Performance-Aware Self-Configurable Multi-Agent Networks: A Distributed Submodular Approach for Simultaneous Coordination and Network Design

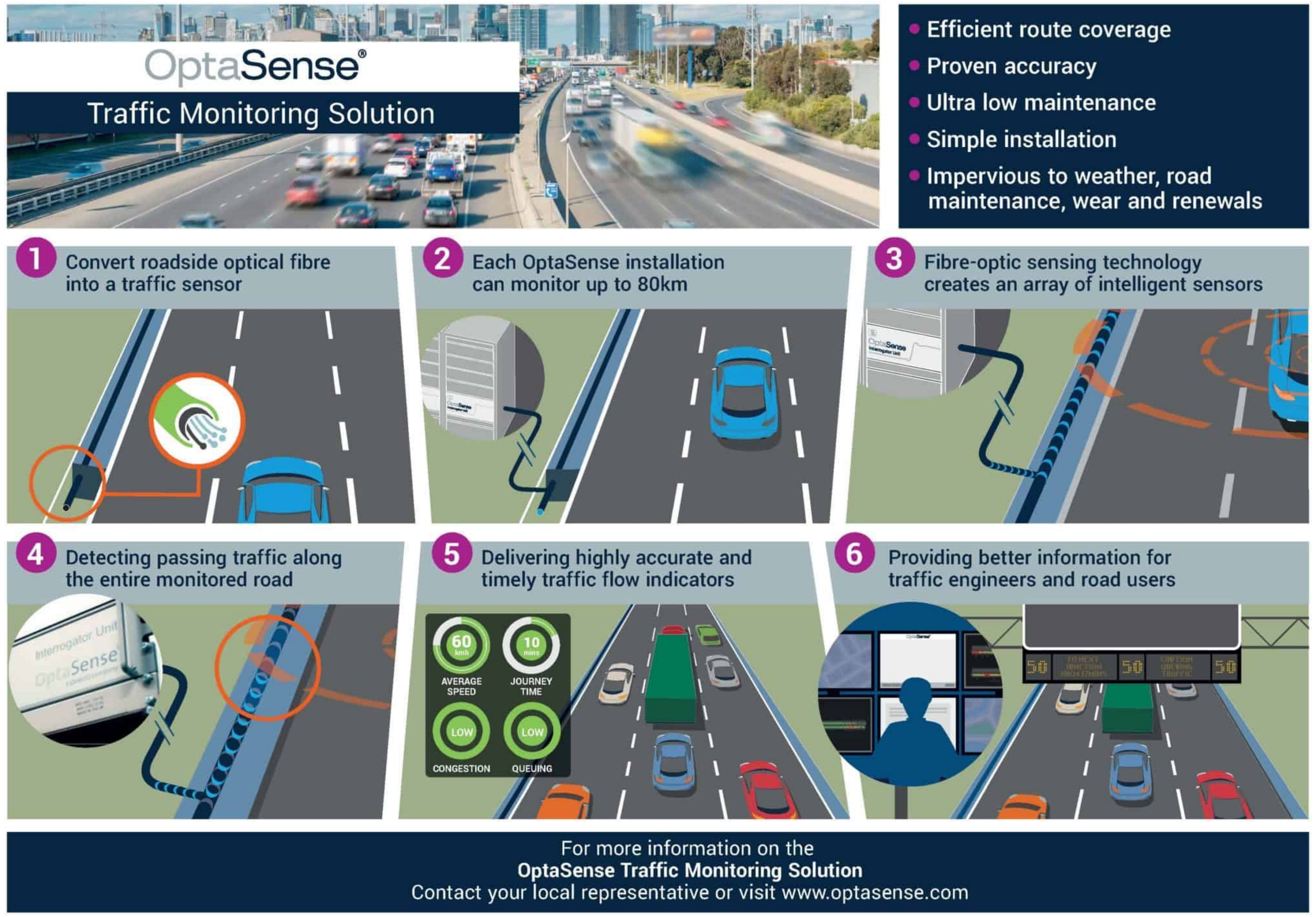
Zirui Xu

Vasileios Tzoumas

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Traffic Monitoring

Goal: Maximize knowledge of traffic conditions based on live data collected by distributed traffic sensors

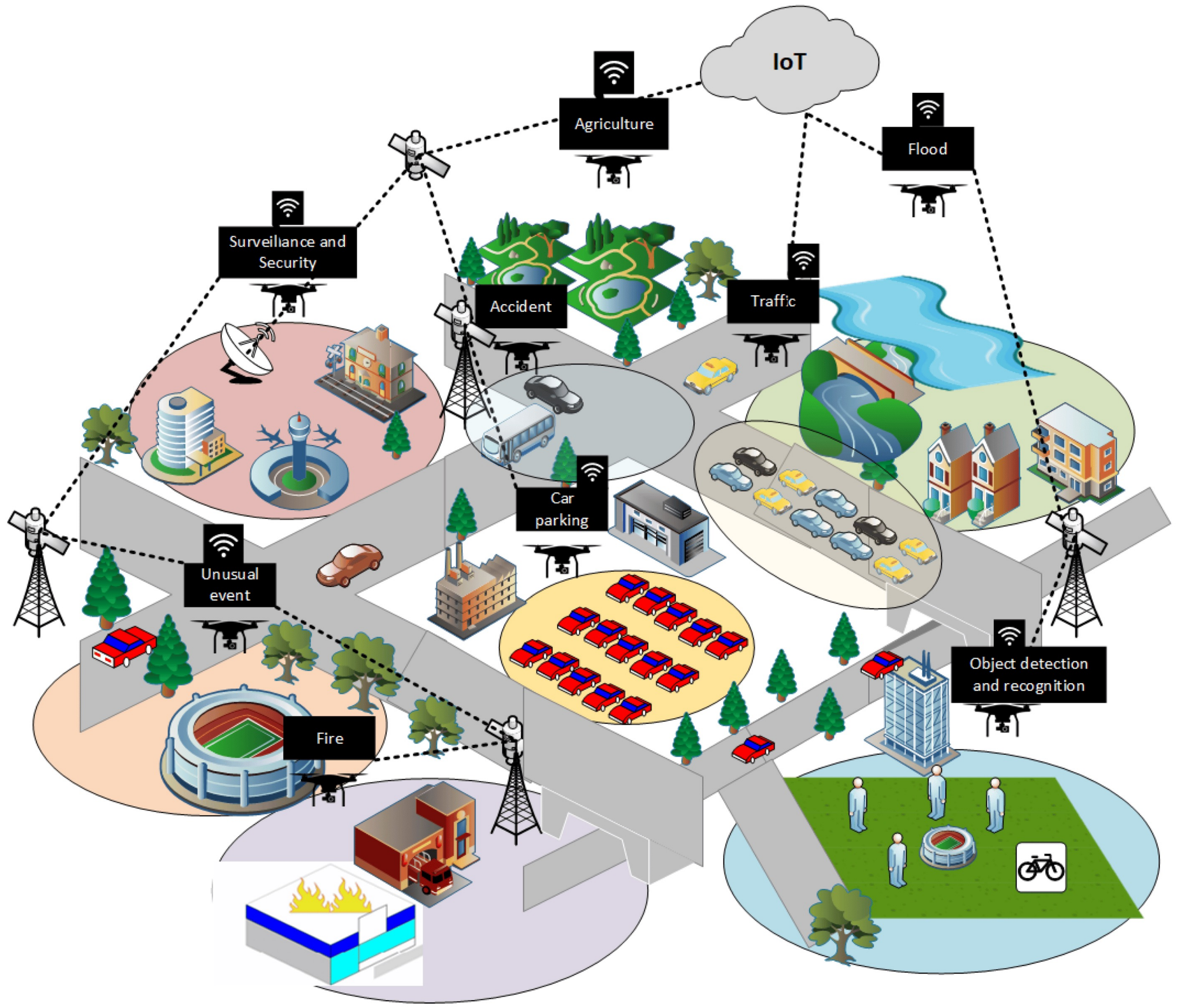


Problem: How should the sensors coordinate where to observe to achieve the goal?[1]

[1] Li, Mehr, Horowitz, Transportation Research B '23

Event Detection

Goal: Maximize number of detected events based on live data collected by wireless sensor networks



Problem: How should the sensors coordinate where to observe to achieve the goal?[2]

[2] Kumar, Rus, Singh, IEEE Pervasive Computing '04

Wildfire Management

Goal: Maximize knowledge of wildfire influence based on live data collected by distributed remote sensors

Problem: How should the sensors coordinate where to observe to achieve the goal?[3]

[3] Afghah, Razi, Chakareski, Ashdown, IEEE INFOCOM '19

All above are submodular maximization problems[1]

Given:

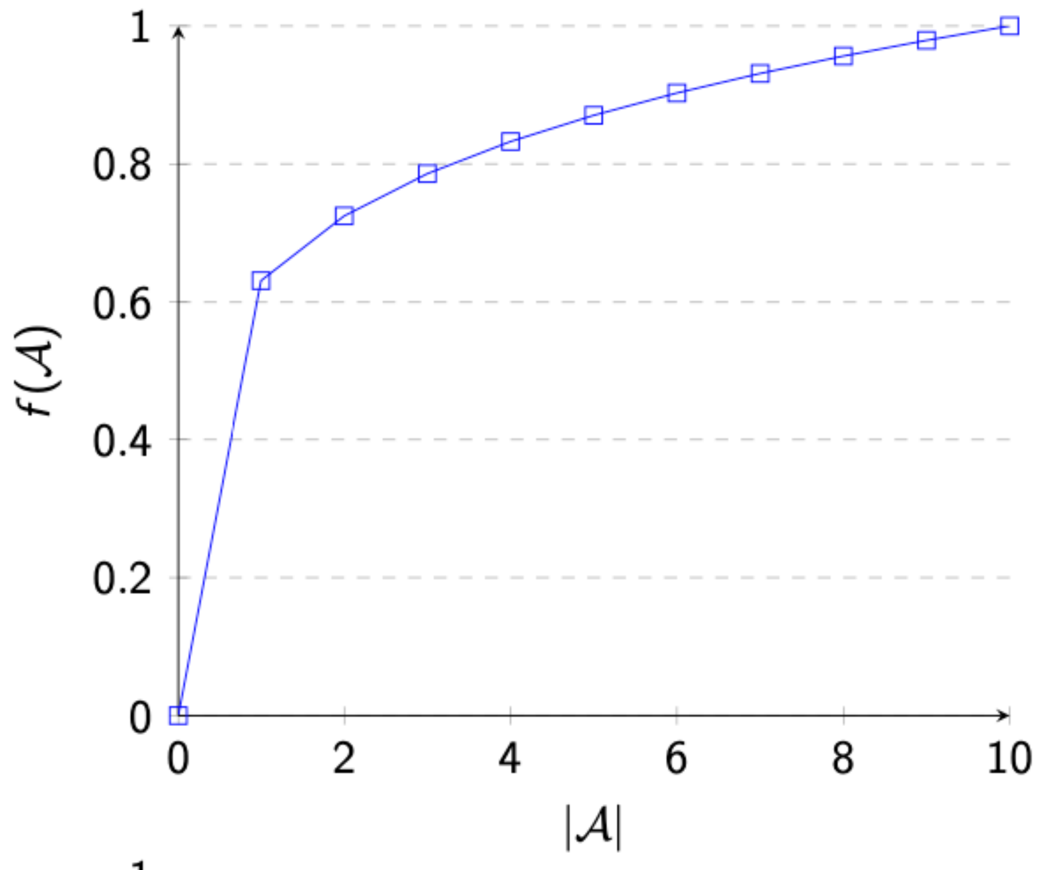
- agents $\mathcal{N}$
- finite available action sets $\mathcal{V}_i$, $\forall i \in \mathcal{N}$
  (e.g., a set of FOV's for a camera to choose from)
- set function $f : 2^{\prod_{i\in\mathcal{N}} \mathcal{V}_i} \mapsto \mathbb{R}$

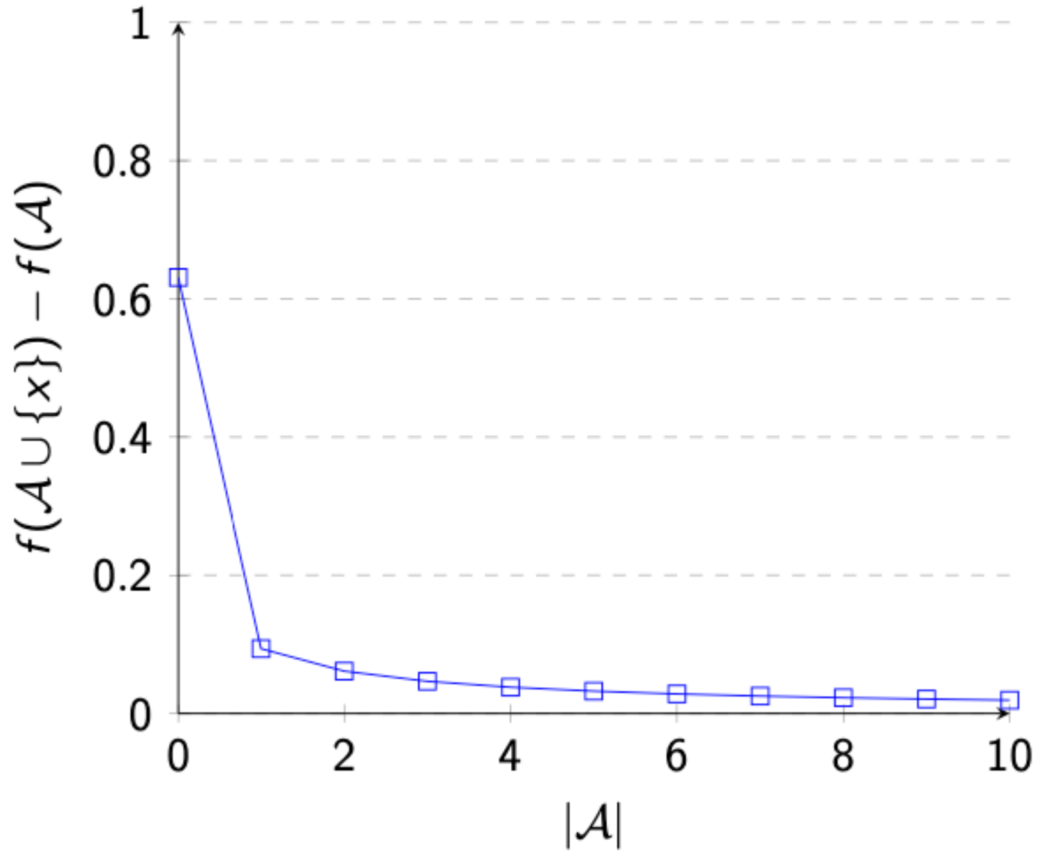
the agents $\mathcal{N}$ select actions $\{a_i\}_{i\in\mathcal{N}}$ to solve

$$\max_{a_i \in \mathcal{V}_i, \forall i \in \mathcal{N}} f(\{a_i\}_{i\in\mathcal{N}})$$

non-decreasing

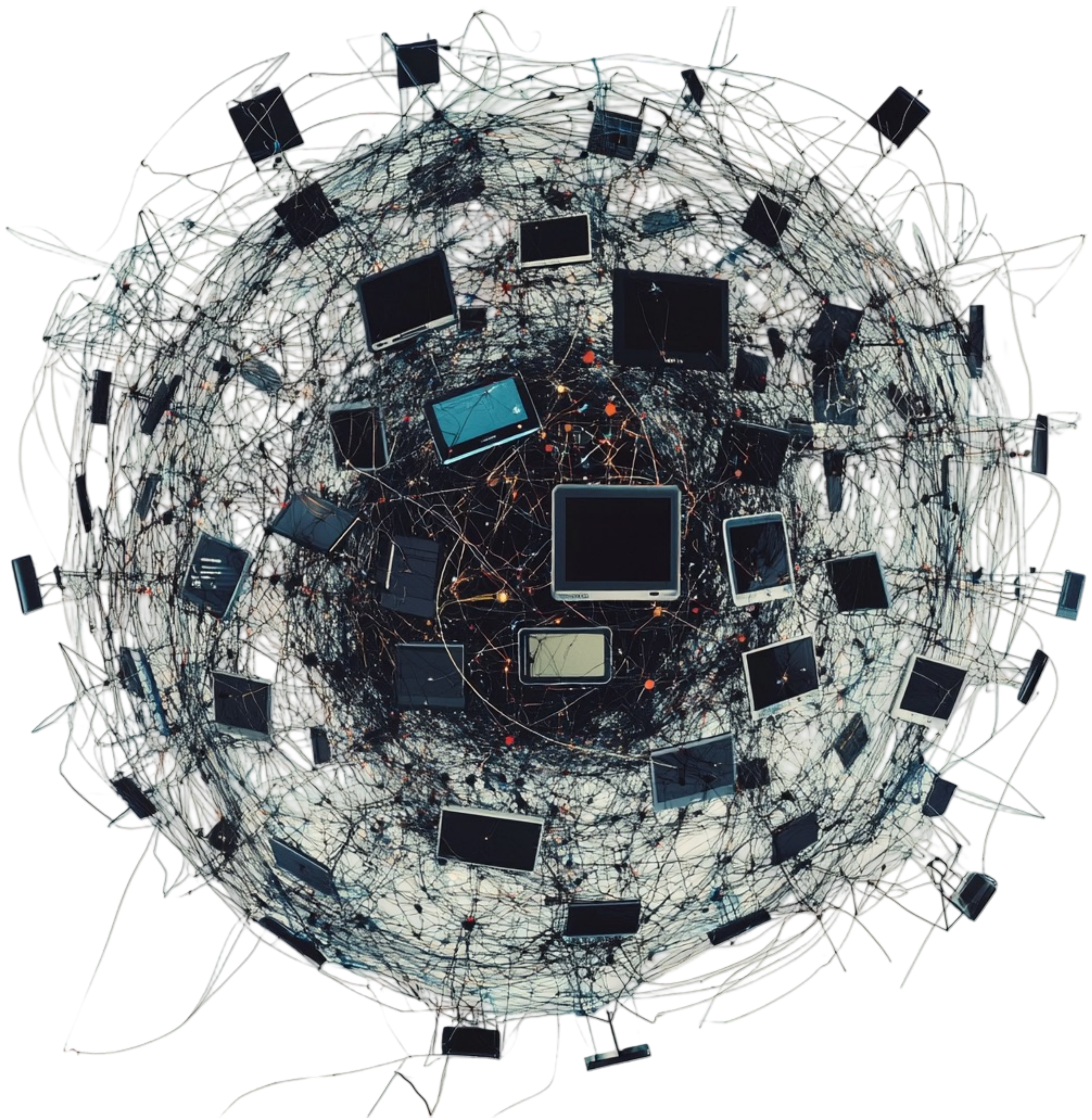
submodular
(and 2nd-order submodular)

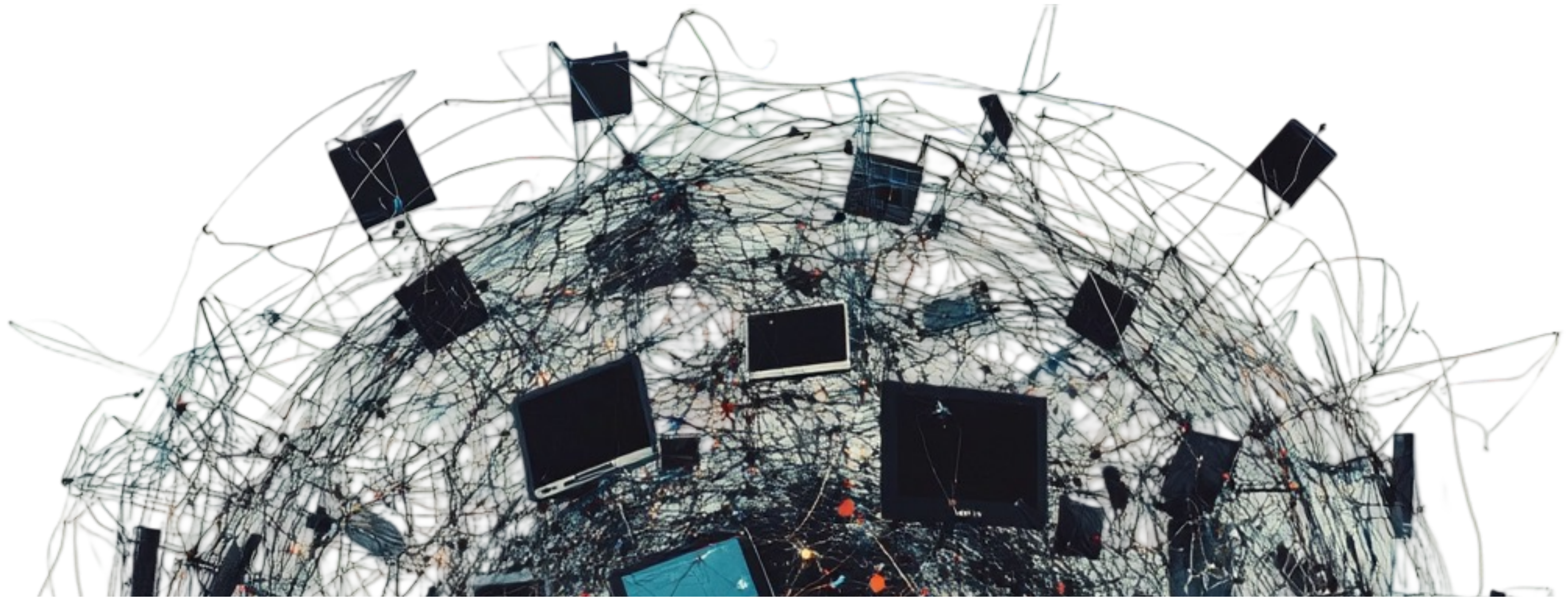




[1] Atanasov; Bilmes; Bushnell; Calinescu; Chekuri; Clark; Corah; Gharesifard; Hassani; Hespanha; Iyer; Karbasi; Kia; Konda; Krause; Li; Marden; Martinez; Michael; Mirzasoleiman; Mokhtari; Poovendran; Rezazadeh; Robey; Smith; Sundaram; Tokekar; ...

Agents Have Limited Communication Bandwidth and Data Rate





Having all sensors coordinating with all other is often impractical:

- Impractical communication and computation loads/requirements
- Impractical decision time

Agents Have Limited Communication Bandwidth and Data Rate

Research Question:
Given the agents' communication constraints, what is the best network configuration?

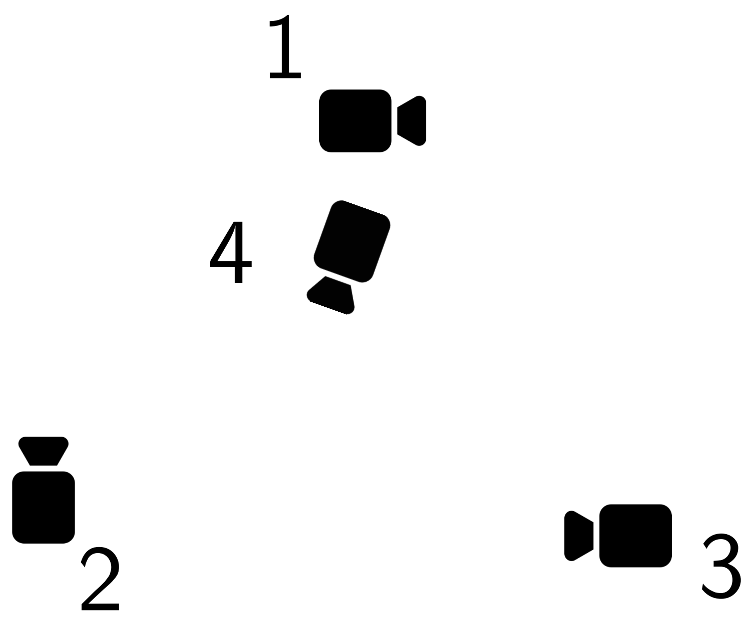
EMERGING INDUSTRIES (continued)

Artificial intelligence: Data-intensive science – the interaction between people and technology – and the use of a convergent approach to research come together in the research area of AI. For example, imagine a future in which autonomous vehicles fill the roads and the sky, all the while constantly communicating with each other, the roadway and traffic control signals. Understanding the behavior of this "swarm" and ensuring that terrestrial and aerial traffic flow evolves safely and efficiently is a research challenge that requires insights from biology, mathematics, engineering, human psychology and computer science. The research addresses the problem of how to integrate large flows of data from sensors in vehicles and embedded in the roadway and visual information from cameras. The traffic flow of highways and skyways of the future is just one example of the ways in which research that provides the ability to deploy AI on a large scale will transform our lives. Other potential outcomes from research on AI and cloud computing include robot assistants for the home-bound, diagnostic systems to aid physicians, improved factory automation and, when coupled with novel approaches to the analysis of large data-streams, new tools for the intelligence community.

NSF Strategic Plan 2022-2026

Network Configuration Matters to Optimization Performance

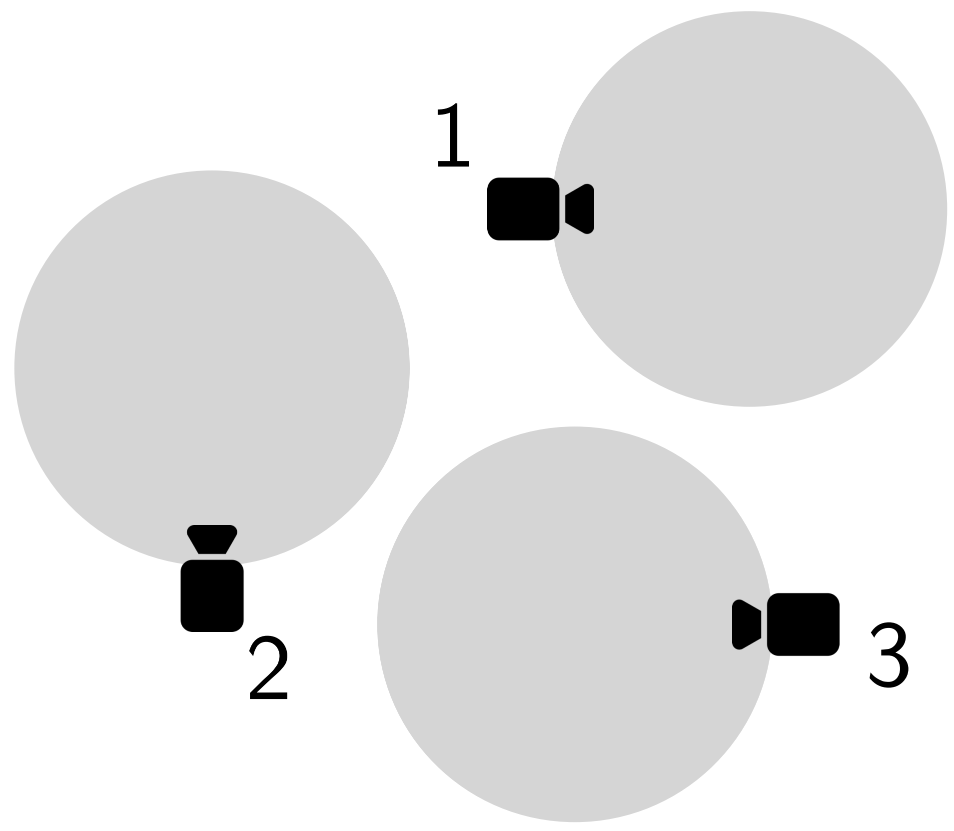
Area Monitoring Example:
- 4 cameras select their FOV's to collaboratively maximize total covered area
- Each camera is able to receive information from only up to 2 others



Network Configuration Matters to Optimization Performance

Area Monitoring Example:

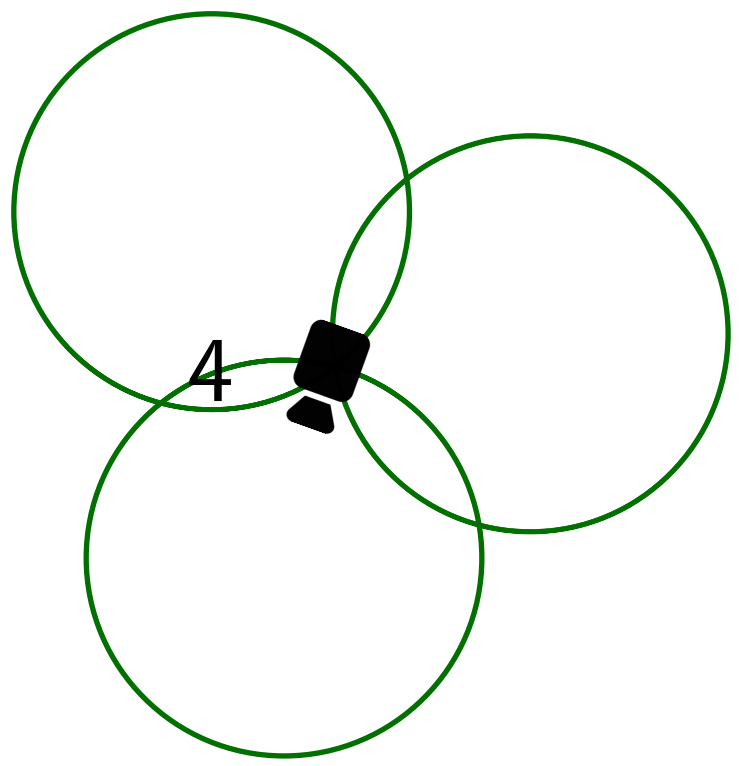
- 4 cameras select their FOV's to collaboratively maximize total covered area
- Each camera is able to receive information from only up to 2 others



cameras 1-3 already decided

Area Monitoring Example:

- 4 cameras select their FOV's to collaboratively maximize total covered area
- Each camera is able to receive information from only up to 2 others



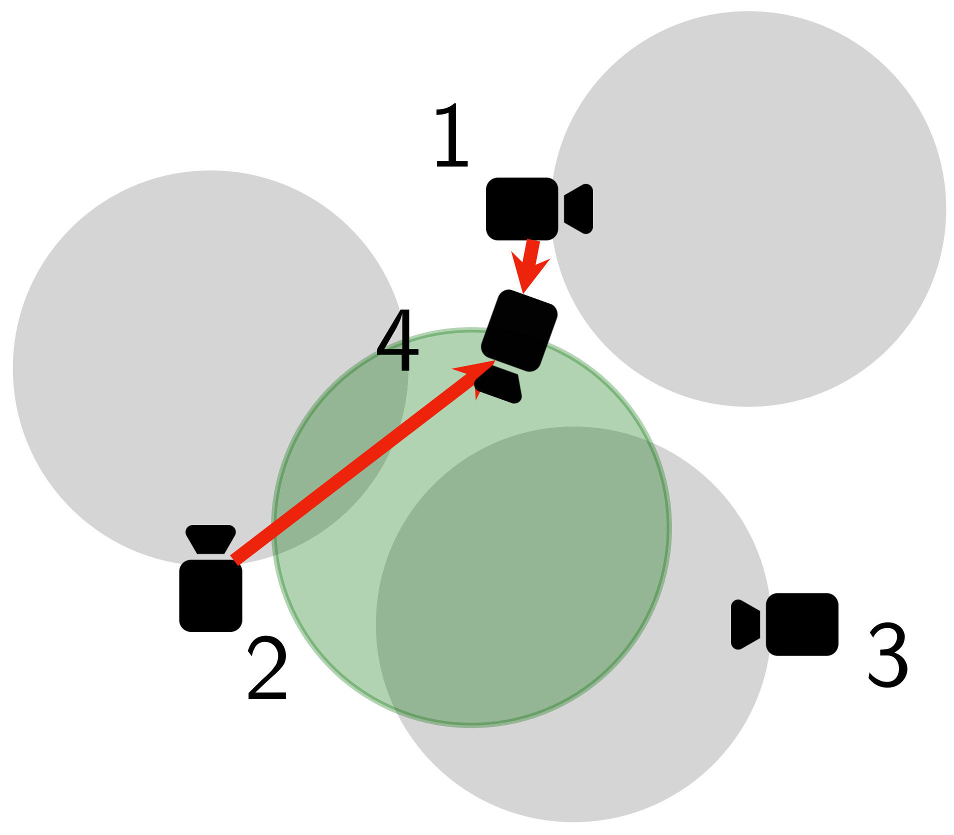
and camera 4 has 3 options

Area Monitoring Example:

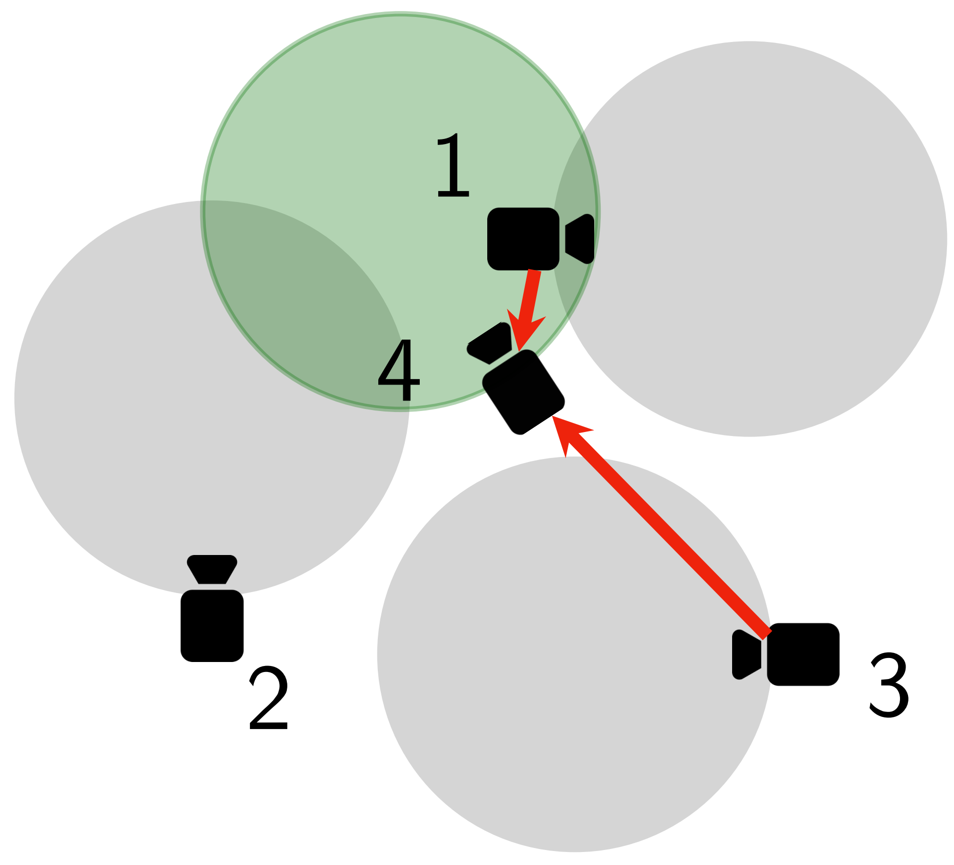
- 4 cameras select their FOV's to collaboratively maximize total covered area
- Each camera is able to receive information from only up to 2 others

All possible local network configurations for camera 4:

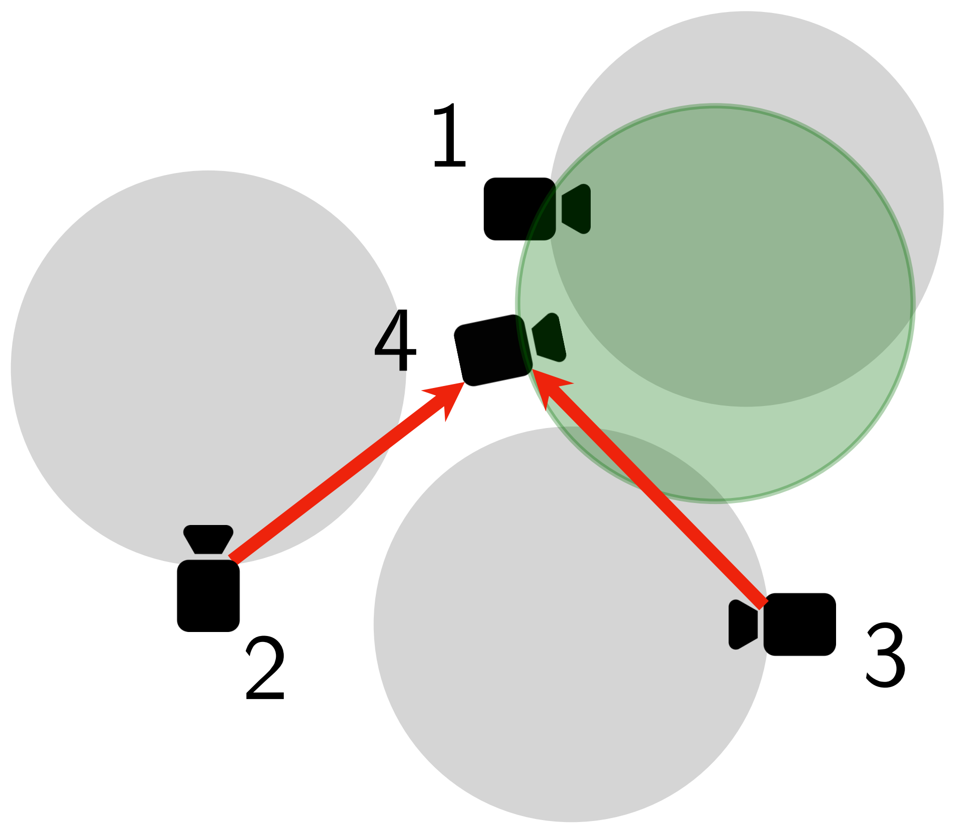
Scenario 1:
Listen to 1 & 2, no info about 3



Scenario 2:
Listen to 1 & 3; no info about 2



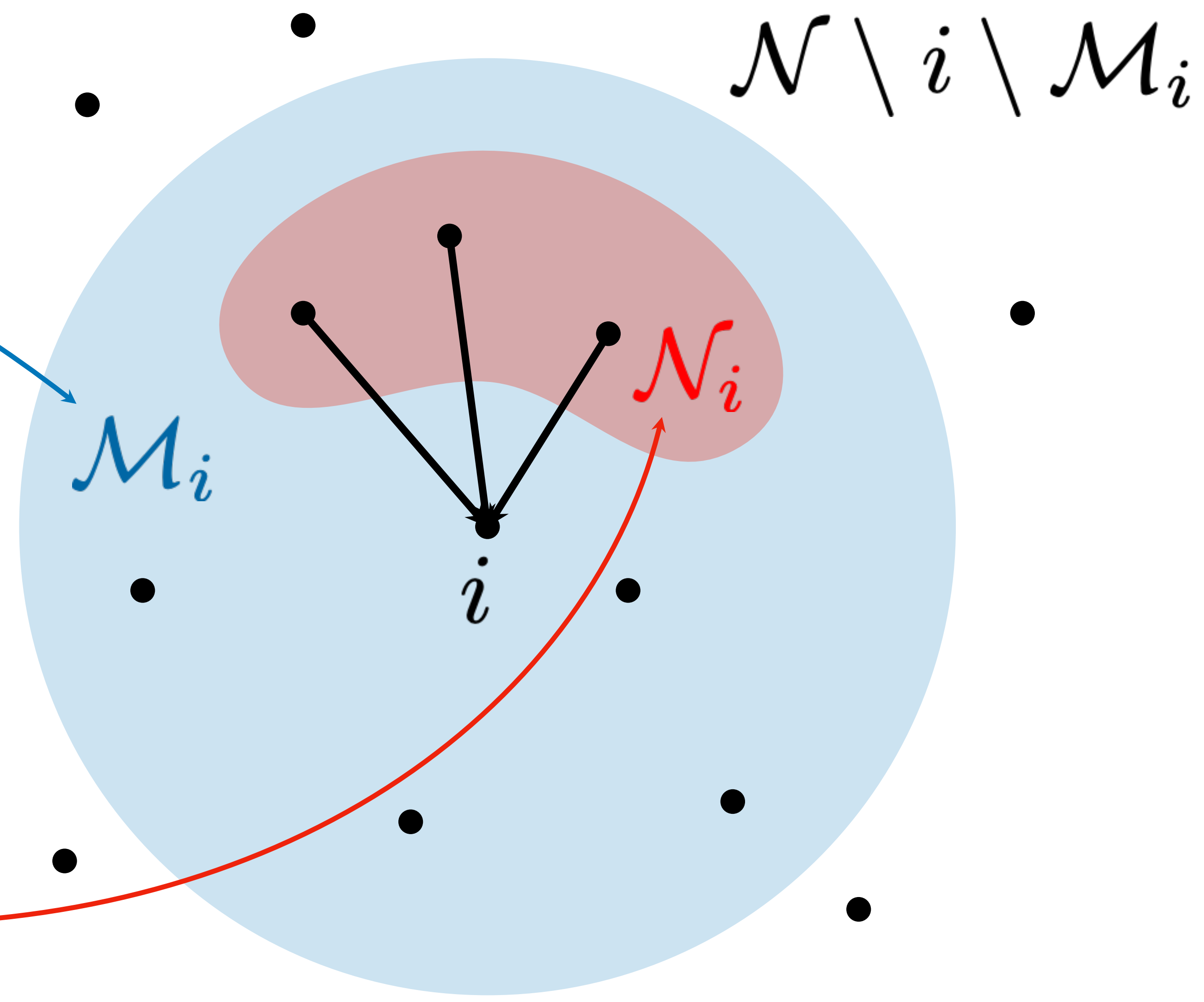
Scenario 3:
Listen to 2 & 3; no info about 1



Best network: gives largest total covered area

For each agent $i \in \mathcal{N}$, given:

- reachable neighbor candidates $\mathcal{M}_i \subseteq \mathcal{N} \setminus i$
- communication bandwidth budget α_i
- finite action sets $\mathcal{V}_i$, $\forall i \in \mathcal{N}$
- set function $f\colon 2^{\prod_{i\in\mathcal{N}} \mathcal{V}_i} \mapsto \mathbb{R}$



agent i needs to select neighborhood $\mathcal{N}_i$ and action a_i to collaboratively solve

$$\max_{\mathcal{N}_i \subseteq \mathcal{M}_i,\, |\mathcal{N}_i| \leq \alpha_i,\, \forall i \in \mathcal{N}} \; \max_{a_i \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} \; f(\{a_i\}_{i\in\mathcal{N}})$$

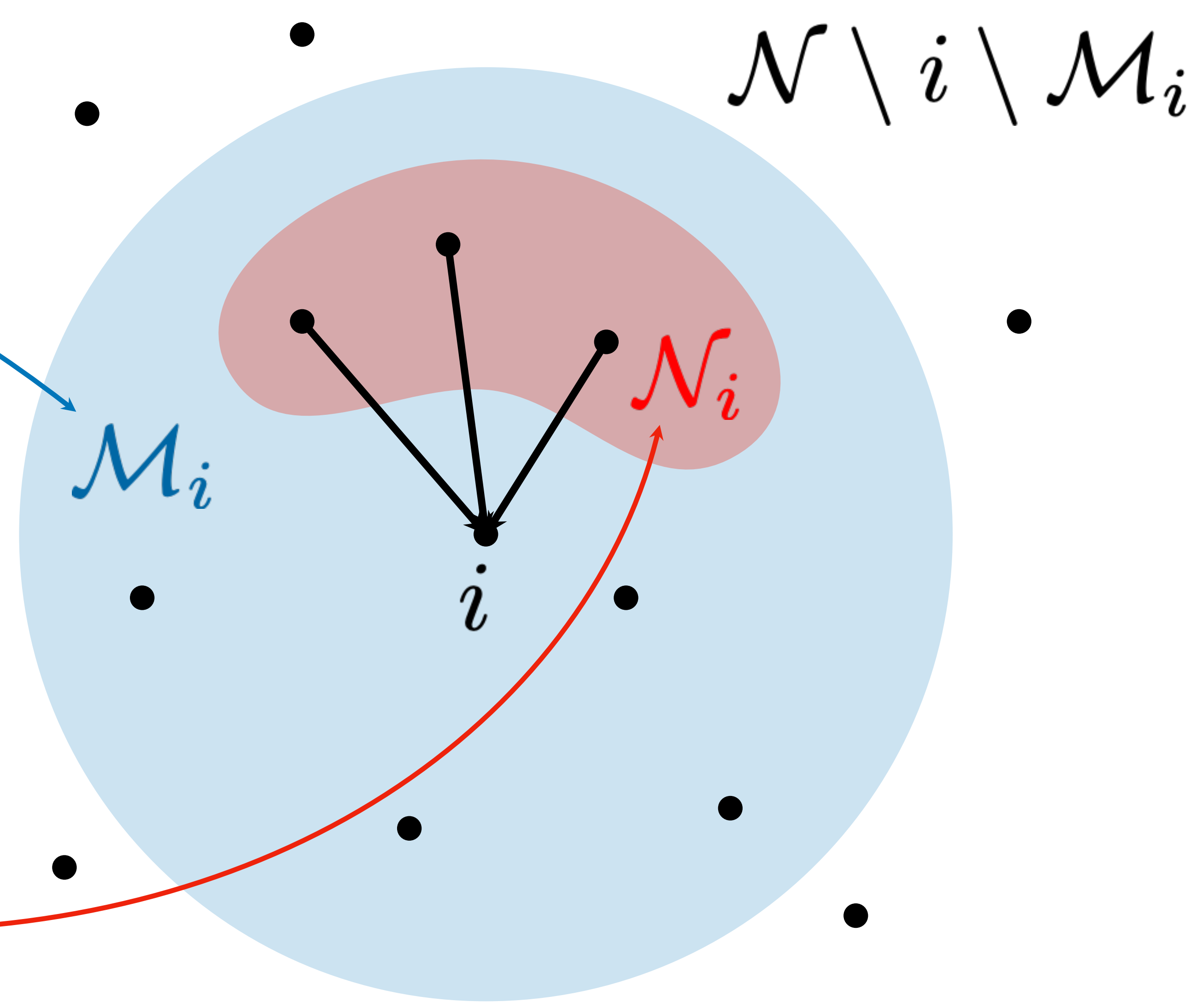
Distributed Simultaneous Action Coordination & Network Self-Configuration

For each agent $i \in \mathcal{N}$, given:

- reachable neighbor candidates $\mathcal{M}_i \subseteq \mathcal{N} \setminus i$
- communication bandwidth budget α_i
- finite action sets $\mathcal{V}_i$, $\forall i \in \mathcal{N}$
- set function $f\colon 2^{\prod_{i\in\mathcal{N}} \mathcal{V}_i} \mapsto \mathbb{R}$

agent i needs to select neighborhood $\mathcal{N}_i$ and action a_i to collaboratively solve

$$\max_{\mathcal{N}_i \subseteq \mathcal{M}_i,\, |\mathcal{N}_i| \leq \alpha_i,\, \forall i \in \mathcal{N}} \; \max_{a_i \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} \; f(\{a_i\}_{i\in\mathcal{N}})$$



if $\mathcal{M}_i = \mathcal{N} \setminus i$ and $\alpha_i = \infty$, then becomes the original maximization problem

Current Distributed Submodular Maximization Approaches

$$\max_{\mathcal{N}_i \subseteq \mathcal{M}_i,\, |\mathcal{N}_i| \leq \alpha_i,\, \forall i \in \mathcal{N}} \; \max_{a_i \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} f(\{a_i\}_{i \in \mathcal{N}})$$ has not been studied before

Distributed algorithms for $\max_{a_i \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} f(\{a_i\}_{i \in \mathcal{N}})$:

- Sequential Greedy and SOTA variants:[1–6]
  - Suboptimality guarantee 1-1/e or 1/2
  - Decision time including communication delays:
    - $O(|\mathcal{N}|^2\, \mathrm{diam}(\mathcal{G}))$ time for connected, directed graphs[2]
    - $O(|\mathcal{N}|^2)$ time for connected, directed graphs | $O(|\mathcal{N}|^3)$ time for strongly connected, undirected graphs[3]

agents must relay info across the network

[1] Fisher, Nemhauser, Wolsey, Math Prog Studies ’78
[2] Liu, Zhou, Tokekar, RAL ’20
[3] Konda, Grimsman, Marden, ACC ’22
[4] Robey, Adibi, Schlotfeldt, Hassani, Pappas, L4DC ’21
[5] Du, Qian, Claudel, Sun, TAC ’22
[6] Rezazadeh, Kia, Automatica ’23

Current Distributed Submodular Maximization Approaches

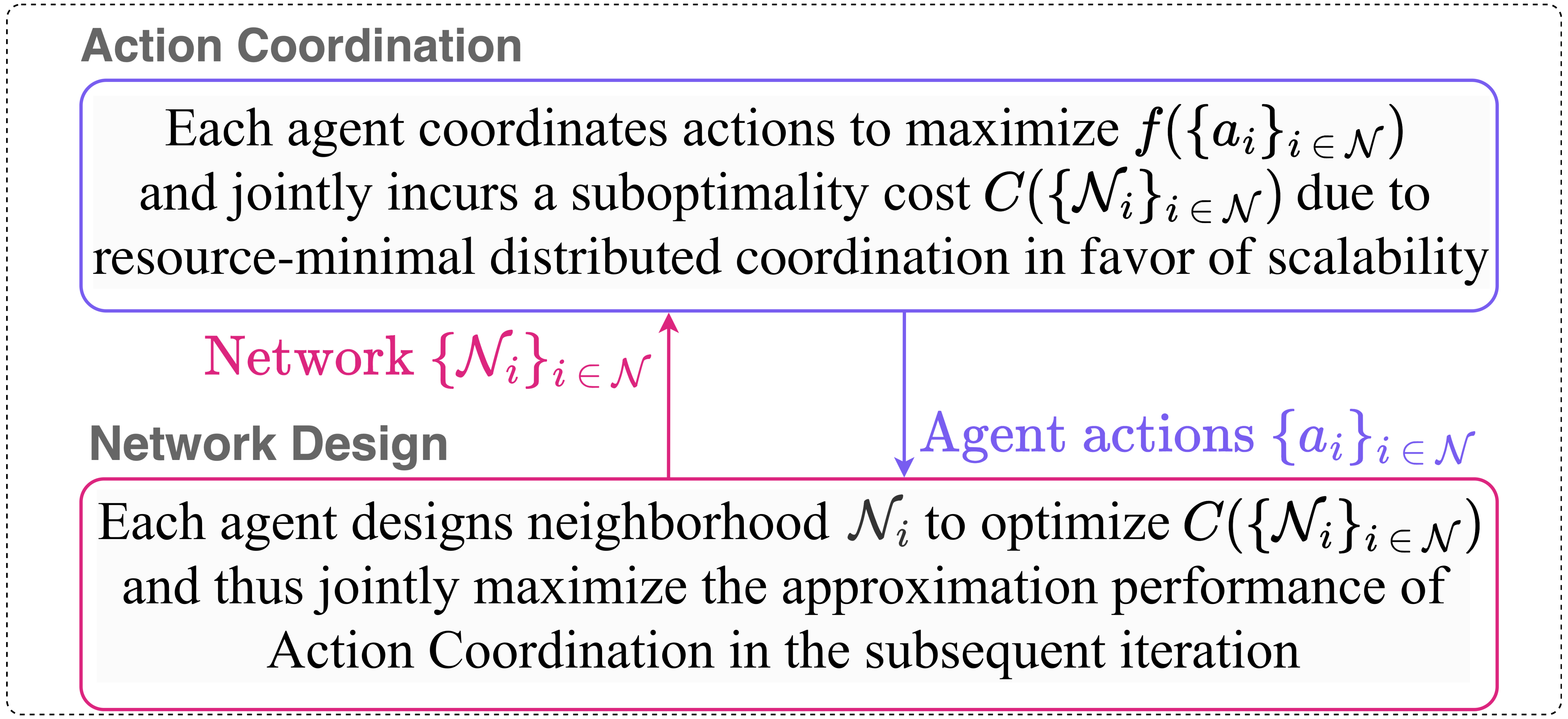
$$\max_{\mathcal{N}_i \subseteq \mathcal{M}_i,\, |\mathcal{N}_i| \leq \alpha_i,\, \forall i \in \mathcal{N}} \; \max_{a_i \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} \; f(\,\{a_i\}_{i \in \mathcal{N}}\,)$$ has not been studied before

Distributed algorithms for $\max_{a_i \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} \; f(\,\{a_i\}_{i \in \mathcal{N}}\,)$:

- Sequential Greedy and SOTA variants:[1–6]
  - Suboptimality guarantee 1-1/e or 1/2
  - Decision time including communication delays: agents must relay info across the network
    - $O(|\mathcal{N}|^2\, \mathrm{diam}(\mathcal{G}))$ time for connected, directed graphs[2]
    - $O(|\mathcal{N}|^2)$ time for connected, directed graphs | $O(|\mathcal{N}|^3)$ time for strongly connected, undirected graphs[3]
- Works that examine impact of limited information access to suboptimality bound:[7–9]
  - Quantification for worst-case f : Suboptimality guarantee $1/(\alpha+1)$[7–8] degrades proportionally to number of agents that select actions independently

[1]Fisher, Nemhauser, Wolsey, Math Prog Studies '78
[2]Liu, Zhou, Tokekar, RAL '20
[3]Konda, Grimsman, Marden, ACC '22
[4]Robey, Adibi, Schlotfeldt, Hassani, Pappas, L4DC '21
[5]Du, Qian, Claudel, Sun, TAC '22
[6]Rezazadeh, Kia, Automatica '23
[7]Gharesifard and Smith, TCNS '18
[8]Grimsman, Ali, Hespanha, Marden, TCNS '19
[9]Corah, Michael, CDC '18

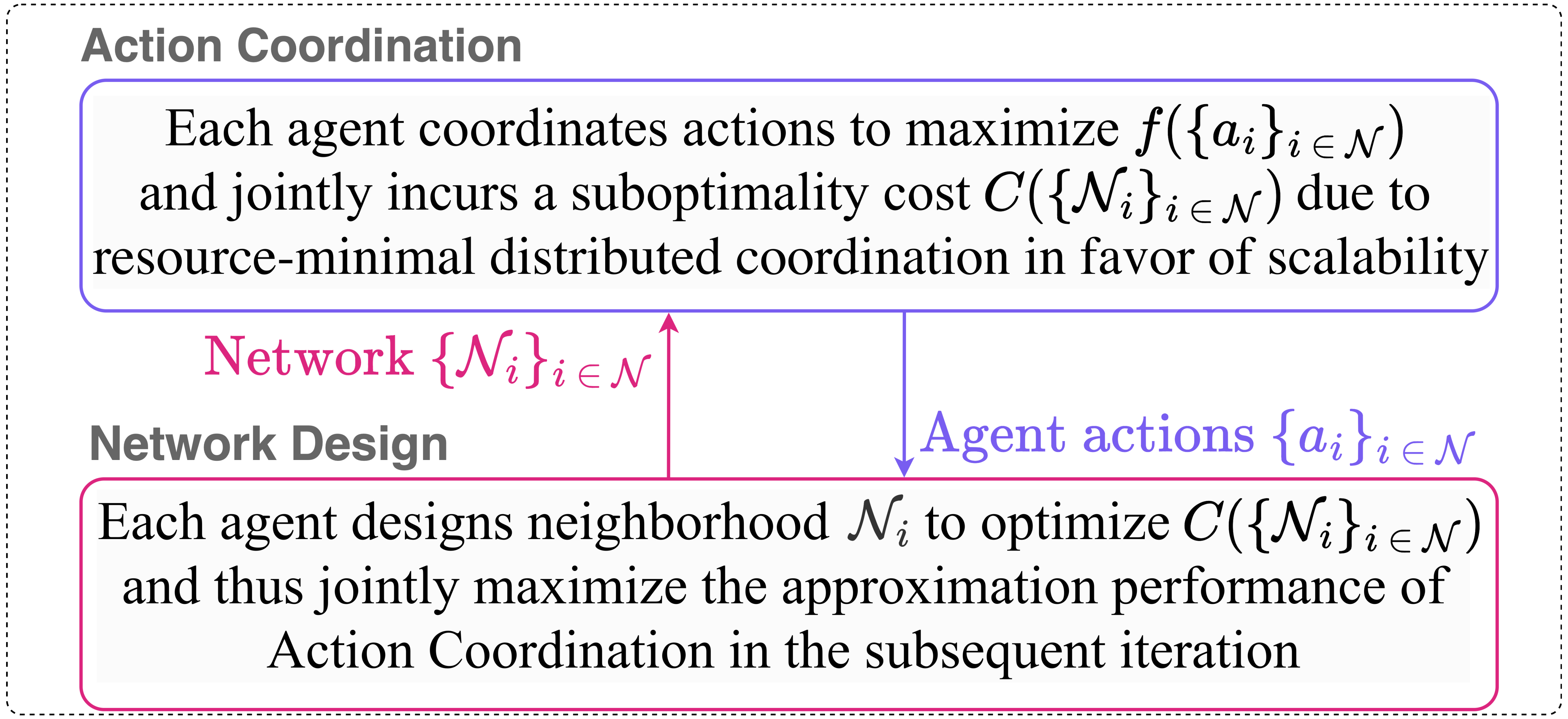
Introduced distributed optimization algorithm:



Algorithm 1: AlterNAting COordination and Network-Design Algorithm (Anaconda)

Our Contributions

Introduced distributed optimization algorithm:

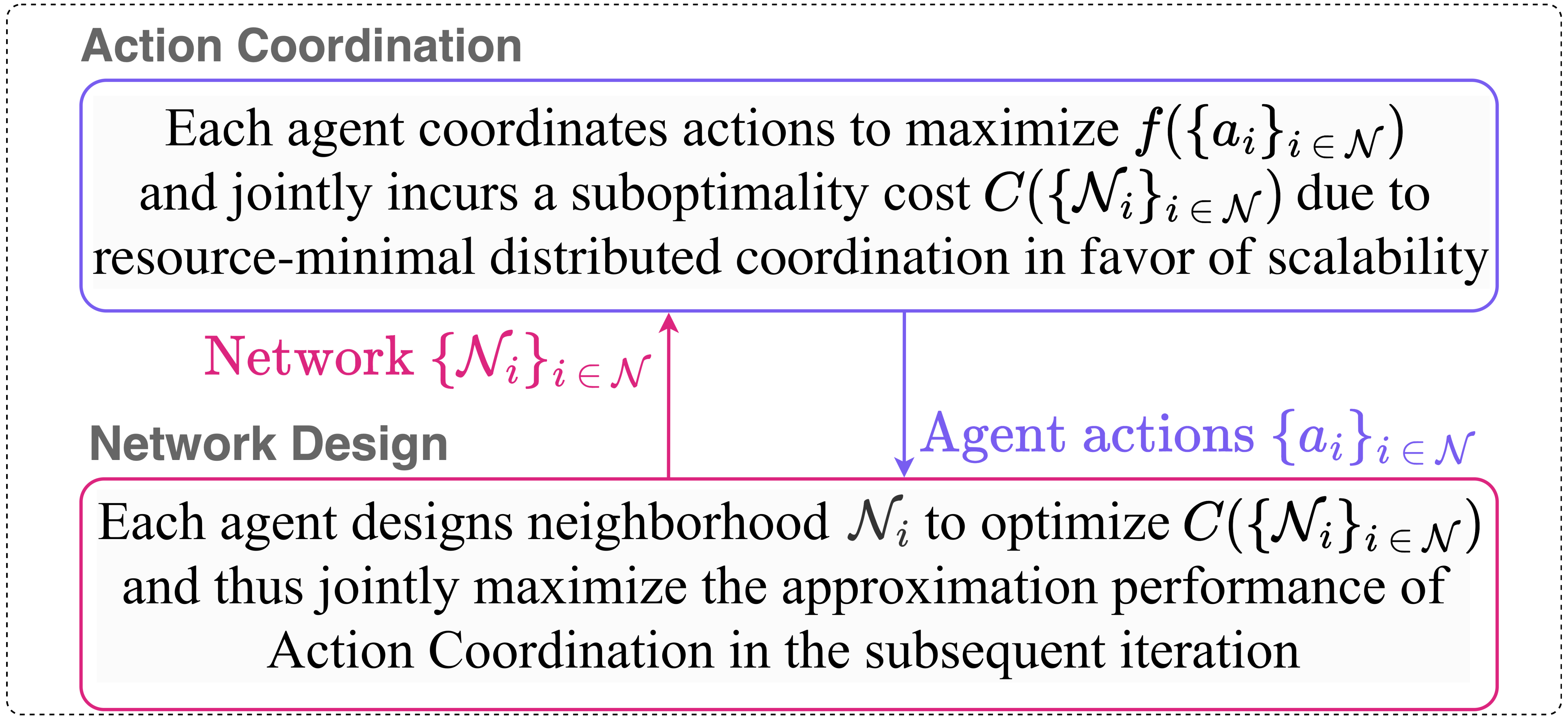


Algorithm 1: AlterNAting COordination and Network-Design Algorithm (Anaconda)

Key ideas in making Algorithm 1 scalable and near-optimal:

- $\boxed{\text{Design Action Coordination and Network Design steps to require minimal communications}}$
- Quantify impact of network configuration $\{\mathcal{N}_i\}_{i\in\mathcal{N}}$ to suboptimality of action coordination
- Use quantification to optimize network configuration via alternating optimization

Introduced distributed optimization algorithm:



Algorithm 1: AlterNAting COordination and Network-Design Algorithm (Anaconda)

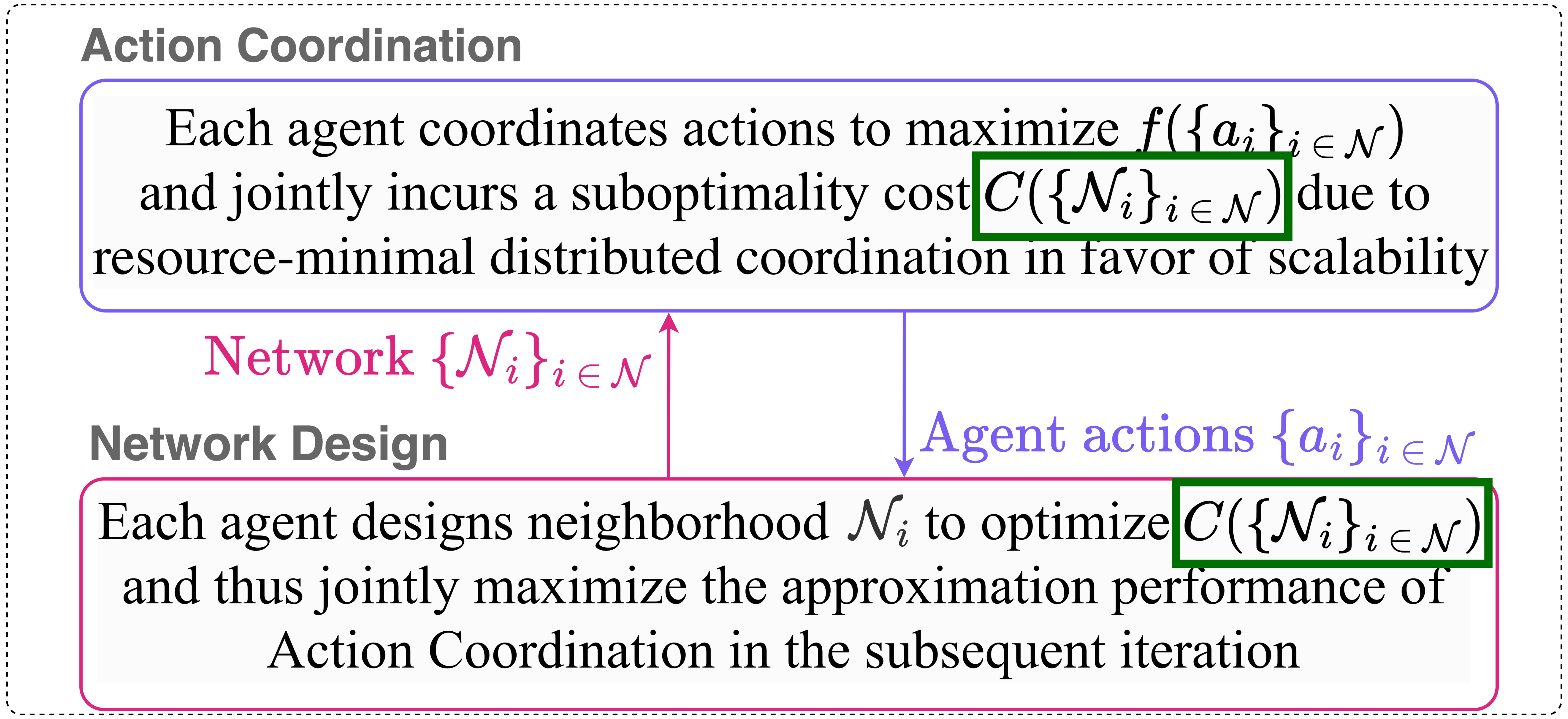
Key ideas in making Algorithm 1 scalable and near-optimal:

- Design Action Coordination and Network Design steps to require minimal communications
- Quantify impact of network configuration $\{\mathcal{N}_i\}_{i\in\mathcal{N}}$ to suboptimality of action coordination
- Use quantification to optimize network configuration via alternating optimization

denote it

$C(\{\mathcal{N}_i\}_{i\in\mathcal{N}})$

Introduced distributed optimization algorithm:



Algorithm 1: AlterNAting COordination and Network-Design Algorithm (Anaconda)

Key ideas in making Algorithm 1 scalable and near-optimal:

- Design Action Coordination and Network Design steps to require minimal communications
- Quantify impact of network configuration $\{\mathcal{N}_i\}_{i\in\mathcal{N}}$ to suboptimality of action coordination
- Use quantification $C(\{\mathcal{N}_i\}_{i\in\mathcal{N}})$ to optimize network configuration via alternating optimization

At iteration t:

Stage 1 (ActionCoordination). *Simultaneously* sample action $a_{i,t} \in \mathcal{V}_i$ based on past rewards to solve

$$\max_{a_{i,t} \in \mathcal{V}_i, \forall i \in \mathcal{N}} f(\{a_{i,t}\}_{i \in \mathcal{N}})$$

Stage 2 (NeighborSelection). *Simultaneously* sample neighborhood $\mathcal{N}_{i,t} \subseteq \mathcal{M}_i$ based on past rewards to solve

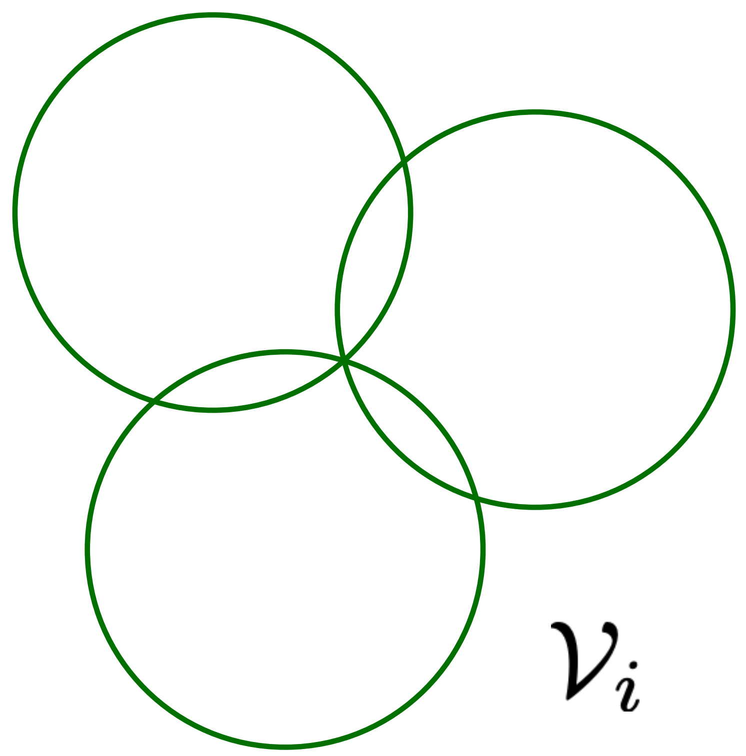
$$\max_{\mathcal{N}_{i,t} \subseteq \mathcal{M}_i, |\mathcal{N}_{i,t}| \leq \alpha_i, \forall i \in \mathcal{N}} \; \max_{a_{i,t} \in \mathcal{V}_i, \forall i \in \mathcal{N}} f(\{a_{i,t}\}_{i \in \mathcal{N}})$$

Given neighborhood $\mathcal{N}_{i,t-1}$ selected in **Stage 2 of time step** $t-1$, each agent i:

1. Compute rewards $r_{a,t-1} = f(a \,|\, \{a_{j,t-1}\}_{j \in \mathcal{N}_{i,t-1}}), \forall a \in \mathcal{V}_i$
2. Update probability distribution $p_{i,t}$ over $\mathcal{V}_i$ using $\{r_{a,t-1}\}_{a \in \mathcal{V}_i}$
3. Sample action $a_{i,t} \in \mathcal{V}_i$ from $p_{i,t}$

marginal gain in f of selecting action a given neighborhood $\mathcal{N}_{i,t-1}$

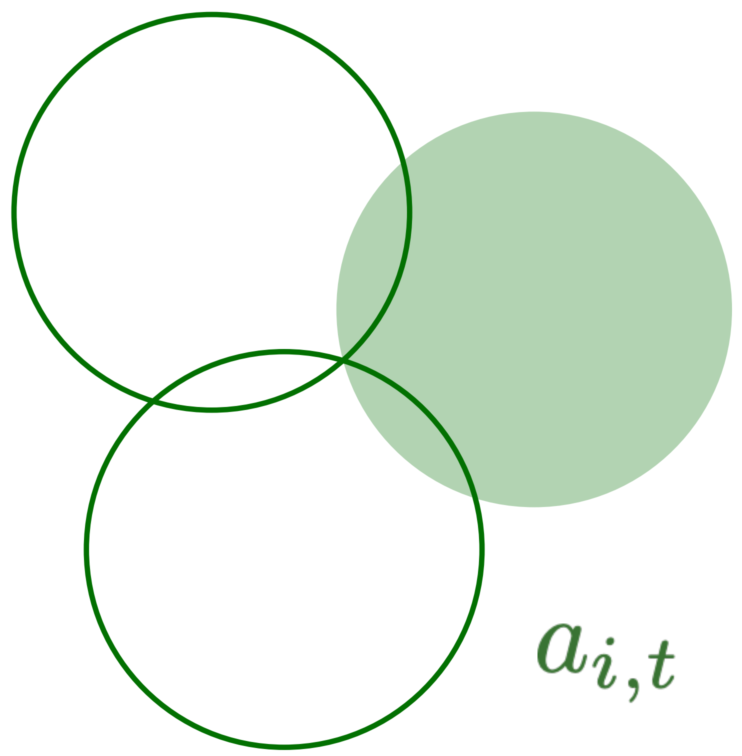
Multiplicative Weights Update[1]



[1] Arora, Hazan, Kale, Theory of Computing, '12

Given neighborhood $\mathcal{N}_{i,t-1}$ selected in **Stage 2 of time step** $t-1$, each agent i:

1. Compute rewards $r_{a,t-1} = f(a \,|\, \{a_{j,t-1}\}_{j \in \mathcal{N}_{i,t-1}}), \forall a \in \mathcal{V}_i$ $\longrightarrow$ marginal gain in f of selecting action a given neighborhood $\mathcal{N}_{i,t-1}$
2. Update probability distribution $p_{i,t}$ over $\mathcal{V}_i$ using $\{r_{a,t-1}\}_{a \in \mathcal{V}_i}$ $\longrightarrow$ **Multiplicative Weights Update**[1]
3. Sample action $a_{i,t} \in \mathcal{V}_i$ from $p_{i,t}$



[1] Arora, Hazan, Kale, Theory of Computing, '12

Suboptimality Performance of ActionCoordination

Definition (Suboptimality Cost of ActionCoordination)

The suboptimality cost due to network decentralization is $C(\{\mathcal{N}_{i,t}\}_{i\,\in\,\mathcal{N}}) = \sum_{i\,\in\,\mathcal{N}} C_{i,t}(\mathcal{N}_{i,t})$, where

$$C_{i,t}(\mathcal{N}_{i,t}) \triangleq f(a_{i,t}\,|\,\{a_{j,t}\}_{j\,\in\,\mathcal{N}_{i,t}}) - f(a_{i,t})$$

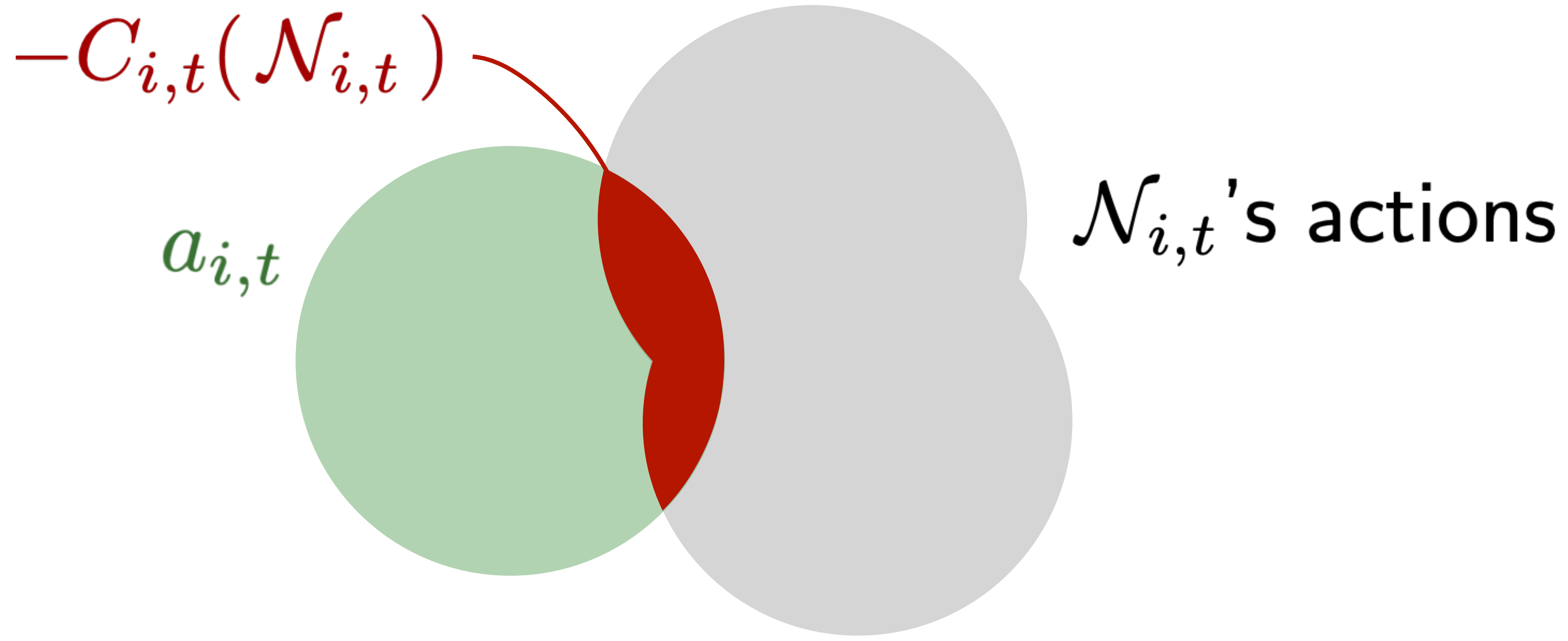
Definition (Suboptimality Cost of ActionCoordination)

The suboptimality cost due to network decentralization is $C(\{\mathcal{N}_{i,t}\}_{i\in\mathcal{N}}) = \sum_{i\in\mathcal{N}} C_{i,t}(\mathcal{N}_{i,t})$, where

$$C_{i,t}(\mathcal{N}_{i,t}) \triangleq f(a_{i,t} \mid \{a_{j,t}\}_{j\in\mathcal{N}_{i,t}}) - f(a_{i,t})$$

Intuition:

$-C_{i,t}$ captures the overlap of i's action and $\mathcal{N}_{i,t}$'s actions:



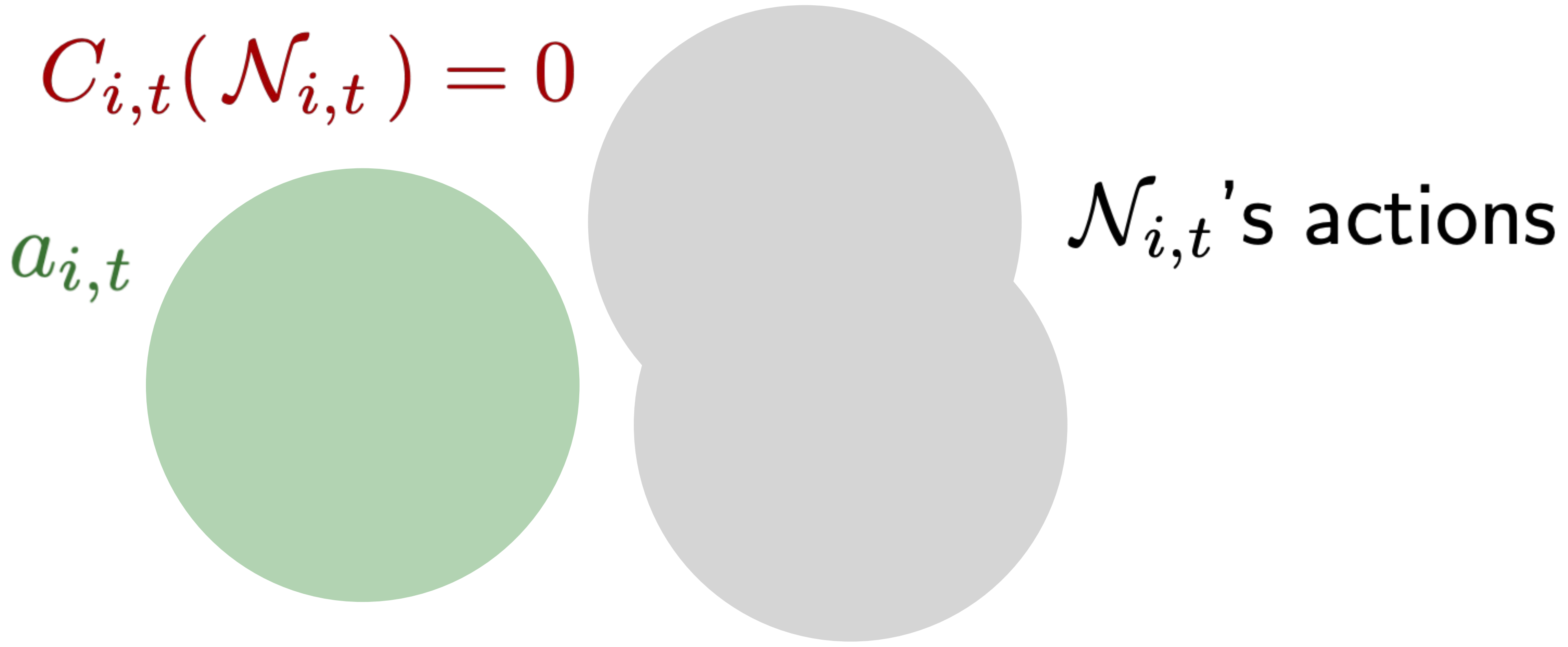
Definition (Suboptimality Cost of ActionCoordination)

The suboptimality cost due to network decentralization is $C(\{\mathcal{N}_{i,t}\}_{i\in\mathcal{N}}) = \sum_{i\in\mathcal{N}} C_{i,t}(\mathcal{N}_{i,t})$, where

$$C_{i,t}(\mathcal{N}_{i,t}) \triangleq f(a_{i,t} \,|\, \{a_{j,t}\}_{j\in\mathcal{N}_{i,t}}) - f(a_{i,t})$$

Intuition:

$-C_{i,t}$ captures the overlap of i's action and $\mathcal{N}_{i,t}$'s actions:



Suboptimality Performance of ActionCoordination

Proposition 1

At each time step t, ActionCoordination instructs each agent $i \in \mathcal{N}$ to select action $a_{i,t}$ such that when $t \geq \tilde{O}(1/\epsilon^2)$, $C_{i,t}(\mathcal{N}_{i,t}\,)$ is minimized in expectation with respect to $a_{i,t}$, i.e.,

$$\mathbb{E}\left[C_{i,t}(\mathcal{N}_{i,t}\,)\right] \leq \min_{a \,\in\, \mathcal{V}_i} \left[f(a \,|\, \{a_{j,t}\}_{j \,\in\, \mathcal{N}_{i,t}}) - f(a)\right] + \epsilon$$

Proposition 1

At each time step t, ActionCoordination instructs each agent $i \in \mathcal{N}$ to select action $a_{i,t}$ such that when $t \geq \tilde{O}(1/\epsilon^2)$, $C_{i,t}(\mathcal{N}_{i,t})$ is minimized in expectation with respect to $a_{i,t}$, i.e.,

$$\mathbb{E}\left[C_{i,t}(\mathcal{N}_{i,t})\right] \leq \min_{a \in \mathcal{V}_i} \left[f(a \,|\, \{a_{j,t}\}_{j \in \mathcal{N}_{i,t}}) - f(a)\right] + \epsilon$$

and that when $t \geq \tilde{O}(|\mathcal{N}|^2/\epsilon^2)$, for all agents' actions $\mathcal{A}_t \triangleq \{a_{i,t}\}_{i \in \mathcal{N}}$,

$$\mathbb{E}\left[f(\mathcal{A}_t)\right] \geq (1-\kappa_f) \left[f(\mathcal{A}^{\text{OPT}}) - \sum_{i \in \mathcal{N}} \mathbb{E}\left[C_{i,t}(\mathcal{N}_{i,t})\right]\right] - \epsilon$$

f's **curvature** $\kappa_f \triangleq 1 - \min_{z \in \mathcal{V}} \frac{f(\mathcal{V}) - f(\mathcal{V} \setminus z)}{f(z)} \in [0, 1]$.
κ_f measures how $\mathcal{V}$'s elements *substitute* each other:

- $\kappa_f = 0 \Leftrightarrow f(\mathcal{A}) = \sum_{z \in \mathcal{A}} f(z)$;
- $\kappa_f = 1 \Leftrightarrow \exists\, z \in \mathcal{V}$ s.t. $f(\mathcal{V}) = f(\mathcal{V} \setminus z)$.

Problem (Neighbor Selection)

At each time step t, each agent $i \in \mathcal{N}$ needs to select neighborhood $\mathcal{N}_{i,t} \subseteq \mathcal{M}_i$ *online* to solve

$$\min_{\mathcal{N}_{i,t} \subseteq \mathcal{M}_i,\, |\mathcal{N}_{i,t}| \leq \alpha_i} C_{i,t}(\mathcal{N}_{i,t}),$$

being able to evaluate $C_{i,t}(\mathcal{M})$, $\forall \mathcal{M} \subseteq \mathcal{M}_i$ only after having selected $\mathcal{M}$ as neighbors and received actions $\{a_{j,t}\}_{j \in \mathcal{M}}$ from them.

Neighbor Selection: A Bandit Supermodular Minimization Approach

Problem (Neighbor Selection)

At each time step t, each agent $i \in \mathcal{N}$ needs to select neighborhood $\mathcal{N}_{i,t} \subseteq \mathcal{M}_i$ *online* to solve

bandit feedback

$$\min_{\mathcal{N}_{i,t} \subseteq \mathcal{M}_i, \, |\mathcal{N}_{i,t}| \leq \alpha_i} C_{i,t}(\mathcal{N}_{i,t}),$$

being able to evaluate $\boxed{C_{i,t}(\mathcal{M}), \; \forall \mathcal{M} \subseteq \mathcal{M}_i}$ only after having selected $\mathcal{M}$ as neighbors and received actions $\{a_{j,t}\}_{j \in \mathcal{M}}$ from them.

Lemma (Monotonicity and Supermodularity of $C_{i,t}$)

Given non-decreasing and 2nd-order submodular function f and action $a_{i,t}$, $C_{i,t}(\mathcal{N}_{i,t})$ is non-increasing and supermodular in $\mathcal{N}_{i,t}$

Neighbor Selection: A Bandit Supermodular Minimization Approach

Problem (Neighbor Selection)

At each time step t, each agent $i \in \mathcal{N}$ needs to select neighborhood $\mathcal{N}_{i,t} \subseteq \mathcal{M}_i$ *online* to solve

bandit feedback

$$\min_{\mathcal{N}_{i,t} \subseteq \mathcal{M}_i,\, |\mathcal{N}_{i,t}| \leq \alpha_i} C_{i,t}(\mathcal{N}_{i,t}),$$

being able to evaluate $\boxed{C_{i,t}(\mathcal{M}),\ \forall \mathcal{M} \subseteq \mathcal{M}_i}$ only after having selected $\mathcal{M}$ as neighbors and received actions $\{a_{j,t}\}_{j \in \mathcal{M}}$ from them.

Lemma (Monotonicity and Supermodularity of $C_{i,t}$)

Given non-decreasing and 2nd-order submodular function f and action $a_{i,t}$, $C_{i,t}(\mathcal{N}_{i,t})$ is non-increasing and supermodular in $\mathcal{N}_{i,t}$

Difficulty: **NP-Hard** to achieve approximation bound better than $\kappa_f^{-1}(1 - e^{-\kappa_f})(\geq 1 - 1/e)$ even when $C_{i,t}(\mathcal{M}),\ \forall \mathcal{M} \subseteq \mathcal{M}_i$ can be evaluated before $\mathcal{M}$ is selected

At iteration t:

Stage 2 (NeighborSelection). *Simultaneously* sample neighborhood $\mathcal{N}_{i,t} \subseteq \mathcal{M}_i$ based on past rewards to solve

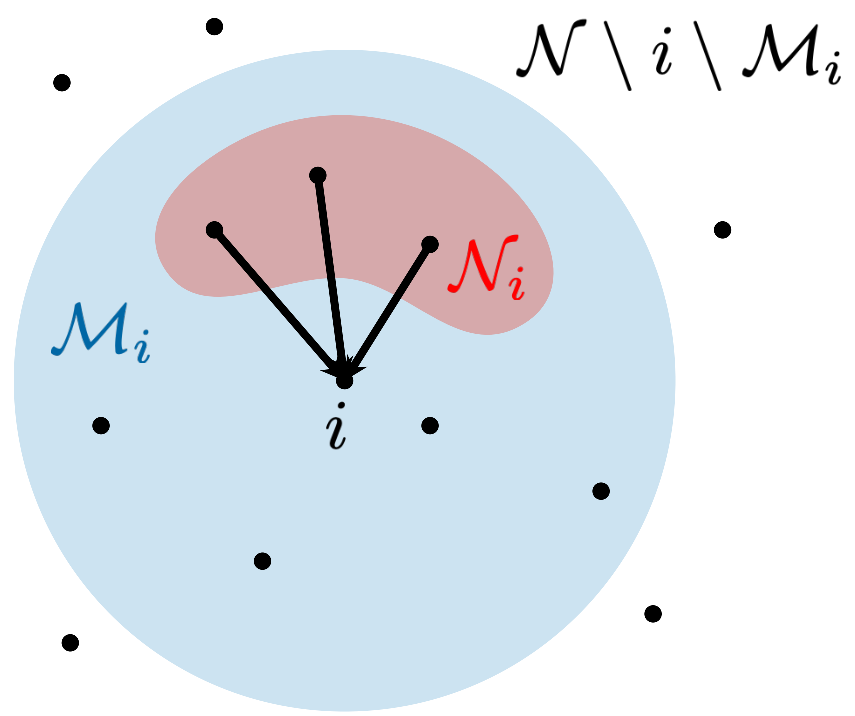
$$\max_{\mathcal{N}_{i,t} \subseteq \mathcal{M}_i,\, |\mathcal{N}_{i,t}| \leq \alpha_i,\, \forall i \in \mathcal{N}} \; \max_{a_{i,t} \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} f(\{a_{i,t}\}_{i \in \mathcal{N}}) \longrightarrow \min_{\mathcal{N}_{i,t} \subseteq \mathcal{M}_i,\, |\mathcal{N}_{i,t}| \leq \alpha_i} C_{i,t}(\mathcal{N}_{i,t})$$

NeighborSelection

Given action $a_{i,t}$ selected in **Stage 1 of time** t and $\mathcal{N}_{i,t} \leftarrow \emptyset$, for $k = 1, \dots, \alpha_i$, each agent i:

1. Update probability distribution $q_{i,t}^{(k)}$ over $\mathcal{M}_i$ using past reward $r_{j_{t-1}^{(k)},t-1}$ $\longrightarrow$ EXP3-IX algorithm[1]
2. Sample k-th neighbor $j_t^{(k)} \in \mathcal{M}_i$ from $q_{i,t}^{(k)}$ and receive its action $a_{j_t^{(k)},t}$
3. Compute reward $r_{j_t^{(k)},t} = C_{i,t}(\mathcal{N}_{i,t}) - C_{i,t}(\mathcal{N}_{i,t} \cup j_t^{(k)})$ and $\mathcal{N}_{i,t} \leftarrow \mathcal{N}_{i,t} \cup j_t^{(k)}$

marginal loss in $C_{i,t}$ of adding neighbor $j_t^{(k)}$ to neighborhood $\mathcal{N}_{i,t}$



[1] Neu, NeurIPS, '15

Suboptimality Performance of NeighborSelection

Proposition

At each time step t, NeighborSelection instructs each agent $i \in \mathcal{N}$ to select neighborhood $\mathcal{N}_{i,t}$ such that when $t \geq \tilde{O}(|\mathcal{M}_i|\, \alpha_i^2 \,/\, \epsilon^2)$,

$$\mathbb{E}\left[C_{i,t}(\mathcal{N}_{i,t})\right] \leq \kappa_f^{-1}(1 - e^{-\kappa_f})\, \mathbb{E}\left[C_{i,t}(\mathcal{N}_i^\star)\right] + \epsilon$$

Suboptimality Performance of NeighborSelection

Proposition

At each time step t, NeighborSelection instructs each agent $i \in \mathcal{N}$ to select neighborhood $\mathcal{N}_{i,t}$ such that when $t \geq \tilde{O}(|\mathcal{M}_i|\, \alpha_i^2 \,/\, \epsilon^2)$,

$$\mathbb{E}\left[C_{i,t}(\mathcal{N}_{i,t})\right] \leq \kappa_f^{-1}(1 - e^{-\kappa_f})\, \mathbb{E}\left[C_{i,t}(\mathcal{N}_i^{\star})\right] + \epsilon$$

$\mathcal{N}_i^{\star}$ is agent i's optimal neighborhood given actions $\{a_{i,t}\}_{t \in [T]}$ selected by ActionCoordination during horizon T:

$$\mathcal{N}_i^{\star} \in \arg \min_{\mathcal{N}_{i,t} \subseteq \mathcal{M}_i,\, |\mathcal{N}_{i,t}| \leq \alpha_i} \mathbb{E}\left[C_{i,t}(\mathcal{N}_{i,t})\right]$$

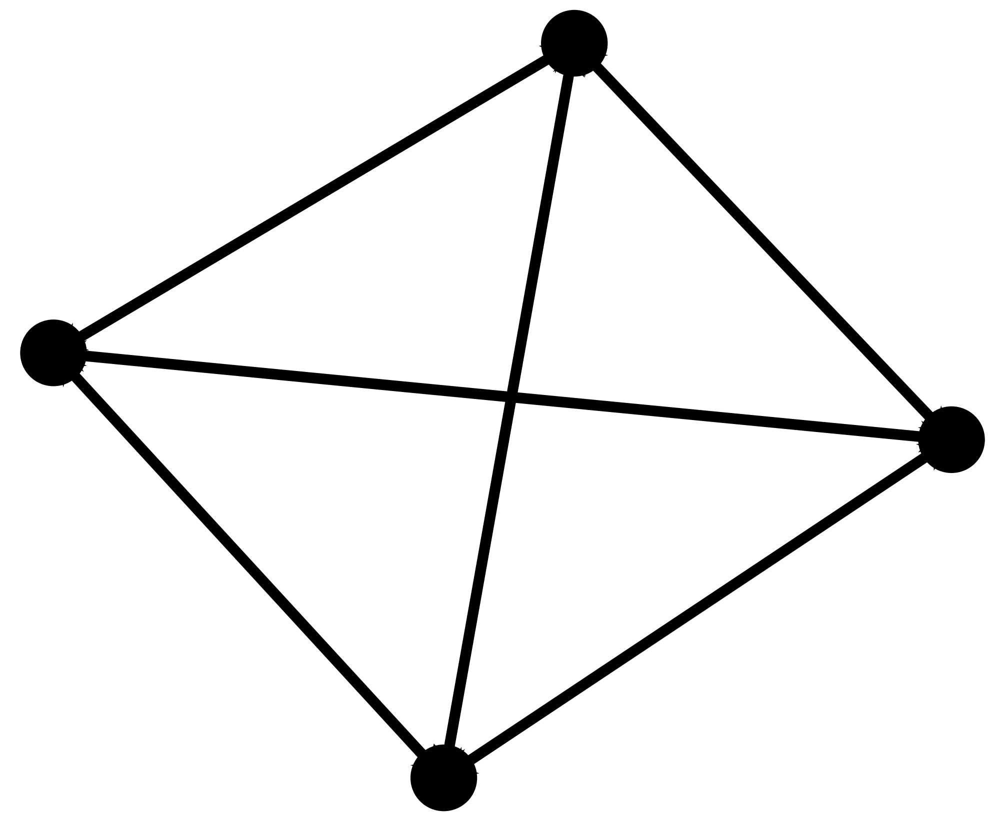
Theorem 1 (Fully Centralized Networks)

At each time step t, Anaconda instructs each agent $i \in \mathcal{N}$ to select action $a_{i,t}$ and neighborhood $\mathcal{N}_{i,t}$ such that

- If the emergent network is *fully centralized* with $\mathcal{N}_{i,t} \equiv \mathcal{N} \setminus i$, when $t \geq \tilde{O}(|\mathcal{N}|^2 \,/\, \epsilon^2)$,

$$\mathbb{E}\left[f(\mathcal{A}_t)\right] \geq \frac{1}{1+\kappa_f} f(\mathcal{A}^{\mathsf{OPT}}) - \epsilon$$

Match the tight bound of Sequential Greedy [Conforti and Cornoejols, '78]



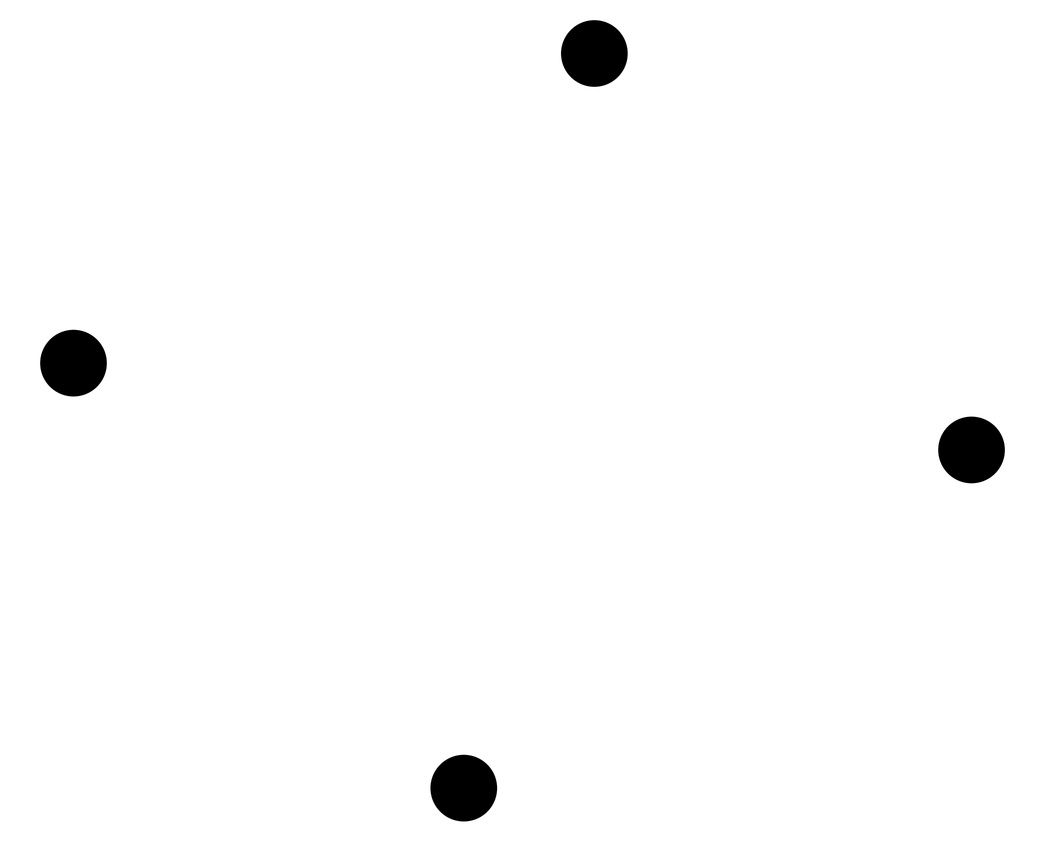
No need to select neighbors since every agent has enough bandwidth budget to hear all others

Theorem 1 (Fully Disconnected Networks)

- If the emergent network is *fully decentralized* with $\mathcal{N}_{i,t} \equiv \emptyset$, when $t \geq \tilde{O}(|\mathcal{N}|^2 \,/\, \epsilon^2)$,

$$\mathbb{E}\left[f(\mathcal{A}_t)\right] \geq (1 - \kappa_f)\, f(\mathcal{A}^{\mathsf{OPT}}) - \epsilon$$

Match the bound [Sviridenko et al. '17]



No agent can hear others so all actions are myopically selected

Theorem 1 (Arbitrary Decentralized Network)

- If the emergent network is *anything in between*, when $t \geq \tilde{O}(|\bar{\mathcal{M}}|\,|\mathcal{N}|^2\,\bar{\alpha}^2\,/\,\epsilon^2)$,

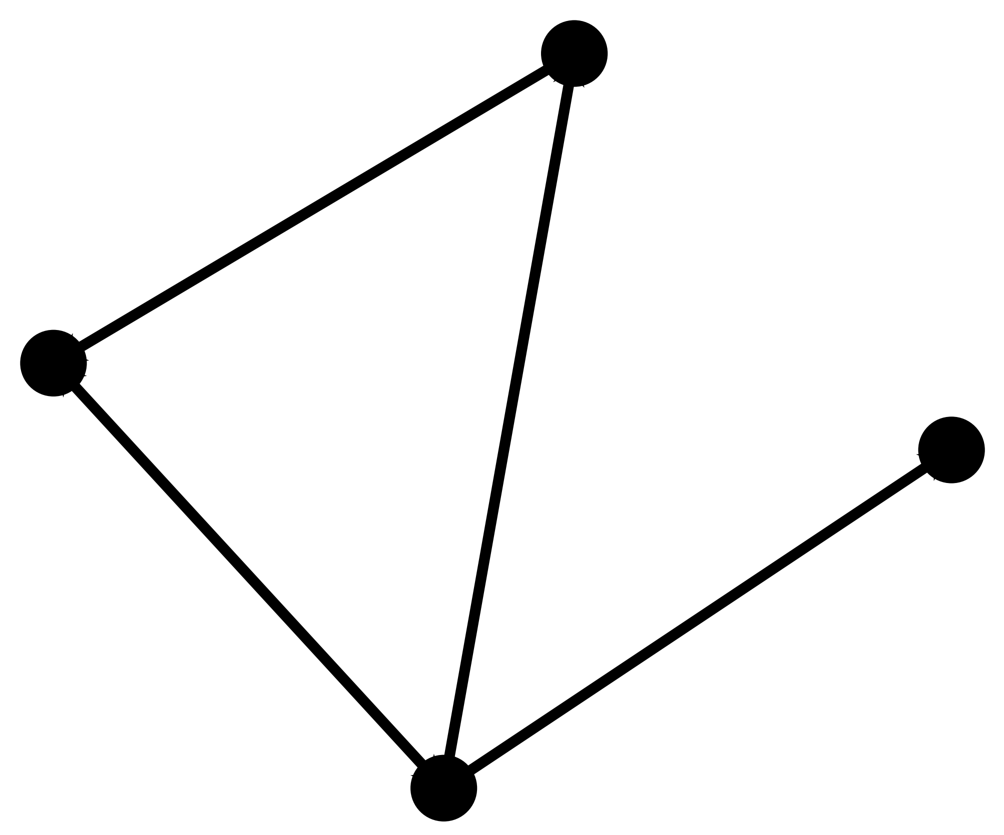
$$\mathbb{E}\left[f(\mathcal{A}_t)\right] \geq (1-\kappa_f)\, f(\mathcal{A}^{\mathsf{OPT}}) - \frac{1-\kappa_f}{\kappa_f}(1-e^{-\kappa_f}) \sum_{i\in\mathcal{N}} \mathbb{E}\left[C_{i,t}(\mathcal{N}_i^{\star})\right] - \epsilon$$

$$\geq \beta\, f(\mathcal{A}^{\mathsf{OPT}}), \quad \beta \in \left[1-\kappa_f, \frac{1-\kappa_f}{1-\kappa_f(1-e^{-\kappa_f})}\right]$$

$$|\bar{\mathcal{M}}| = \max_{i\in\mathcal{N}} |\mathcal{M}_i|$$

$$\bar{\alpha} = \max_{i\in\mathcal{N}} \alpha_i$$

β increases with larger α_i and "better" $\mathcal{M}_i$ [Xu and Tzoumas, arXiv '24]



Network that is not fully connected emerges after agents select neighbors individually

Decision Speed of Anaconda

Theorem 2

Anaconda terminates in $O\left[(\tau_f\, \overline{|\mathcal{V}_i| + \alpha_i} + \tau_c)\, |\bar{\mathcal{M}}|\, |\mathcal{N}|^2 / \epsilon\right]$ time.

function evaluation delay τ_f

communication delay τ_c

$$\overline{|\mathcal{V}_i| + \alpha_i} = \max_{i \in \mathcal{N}} (|\mathcal{V}_i| + \alpha_i)$$

Decision Speed of Anaconda

Theorem 2

Anaconda terminates in $O\left[\left(\tau_f\,\overline{|\mathcal{V}_i| + \alpha_i} + \tau_c\right)|\bar{\mathcal{M}}|\,|\mathcal{N}|^2 / \epsilon\right]$ time.

Comparison to SOTA decision time: In sparse directed networks where $|\mathcal{M}_i| = o(|\mathcal{N}|)$,

- Anaconda: $O(|\mathcal{N}|^2 / \epsilon)$
- Sequential Greedy with depth-first search:[1] $O(|\mathcal{N}|^3)$

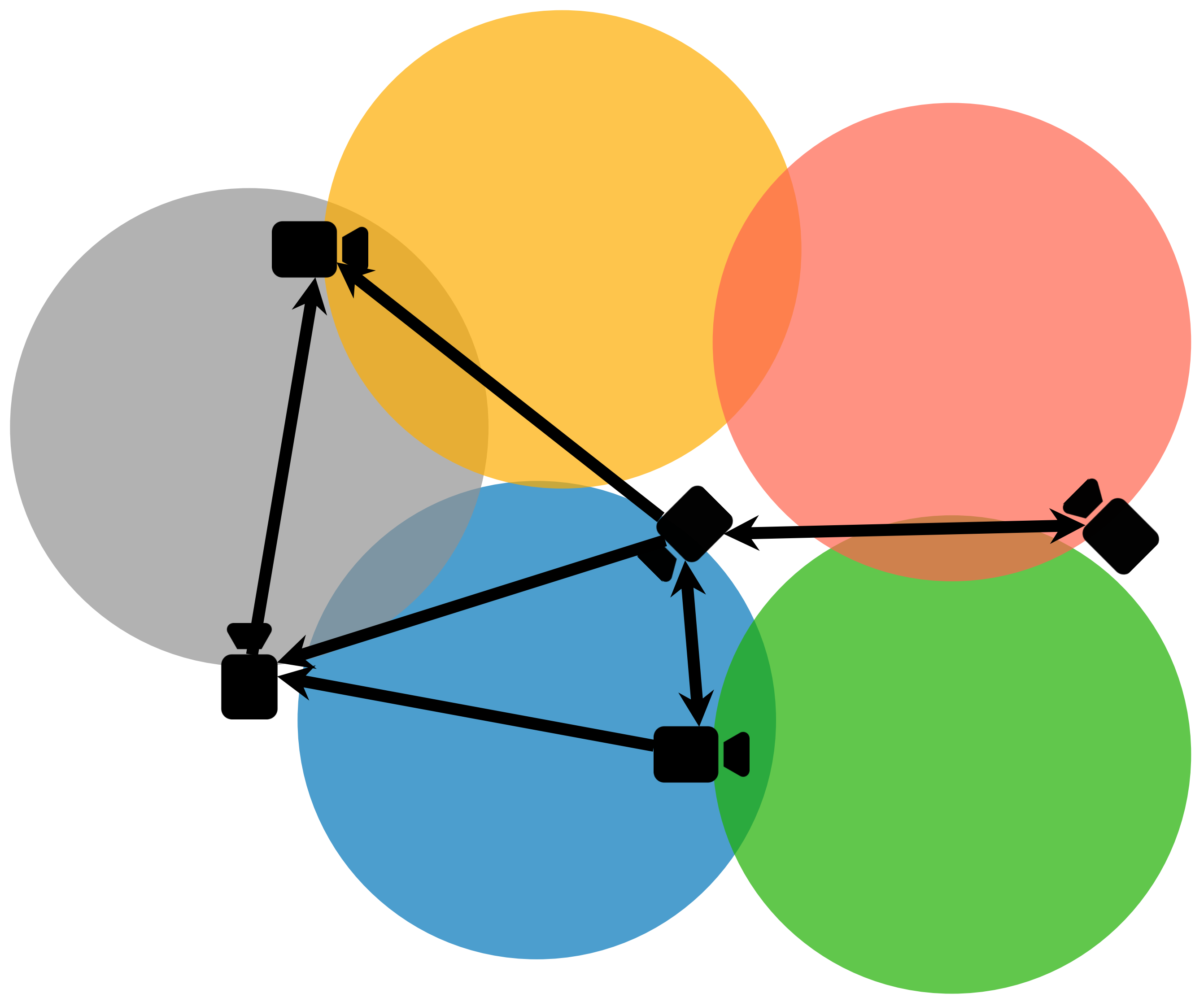
[1]Konda, Grimsman, Marden, ACC '22

Agents: 60 downward-facing cameras $\mathcal{N}$ randomly located above a 100×100 static environment, each with a circular FOV of radius 7

Actions: Directions $\mathcal{V}_i$ to place FOV (i has no knowledge of $\mathcal{V}_j$, $\forall j \neq i$)

Communication: Each camera $i \in \mathcal{N}$ can receive images of FOV from other cameras within its communication range, denoted as $\mathcal{M}_i$

Objective function: Area $f(\{a_{i,t}\}_{i \in \mathcal{N}})$ covered by all cameras' FOV's, each in direction $a_{i,t} \in \mathcal{V}_i$, $\forall t$



Compared algorithms:

a. **Anaconda with different uniform bandwidth budgets**: $|\mathcal{N}_{i,t}| \leq \alpha_i,\ \alpha_i = \alpha \in \{0, 1, 3, 5\},\ \forall i \in \mathcal{N}$

b. **DFS-SG** [Konda et al., ACC '22]: $\mathcal{N}_{i,t} \equiv \mathcal{M}_i$

- Sequential Greedy [Fisher et al., Math Prog Studies '78] on a pre-defined connected network
- Agents' action-selection order determined by depth-first search (DFS)

Compared algorithms:

a. **Anaconda with different uniform bandwidth budgets**: $|\mathcal{N}_{i,t}| \leq \alpha_i,\ \alpha_i = \alpha \in \{0, 1, 3, 5\},\ \forall i \in \mathcal{N}$

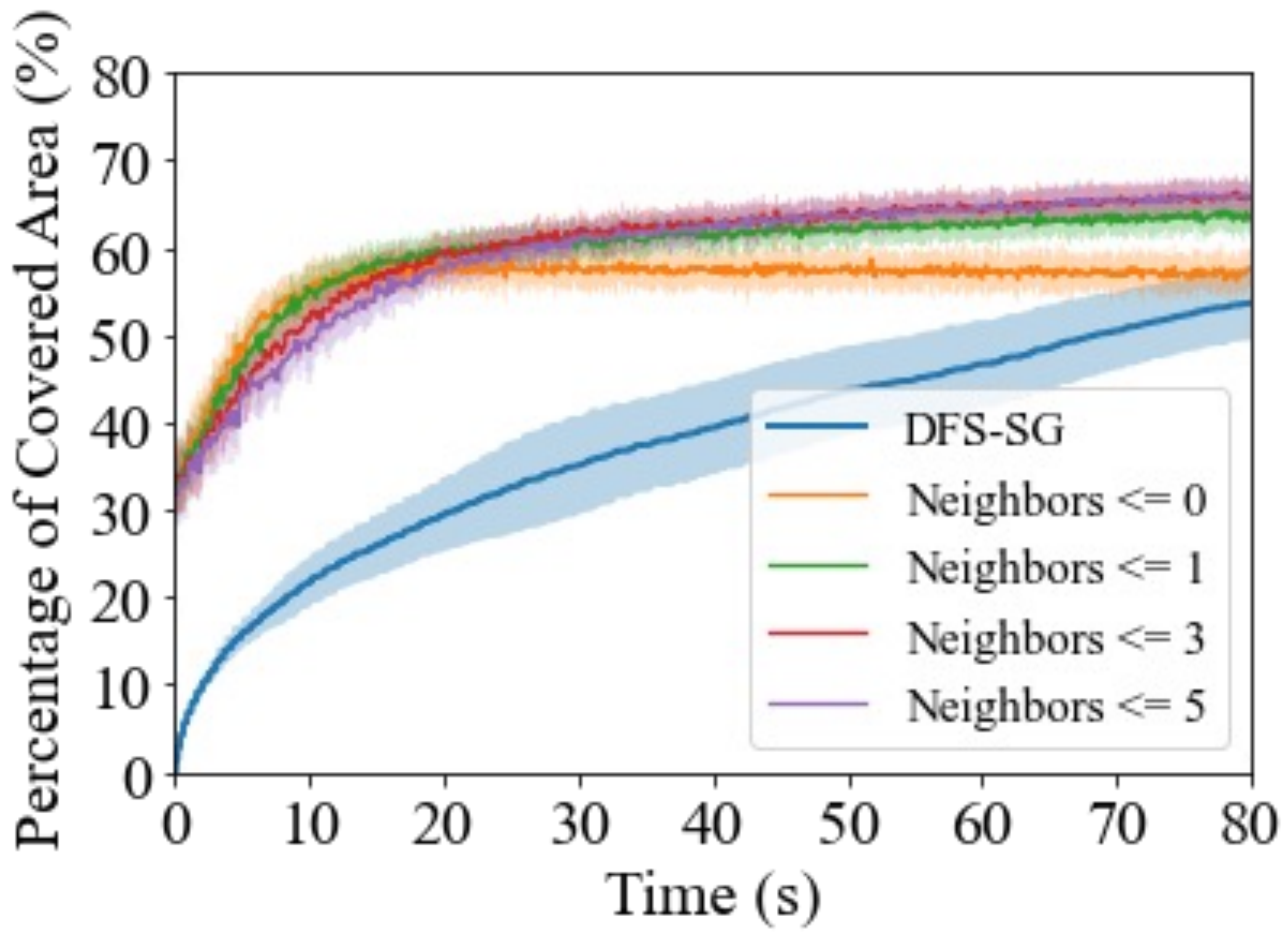
b. **DFS-SG** [Konda et al., ACC '22]: $\mathcal{N}_{i,t} \equiv \mathcal{M}_i$

- Sequential Greedy [Fisher et al., Math Prog Studies '78] on a pre-defined connected network
- Agents' action-selection order determined by depth-first search (DFS)

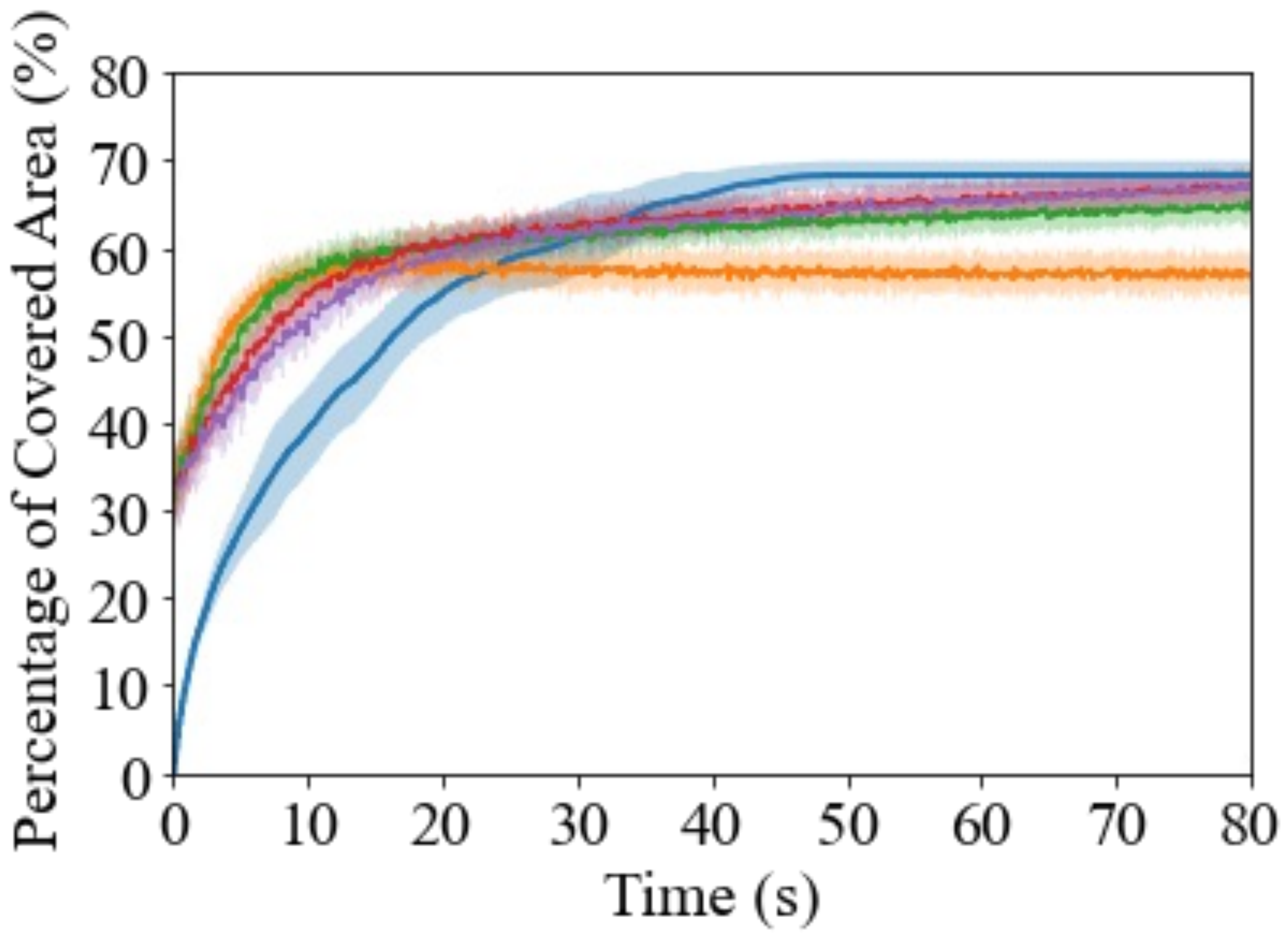
Monte-Carlo simulation setup:

a. **Scenarios with different computation & communication loads**: 30 trials × 5 algorithms × 3 combinations of function evaluation delay τ_f & communication delay τ_c

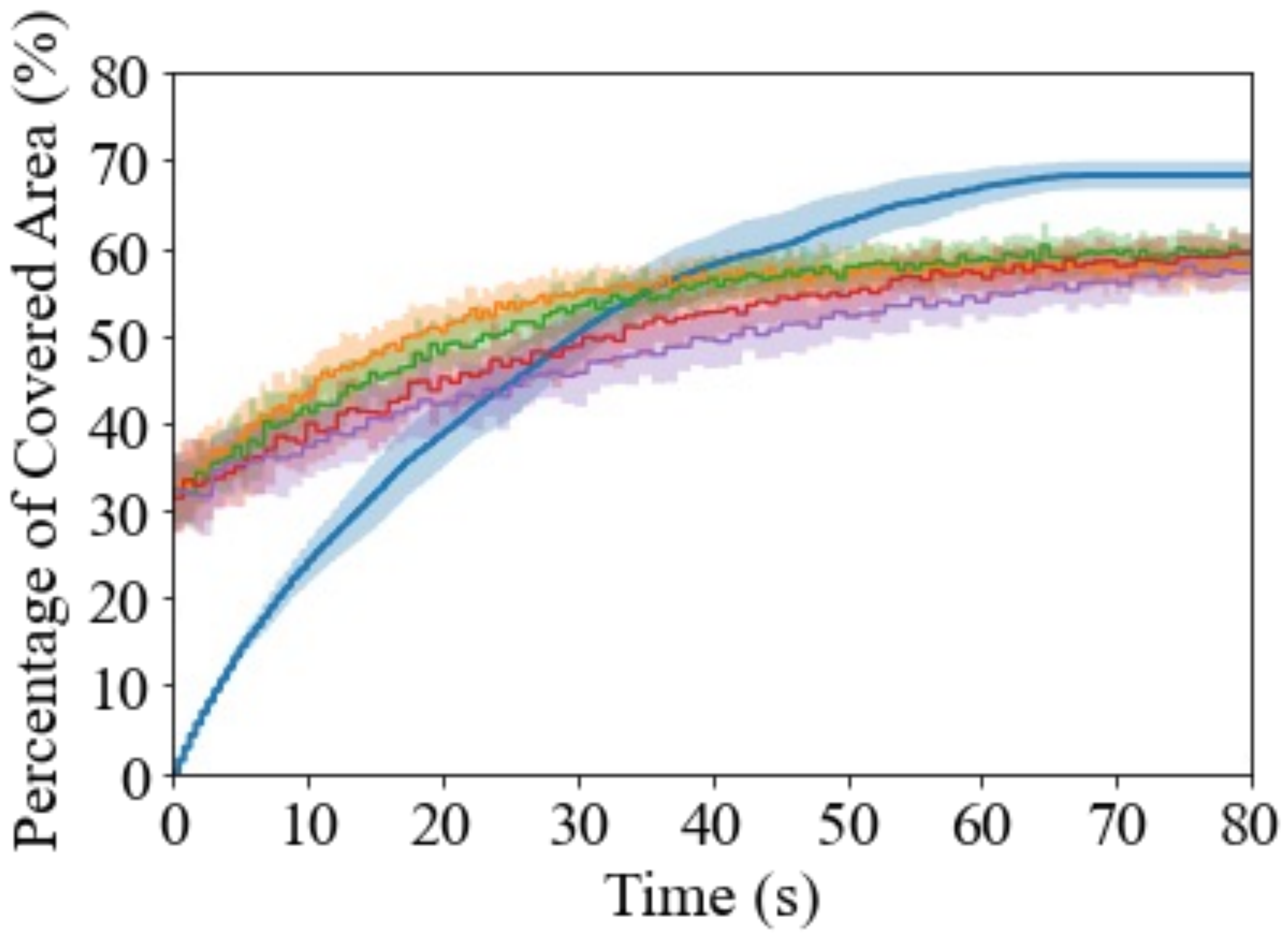
b. **Network connectivity**: for all trials, sample communication range $c_i \in [15, 20], \forall i \in \mathcal{N}$ for network connectivity (required by DFS-SG only)



$\tau_f = 0.01s,\ \tau_c = 0.05s$

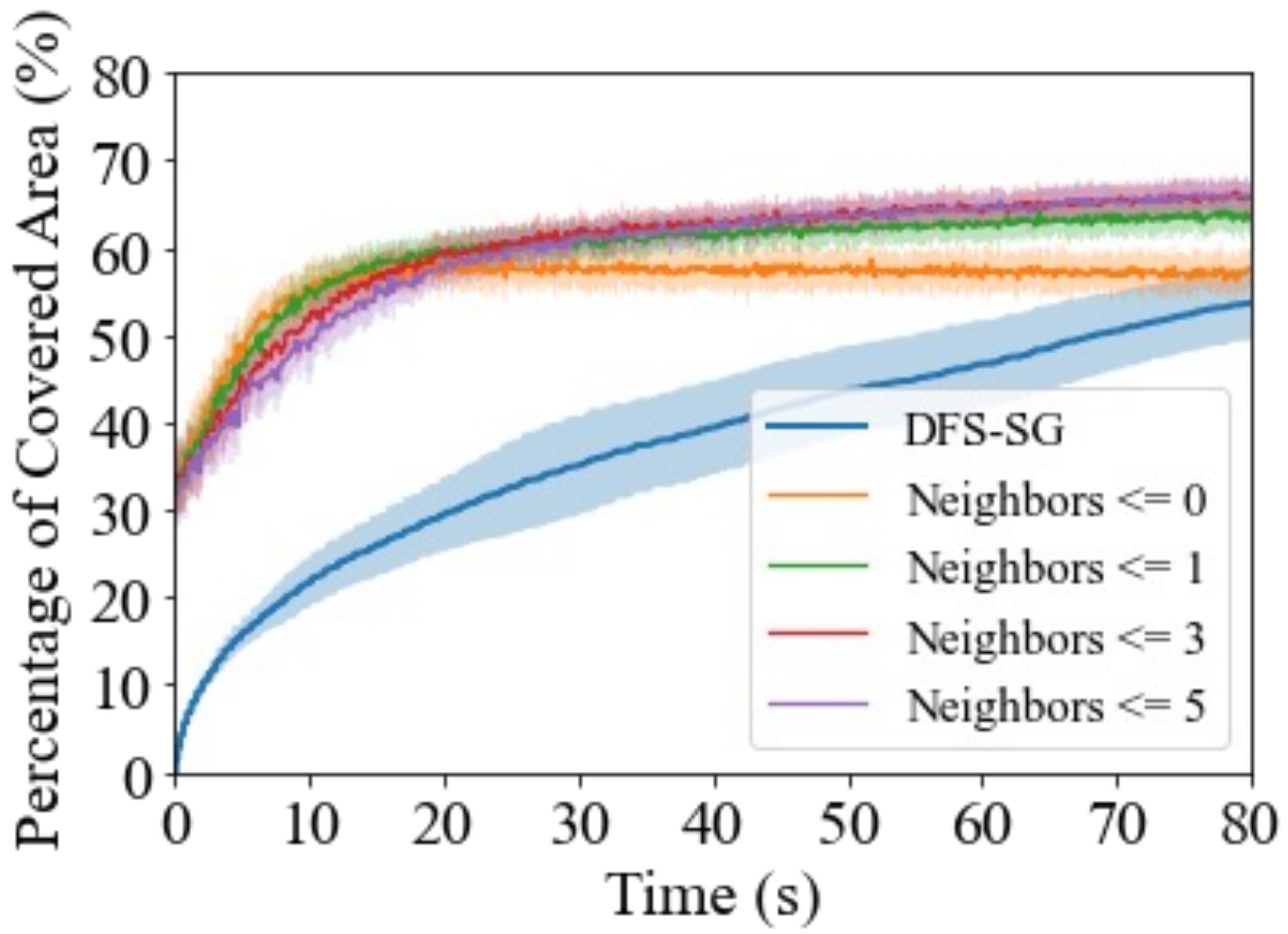


$\tau_f = 0.01s,\ \tau_c = 0.01s$

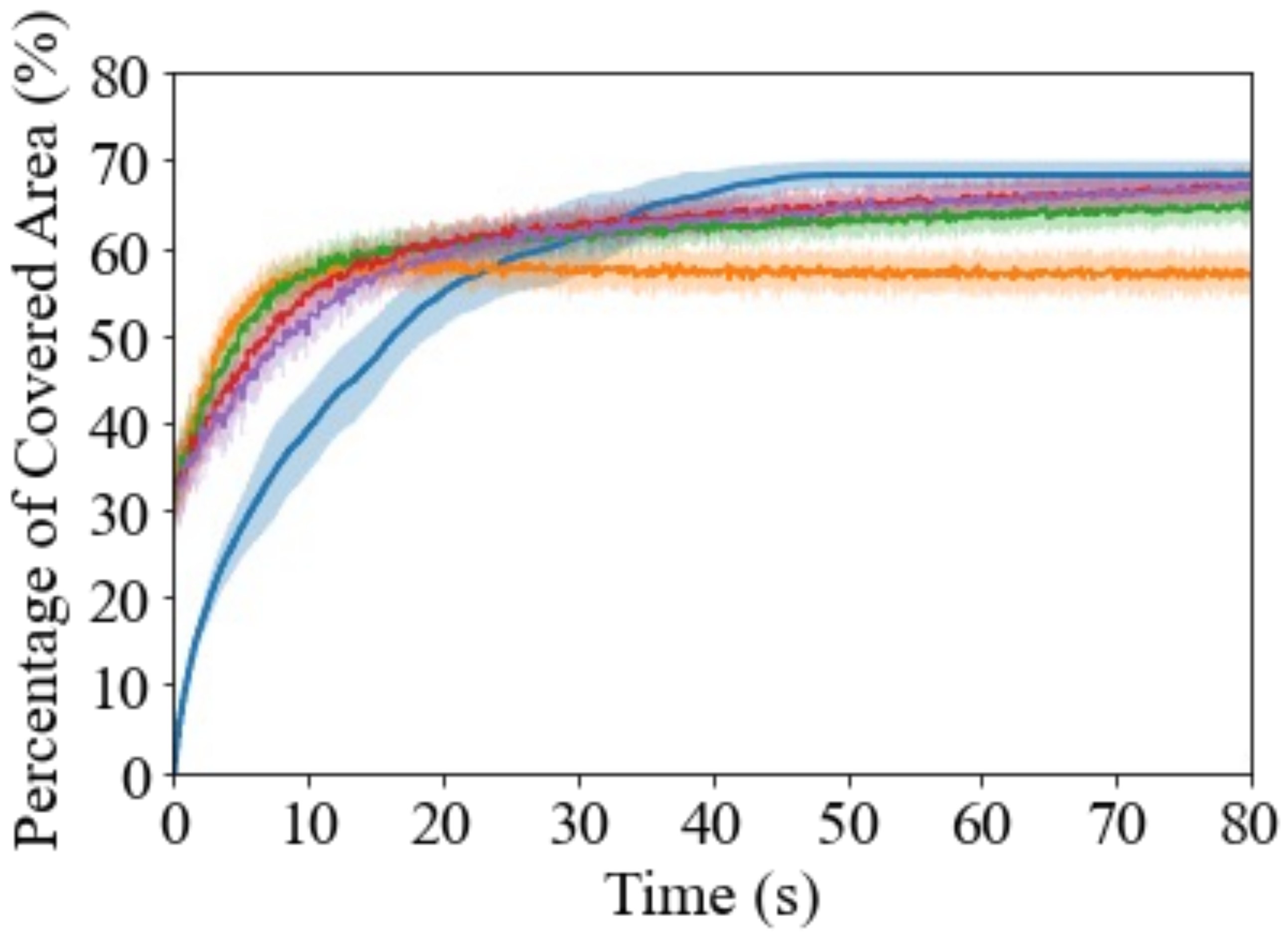


$\tau_f = 0.05s,\ \tau_c = 0.01s$

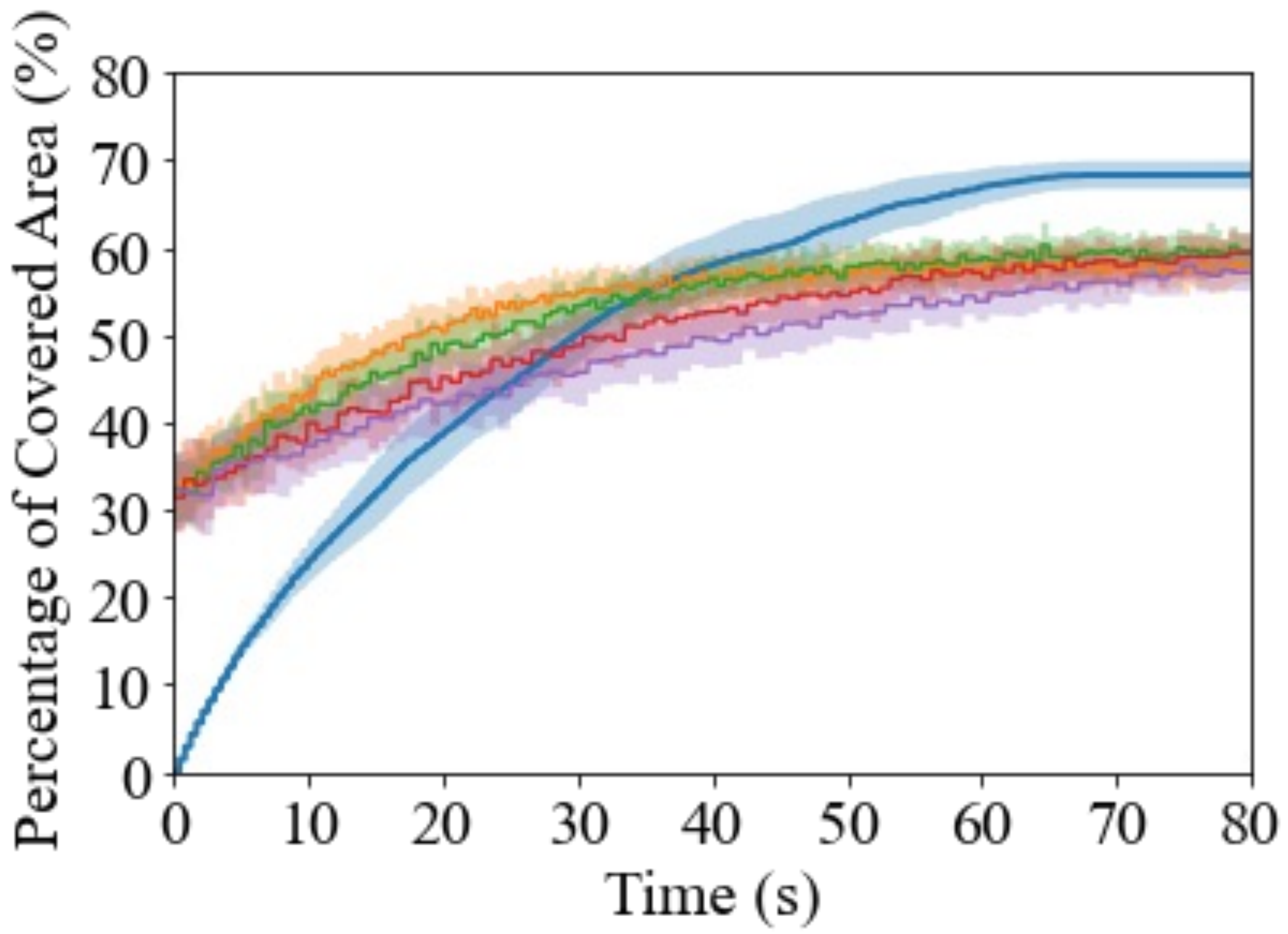
Anaconda with different uniform bandwidth budgets: As bandwidth budget increases, Anaconda's coverage performance increases (Theorem 1) and convergence speed decreases (Proposition 2)



$\tau_f = 0.01s,\ \tau_c = 0.05s$



$\tau_f = 0.01s,\ \tau_c = 0.01s$



$\tau_f = 0.05s,\ \tau_c = 0.01s$

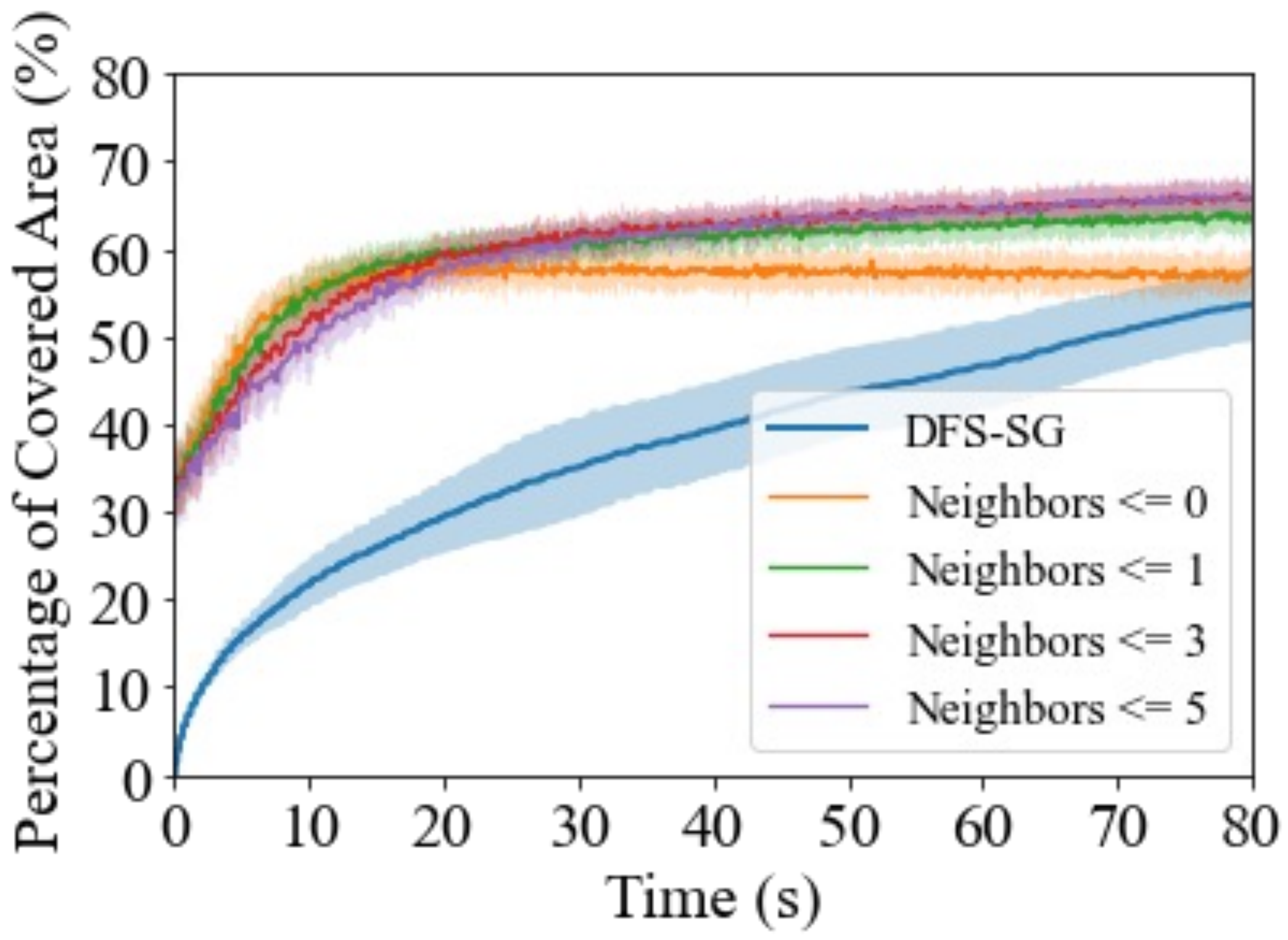
Anaconda with different uniform bandwidth budgets: As bandwidth budget increases, Anaconda's coverage performance increases (Theorem 1) and convergence speed decreases (Proposition 2)

Comparison with DFS-SG: achieve comparable coverage performance after convergence

- **Anaconda starts with higher coverage performance**: cameras all select FOV's from the start per Anaconda but sequentially per DFS-SG
- **Anaconda's coverage performance converges faster when communication delay τ_c is high**: e.g., when $\tau_c > \tau_f$

Summary and Extensions

- Introduce $\max_{\mathcal{N}_i \subseteq \mathcal{M}_i,\, |\mathcal{N}_i| \leq \alpha_i,\, \forall i \in \mathcal{N}} \; \max_{a_i \in \mathcal{V}_i,\, \forall i \in \mathcal{N}} \; f(\{a_i\}_{i \in \mathcal{N}})$
- Provide distributed alternating optimization algorithm
  - Neighborhood self-configuration
  - Quantify impact of network configuration to suboptimality
  - Scalability vs. optimality trade-off
- One-order faster than SOTA when accounting for inter-agent communication delay



Extensions:
- Unknown a priori f
- Linear decision time
- Tight approximation bound