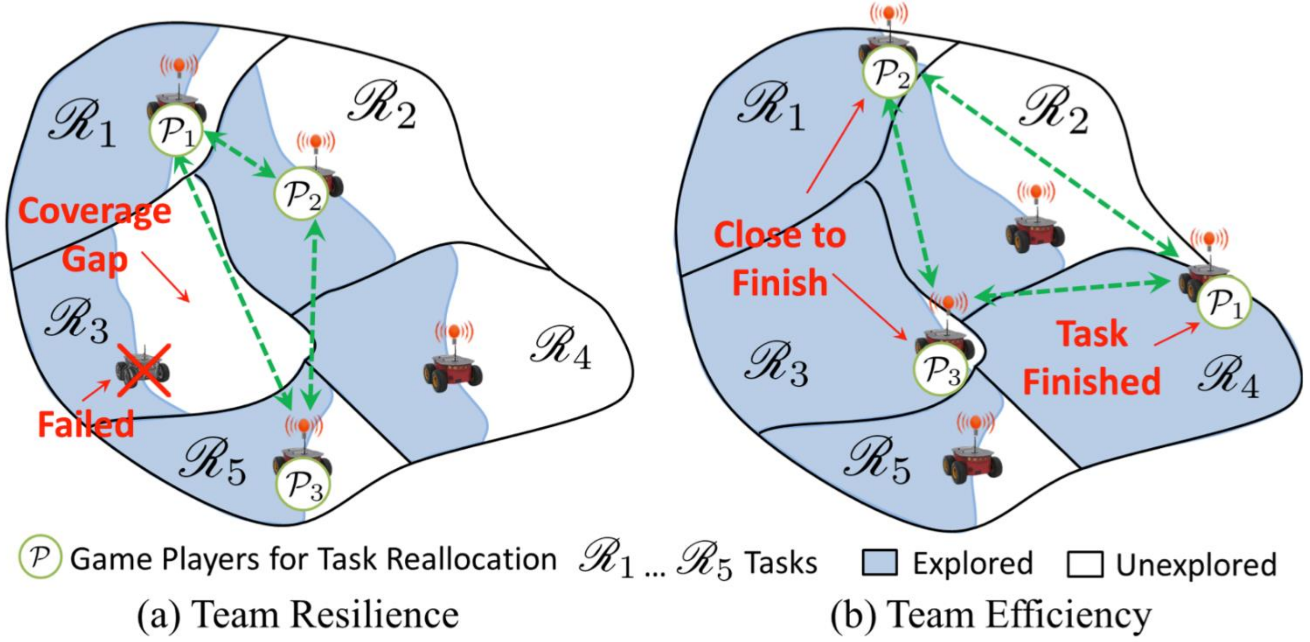
# CARE: *C*ooperative *A*utonomy for *R*esilience and *E*fficiency of Robot Teams for Complete Coverage in Unknown Environment

Authors: Junnan Song and Shalabh Gupta

❖ **Objective:** Develop a multi-robot resilient and efficient algorithm for complete coverage in unknown environment.

❖ **Challenges:**

- Scalability: distributed vs. centralized
- Dynamically changing conditions
- Resilience: complete coverage under failures
- Efficiency: prevent robot idling
- Optimization factors during task reallocations:
  - Task worth (e.g., unfound targets)
  - Probabilities of success of the available robots in finishing the contested tasks
- Connection between local and global objectives: the local optimization must not only benefit the involved robots but also the whole team
- Real-time execution




Requirements for resilience and efficiency

J. Song and S. Gupta, "CARE: Cooperative Autonomy for Resilience and Efficiency of Robot Teams for Complete Coverage of Unknown Environments under Robot Failures", Autonomous Robots, Vol .44, pp. 647-671, 2020.

# The CARE Algorithm

## The Autonomous Vehicle and the Search Area

❖ **The Robot Team $V = \{v_\ell, \ell = 1, \dots N\}$**

1. Localization system, range detector and tasking sensor.
2. Wireless Communication Device
   - Allows periodic information exchange between all pairs of robots. The communication is assumed to be perfect.

❖ **Battery Reliability**

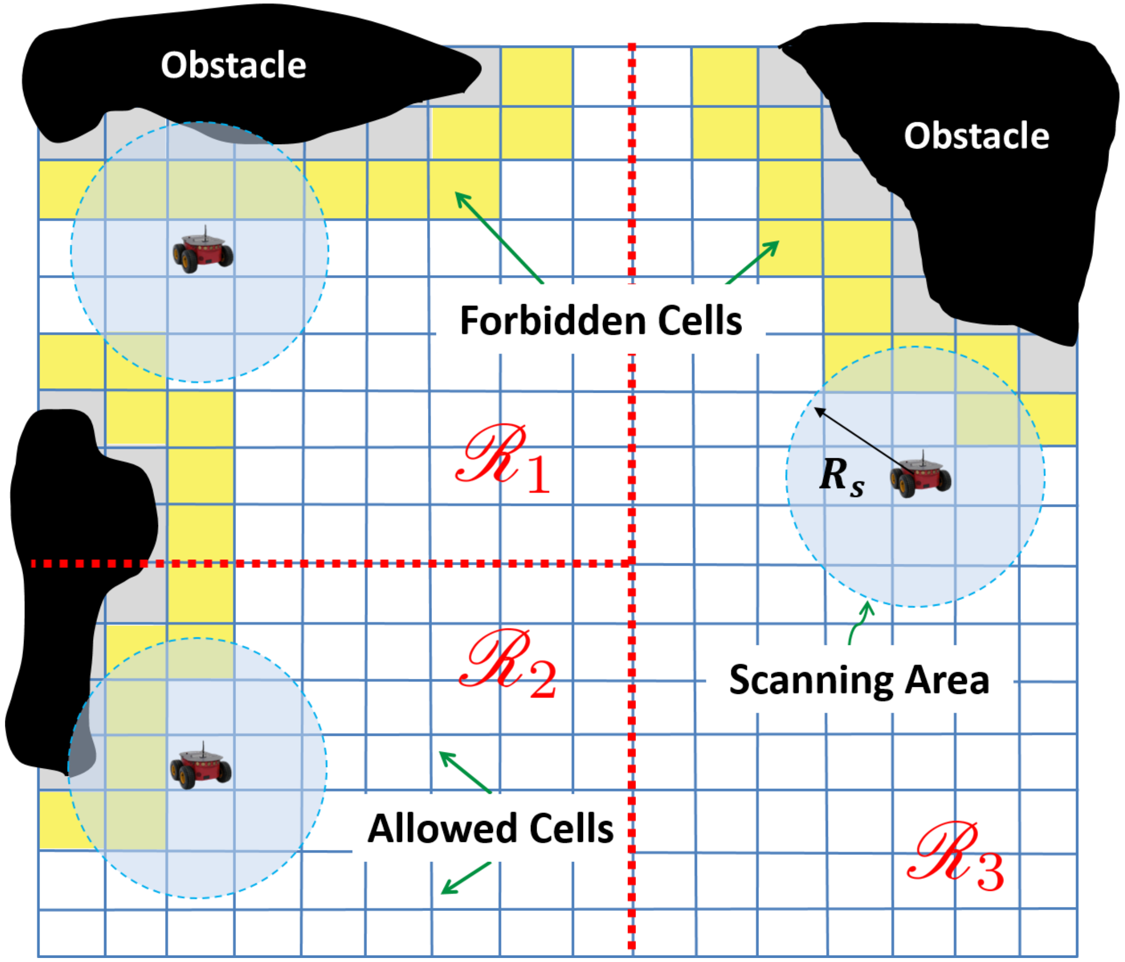
Each robot $v_\ell \in V$ is assumed to carry a battery, whose reliability is computed as[1][2]

$$R_{v_\ell}(t) = \frac{1}{1 + e^{\rho_0 (t - \rho_1)}}$$

where $\rho_0$ and $\rho_1$ are determined based on specific batteries.

❖ **Initial Task Allocation**

The tiling is grouped into $M$ disjoint regions $\{\mathcal{R}_r, r = 1, \dots M\}$, s.t. $\mathcal{R} = \cup_{r=1}^{M} \mathcal{R}_r$. Each robot can work on one task at a time, but one task can be assigned to multiple robots.




An example of the area $\mathcal{R}$ with $M = 3$


[1] A. Islam, A. Alim, C. Hyder, and K. Zubaer,“Diggingtheinnatereliabilityofwireless networked systems,” in *2015 International Conference on Networking Systems and Security*, pp. 1–10, IEEE, 2015.
[2] M. Jongerden and B. Haverkort, “Which battery model to use?,” IET software, vol. 3, no. 6, pp. 445–457, 2009.

# The CARE Algorithm

## Complete Coverage and Performance Metrics

### ❖ Complete Coverage

Let $\epsilon_\ell(k) \in T$ be the $\epsilon$-cell that is visited and explored by robot $v_\ell$ at time $k$. Then the team $V$ is said to achieve complete coverage, if $\exists K \in \mathbb{N}$, s.t. the sequences $\{\epsilon_\ell(k), k = 1, \dots K\}, \forall \ell = 1, \dots N$, satisfy

$$\mathcal{R}(T^a) \subseteq \bigcup_{\ell=1}^{N} \bigcup_{k=1}^{K} \epsilon_\ell(k)$$

### ❖ Performance Metrics

- Coverage Ratio ($\boldsymbol{CR}$):

$$CR = \frac{\left(\bigcup_{\ell=1}^{N} \bigcup_{k=1}^{K} \epsilon_\ell(k)\right) \cap \mathcal{R}(T^a)}{\mathcal{R}(T^a)} \in [0,1]$$

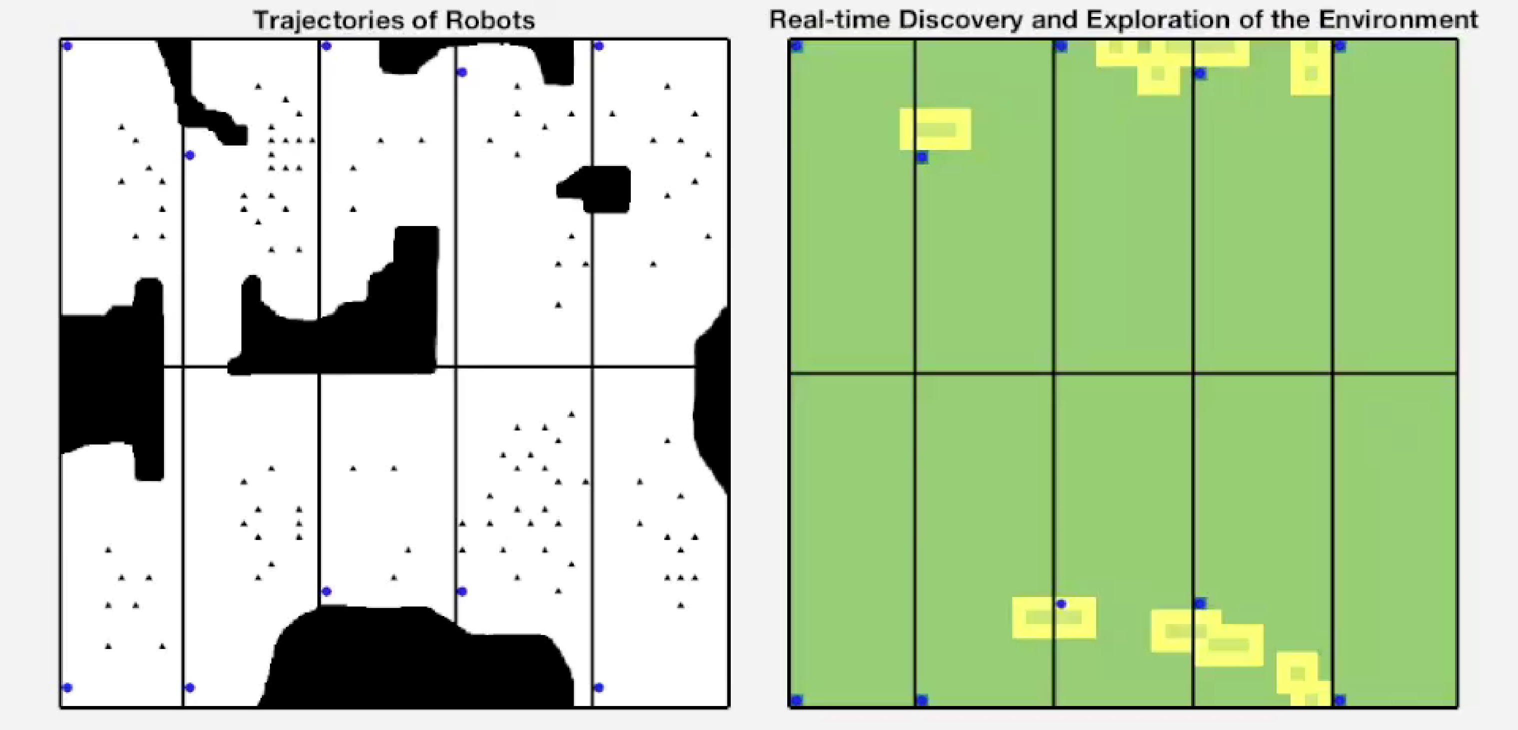
- Coverage Time ($\boldsymbol{CT}$): measured by the last finishing robot
- Remaining Reliability ($\boldsymbol{RR}$): the average remaining reliability of all live robots by the end of the operation
- Number of Targets Found ($\boldsymbol{NoTF}$): total number of targets discovered by the whole team
- Time of Target Discovery ($\boldsymbol{ToTD}$): time for the team to discover a certain percentage of all targets

***Objective***: To achieve $CR = 1$ (even under a few robot failures), while minimizing $CT$ and $ToTD$, and maximizing $RR$ and $NoTF$ .

# The CARE Algorithm

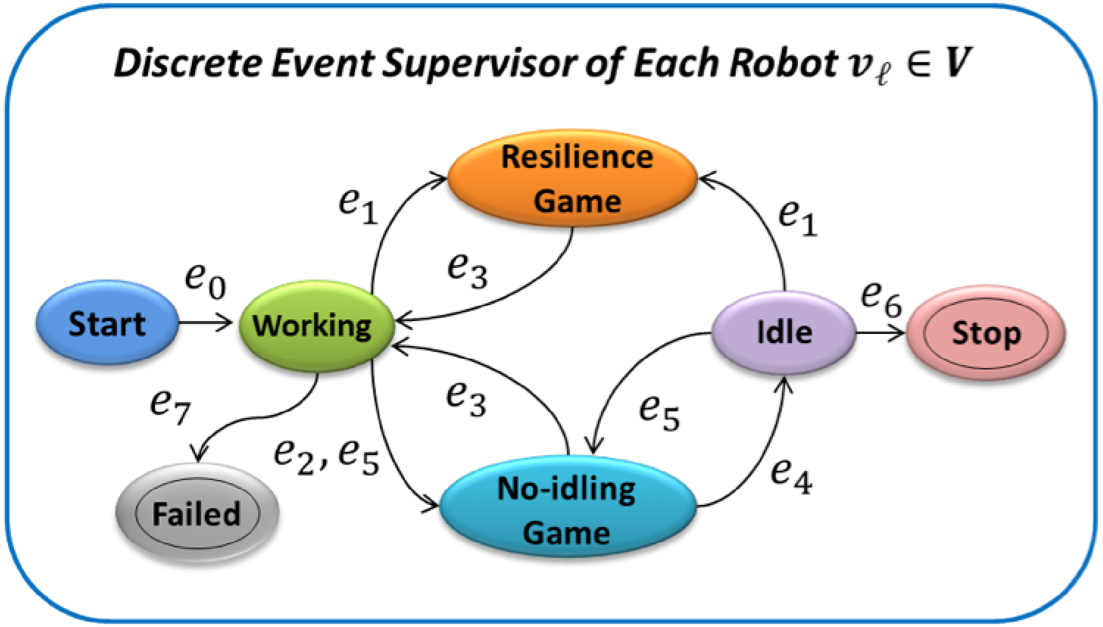
❖ **Features:**

- Distributed multi-robot control
- Complete coverage under failures
- Proactive event-driven task reallocation upon robot failures or robot idling
- Task reallocation considers various optimization factors, including task worth, robot remaining energy, and relative locations.
- Local task reallocation decision is aligned with the global objective of the whole team



# The CARE Algorithm

## Distributed Discrete Event Supervisory Control Structure




| *Events* | *Description* | *Generation Condition* |
|---|---|---|
| $e_0$ | Start exploration | Robot $v_\ell$ is turned on |
| $e_1$ | Neighbor failed | Confirmation of neighbor failure |
| $e_2$ | Own task completed | $n_U(r_c(v_\ell)) = 0$ |
| $e_3$ | Task assigned | Task assigned by the Optimizer |
| $e_4$ | No task assigned | No task assigned by the Optimizer |
| $e_5$ | Neighbor task completed | Some neighbor completes its task and $t_c(r_c(v_\ell)) \leq \eta$ |
| $e_6$ | All tasks completed | $\sum n_U(r) = 0$ |
| $e_7$ | Robot failed | Robot $v_\ell$ is diagnosed as failed |

- **Failure Detection**: use a standard mechanism of heartbeat signals[1], where each robot $v_\ell$ periodically broadcasts heartbeat signals indicating its healthiness, and also listening from others.
- A robot is detected as **failed** if its heartbeat signal cannot be received constantly for a certain period of time.

- $r_c: V \rightarrow \{1, \dots M\}$ is the allocation function that indicates the current task allocations of robots
- $t_c: \{1, \dots M\} \rightarrow [0, \infty)$ is the remaining estimated time to finish a given task by its assigned robots
- $n_U: \{1, \dots M\} \rightarrow \mathbb{N}$ is the number of unexplored cells in a task
- $\eta \in \mathbb{R}^+$: the threshold to define a task as *close-to-finish*.

[1] W. Chen, S. Toueg, and M. Aguilera, "On the quality of service of failure detectors," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 561–580, 2002.

# The CARE Algorithm

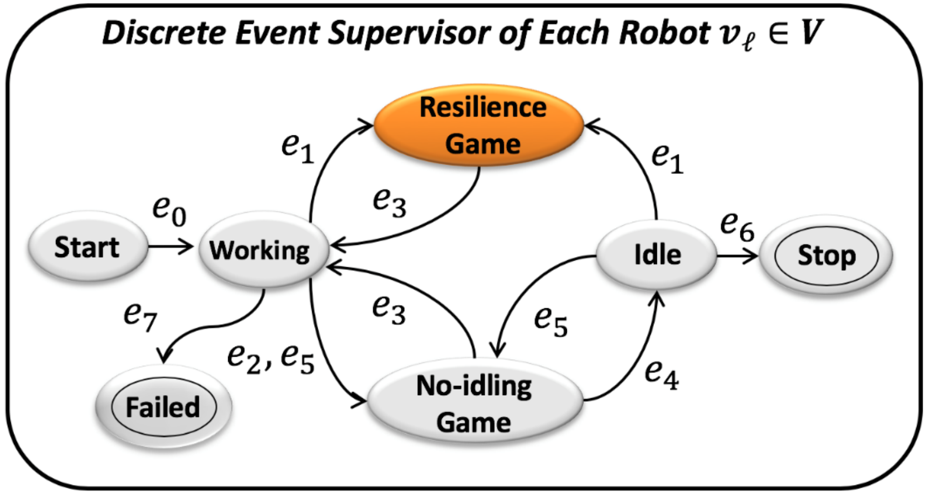
## Game-theoretic Event-driven Distributed Optimizer

**When is it triggered?**

Upon detection of a neighbor failure (i.e., $e_1$)

**What does it do?**

Re-optimize local task allocations to fill coverage gap.

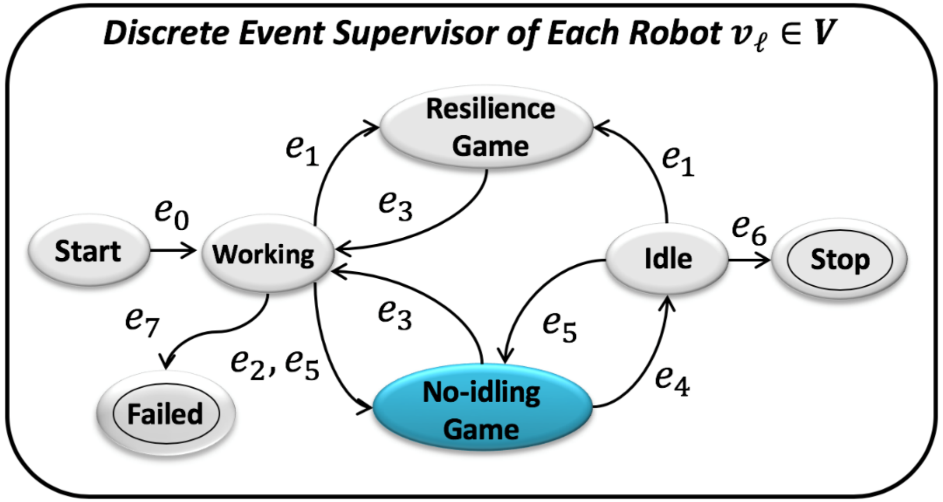



**When is it triggered?**

Upon completion of own task (i.e., $e_2$), or a neighbor's task (i.e., $e_5$)

**What does it do?**

Compute for new tasks (if available).




Task re-allocation is required!

# The CARE Algorithm

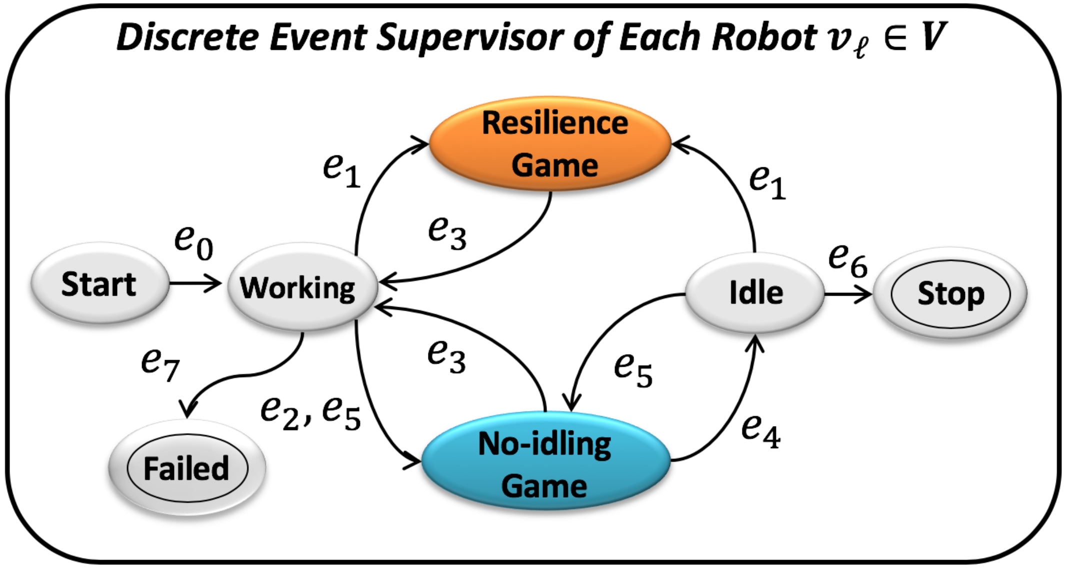
## Game-theoretic Event-driven Distributed Optimizer

❖ **Requirements for Task Reallocation:**

- Distributed framework for scalability
- Optimization over both task worth and robot reliabilities
- Local reallocation decision must also benefit the whole team
- Complete coverage under failures

❖ **Potential Games**

- Fits the distributed framework
- Utility of each player is perfectly aligned with the global potential function $\phi$ for the whole team.
- Existence of solutions: at least one pure Nash Equilibrium exists, which is a maximizer to $\phi$
- Max-Logit can quickly converge to the (sub-)optimal equilibrium



# The CARE Algorithm

## Potential Games



**Game Components:**

- Players $\mathcal{P} = \{\mathcal{P}_i \in V, i = 1, \dots |\mathcal{P}|\}$
  - The available robots to be reallocated.
- Action set $\mathcal{A}_i$ for each player $\mathcal{P}_i$
  - Each action $a_i \in \mathcal{A}_i$ refers to the index of an available task. Also, it has $\tilde{\mathcal{A}} = \mathcal{A}_i = \mathcal{A}_j, \forall i \neq j$.
- The *utility function* for each player $\mathcal{P}_i$, defined as:

$$\mathcal{U}_i : \mathcal{A}_\mathcal{P} \to \mathbb{R}$$

where $\mathcal{A}_\mathcal{P} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_{|\mathcal{P}|}$ denotes the set of joint actions for all players, and the joint action $a_\mathcal{P} = (a_i, a_{-i}) \in \mathcal{A}_\mathcal{P}$ indicates a task reallocation for the team.

- Note: players other than $\mathcal{P}_i$ have joint action $a_{-i} \in \mathcal{A}_{-i}$

**Nash Equilibrium:** A joint action $a_\mathcal{P}^\star = (a_i^\star, a_{-i}^\star) \in \mathcal{A}_\mathcal{P}$ is a pure Nash Equilibrium if:

$$\boxed{\mathcal{U}_i(a_i^\star, a_{-i}^\star) = \max_{a_i \in \tilde{\mathcal{A}}} \mathcal{U}_i(a_i, a_{-i}^\star), \forall \mathcal{P}_i \in \mathcal{P}}$$

**Potential Game:** a potential game $G$ in strategic form is a potential game if and only if a potential function $\phi : \mathcal{A}_\mathcal{P} \to \mathbb{R}$ exists, s.t., $\forall \mathcal{P}_i \in \mathcal{P}$

$$\boxed{\mathcal{U}_i(a_i', a_{-i}) - \mathcal{U}_i(a_i'', a_{-i}) = \phi(a_i', a_{-i}) - \phi(a_i'', a_{-i})}$$

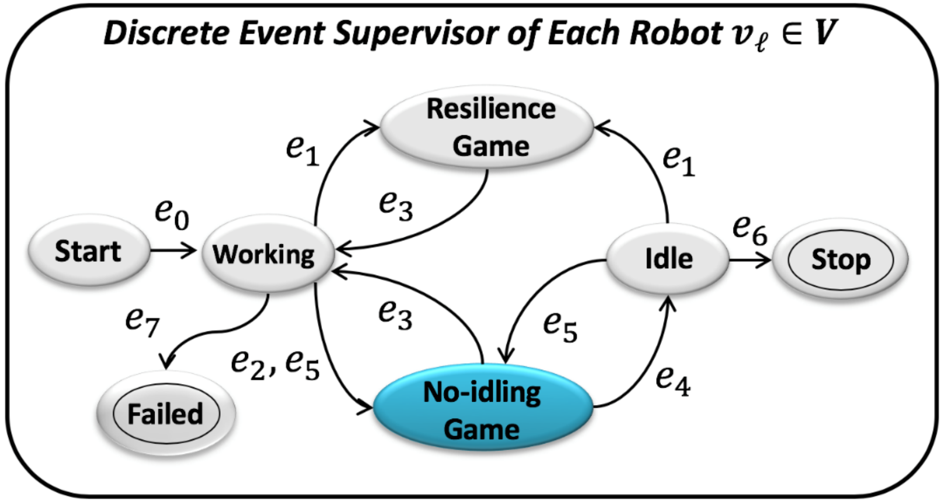
$\forall a_i', a_i'' \in \mathcal{A}_i$ and $a_{-i} \in \mathcal{A}_{-i}$.

# The CARE Algorithm

## Specifics of the No-idling Games

❖ **No-idling Games**

- **Condition**: if some robot $v_{id} \in V$ completes its current task and becomes idle.
- **Game Specifics**: It calls the $\kappa_1$ nearest neighbors $N_{\kappa_1}^{v_{id}}$, that are close to finish their tasks to participate.
  - Players: $\mathcal{P} = \{v_{id}\} \cup \{v_\ell \in N_{\kappa_1}^{v_{id}} : t_c(r_c(v_\ell)) \le \eta\}$
  - Actions: $\mathcal{A}_i = \{r \in \{1, \dots M\} : t_c(r) \ge \gamma \in \mathbb{R}^+\}$, i.e., the incomplete tasks with sufficient work left.



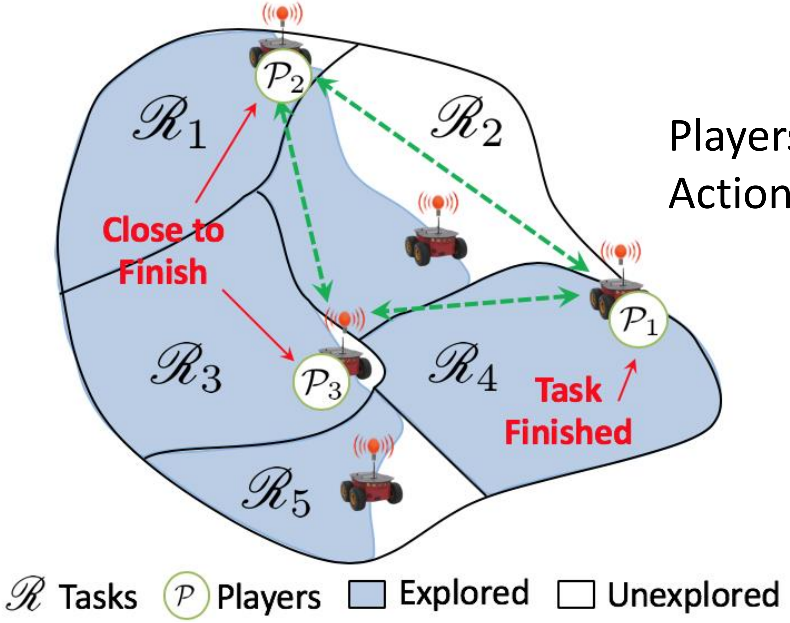




**Figure**. An example of the no-idling game when $\kappa_1 = 3$

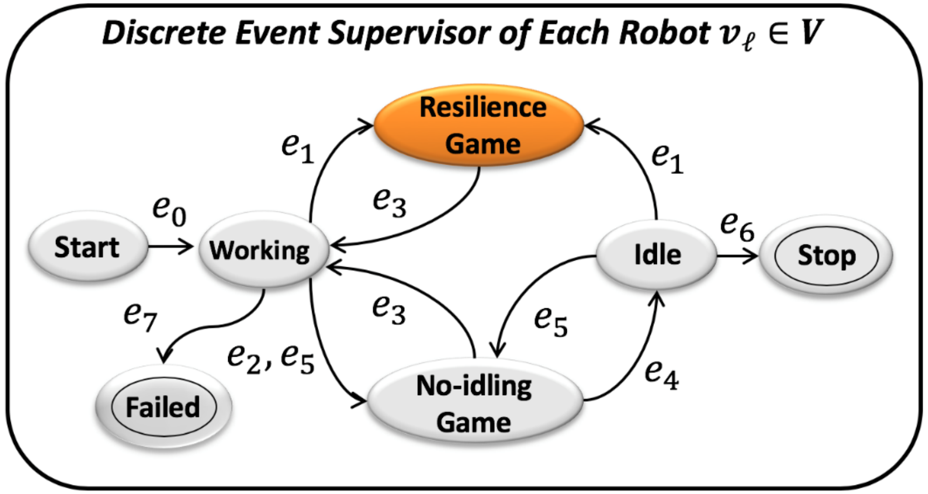
Players: $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$
Actions: $\mathcal{R}_2, \mathcal{R}_5$

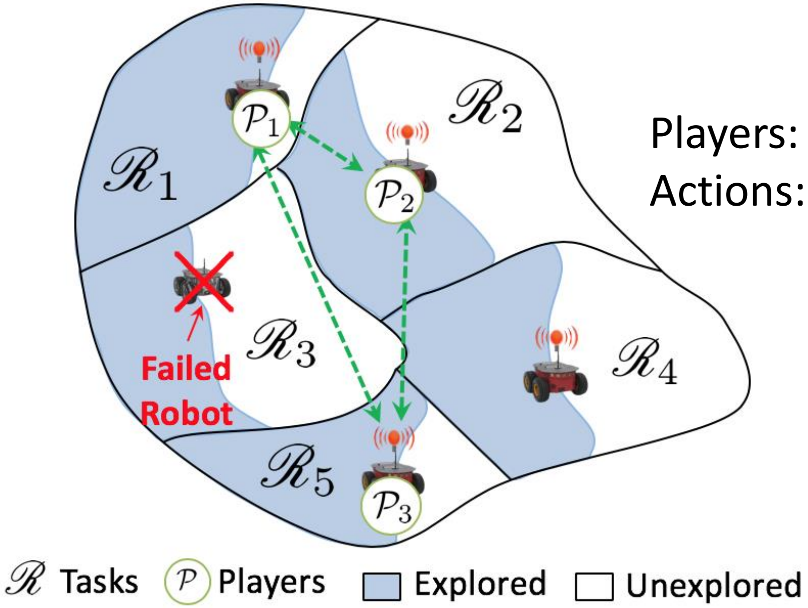
# The CARE Algorithm

## Specifics of the Resilience Games

❖ **Resilience Games**

- **Condition**: if some robot $v_f \in V$ fails.
- **Game Specifics**: the $\kappa_2$ nearest neighbors of $v_f$, $N_{\kappa_2}^{v_f}$, are involved.
  - ➢ Players: $\mathcal{P} = N_{\kappa_2}^{v_f}$
  - ➢ Actions: $\mathcal{A}_i = \{r_c(v_f)\} \cup \{r_c(v_\ell), v_\ell \in N_{\kappa_2}^{v_f} : t_c(r_c(v_\ell)) \geq \eta\}$, i.e., the current tasks of all players and the failed robot.




Players: $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$
Actions: $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_5$

**Figure**. An example of the resilience game when $\kappa_2 = 3$

# The CARE Algorithm

## Potential Function and Utility Function

### ❖ Potential Function $\boldsymbol{\phi(a_{\mathcal{P}})}$

$$\phi(a_{\mathcal{P}}) = \sum_{r \in \tilde{\mathcal{A}}} w_r \left(1 - \prod_{\mathcal{P}_i \in \{\mathcal{P}\}_r} [1 - p_r(\mathcal{P}_i)]\right)$$

Joint probability to successfully finish task $r$

where:

- $w_r$: estimated worth (i.e., undiscovered number of targets) of task $r \in \mathcal{A}_i$
- $\{\mathcal{P}\}_r = \{\mathcal{P}_i \in \mathcal{P}: a_i = r\}$: set of players that choose task $r$ in the joint action $a_{\mathcal{P}}$
- $p_r(\mathcal{P}_i)$: the success probability for player $\mathcal{P}_i$ to finish task $r$

### ❖ Utility Function

The utility function is obtained using *Marginal Contribution*:

$$\mathcal{U}_i(a_i, a_{-i}) = \phi(a_i, a_{-i}) - \phi(\emptyset, a_{-i})$$
$$= w_{a_i} \cdot p_{a_i}(\mathcal{P}_i) \cdot \prod_{\mathcal{P}_j \in \{\mathcal{P}\}_{a_i} \setminus \mathcal{P}_i} [1 - p_{a_i}(\mathcal{P}_i)]$$

Where $\emptyset$ represents the null action.

Joint failure probability for task $r$ by other players

**Physical Meaning of $\boldsymbol{\phi(a_{\mathcal{P}})}$**: the total expected worth for the joint action $a_{\mathcal{P}} \in \mathcal{A}_{\mathcal{P}}$

Proposition 1: The game with potential function $\phi$ and and utility function $\mathcal{U}_i$ constitute a potential game[1].

[1] J. Song and S. Gupta, "CARE: Cooperative Autonomy for Resilience and Efficiency of Robot Teams for Complete Coverage of Unknown Environment under Robot Failures," *Autonomous Robots*, Under review, 2018.

# The CARE Algorithm

## Success Probability

**Success Probability $p_r(\mathcal{P}_i)$**: the probability for player $\mathcal{P}_i$ to successfully finish a contested task $r$

❖ **Compute $p_r(\mathcal{P}_i)$**

It is evaluated based on robot reliability

$$p_r(\mathcal{P}_i) = R_{\mathcal{P}_i}(\tilde{t})$$

Where:

- Battery reliability: $R_{\mathcal{P}_i}(t) = \frac{1}{1+e^{\rho_0(t-\rho_1)}}$
- Expected total time of tasking and traveling:

$$\tilde{t} = t_k + t_{tr} + t_r$$

  - $t_k$: the total tasking time of $\mathcal{P}_i$ since the beginning until game is initiated.
  - $t_{tr} = \frac{Dist(\mathcal{P}_i, r)}{u}$: the traveling time to task $r$, and $u \in \mathbb{R}^+$ is the traveling speed.
  - $t_r = \frac{N_U(r)}{\omega}$: the estimated time to finish task $r$, and $\omega \in \mathbb{R}^+$ is the speed of tasking a cell.
  - Additionally, if a player has small portion of work left in its current task (i.e., $t_c(r_c(\mathcal{P}_i)) \leq \eta$), then $t_c$ is also included into $\tilde{t}$.

# The CARE Algorithm

## Task Worth $w_r$

**Task Worth $w_r$**: the expected number of undiscovered targets in task $r$ that are available to the players.

❖ **Total Worth of Task $r$**

- Let $x_r$ be a random variable that denotes the total number of targets in task $r$. It is assumed to follow the *Poisson distribution* with a *known* mean $\lambda_r$:

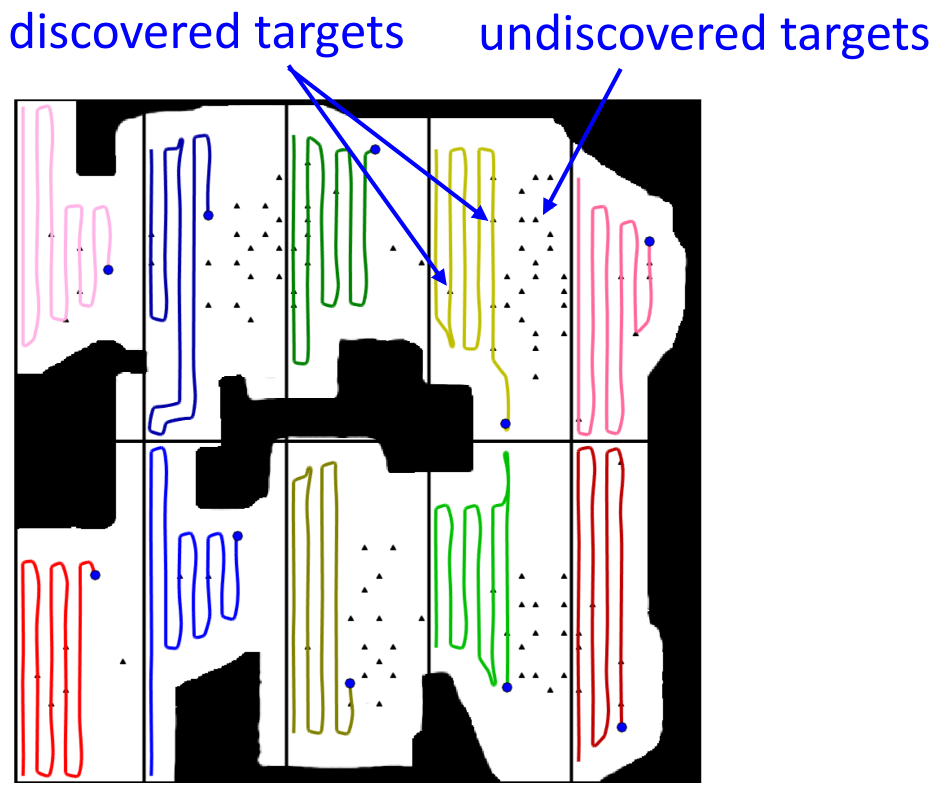
$$P_r(x_r = x) = e^{-\lambda_r} \cdot \frac{\lambda_r^x}{x!}, x = 0,1,2 \ldots$$

- If $\xi$ have already been discovered, then the estimated remaining number of undiscovered targets are:

$$\widetilde{w_r} = \sum_{x=\xi+1}^{\infty} (x - \xi) \cdot e^{-\lambda_r} \cdot \frac{\lambda_r^x}{x!}$$

Using $\sum_{x=0}^{\infty} x \cdot e^{-\lambda_r} \cdot \frac{\lambda_r^x}{x!} = \lambda_r$, and $\sum_{x=0}^{\infty} e^{-\lambda_r} \cdot \frac{\lambda_r^x}{x!} = 1$, then:

$$\boxed{\widetilde{w_r} = (\lambda_r - \xi) + e^{-\lambda_r} \cdot \sum_{x=0}^{\xi} (\xi - x) \cdot \frac{\lambda_r^x}{x!}}$$




Robot exploration with target discovery

# The CARE Algorithm

## Task Worth $\boldsymbol{w_r}$

**But…** there can be non-player robots currently working in task $r$, and they are also discovering targets but not participating the game.

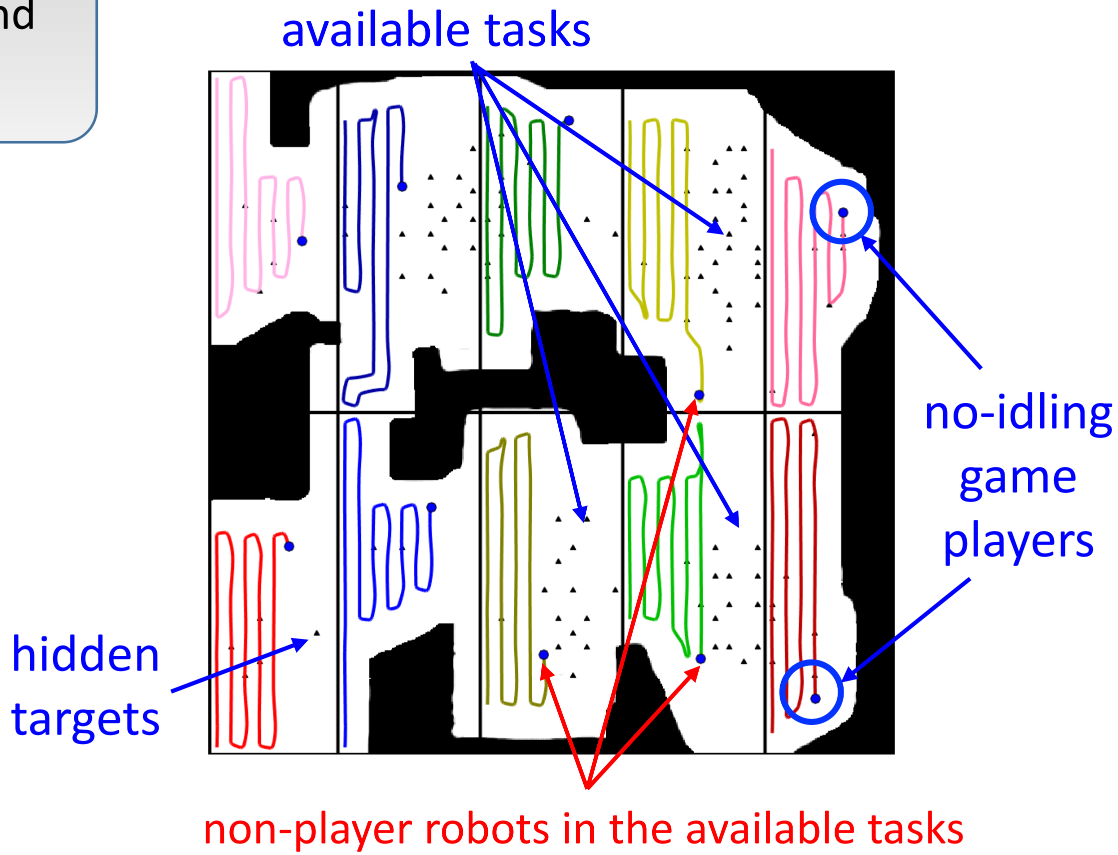
❖ **Task Worth $\boldsymbol{w_r}$ for Players**

- Let $\bar{\mathcal{P}} = V \setminus \mathcal{P}$ denote the set of non-players.
- Let $\{\bar{\mathcal{P}}\}_r$ denote the set of non-players currently in task $r$.
- These non-players have joint success probability for task $r$:

$$q(r) = 1 - \prod_{v_\ell \in \{\bar{\mathcal{P}}\}_r} [1 - p_r(v_\ell)]$$

- The task worth available to the players

$$\boxed{w_r = \widetilde{w_r} \cdot (1 - q(r))}$$



# The CARE Algorithm

## Connection between Local Games and Whole Team

❖ **Total Team Potential $\Phi(a)$**

$$\Phi(a) = \sum_{r=1}^{M} \widetilde{w_r} \cdot \left(1 - \prod_{v_\ell \in \{V\}_r} [1 - p_r(v_\ell)]\right)$$

where:

- $a = (a_{\mathcal{P}}, a_{\bar{\mathcal{P}}})$ is the joint action of both players $a_{\mathcal{P}}$ and non-players $a_{\bar{\mathcal{P}}}$
- $\{V\}_r = \{\mathcal{P}\}_r \cup \{\bar{\mathcal{P}}\}_r$ is the set of all robots assigned to task $r$

❖ Once local players reach equilibrium $a_{\mathcal{P}}^{\star}$, denote the new allocation for the whole team as $a^{\star} = (a_{\mathcal{P}}^{\star}, a_{\bar{\mathcal{P}}})$.

Remark: $\Phi(a)$ is the potential for the whole team; while $\phi(a_{\mathcal{P}})$ is the potential for the local players

Since the players and non-players are mixed and distributed over different tasks, how does the total team potential $\Phi$ change when local potential $\phi$ is increased?

# The CARE Algorithm

## Connection between Local Games and Whole Team

Theorem[1]: The optimal equilibrium $a^\star$ increases the total team potential $\Phi(a)$, i.e., $\Phi(a^\star) \geq \Phi(a)$

**Sketch of Proof:**

$$\Phi(a) = \sum_{r=1}^{M} \widetilde{w_r} \cdot \left(1 - \prod_{\mathcal{P}_i \in \{\mathcal{P}\}_r} [1 - p_r(\mathcal{P})] \cdot \prod_{v_\ell \in \{\bar{\mathcal{P}}\}_r} [1 - p_r(v_\ell)]\right)$$

$$= \sum_{r=1}^{M} \widetilde{w_r} \cdot (1 - [1 - p(r)][1 - q(r)])$$

$$= \sum_{r=1}^{M} \widetilde{w_r}[1 - q(r)] \cdot p(r) + \sum_{r=1}^{M} \widetilde{w_r} \cdot q(r)$$

- Joint success probability for players: $p(r) = 1 - \prod_{\mathcal{P}_i \in \{\mathcal{P}\}_r} [1 - p_r(\mathcal{P})]$
- Joint success probability for non-players: $q(r) = 1 - \prod_{v_\ell \in \{\bar{\mathcal{P}}\}_r} [1 - p_r(v_\ell)]$

Substitute: $w_r = \widetilde{w_r}[1 - q(r)]$

Then, it can be written as

$$\Phi(a) = \sum_{r=1}^{M} w_r \cdot p(r) + \sum_{r=1}^{M} \widetilde{w_r} \cdot q(r)$$

$$= \left(\sum_{r \in \tilde{\mathcal{A}}} w_r \cdot p(r) + \sum_{r \notin \tilde{\mathcal{A}}} w_r \cdot p(r)\right) + \sum_{r=1}^{M} \widetilde{w_r} \cdot q(r)$$

$$= \phi(a_{\mathcal{P}}) + \sum_{r \notin \tilde{\mathcal{A}}} w_r \cdot p(r) + \sum_{r=1}^{M} \widetilde{w_r} \cdot q(r)$$

Substitute: $\phi(a_{\mathcal{P}}) = \sum_{r \in \tilde{\mathcal{A}}} w_r \cdot p(r)$

The players should finish their current small left-over tasks

Not affected for non-players

[1] J. Song and S. Gupta, "CARE: Cooperative Autonomy for Resilience and Efficiency of Robot Teams for Complete Coverage of Unknown Environment under Robot Failures," *Autonomous Robots*, Under review, 2018.

# The CARE Algorithm
## The Reallocation Decision

**Learning for equilibrium $a^{\star}_{\mathcal{P}}$**: Max-Logit can fast converge to the optimal equilibrium (i.e., maximizer to $\phi$).

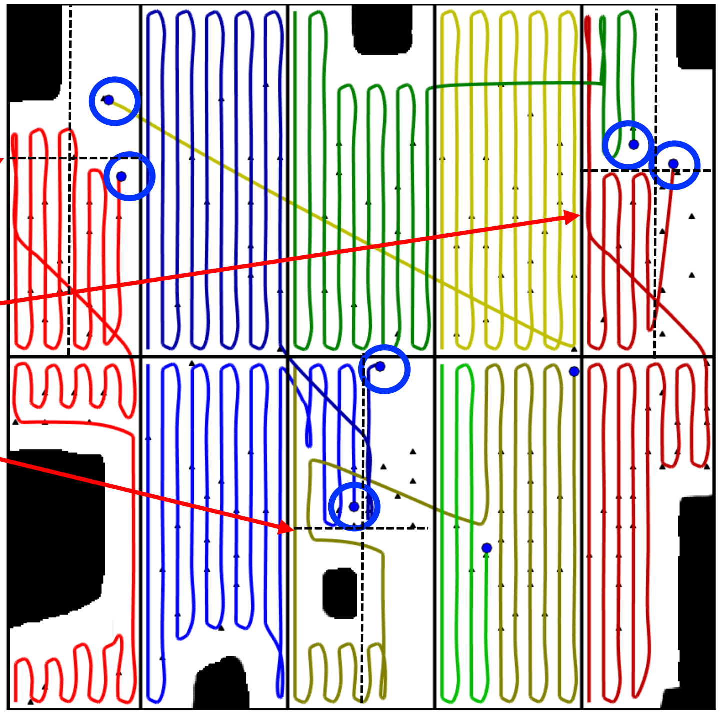
**But…** what if multiple robots are assigned to the same task, and/or what if there already are some non-player robots there?

**Post-game Coordination**

- First, evenly partition task $r$ into $n_{\max} \in \mathbb{N}^{+}$sub-regions
- Non-players continues search within the sub-region determined by its current location
- Each player $\mathcal{P}_i \in \{\mathcal{P}_i \in \mathcal{P} : a_i^{\star} = r\}$ chooses the closest sub-region in an order based on its success probability $p_r(\mathcal{P}_i)$

$n_{\max} = 4$

Division into $n_{\max} = 4$ sub-regions

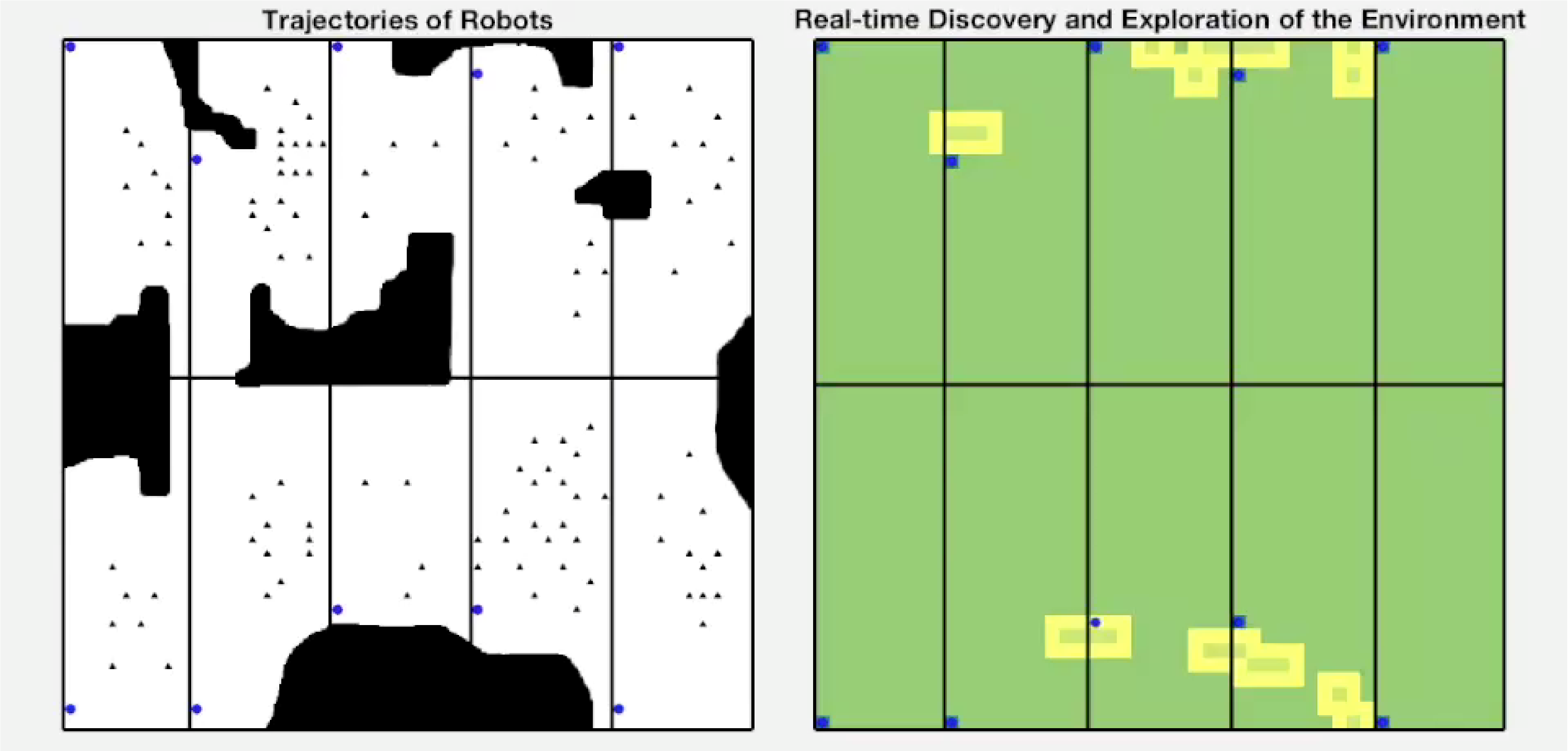


Theorem 2[1]: Complete coverage is guaranteed as long as at least one robot is still alive.

[1] J. Song and S. Gupta, "CARE: Cooperative Autonomy for Resilience and Efficiency of Robot Teams for Complete Coverage under Failures", Autonomous Robots (Under Review).

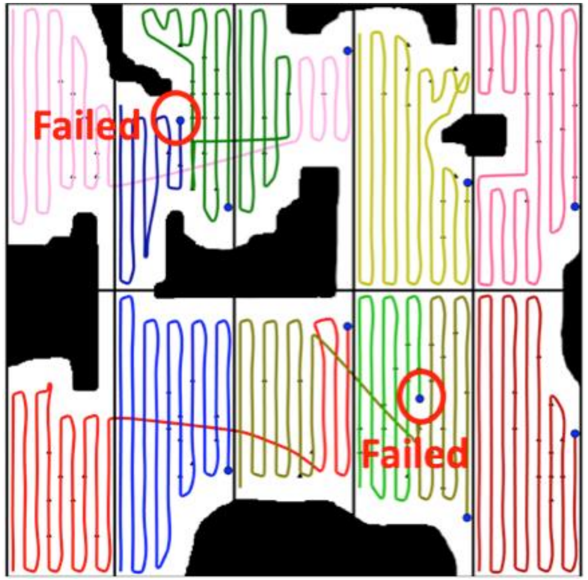
# Simulation Validations

- The search area is of size $50\text{m} \times 50\text{m}$, partitioned into $M = 10$ tasks. A team of $N = 10$ robots are deployed.
- Vehicle: battery reliability parameters $\rho_0 \sim N(3 \times 10^{-3}, 7.5 \times 10^{-5})$ and $\rho_1 \sim N(1400{,}35)$; laser range: $5\text{m}$; $u = 1\text{m/s}$; $\omega = 0.32$ cell/s
- Game parameters: $\kappa_1 = 6$, $\kappa_2 = 3$, $\eta = 30\text{s}$, $\gamma = 200\text{s}$, number of game computation cycles: $50$



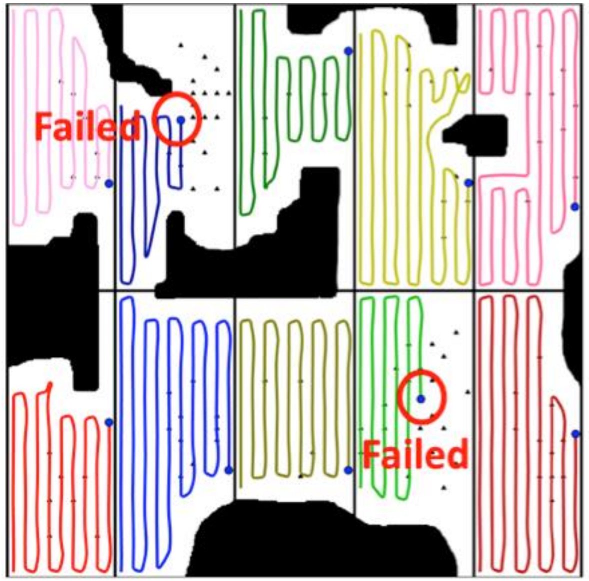
# Simulation Validations

## Coverage Trajectories using Alternative Methods

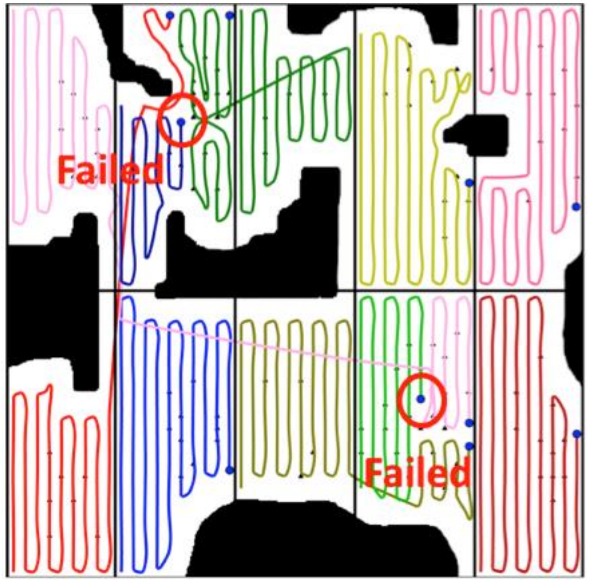



(a) CARE

**Incomplete coverage**

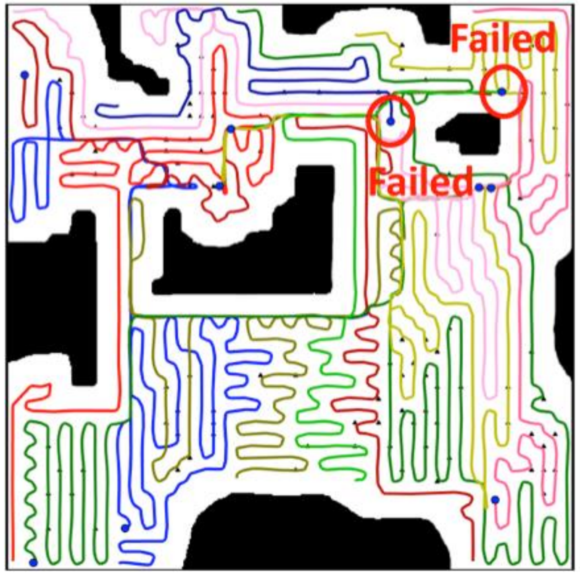



(b) Non-cooperative Coverage




(c) First-responder Coverage

**No task partition**

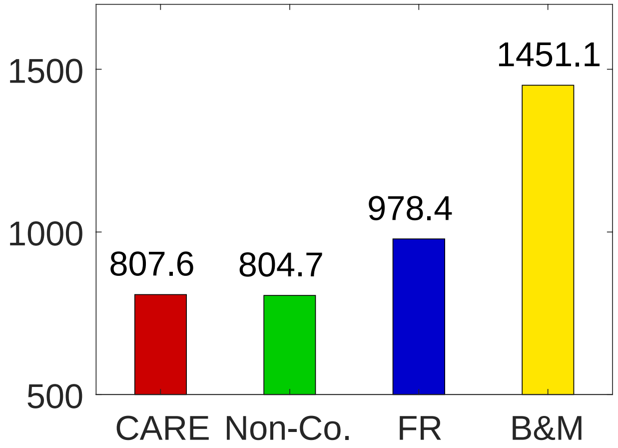



(c) Multi-robot Brick-and-Mortar

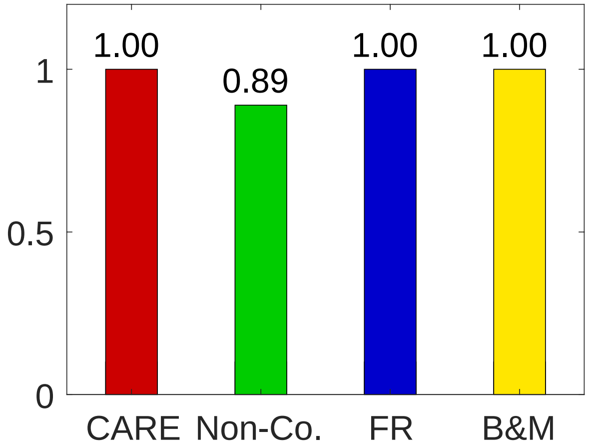
# Simulation Validations

## Performance Comparison with Alternative Methods

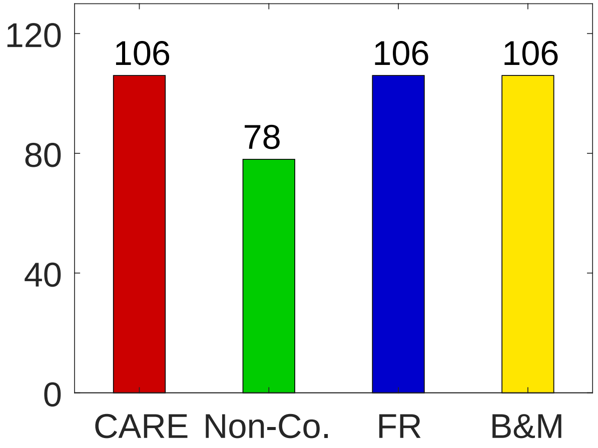
### Coverage Time ($CT$)



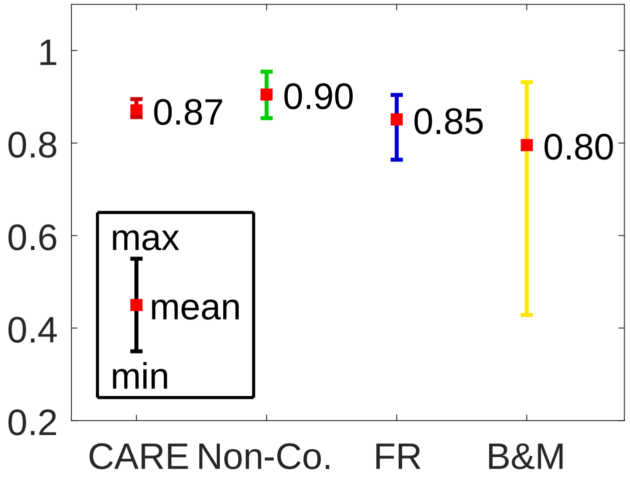

### Coverage Ratio ($CR$)




### Number of Targets Found ($NoTF$)




### Remaining Reliability ($RR$)




### Time of Target Discovery ($ToTD$)