a tree, and let $r = r(T)$ and $\Delta = \Delta(T) \geq 3$. Then we have
(i) $s^*(T) \leq r(\Delta - 1) + 2$ for $r = 1, 2$.
(ii) $s^*(T) \leq \left(\lceil \frac{\Delta - 1}{2} \rceil + 1 \right) (\Delta - 1) + u$ for $r = 3$, where $u = 3$ if $\Delta = 3$ and $u = 2$ for $\Delta \geq 4$.
(iii) $s^*(T) \leq \left(\lceil \frac{\Delta}{2} \rceil + 1 \right) (\Delta - 1) + 2$ for $r \geq 4$.
The bounds are best possible in the sense that for any $r = 3, 4$ there exists a tree for which a lower bound can be shown that differs by 1 and for any $r \notin \{3, 4\}$ the lower bound exactly matches the corresponding upper bound in (i) and (iii).

Optimal strategies when the target is restricted:

The target can change its position at most t times

Thus, there are at most t distinct elements in a target walk $(d_1, \ldots, d_n)$. The notations $N_c(n,s,t)$ and $N_p(n,s,t)$ have the same meaning as $N_c(n,s)$ and $N_p(n,s)$ but for the restricted version of problem

Theorem

For $s \geq 4$ we have

$$N_c(n,s,t) \geq (s-4)2^n + 2^{n-t+2} = N_c(n,s) + 4(2^{n-t} - 1)$$

It is easy to see that for paths we have

$$N_p(n,s,t) \geq N_c(n,s,t) + 2t$$

For cases $t = 1, 2$ and $s = 3$ we give optimal strategies.

Theorem

(i) For $n \geq 2$

$$N_c(n, 3, 1) = 2^n$$

(ii) For $n \geq 4$ we have

$$N_c(n, 3, 2) = 2^{n-2}$$

Conclusion and Open problems

We considered a new model of combinatorial two-sided search

We described optimal search strategies for path graphs and cycles.

For arbitrary tree topologies we characterized the minimum possible size of a target set.

An interesting open question is the characterization of strategies of minimum length for trees
We also find it interesting to

- Consider the problem for other popular network topologies like grids, n-cubes etc
- Study of probabilistic models of two-sided search
- Consider two-sided search models in the case where some of test results are incorrect, and the related coding problems.

Thank you for your attention!