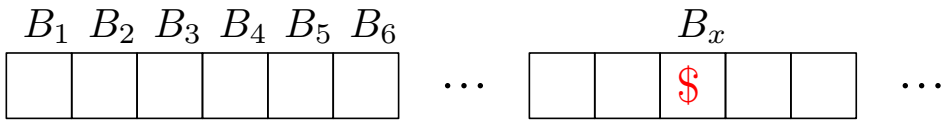
Treasure Hunt with Barely Communicating Deterministic Agents

Stefan Dobrev, Rastislav Královič, Dana Pardubská

OPODIS 2017

Treasure Hunt



- a linear (potentially infinite) sequence of boxes $B_1, B_2, \ldots$
- one of them (B_x) contains a treasure, x is unknown
- k agents search in parallel for the treasure by opening boxes
  - direct access to any box
- search is completed when the treasure is found
- synchronous system, discrete time steps

Goal: minimize the time to find treasure, as a function of x and k

- k is a fixed constant, x goes to infinity
- trivial if there are no failures: $\lceil \frac{x}{k} \rceil$

Treasure Hunt with Failures

Introduced in Freigniaud et.al in [STOC 2016]:

- up to f agents can fail, $f \leq k - 1$
- agents are anonymous, randomized and without any communication

Goal: minimize **expected** search time

Result:

- a simple randomized algorithm with expected search time $\frac{4x}{k} + o(\frac{x}{k})$
  - the penalty paid for non-coordination is only a factor of 4

Questions that popped in our heads (and resulted in this paper):

Question 1:

- **What is the minimum coordination needed to achieve expected search time $\frac{x}{k} + o(\frac{x}{k})$?**

Question 2:

- **Do we really need randomization?**

Question 3:

- **What happens if the failures are due to black holes?**
  - up to f boxes contain black holes
  - if a agent opens a box with a black hole, it fails immediately
    - the other agents are not notified of its failure
    - if the algorithm is not careful, many agents can fail
  - no other agents fail

Motivation – Treasure Hunt

- yet another take on search
- huge search spaces need
  - parallel/distributed approach
  - fault-tolerance
- can be seen as an abstraction of exhaustive search done on a distributed platform, e.g. BOINC
  - Berkeley Open Infrastructure for Network Computing
    - made famous for SETI@home
- boxes correspond to jobs
- treasure is the SETI signal, cryptographic key, ...
- linear order corresponds to
  - likelyhood of success
  - payout (preference of the solution)
  - inverse of cost

Motivation – Weak Coordination

- tight coordination is costly when needed on huge scale
- adds another failure-prone layer
- environmental communication channels (e.g. whiteboards) can be difficult to add to legacy systems
- direct inter-agent communication might be undesirable from the privacy and security point of view
  - an ideal setting is a very limited communication facilitated via the central job server

Motivation – Weak Coordination

- tight coordination is costly when needed on huge scale
- adds another failure-prone layer
- environmental communication channels (e.g. whiteboards) can be difficult to add to legacy systems
- direct inter-agent communication might be undesirable from the privacy and security point of view
  - an ideal setting is a very limited communication facilitated via the central job server

Still, there is a basic coordination mechanism which is always present, by the very nature of the setting: For each work task, the server knows whether it has been already solved or not.

Motivation – Weak Coordination

- tight coordination is costly when needed on huge scale
- adds another failure-prone layer
- environmental communication channels (e.g. whiteboards) can be difficult to add to legacy systems
- direct inter-agent communication might be undesirable from the privacy and security point of view
  - an ideal setting is a very limited communication facilitated via the central job server

Still, there is a basic coordination mechanism which is always present, by the very nature of the setting: For each work task, the server knows whether it has been already solved or not.

In this paper we investigate how can this mechanism be used to efficiently solve the Treasure Hunt problem.

Barely Communicating Deterministic Agents

Barely Communicating Agents

- no direct communication between agents
- when an agent opens a box for the first time, the box changes state from "closed" to "opened" and remains "opened" afterwards
- opening the box returns its content (empty, treasure) and its previous state (closed or opened)

Deterministic Agents

- the agents need unique IDs
  - for simplicity of presentation, we assume the IDs are from 1 to k
  - a preprocessing phase of cost $O(f \log k \log \mathrm{MAXID})$ using standard parallel processing techniques can compact the IDs
- hence, we assume k agents $A_1, A_2, \ldots A_k$
- non-conflicting schedules (no two agents open the same box at any given time)

Failing Agents Model

- *Failure Pattern* $\mathcal{F}$: the identities of the failing agents, and the times of their failures
- $OPT_{\mathcal{F}}(k, x)$: the hunt time of the optimal k-agent algorithm, when treasure is in box x, and executing under the failure pattern $\mathcal{F}$
  - optimal algorithm: all agents are immediately notified of any failures
- $T_{\mathcal{F}}(k, x)$: the hunt time (same setting) of the algorithm under consideration

Theorem

There exists a deterministic k-agent treasure hunt algorithm with

$$T_{\mathcal{F}}(k, x) \leq OPT_{\mathcal{F}}(k, x) + O(k^3\sqrt{x})$$

Black Holes Model – Non-communicating Agents

Theorem

Consider a system with k non-communicating agents with a single black hole. There is no algorithm that always successfully finds the treasure.

Having a finite sequence of boxes changes things significantly:

Theorem

If $n \geq f^4$, $O(\sqrt{n \ln n})$ non-communicating agents can perform treasure hunt in the presence of f black holes in a sequence of n boxes (in time $\tilde{O}(n^{\frac{3}{4}})$).

Black Holes Model – Barely Communicating Agents I

Lower Bound:

Theorem

$2f + 1$ agents are needed to perform treasure hunt in the presence of f black holes

Upper Bound:

Theorem

$4f + 1$ agents are sufficient to perform treasure hunt in the presence of f black holes

- however, the speed is $O(1/f^2)$
- if treasure is in B_x, boxes up to B_{xf^2} are opened
  - not suitable for finite sequences

Black Holes Model – Barely Communicating Agents II

Finite Sequences:

Theorem

$O(f^2 \ln^2 f)$ agents can locate the treasure in the presence of f black holes, even if the number of boxes is limited, provided that it is at least $x + O(f^2 \ln^2 f)$, where x is the location of the treasure.

Fast Hunt:

Theorem

In the presence of one black hole, $k > 2$ and even, k agents can locate the treasure in time $T(k, x) \leq x/(k-2) + O(k + \sqrt{x}) \leq OPT(k-2, x) + O(k + \sqrt{x})$.

Related Work I

- huge literature on all kinds of variants of search, with some problems/models matching some aspects of our setting
- cow path problem – linear search space, competitive ratio as a function of the treasure(gate) distance
  - but has to pay for accessing location
- ANTS with one pheremone – destructive read
  - ants foraging in 2D space searching for food
  - no inter-ant communication, just marking the visited cell by pheremone(s)
  - investigating the speedup, impact of ant's computational power, failures
- do-all problem
  - p processors must perform n tasks such that each task is done by at least one processor

Related Work II

- Black Hole search – Black Holes destroying agents
  - paying for movements vs direct access to boxes
  - need to locate all black holes vs stop when the treasure is found
- TreasureHunt of Freigniaud, Korman, Rodeh – the closest match
  - probablistic vs our deterministic
  - no communication vs barely communicating agents
  - they studied also the case of the treasure placed uniformly at random in a finite space, and showed speedup of $k/3+o(1)$
  - also known non-uniform distribution of treasure placement recently investigated [Korman,Rodeh: SIROCCO 2017]

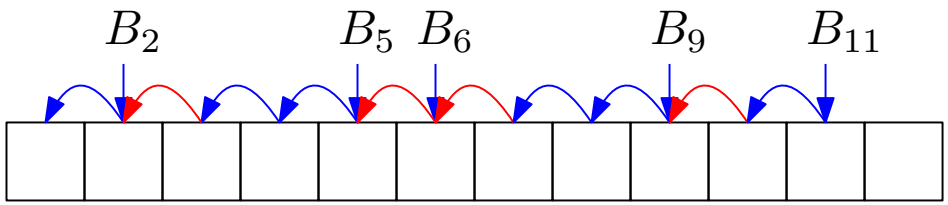
Communication via Scratchpads

- how to pass information from A_i to A_j?
  - agree on an unopened box B_z
  - A_i opens B_z if it wants to inform A_j
  - later, A_j opens B_z, if it finds it already opened, it knows that A_i wanted to inform it
- the details of the transmission can differ, the basic idea is that for each such transmission, we need a new unopened box

Scratchpad:

- an agreed-upon set of unopened boxes
- can't be reused, always use a new set
  - allocate a new scratchpad

Leader Election



- allocate a scratchpad S of size k
  - A_i opens S_i, then keeps opening to the left until S_1 is reached or an opened box is found
  - the agent opening S_1 as first is the leader
- if there are no failures, elects a leader in k steps
- if there are failures, the leader might be dead
  - but we never have two or more leaders

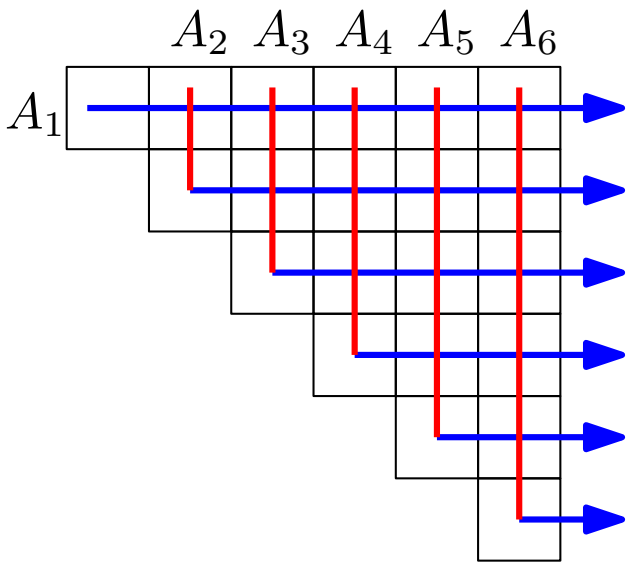
Basic Coordination

- some subset of agents is alive
- one agent is a designated leader (w.l.o.g. A_1, otherwise rotate)
- if the leader is dead at the beginning of the protocol, the output should be **false**
- if the leader is alive at the beginning of the protocol and there were no failures, the output should be **true**
- all surviving agents should return the same value

Theorem

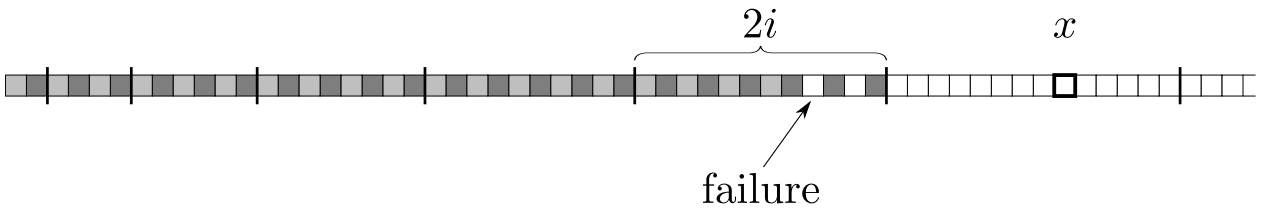
The coordination problem can be solved in $2k-2$ steps, provided that the agents agree on a set of $k(k-2)/2$ unopened boxes.

Basic Coordination II



- use a matrix $M[i, j]$ of unopened boxes to communicate information about leader's state
- $M[i, j]$ being open means A_i tells A_j that it knows that A_1 has been alive

Failing Agents – 2 Agents



- the algorithm proceeds in phases, in phase i, a block of $2i$ boxes is cleared
  - one agent takes care of odd boxes, the other of even boxes
- at the end of the phase, each agent checks whether the other agent has opened its last block
- if not, it opens the boxes that the other agent should have opened in this phase and then proceeds by opening boxes sequentially

Failing Agents – 2 Agents

- if there is no failure during the current phase, the extra delay is one step for the check at the end of the phase
- if there is a failure during the current phase, the extra delay is equal to the block size
- the number of phases is $O(\sqrt{\min(x, f)})$, where f is the location of the agent's failure
  - hence the overall delay is $O(\sqrt{\min(x, f)})$

Failing Agents – k Agents

- the same general idea
- just the checking and future work re-distribution at the end of the phases is more complicated
- in each phase, the agents maintain the set of alive agents
- the agreement protocol (invoked for each agent of this set) is used to verify whether all of them survived
- if a failure of some agent is detected, all agents re-do the work of the current phase, and the set of alive agents is re-computed

Impossibility for Non-communicating Agents

- signature τ_x of a box x: $[t_{x,1}, \ldots, t_{x,k}]$
  - $t_{x,k}$ is the time an agent A_k visits x (∞ if never) in a fault-free execution
- **pattern** of a signature: the set of positions with ∞ in them

Key Observation: If τ_i and τ_j have the same signature and $\tau_i < \tau_j$, then placing the black hole in B_i ensures that B_j is never opened.

- there are 2^k patterns, so there must be a pattern with infinitely many signatures with this pattern
- let τ_b be a signature of such pattern; there can be only finitely many signatures τ_i of this pattern such that $\tau_b \not< \tau_i$
  - each finite number occurs in a given position at most once among all signatures

Lower Bound of $2f + 1$ agents

For any algorithm $\mathcal{A}$, we construct a configuration in which each BH kills at least two agents

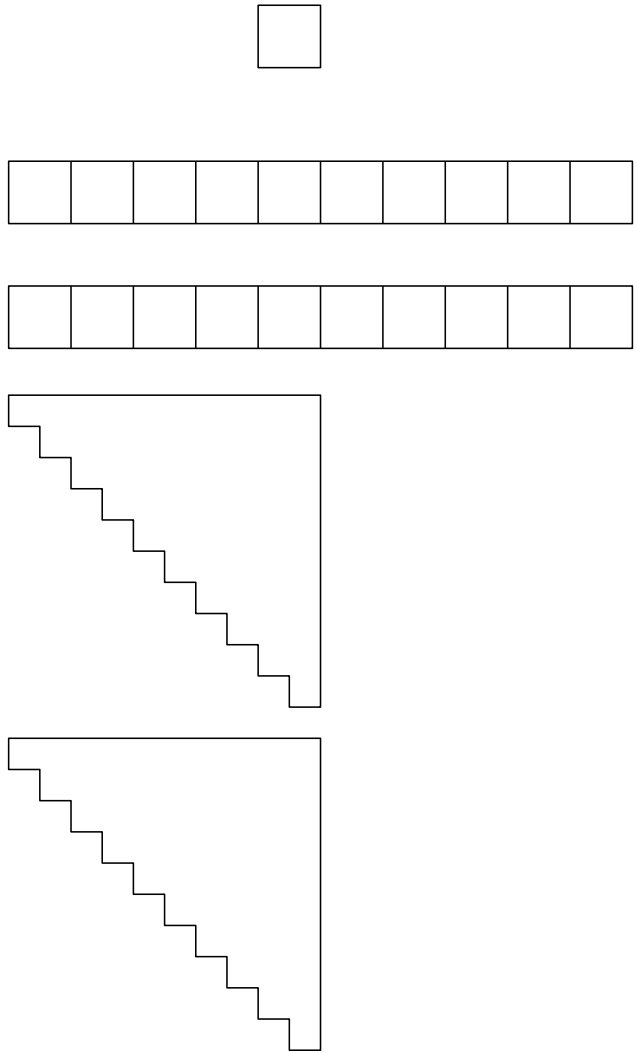
- consider computation C_0 of $\mathcal{A}$ without BHs, there must be a box checked by at least two agents
  - if there is no such box, then there must be two boxes B_i and B_j checked by the same single agent
  - w.l.o.g. B_i is checked first; place BH in B_i, then B_j will never be checked
- let t_1 be the first time an already opened box x_1 is opened
  - construct C_1 by placing the BH in x_1 (this kills two agents)
- let S_1 be the set of unopened boxes at time t_1
  - apply the same argument starting from time t_1 for boxes in S_1
  - in this way, each black hole can kill two agents

Upper Bound of $4f + 1$ agents

- works in phases; phase i ensures that B_i has been opened
- basic scheme:
  - elect a leader
  - the leader opens B_i
  - run coordination to agree whether the leader survived
  - if the answer is yes, proceed to the next phase
- the problem: what if B_i contains a BH?
  - we want to proceed to the next phase in such case
  - but must be sure that if the leader is dead, it died in B_i and not before opening it
    - otherwise B_i might remain unopened

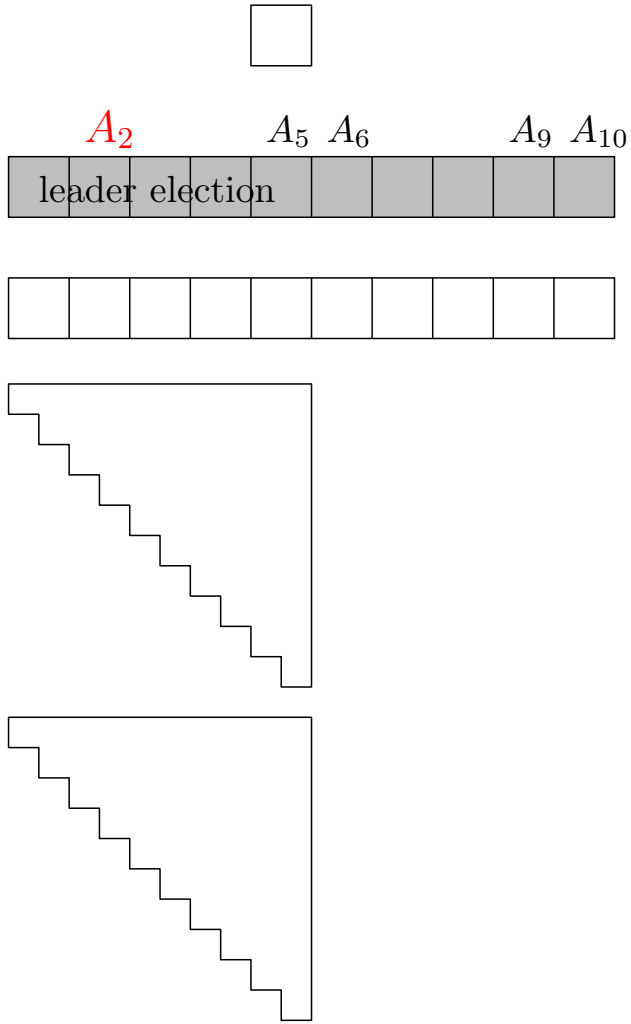
Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



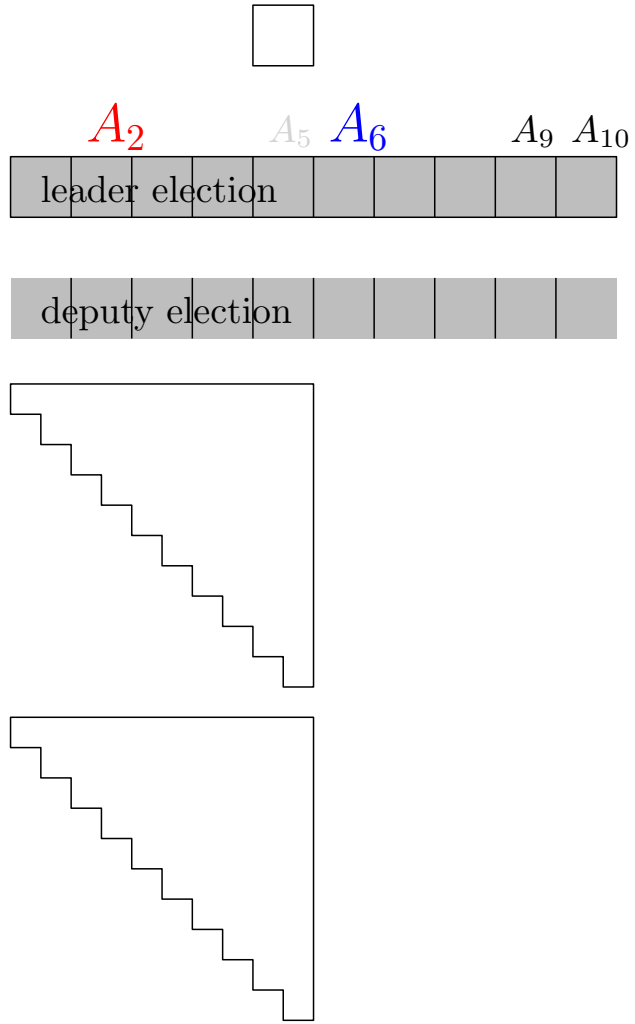
Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



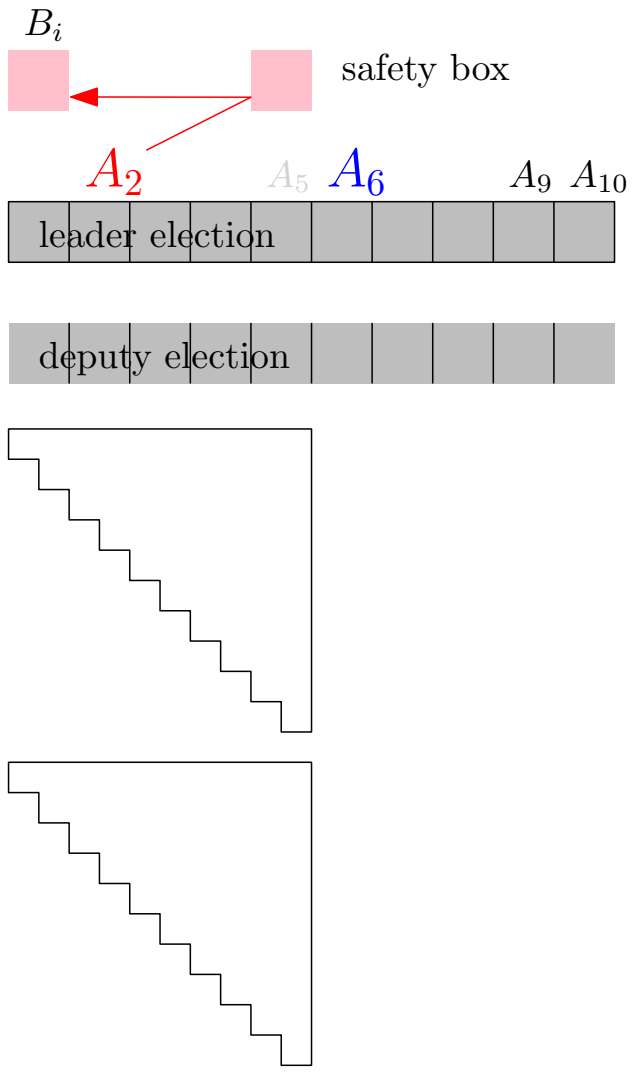
Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



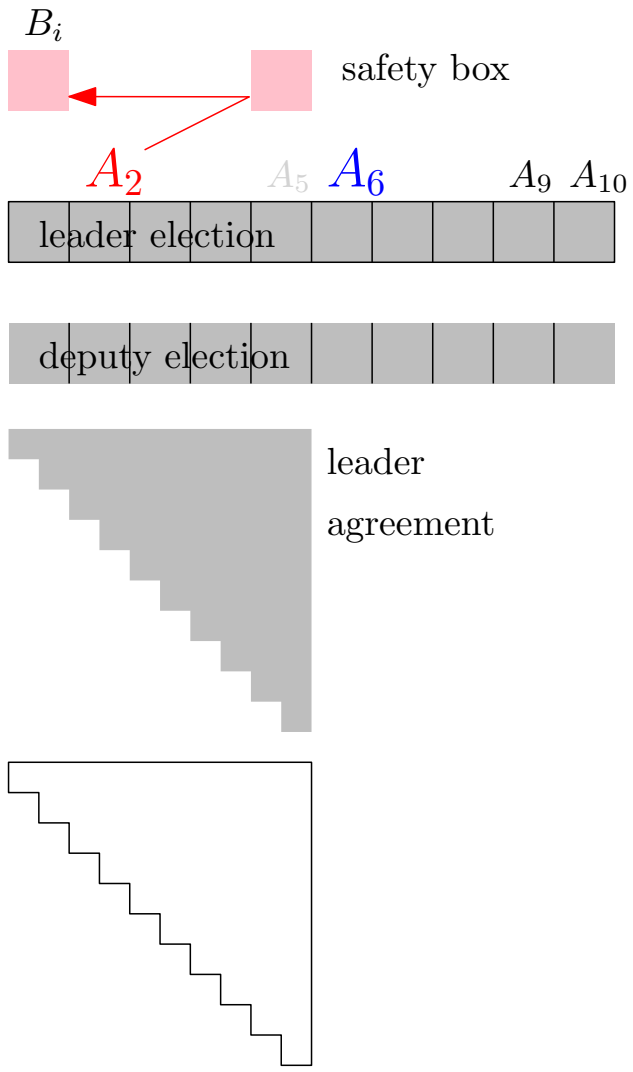
Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



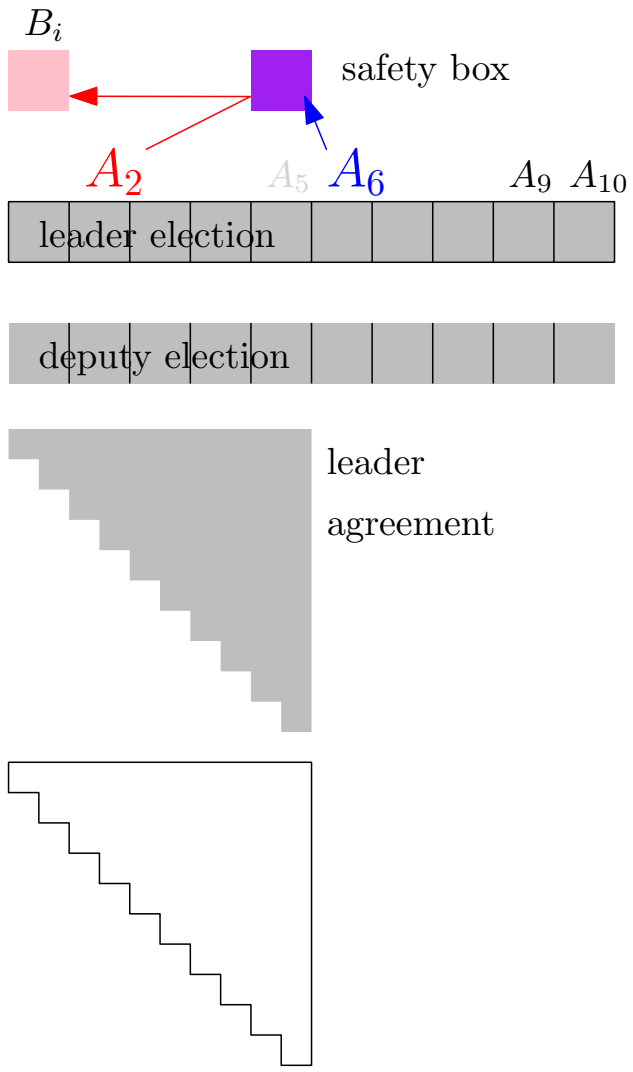
Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



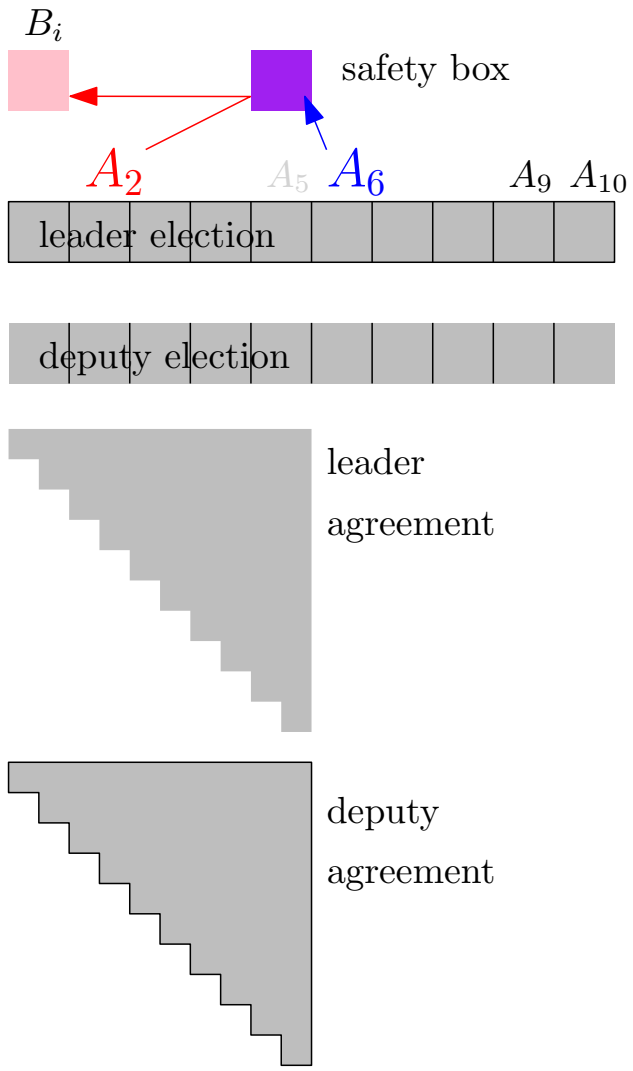
Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



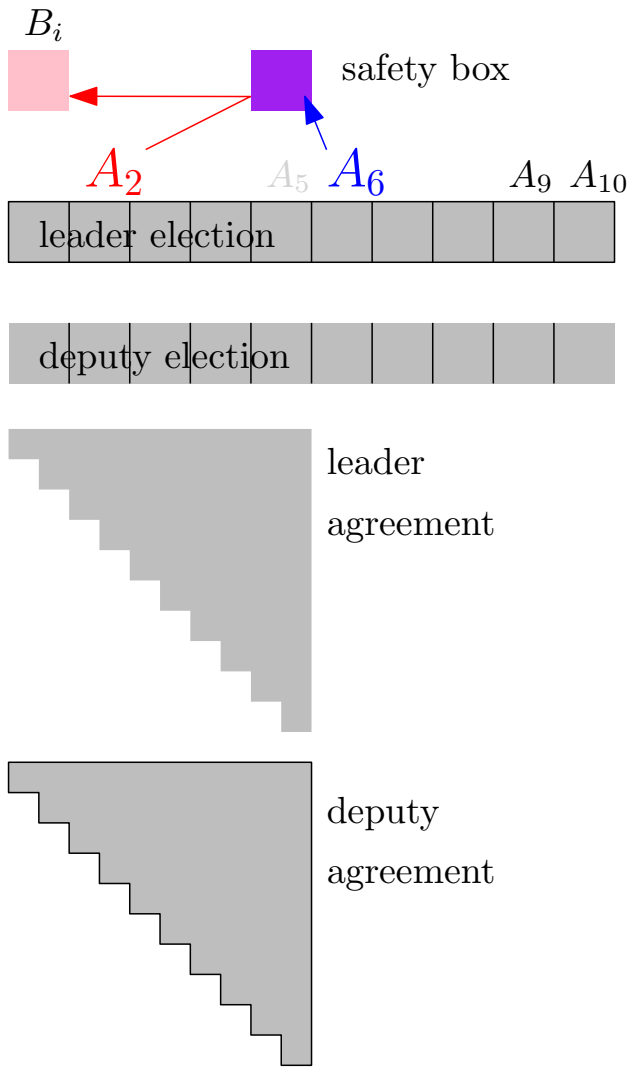
Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



Upper Bound of $4f + 1$ agents, take II

- allocate a scratchpad of size $k^2 + k + 1$
- elect a leader
- among the remaining agents, elect a deputy
- the leader opens the safety box, then opens B_i
- run coordination to agree whether the leader survived; if yes, proceed to the next phase
- otherwise the deputy checks the safety box
- run coordination to agree whether the deputy has seen an opened safety box and survived
- if yes, proceed to the next phase
- otherwise repeat the phase
  - happens only if there were failures during phase



Upper Bound of $4f + 1$ agents – Why it works?

Correctness:

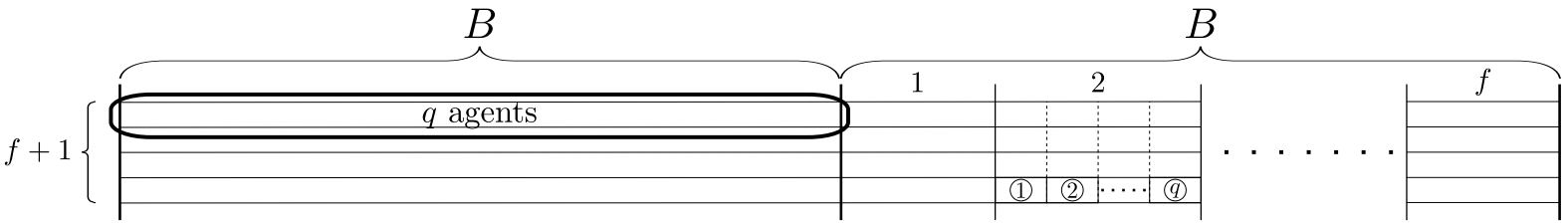
- if the leader survives, B_i has been opened
- if the deputy survived and has seen opened safety box, then the leader has opened B_i

The number of agents

- each scratchpad box is visited by at most 2 agents during coordination/leader election
  - each BH can get credit at most 2 for killing in a scratchpad
- a BH in B_i gets another credit for killing the leader in round i
- when/if the phase i is repeated, the BH in B_i can kill more leaders
  - however, that happens only if there was a BH in the scratchpad used in the previous run
  - in such case, we credit the kill not to the BH in B_i, but to the BH in the scratchpad that caused the repeat of the phase
    - each BH can receive at most one such credit

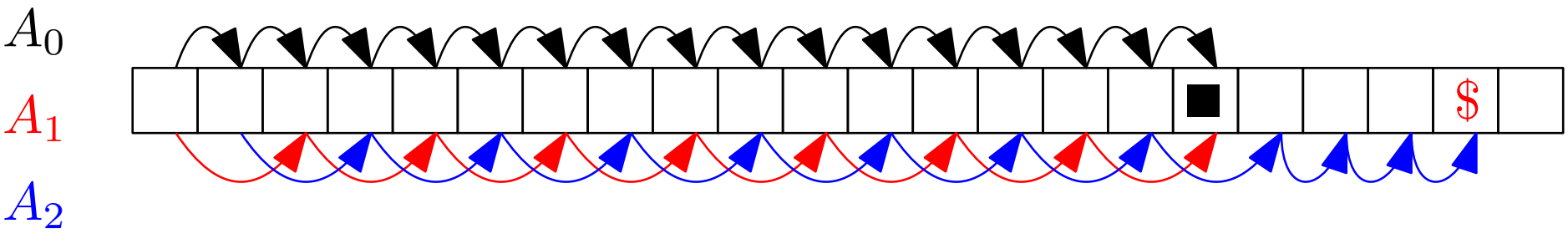
Upper Bound for Finite Size

- the previous algorithm allocates boxes upto xf^2, where B_x contains the treasure
- not usable for finite search spaces
- it is possible to design an algorithm that works for finite search spaces
  - however, it needs much more ($f^2 \ln^2 f$) agents
  - uses as a building block the algorithm for treasure hunt without coordination



Simple Fast Solution for One Black Hole

- A_0: open box i at time i
- A_1: open box $2i-1$ at time $2i$; if empty, start opening boxes sequentially
- A_2: open box $2i$ at time $2i+1$; if empty, start opening boxes sequentially



Conclusions and Future Directions

- design a **fast** treasure hunt algorithm dealing with f Black Holes
  - interesting even for $f = 2$
  - perhaps use more agents that strictly necessary
- treasure hunt with black holes using randomized non-coordinated agents?
  - how many agents are needed and what would be the speedup?
  - what about allowing both weak coordination and randomization?
- our solutions are not robust w.r.t. to timing and ordering of boxes
  - how do you make them robust and at what cost?

QUESTIONS?

THANK YOU

time for reception . . .