

Cooperative Autonomous Distributed Robotic Exploration (CADRE)

Lunar Leader: Persistent, Optimal Leader Election for Multi-Agent Exploration Teams

Keenan Albee, Sriramya Bhamidipati, Federico Rossi, Joshua Vander Hook, Jean-Pierre de la Croix

Jet Propulsion Laboratory, California Institute of Technology

Presented at the

International Workshop on Autonomous Agents and Multi-Agent Systems for Space Applications (MASSpace '24)

Auckland, New Zealand

May 7, 2024





© 2024 California Institute of Technology. Government sponsorship acknowledged.

# Coordination Architectures Elected Leader



## **Coordination Architectures: figures of merit**

1. Bandwidth	2. Interpretability	3. Expressivity
How many messages are exchanged among agents? Monarch: O(n); flooding: O(n <sup>2</sup> )	How easy is it to understand a given system behavior? Is it emerging, or can it be attributed to a specific line of code on an agent?	How hard is it to encode our problem in the distributed system? Do we need to carefully design local behaviors, or can we just solve a centralized optimization problem?
4. Scalability	5. Consistency	6. Resilience

#### **Coordination Architectures**



#### **Coordination Architectures**





### **Assumptions and Requirements**

#### Assumptions

- Agents have unique IDs
- Each agent has a list of neighbors
- Communication is asynchronous but reliable (messages are not lost)

#### Requirements

- All agents agree on one leader
- If leader is lost, a new leader is selected within specified timeframe (nominally, 10 [s])
- Leader is selected based on health metrics



# Concept of Operations

- 1. Leader election runs continuously in background
- 2. A unique agent is selected as "Appointer"
- 3. Appointer queries all other agents for health metrics (convergecast)
- 4. Appointer appoints leader and informs all agents (broadcast)
- 5. Process repeats periodically as a failsafe (e.g., leader has died)

*Rationale:* decouple (i) finding a unique agent from (ii) finding the best leader, making leader selection a *centralized* problem



- Each rover and the base station is a GHS agent that only interacts with the world through messages from/to neighbors
- Messages are delivered in order but can have arbitrary delays
- Each agent only knows its neighbors' IDs
- Somehow, an algorithm needs to allow these uncoordinated agents to all agree on a unique "Appointer"



*Distributed* Minimum Spanning Tree using Gallager, Humblet, and Spira;s algorithm

Intuition: recursively merge trees log(n) times.

0. Everyone is root (■) of its own onenode subtree

- 1. Ping neighbors to find edge cost, send to root along own subtree's MST
- 2. Nearest\* neighbor subtrees merge \*defined by edge cost
- 3. Repeat up to log(n) times



R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," ACM Trans. Program. Lang. Syst., vol. 5, no. 1, pp. 66–77, Jan. 1983, doi: <u>10.1145/357195.357200</u>. <u>12 jpl.nasa.gov</u>

• Start with a forest of single GHS agents; initialize each to a level; set leader of this component



- Build up using repeated merge and absorb operations
  - Very important: operations can be asynchronous



• Continue until no new components to probe



• GHS output: a rooted MST, which can be used for leader election using a unique "Appointer" node



## Selecting the best leader

- Appointer requests health information from all agents
  - Broadcast request on MST
  - Wait for convergecast via MST
- Appointer decides best leader
- Appointer informs all agents of new leader via MST broadcast



#### **End-to-end leader election**



#### Leader re-election

- Idea: re-run leader election periodically
- Concern 1: how do we ensure that messages from multiple leader election epochs do not overlap?
  - Append epoch ID to messages
  - Reject messages with old epoch IDs



#### Leader re-election

- Idea: re-run leader election periodically
- Concern 2: what if agents' clocks are not perfectly synchronized?
  - Buffer messages near the beginning of a re-election period





—0 "ubuntu-linux-20-04-desktop"————————————————————————————————————	
500000, w_temp: 0.500000, soc_norm: 0.000000, temp_norm: 0.738095	.500000, w_temp: 0.500000, soc_norm: 0.000000, temp_norm: 0.738095
EVENT: (26887) (2024-04-29T19:08:34.814219) ACTIVITY_LO: (leaderElection) leaderElectionOutputs : ***Lead	EVENT: (26887) (2024-04-29T19:08:34.611164) ACTIVITY_LO: (leaderElection) leaderElectionOutputs : ***Lead
er election is complete.*** phase: 2, start_round_flag_: 1, recv_queue: 0, leader_id: 4	er election is complete.*** phase: 2, start_round_flag_: 1, recv_queue: 0, leader_id: 4
VENT: (26888) (2024-04-29T19:08:35.617392) ACTIVITY_LO: (leaderElection) leaderIdTransmitted : Sent lead	EVENT: (26888) (2024-04-29T19:08:36.220444) ACTIVITY_LO: (lenderElection) leaderIdTransmitted : Sent lead
r change to listening components (leader_id: 4)	er change to listening components (leader_id: 4)
VENI: (2688/) (2024-04-29119:08:40.038580) ACTIVITY_LO: (LeaderElection) LeaderElectionOutputs : Reset L	EVENI: (20887) (2024-04-29119:08:40.038580) ACTIVITY_LO: (LeaderElection) LeaderElectionOutputs : Reset L
cycle. phase: 0, start round_ttag_: 0, recv_queue: 0, teader_10: 4	E cycle, phase: 0, start round_tag: 0, recv_queue: 0, leader 10: 4
veni: (20800) (2024-04-29119:08:40.038009) ACTIVIT_LU: (LeaderElection) Legenericevent : connected agen	to list, 1, 2, 4
is List: 2, 3, 4 VENT. (26887) (2024.04-20110.08.41 043012) (CITVITY LO. (leaderElection) leaderElectionOutputs · Started	USELT, 1, 3, 4 EVENT, (2007) (2024-04-2010-08-41 043026) ACTIVITY 10. (leaderElection) leaderElectionOutputs · Started
new IF cycle after quiescent period, phase: 0, start round flag : 1, rev queue: 0, leader id: 4	new LE cycle after quiescent period, phase 0, start round flag : 1, recy queue 0, leader id: 4
VENT: (26880) (2024-04-29119:08:44.255308) ACTIVITY LO: (leaderElection) LEGenericEvent : [LE]: soc: 0.0	EVENT: (26880) (2024-04-29T19:08:44.657048) ACTIVITY LO: (leaderElection) LEGenericEvent: [LE]: soc: 0.0
00000, valid: 1, temp: 0.000000, valid: 1, health: 0.369048	00000, valid: 1, temp: 0.000000, valid: 1, health: 0.369048
VENT: (26880) (2024-04-29T19:08:44.255331) ACTIVITY LO: (leaderElection) LEGenericEvent : [LE]: w soc: 0	EVENT: (26880) (2024-04-29T19:00:44.657067) ACTIVITY LO: (leaderElection) LEGenericEvent : [LE]: w soc: 0
500000, w_temp: 0.500000, soc_norm: 0.000000, temp_norm: 0.738095	.500000, w_temp: 0.500000, soc_norm: 0.000000, temp_norm: 0.738095
VENT: (26887) (2024-04-29T19:08:44.858952) ACTIVITY_L0: (leaderElection) leaderElectionOutputs : ***Lead	EVENT: (26887) (2024-04-29T19:08:44.657197) ACTIVITY_L0: (leaderElection) leaderElectionOutputs : ***Lead
r election is complete.*** phase: 2, start_round_flag_: 1, recv_queue: 0, leader_id: 4	er election is complete.*** phase: 2, start_round_flag_: 1, recv_queue: 0, leader_id: 4
VENT: (26888) (2024-04-29T19:08:45.663600) ACTIVITY_LO: (leaderElection) leaderIdTransmitted : Sent lead	EVENT: (26888) (2024-04-29T19:08:46.266369) ACTIVITY_LO: (leaderElection) leaderIdTransmitted : Sent lead
r change to listening components (leader_id: 4)	er change to listening components (leader_id: 4)
VENT: (26887) (2024-04-29119:08:50.087549) ACTIVITY_LO: (LeaderElection) LeaderElectionOutputs : Reset L	EVENI: (2088/) (2024-04-29119:08:50.08252/) ACTIVITY_LO: (leaderElection) leaderElectionOutputs : Reset L
Cycle. phase: 0, start round_tlag_: 0, recv_queue: 0, teader_10: 4	E cycle, phase: 0, start round_tag : 0, recv_queue: 0, leader 10: 4
veni: (2000) (2024-04-29119:00:30.00/3/2) ACTIVIT_LU: (Leaderetettion) Ledeneticevent : connected agen	te list 1 2 4
S (13(, 2, 3, 4)) (2024-04-29T19-08-51 096650) ACTIVITY 10- (leaderFlection) leaderFlectionOutputs · Started	EVENT (26837) (2024-04-29T19-08-51 087368) ACTIVITY 10. (leaderElection) leaderElectionOutputs · Started
new Le cycle after quiescent period, phase: 0, start round flag : 1, recy queue: 0, leader id: 4	new Le cycle after quiescent period, phase: 0. start round flag : 1. recy queue: 0. leader id: 4
-2 "ubuntu-linux-20-04-desktop"	8 "ubuntu-linux-20-04-desktop"
EVENT: (26887) (2024-04-29T19:08:34.959570) ACTIVITY_LO: (leaderElection) leaderElectionOutputs : ***Lead	EVENT: (26887) (2024-04-29T19:08:35.060013) ACTIVITY_LO: (leaderElection) leaderElectionOutputs : ***Lead
r election is complete.*** phase: 2, start_round_flag_: 1, recv_queue: 0, leader_id: 4	er election is complete.*** phase: 2, start_round_flag_: 1, recv_queue: 0, leader_id: 4
EVENT: (26888) (2024-04-29T19:08:39.887698) ACTIVITY_LO: (leaderElection) leaderIdTransmitted : Sent lead	EVENT: (26888) (2024-04-29T19:08:36.569942) ACTIVITY_LO: (leaderElection) leaderIdTransmitted : Sent lead
r change to listening components (leader_id: 4)	er change to listening components (leader_id: 4)
VENT: (26887) (2024-04-29119:08:40.089345) ACTIVITY LO: (LeaderElection) LeaderElectionOutputs : Reset L	EVENT: (26887) (2024-04-29119:08:40.089400) ACTIVITY_LO: (leaderElection) leaderElectionOutputs : Reset L
Cycle. phase: 0, start round_itag_: 0, recv_queue: 0, teader_10: 4	E cycle, phase: 0, start round_itag: 0, recv queue: 0, teader 10: 4
vent: (2000) (2024-04-29119:00:40.0093//) ACTIVITELD: (Leader Election) Levenet icevent : connected agen	te list 1 2 3
VENT. (26887) (2024-04-29T19.08.41.093328) ACTIVITY 10. (leaderFlection) leaderFlectionOutputs . Started	EVENT (26837) (2024-04-29T19-08-41 093080) ACTIVITY 10. (leaderElection) leaderElectionOutputs · Started
new LE cycle after guiescent period, phase: 0, start round flag : 1, recy gueue: 0, leader id: 4	new Le cycle after guiescent period, phase: 0, start round flag : 1, recy gueue: 0, leader id: 4
error] LE: Could not call ghs.process(): Received SRCH msg from parent, while still waiting for replies-	[error] LE: Could not call ghs.process(): Received SRCH msg from parent, while still waiting for replies-
new round?	-new round?
EVENT: (26880) (2024-04-29T19:08:44.004215) ACTIVITY_LO: (leaderElection) LEGenericEvent : [LE]: soc: 0.0	EVENT: (26880) (2024-04-29T19:08:44.004215) ACTIVITY_LO: (leaderElection) LEGenericEvent : [LE]: soc: 1.0
00000, valid: 1, temp: 0.000000, valid: 1, health: 0.369048	00000, valid: 1, temp: 0.000000, valid: 1, health: 0.869048
<pre>VENT: (26880) (2024-04-29T19:08:44.004233) ACTIVITY_L0: (leaderElection) LEGenericEvent : [LE]: w_soc: 0</pre>	EVENT: (26880) (2024-04-29T19:08:44.004233) ACTIVITY_LO: (leaderElection) LEGenericEvent : [LE]: w_soc: 0
500000, w_temp: 0.500000, soc_norm: 0.0000000, temp_norm: 0.738095	500000, w temp: 0.500000, soc_norm: 1.0000000, temp_norm: 0.738095
VENI: (26887) (2024-04-29119:08:45.110390) ACTIVITY LO: (LeaderElection) LeaderElectionOutputs : ***Lead	EVENI: (2088/) (2024-04-29119:08:45.110425) ACTIVITY LO: (leaderElection) leaderElectionOutputs : ***Lead
er election is complete.*** phase: 2, start_round_flag_: 1, recv queue: 0, leader_10: 4	er election is complete.*** phase: 2, start round tiag : 1, recv queue: 0, leader in: 4
vert. (2000) (2024-04-2919:00:49.951009) ACTIVIT_U: (Leader Etection) Leader Idifansmilled : Sent Lead	er change to listening components (leader id: 4)
Vent (26887) (2024-04-2019) 08:50 032459) ACTIVITY 10: (leaderElection) leaderElectionOutputs - Reset 1	EVENT (26887) (2024-04-2019-08-56 032584) ACTIVITY 10. (leaderElection) leaderElectionOutputs . Peset +
cycle, phase 0, start round flag : 0, recy queue: 0, leader id: 4	E cycle, phase 0, start round flag : 0, recy queue: 0, leader id: 4
VENT: (26880) (2024-04-29T19:08:50.032490) ACTIVITY LO: (leaderElection) LEGenericEvent : Connected agen	EVENT: (26880) (2024-04-29T19:08:50.032604) ACTIVITY LO: (leaderElection) LEGenericEvent : Connected agen
s list: 1, 2, 4	ts list: 1, 2, 3
EVENT: (26887) (2024-04-29T19:08:51.037271) ACTIVITY_L0: (leaderElection) leaderElectionOutputs : Started	EVENT: (26887) (2024-04-29T19:08:51.037360) ACTIVITY LO: (leaderElection) leaderElectionOutputs : Started
new LE cycle after quiescent period. phase: 0, start_round_flag_: 1, recv_queue: 0, leader id: 4	_new 💵 cycle after quiescent period. 🏻 phase: 0, start round flag : 1, recv queue: 0, leader id: 4

## Message complexity



jpl.nasa.gov

