Black Hole Search in arbitrary networks

Matoula Petrolia

October 17, 2011

Introduction

- The **Black Hole Search** is the task of locating all black holes in a network by exploring it with mobile agents.
- Given the map of the network and the starting node we want to design the *fastest* Black Hole Search.

The Black Hole Search problem

- **Black hole**: a node containing a stationery process that destroys all mobile agents visiting the node, without leaving any trace.
- Agents try to identify the black hole and then avoid it.
- At most one black hole in the network.
- Exactly two agents starting from the same node.
- The network is *synchronous*, i.e. there is an upper bound on the time needed by an agent for traversing any edge (assume 1 unit).

Model and Terminology

- **Network**: a connected, undirected graph $G(V, E)$. No multiple edges. No self loops.
- $B \subsetneq V$: the set of black holes in G.
- $V \setminus B$: safe nodes in G.
- **Starting node** $s \in V \setminus B$ (the only one known to be safe).
- $|B| \leq 1$ (one or no black holes in G).
- Two agents (*Agent-1*,*Agent-2*).
- Agents communicate only when they are in the same node.
- Synchronous network.

Formalizing the problem

MINIMUM COST BHS PROBLEM or BHS

Instance: a connected undirected graph $G(V, E)$ and a node $s \in V$.

Solution: an **exploration scheme** $\mathcal{E}_{G,s} = (\mathbb{X}, \mathbb{Y})$ for G and s, $\mathbb{X} = \langle x_0, x_1, \ldots, x_T \rangle$, $\mathbb{Y} = \langle y_0, y_1, \ldots, y_T \rangle$ two equal-length sequences of nodes in G. The exploration scheme must satisfy some costraints given below. T is the length of $\mathcal{E}_{G,s}$.

Measure: the cost of the BHS based on $\mathcal{E}_{G,s}$.

- *Agent-1* and *Agent-2* follow the paths defined by $\mathbb{X}$ and $\mathbb{Y}$, respectively.

- *Agent-1* and *Agent-2* follow the paths defined by $\mathbb{X}$ and $\mathbb{Y}$, respectively.
- When an agent deduces the existence of a black hole and its exact location, aborts the exploration and returns to s traversing safe nodes.

- *Agent-1* and *Agent-2* follow the paths defined by $\mathbb{X}$ and $\mathbb{Y}$, respectively.
- When an agent deduces the existence of a black hole and its exact location, aborts the exploration and returns to s traversing safe nodes.
- Exploration is deterministic and the scheme is calculated before the exploration starts.

- *Agent-1* and *Agent-2* follow the paths defined by $\mathbb{X}$ and $\mathbb{Y}$, respectively.
- When an agent deduces the existence of a black hole and its exact location, aborts the exploration and returns to s traversing safe nodes.
- Exploration is deterministic and the scheme is calculated before the exploration starts.
- The agents must follow the sequences until one realizes that the other has died.

Some definitions

- Given an exploration scheme $\mathcal{E}_{G,s}$, for each $i = 0, 1, \ldots, T$, the set

$$S_i = \begin{cases} \bigcup_{j=0}^{i}\{x_j\} \cup \bigcup_{j=0}^{i}\{y_j\}, & \text{if } \; x_i = y_i \\ S_{i-1}, & \text{otherwise.} \end{cases}$$

  is called the **explored territory** at step i.
  A node v is **explored** at step i if $v \in S_i$ or **unexplored** otherwise.
  Note that $S_{j-1} \subseteq S_j$.

Some definitions

- Given an exploration scheme $\mathcal{E}_{G,s}$, for each $i = 0, 1, \ldots, T$, the set

$$S_i = \begin{cases} \bigcup_{j=0}^{i}\{x_j\} \cup \bigcup_{j=0}^{i}\{y_j\}, & \text{if } \ x_i = y_i \\ S_{i-1}, & \text{otherwise.} \end{cases}$$

  is called the **explored territory** at step i.
  A node v is **explored** at step i if $v \in S_i$ or **unexplored** otherwise.
  Note that $S_{j-1} \subseteq S_j$.
- **Meeting step** is every step $0 \leq j \leq T$ such that $S_j \neq S_{j-1}$.
- For each meeting step we have $x_i = y_i$ and that node is called a **meeting point**.
- A sequence of steps $\langle j+1, \ldots, k\rangle$ where j and k are two consecutive meetings is called a **phase** of length $k - j$.

Note that:

- A node may have been visited by an agent but becomes explored only when the agents meet.
- The explored territory is defined for an exploration scheme $\mathcal{E}_{G,s}$, not for the BHS based on it: it doesn't take into account the possible existence of a black hole.

The Constraints

$\mathcal{E}_{G,s} = (\mathbb{X}, \mathbb{Y})$ is a feasible exploration scheme for G and s if the following constraints are satisfied:

Constraint 1: $x_0 = y_0 = s$, $x_T = y_T$.

The Constraints

$\mathcal{E}_{G,s} = (\mathbb{X}, \mathbb{Y})$ is a feasible exploration scheme for G and s if the following constraints are satisfied:

Constraint 1: $x_0 = y_0 = s$, $x_T = y_T$.

Constraint 2: for each $i = 0, \ldots, T-1$, either $x_{i+1} = x_i$ or $(x_i, x_{i+1}) \in E$ and similarly for the y_i's.

The Constraints

$\mathcal{E}_{G,s} = (\mathbb{X}, \mathbb{Y})$ is a feasible exploration scheme for G and s if the following constraints are satisfied:

Constraint 1: $x_0 = y_0 = s$, $x_T = y_T$.

Constraint 2: for each $i = 0, \ldots, T-1$, either $x_{i+1} = x_i$ or $(x_i, x_{i+1}) \in E$ and similarly for the y_i's.

Constraint 3: $\bigcup_{i=0}^{T}\{x_i\} \cup \bigcup_{i=0}^{T}\{y_i\} = V$.

The Constraints

$\mathcal{E}_{G,s} = (\mathbb{X}, \mathbb{Y})$ is a feasible exploration scheme for G and s if the following constraints are satisfied:

Constraint 1: $x_0 = y_0 = s$, $x_T = y_T$.

Constraint 2: for each $i = 0, \ldots, T-1$, either $x_{i+1} = x_i$ or $(x_i, x_{i+1}) \in E$ and similarly for the y_i's.

Constraint 3: $\bigcup_{i=0}^{T}\{x_i\} \cup \bigcup_{i=0}^{T}\{y_i\} = V$.

Constraint 4: for each phase $\langle j+1, \ldots, k\rangle$,

(a) $|\{x_{j+1}, \ldots, x_k\} \setminus S_j| \leq 1$ and $|\{y_{j+1}, \ldots, y_k\} \setminus S_j| \leq 1$; and
(b) $\{x_{j+1}, \ldots, x_k\} \setminus S_j \neq \{y_{j+1}, \ldots, y_k\} \setminus S_j$.

In other words...

Constraint 1: both agents start from the same node s and end at the same node.
Constraint 2: during each step, an agent can either wait in the node v where it was, or move to a node adjacent to v.
Constraint 3: each node in V is visited at least once by at least one agent during the exploration.
Constraint 4: during each phase, an agent can visit at most one unexplored node (a) and the two agents cannot visit the same unexplored node during the same phase (b).

Also, observe that:

- $S_0 = \{s\}$, by Constraint 1, and
- $S_T = V$, by Constraints 1 and 3.

Lemma 1

If $k \geq 1$ is a meeting step for $\mathcal{E}_{G,s}$, then $x_k = y_k \in S_{k-1}$.

Proof:

If j is the last meeting step before k, then $S_j = S_{j+1} = \ldots = S_{k-1}$ and by definition, $x_k = y_k \in S_k$. Now, if $x_k = y_k \notin S_{k-1} = S_j$, then, it is in both $\{x_{j+1}, \ldots, x_k\} \setminus S_j$ and $\{y_{j+1}, \ldots, y_k\} \setminus S_j$ and Constraint 4 is violated. □

Lemma 2

Each phase of $\mathcal{E}_{G,s}$ has length at least two.

Proof:

If there were a phase of length 1, there would be two adjacent meeting steps $j, j+1$. But $j+1$ is a meeting iff $S_{j+1} \supsetneq S_j$ and, by Lemma 1, $x_{j+1} = y_{j+1} \in S_j$ and hence $S_{j+1} = S_j$. Contradiction. □

Phases of length 2

Consider a phase of length two, $\langle j+1, j+2\rangle$ at the end of which the explored territory icreases by 2 nodes.

If m is the meeting point at step j, *Agent-1* and *Agent-2* visit unexplored nodes adjacent to m, v_1, v_2 respectively, at step j+1. Then, at step $j+2$ the agents meet:

- in m (**b-split**(m, v_1, v_2)), or
- in a node $m' \neq m$, adjacent to v_1, v_2 (**a-split**(m, v_1, v_2, m')).

Execution time

Execution time:

- If $B = \emptyset$, the execution time is T plus the shortest path from $x_T = y_T$ to s.
- If $B = \{b\}$, execution time equals j plus the shortest path from $x_j = y_j$ to s not including b.
  (j is the first step such that $b \in S_j$.)

In the first case, both agents have to perform the full exploration and get back to s to report there is no black hole in G.

In the second case, one agent doesn't show up at meeting point $x_j = y_j$ so the other knows the exact location of b, as the agents visit only one unexplored node during a phase. Then the surviving agent returns to s.

Cost

Cost: The worst execution time of $\mathcal{E}_{G,s}$, over all possible values of B.

We allow a malicious adversary which exactly knows G and $\mathcal{E}_{G,s}$ to place b, or not place it.

Examples:

- For a tree, the case $B = \emptyset$ gives the maximum execution time.
- For a n-node ring graph, if $B = \emptyset$, then the execution time is $3n + \mathcal{O}(1)$. But if v_{n-1} is the black hole, then execution time is $4n + \mathcal{O}(1)$.

NP-hardness of Black Hole Search

To prove the NP-hardness of BHS, we provide a reduction from an NP-complete problem to the decision version of BHS.

Hamiltonian cycle problem for cubic planar graphs (cpHP)

Instance: a cubic planar 2-edge connected graph $G(V, E)$ and an edge $(x, y) \in E$.
Question: does G contain a Hamiltonian cycle that includes edge (x, y)?

The cpHP problem without the requirement that the Hamiltonian path passes through a given edge is NP-complete and so this one is also NP-complete because of a simple reduction.

Decision Black Hole Search problem for planar graphs (dBHS)

Instance: a planar graph $G' = (V', E')$, a starting node $s \in V'$ and a positive integer X.
Question: does there exist an exploration scheme $\mathcal{E}_{G',s}$ for G' starting from s, such that the BHS based on $\mathcal{E}_{G',s}$ has cost at most X?

Reduction from cpHC to dBHS

$G = (V, E)$ and (x, y) an instance for cpHC.
We construct the corresponding instance of the dBHS problem.

1. Replace edge (x, y) in G with edges (x, s), (s, y), s a new node. Get graph $\overline{G}$.

Reduction from cpHC to dBHS

$G = (V, E)$ and (x, y) an instance for cpHC.
We construct the corresponding instance of the dBHS problem.

1. Replace edge (x, y) in G with edges (x, s), (s, y), s a new node. Get graph $\overline{G}$.
2. $\mathcal{F}$ the set of the faces of $\overline{G}$. Identify each $f \in \mathcal{F}$ with the sequence of the consecutive edges adjacent to f.

Reduction from cpHC to dBHS

$G = (V, E)$ and (x, y) an instance for cpHC.
We construct the corresponding instance of the dBHS problem.

1. Replace edge (x, y) in G with edges (x, s), (s, y), s a new node. Get graph $\overline{G}$.
2. $\mathcal{F}$ the set of the faces of $\overline{G}$. Identify each $f \in \mathcal{F}$ with the sequence of the consecutive edges adjacent to f.
3. For each f and each edge (v, w) adjacent to f, add a new node $z_f^{(v,w)}$ and two edges $(v, z_f^{(v,w)})$, $(w, z_f^{(v,w)})$.

Reduction from cpHC to dBHS

$G = (V, E)$ and (x, y) an instance for cpHC.
We construct the corresponding instance of the dBHS problem.

1. Replace edge (x, y) in G with edges (x, s), (s, y), s a new node. Get graph $\overline{G}$.
2. $\mathcal{F}$ the set of the faces of $\overline{G}$. Identify each $f \in \mathcal{F}$ with the sequence of the consecutive edges adjacent to f.
3. For each f and each edge (v, w) adjacent to f, add a new node $z_f^{(v,w)}$ and two edges $(v, z_f^{(v,w)})$, $(w, z_f^{(v,w)})$.
4. For each $f = \langle e_1, e_2, \ldots, e_3 \rangle \in \mathcal{F}$, add the shortcut edges $(z_f^{e_1}, z_f^{e_2}), (z_f^{e_2}, z_f^{e_3}), \ldots, (z_f^{e_q}, z_f^{e_1})$.

Reduction from cpHC to dBHS

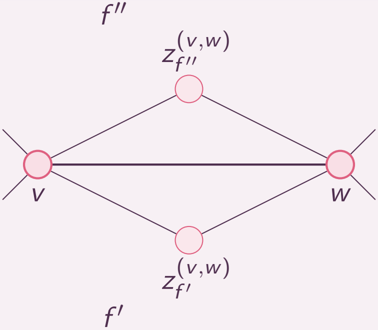
$G = (V, E)$ and (x, y) an instance for cpHC.
We construct the corresponding instance of the dBHS problem.

1. Replace edge (x, y) in G with edges (x, s), (s, y), s a new node. Get graph $\overline{G}$.
2. $\mathcal{F}$ the set of the faces of $\overline{G}$. Identify each $f \in \mathcal{F}$ with the sequence of the consecutive edges adjacent to f.
3. For each f and each edge (v, w) adjacent to f, add a new node $z_f^{(v,w)}$ and two edges $(v, z_f^{(v,w)})$, $(w, z_f^{(v,w)})$.
4. For each $f = \langle e_1, e_2, \ldots, e_3 \rangle \in \mathcal{F}$, add the shortcut edges $(z_f^{e_1}, z_f^{e_2}), (z_f^{e_2}, z_f^{e_3}), \ldots, (z_f^{e_q}, z_f^{e_1})$.
5. For each $v \in V \cup \{s\} \setminus \{x\}$ add a new node v^F (**flag node**) and edge (v, v^F).

Reduction from cpHC to dBHS

$G = (V, E)$ and (x, y) an instance for cpHC.
We construct the corresponding instance of the dBHS problem.

1. Replace edge (x, y) in G with edges (x, s), (s, y), s a new node. Get graph $\overline{G}$.
2. $\mathcal{F}$ the set of the faces of $\overline{G}$. Identify each $f \in \mathcal{F}$ with the sequence of the consecutive edges adjacent to f.
3. For each f and each edge (v, w) adjacent to f, add a new node $z_f^{(v,w)}$ and two edges $(v, z_f^{(v,w)})$, $(w, z_f^{(v,w)})$.
4. For each $f = \langle e_1, e_2, \ldots, e_3 \rangle \in \mathcal{F}$, add the shortcut edges $(z_f^{e_1}, z_f^{e_2}), (z_f^{e_2}, z_f^{e_3}), \ldots, (z_f^{e_q}, z_f^{e_1})$.
5. For each $v \in V \cup \{s\} \setminus \{x\}$ add a new node v^F (**flag node**) and edge (v, v^F).
6. Obtain G'. Set $X = n' - 1 = 5n + 2$, where $n' = n + 1 + 2(e + 1) + n = 5n + 3$ the number of nodes in G', n, e nodes and edges in G respectively ($e = \frac{3}{2}n$).



Each edge e in $\overline{G}$ is adjecent to exactly two faces f', f''. The nodes $z_{f'}^{e}$, $z_{f''}^{e}$ added are called **twin nodes** for e.

G' is planar and can be constructed in linear time.

The nodes in G' inherited from $\overline{G}$ are called **original nodes**.

In Lemma 3 a useful property of G' is stated.

Lemma 3

Let $\langle u, v, w\rangle$ be a path in graph $\overline{G}$. Then there is a path $\langle u, z', z'', w\rangle$ in G' bypassing node v (i.e. $v \notin \{z, z'\}$).

Proof:

The degree of each node in $\overline{G}$ is at most 3 (as $\overline{G}$ was obtained from G by replacing an edge and adding a node and G is a cubic graph). So, there must be a face $f \in \mathcal{F}$ to which both edges (u, v) and (v, w) are adjacent. By the construction og G' the sequence $\langle u, z_f^{(u,v)}, z_f^{(v,w)}, w\rangle$ is a path in G'. $\square$

The following Lemmas prove that graph G has a Hamiltonian cycle passing through edge (x, y) iff there there is an exploration scheme for G' and the starting node s with cost at most $X = 5n + 2$.

Lemma 4

If G has a Hamiltonian cycle that includes (x, y), then there exists an exploration scheme $\mathcal{E}^*_{G',s}$ on G' from s such that the BHS based on it has cost at most $5n + 2$.

Lemma 5

If there exists an exploration scheme $\mathcal{E}_{G',s}$ on G' starting from s such that the cost of BHS based on it is at most $5n + 2$, then G has a Hamiltonian cycle that includes edge (x, y).

Proof of Lemma 4 (1)

Let $\{v_1 = y, e_1, v_2, \ldots, e_{n-1}, v_n = x, e_n, v_1 = y\}$ be a Hamiltonian cycle in G that includes (x, y). Consider the following exploration scheme $\mathcal{E}^*_{G',s}$:

1. **b-split**(s, s^F, y).

Proof of Lemma 4 (1)

Let $\{v_1 = y, e_1, v_2, \ldots, e_{n-1}, v_n = x, e_n, v_1 = y\}$ be a Hamiltonian cycle in G that includes (x, y). Consider the following exploration scheme $\mathcal{E}^*_{G',s}$:

1. **b-split**(s, s^F, y).
2. **a-split**(s, z_1, z_2, y), z_1, z_2 the twin nodes of (s, y).

Proof of Lemma 4 (1)

Let $\{v_1 = y, e_1, v_2, \ldots, e_{n-1}, v_n = x, e_n, v_1 = y\}$ be a Hamiltonian cycle in G that includes (x, y). Consider the following exploration scheme $\mathcal{E}^*_{G',s}$:

1. **b-split**(s, s^F, y).
2. **a-split**(s, z_1, z_2, y), z_1, z_2 the twin nodes of (s, y).
3. for each v_i of the Hamiltonian cycle $(i = 1, \ldots, n-1)$:
    3.1 let v_j be the third neighbor of v_i (other than v_{i-1}, v_{i+1}); if $j > i$ then **b-split**(v_i, z_1, z_2), z_1, z_2 the twin nodes of (v_i, v_j).

Proof of Lemma 4 (1)

Let $\{v_1 = y, e_1, v_2, \ldots, e_{n-1}, v_n = x, e_n, v_1 = y\}$ be a Hamiltonian cycle in G that includes (x, y). Consider the following exploration scheme $\mathcal{E}^*_{G',s}$:

1. **b-split**(s, s^F, y).
2. **a-split**(s, z_1, z_2, y), z_1, z_2 the twin nodes of (s, y).
3. for each v_i of the Hamiltonian cycle ($i = 1, \ldots, n-1$):
    3.1 let v_j be the third neighbor of v_i (other than v_{i-1}, v_{i+1}); if $j > i$ then **b-split**(v_i, z_1, z_2), z_1, z_2 the twin nodes of (v_i, v_j).
    3.2 **b-split**(v_i, v_i^F, v_{i+1}).

Proof of Lemma 4 (1)

Let $\{v_1 = y, e_1, v_2, \ldots, e_{n-1}, v_n = x, e_n, v_1 = y\}$ be a Hamiltonian cycle in G that includes (x, y). Consider the following exploration scheme $\mathcal{E}^*_{G',s}$:

1. **b-split**(s, s^F, y).
2. **a-split**(s, z_1, z_2, y), z_1, z_2 the twin nodes of (s, y).
3. for each v_i of the Hamiltonian cycle $(i = 1, \ldots, n-1)$:
   - 3.1 let v_j be the third neighbor of v_i (other than v_{i-1}, v_{i+1}); if $j > i$ then **b-split**(v_i, z_1, z_2), z_1, z_2 the twin nodes of (v_i, v_j).
   - 3.2 **b-split**(v_i, v_i^F, v_{i+1}).
   - 3.3 **a-split**(v_i, z_1, z_2, v_{i+1}), z_1, z_2 the twin nodes of (v_i, v_{i+1}).

Proof of Lemma 4 (1)

Let $\{v_1 = y, e_1, v_2, \ldots, e_{n-1}, v_n = x, e_n, v_1 = y\}$ be a Hamiltonian cycle in G that includes (x, y). Consider the following exploration scheme $\mathcal{E}^*_{G',s}$:

1. **b-split**(s, s^F, y).
2. **a-split**(s, z_1, z_2, y), z_1, z_2 the twin nodes of (s, y).
3. for each v_i of the Hamiltonian cycle $(i = 1, \ldots, n-1)$:
   3.1 let v_j be the third neighbor of v_i (other than v_{i-1}, v_{i+1}); if $j > i$ then **b-split**(v_i, z_1, z_2), z_1, z_2 the twin nodes of (v_i, v_j).
   3.2 **b-split**(v_i, v_i^F, v_{i+1}).
   3.3 **a-split**(v_i, z_1, z_2, v_{i+1}), z_1, z_2 the twin nodes of (v_i, v_{i+1}).
4. **a-split**(x, z_1, z_2, s), z_1, z_2 the twin nodes of (x, s).

Proof of Lemma 4 (2)

Length of $\mathcal{E}^*_{G',s}$:

a-split and b-split phases have length 2 and increase the explored territory by 2 nodes, thus the number of phases is $(5n+2)/2$ and so $\mathcal{E}^*_{G',s}$ has length $5n+2$. This number is also the exploration time for $\mathcal{E}^*_{G',s}$ when $B = \emptyset$, ($\mathcal{E}^*_{G',s}$ ends in s).

This is also the cost of the BHS, i.e. there is no allocation of the black hole that yields a larger exploration time.
Observe that the set of the meeting points in $\mathcal{E}^*_{G',s}$ is $\{v_i : 1 \leq i \leq n\} \cup \{s\}$.

Proof of Lemma 4 (3)

Claim: Consider the meeting step when the agents are to meet at a node v_i, $(1 \leq i \leq n)$. If a black hole has just been discovered, then the remaining exploration time for this case is not greater than the remaining exploration time for the case $B = \emptyset$.

Proof:

- If b is the flag node v_i^F or one of the twin edges of (v_{i-1}, v_i) or (v_i, v_j), then the surviving agent can reach s by following the remaining part of the Hamiltonian cycle. Remaining cost at most: $n + 1 - i$.
- If b is node v_{i+1}, then there is a path of length 4 in G' from v_i to v_{i+2} bypassing v_{i+1} (Lemma 3). The surviving agent can use this safe path and then follow the remaining part of the Hamiltonian cycle. Remaining cost at most: $n + 2 - i$.
- If $B = \emptyset$, remaining cost at least $2(n + 1 - i)$.

Proof of Lemma 5 (1)

Each phase of $\mathcal{E}_{G',s}$ has length at least two (Lemma 2) and cannot explore more than two unexplored nodes. G' has $5n+2$ unexplored nodes so $\mathcal{E}_{G',s}$ must end in s, and each of its phases must be either an a-split or a b-split.

Let $M_{\mathcal{E}}$ be the sequence of the meeting points for $\mathcal{E}_{G',s}$ at the end of each a-split, excluding s.
Each meeting point $v_i \in M_{\mathcal{E}}$ must have degree at least 5:

- one neighbor for the initial exploration of v_i,
- two unexplored neighbors for the a-split that ends in v_i,
- two unexplored neighbors for the a-split that leaves v_i.

Thus, only the original nodes of G' can be in $M_{\mathcal{E}}$, since flag nodes have 1 neighbor and twin nodes have 4.

Proof of Lemma 5 (2)

Claim: The nodes x and y must be the two endpoints of $M_{\mathcal{E}}$, node s cannot be in $M_{\mathcal{E}}$ and each node v in G must be in $M_{\mathcal{E}}$.

Proof:

s is the only initially safe node so the first phase must be a b-split from s. The first a-split in $\mathcal{E}_{G',s}$ is from s to x or y and the last one starts from y or x and ends in s.

Now, if s is an indermediate meeting point, then we need another a-split to s. But each of these (four) phases require two unexplored neighbors, thus s must have degree 8. Contradiction, as s has degree 7.

Finally, for each v in G, its flag node v^F has to be explored with a b-split having as meeting point v. Thus v must be in $M_{\mathcal{E}}$. $\square$

Proof of Lemma 5 (3)

$M_{\mathcal{E}}$ **defines a Hamiltonian cycle on** G:

(a) each node of G appears at most once in $M_{\mathcal{E}}$;

(b) if v_i, v_j are consecutive in $M_{\mathcal{E}}$, then (v_i, v_j) must be in G.

Proof of (a):

A node $v_i \in M_{\mathcal{E}}$ needs:

- at least one neighbor for its initial exploration,
- two unexplored neighbors, for each occurence of v_i in $M_{\mathcal{E}}$, for the a-split that ends in it,
- two more unexplored neighbors for the a-split that leaves v_i.
- one more neighbor for the exploration of its flag node (b-split).

If v_i occurs k times in $M_{\mathcal{E}}$ then it must have at least $1 + 4k + 2 = 3 + 4k$ neighbors. But each original node in G' has only 10 neighbors so $k \leq 1$. $\square$

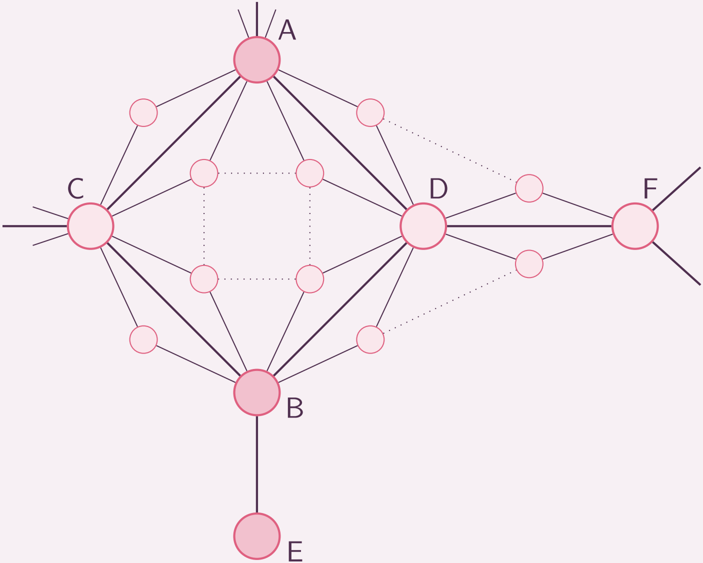
Proof of Lemma 5 (4)

Proof of (b):

According to the structure of G', a-splits with original nodes as meeting points can either:

- explore two twin nodes of an original node, or
- explore two original nodes (big a-split).

$\square$


A
C
D
F
B
E

An approximation algorithm for the BHS problem in arbitrary networks

The idea: Find a spanning tree in G and explore the graph by traversing its edges.
(e.g. Both agents traverse the tree together in depth-first order. One explores a new node while p the other waits in the parent of p.)
This approach guarantees an approximation ratio of 4:

- any exploration of an n-node needs at least $n-1$ steps,
- exploring an n-node tree with the above needs $(4(n-1)-2l)$ steps, l the number of leaves.

For this approach we need:

1. A good exploration scheme for trees.
2. An algorithm for constructing a "good" spanning tree of G.

STE (*Spanning-Tree Exploration*) algorithm returns, for a given graph G and a starting node s, the exploration scheme for the tree computed by algorithm *Generate-Tree(G,s)*.

STE algorithm guarantees an approximation ratio of at most $3\frac{3}{8}$.

Exploration schemes for trees

Let T be an n-node tree rooted at s, $n \geq 2$.

Idea for the scheme:

- p internal node with x children
- two groups: $\lfloor x/2 \rfloor$, $\lceil x/2 \rceil$
- agents follow the depth-first traversal of internal nodes
- when *Agent-1* comes to a new node, it visits all its children in group 1
- *Agent-2* visits the children in group 2.

Types of nodes

We classify all nodes of T except s:

- *type-1* nodes: the leaves;
- *type-3* nodes: the internal nodes with at least one sibling;
- *type-4* nodes: the internal nodes without siblings.

x_t: the number of *type-t* nodes.

Lemma

T a tree rooted at s with $n \geq 2$. The exploration scheme $\mathcal{E}_T = (\mathbb{X}_T, \mathbb{Y}_T)$ for T is feasible, can be constructed in linear time and its cost is at most

$$x_1 + 3x_3 + 4x_4 + 1.$$

The heuristic algorithm *Generate-Tree(G,s)* computes a spanning tree T_G of a given graph $G = (V, E)$, which tries to achieve a relatively small value of $x_1 + 3x_3 + 4x_4 + 1$.

To achieve that, the algorithm tries to avoid creating *type-4* nodes.

Thank You!!!

$\ddot\smile$