EiR/CPR/CAMS 2024/2025

Analysis and Control of Multi-Robot Systems

Optimization-based Task Allocation for Multi-Robot Systems

Andrea Cristofaro

(Slides by Lorenzo Govoni)

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

Task Allocation

- **Which robot should execute which task to cooperatively achieve the global goal?**

- Inputs of the problem
  - *R*: team of mobile robots
  - *T*: set of tasks
  - *U*: set of robots' utilities, where u_{ij} is the utility of robot *i* to execute task *j*

- Output of the problem: find the optimal mapping $A : T \rightarrow R$

- The task allocation problem can be formulated as an optimal assignment problem

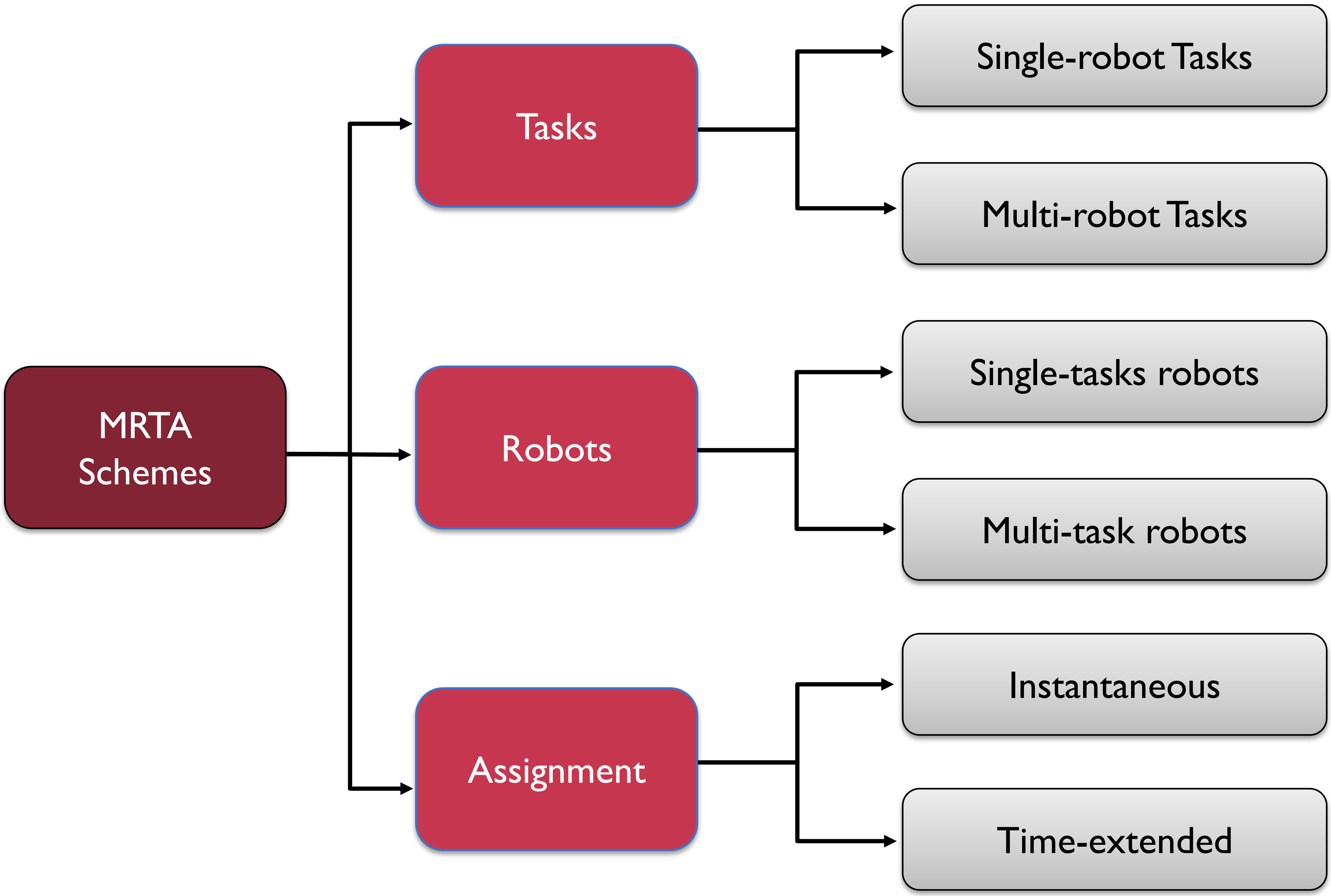
  - Define a cost function encoding the profit of assigning a task to a robot

$$W(r, t) : R \rightarrow T$$

  - The optimal allocation π will be the solution of

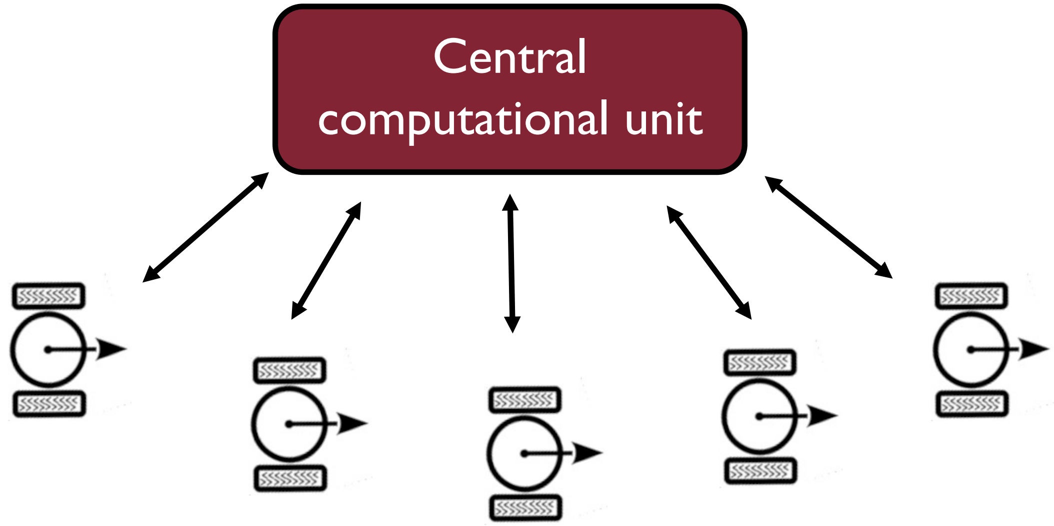
$$\pi^* = \max_\pi \sum_{i=1}^{n} w(r_i, t_{\pi,i})$$

Task Allocation Schemes

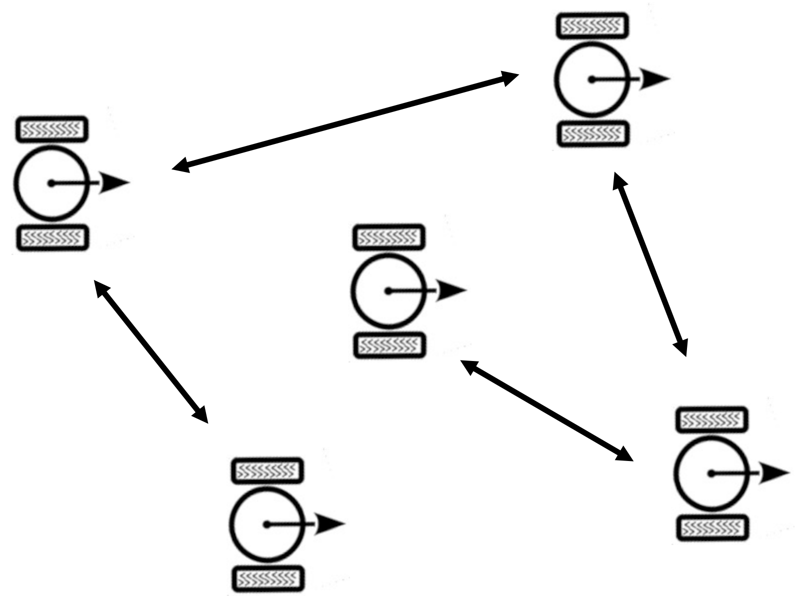


Task Allocation Architecture

- Centralized
  - Pros:
    - Reduction of duplication of effort and resources
    - Saving cost and time
  - Cons:
    - Lack of robustness: single point of failure
    - Scalability

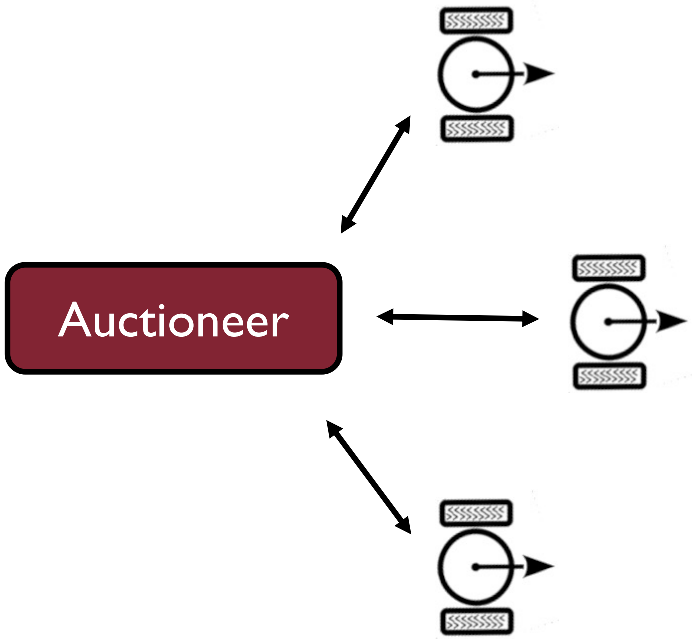


- Decentralized
  - Pros:
    - Robustness
    - Flexibility
    - Low communication demand
  - Cons:
    - Design of highly sub-optimal solutions



Task Allocation Approaches

- Market-based approach
  - It is based on auctions: process of assigning a set of goods or services to a set of bidders
  - Robots bid for tasks based on their capabilities and the auctioneer decides the "winner" of the task
  - Can be either centralized or decentralized



- Optimization-based approach
  - The task allocation problem is seen as an optimal assignment problem
  - Can be either centralized or decentralized
  - Example: minimum-energy task execution

$$\min_{ui,\delta} ||u_i||^2 + ||\delta||^2$$

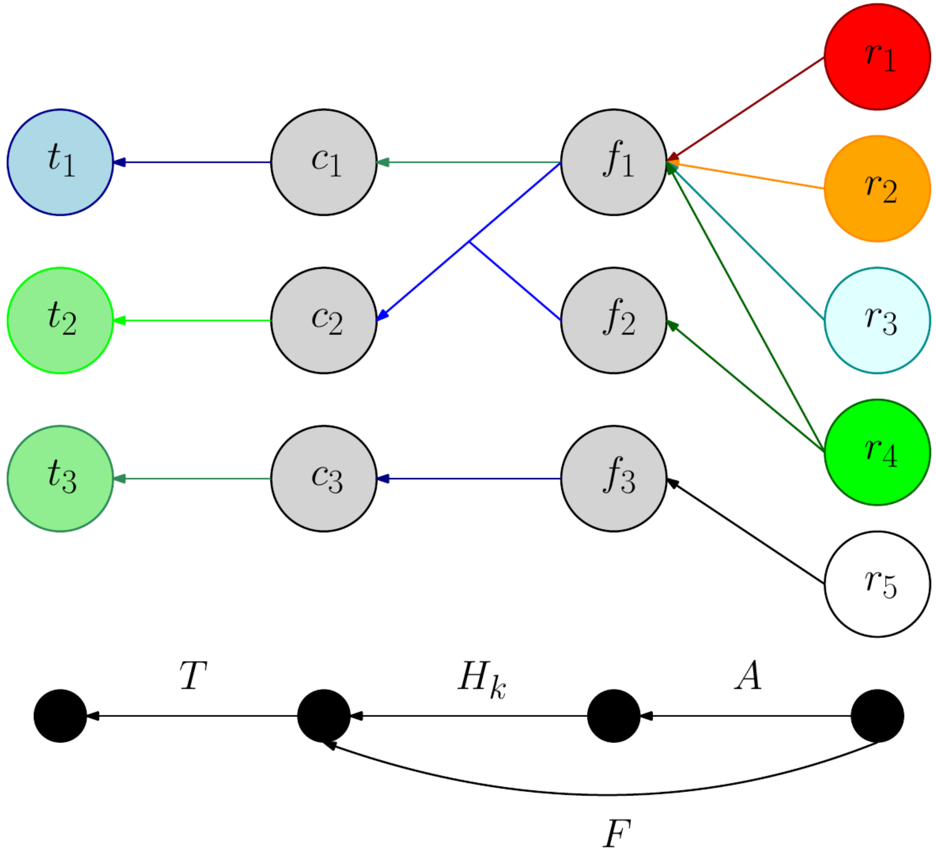
$$\text{s.t} \;\; c_{task_j}(x,u) \geq -\delta_j \;\; \forall j \in \{1,\ldots,n_t\}$$

Optimization-based Task Allocation Framework

- Problem: given a set of n_t tasks and a team composed of n_r heterogenous robots, find the optimal task allocation based on the different nature of each robot.

- **How to encode the heterogeneity of the team ?**

- Features:
  - f1: wheels
  - f2: water propellers
  - f3: air propellers
- Capabilities:
  - c1: ground locomotion
  - c2: mobility in water
  - c3: fly
- Tasks:
  - t1: reach a point on the ground
  - t2: reach a point in a lake
  - t3: hovering



Optimization-based Task Allocation Framework

- **How to encode the execution of a task?** Control Barrier Function (CBF)

- Consider an input-affine system $\dot{x} = f(x) + g(x)u$

- Let us consider a set $\mathcal{C}$ defined as the superlevel set of a continuously differentiable function $h(x)$

$$\mathcal{C} = \{x \in \mathcal{D} \subset \mathbb{R}^n : h(x) \geq 0\}$$
$$\partial\mathcal{C} = \{x \in \mathcal{D} \subset \mathbb{R}^n : h(x) = 0\}$$
$$\mathrm{Int}\mathcal{C} = \{x \in \mathcal{D} \subset \mathbb{R}^n : h(x) > 0\}$$

- We want to find a control that renders $\mathcal{C}$ forward invariant

$$\exists u \text{ s.t. } \dot{h}(x, u) \geq -\alpha(h(x))$$

- Given a set $\mathcal{C}$, a function $h(x)$ is said to be a Control Barrier Function if

$$\sup_{u \in \mathcal{U}} \{L_f h(x) + L_g h(x) u\} \geq -\alpha(h(x)) \;\; \forall x \in \mathcal{X}$$

Optimization-based Task Allocation Framework

- Optimization based control

- Suppose we are given a nominal control u_{nom} that does not satisfy the condition

$$\sup_{u \in \mathcal{U}} \{L_f h(x) + L_g h(x) u\} \geq -\alpha(h(x)) \;\; \forall x \in \mathcal{X}$$

- hence invariance is not guaranteed.

- Idea: perturb in a minimal way u_{nom} in order to find a controller that guarantees forward invariance if the set $\mathcal{C}$.

$$u(x) = \min_{u \in \mathbb{R}^m} \frac{1}{2} ||u - u_{nom}||^2$$
$$\text{s.t.} \;\; L_f h(x) + L_g h(x) u \geq -\alpha(h(x))$$

Optimization-based Task Allocation Framework

- Each task can be defined by means of a positive continuously differentiable cost function $J_m(x)$ and its execution is characterized by the design of a control that minimizes such function.

- We can define the CBF as $h_m(x) = -J_m(x)$, yielding in the safe set $\mathcal{C}$

$$\begin{aligned}\mathcal{C} &= \{x_i \in \mathbb{R}^{n_x} : h_m(x_i) \geq 0\} \\ &= \{x_i \in \mathbb{R}^{n_x} : J_m(x_i) \leq 0\} \\ &= \{x_i \in \mathbb{R}^{n_x} : J_m(x_i) = 0\}\end{aligned}$$

- Each robot *i* can design its own control u_i for executing the task t_m solving the following constrained optimization problem

$$\begin{aligned}&\min_{u_i,\delta_i} ||u_i||^2 + \boxed{\delta_i^2} \\ &\text{s.t. } L_f h_m(x_i) + L_g h_m(x_i) u_i \geq -\gamma(h_m(x_i)) - \boxed{\delta_{im}}\end{aligned}$$

Optimization-based Task Allocation Framework

- Centralized task allocation
- We encode the global task allocation in the variable $\alpha \in \{0,1\}^{n_t \times n_r}$

$$\delta_{in} \geq k(\delta_{im} - \delta_{max}(1 - \alpha_{mi})), n \neq m$$

$$\alpha_{im} = 1 \Rightarrow \delta_{im} \leq \tfrac{1}{k}\delta_{in}$$

$$\alpha_{im} = 0 \Rightarrow \delta_{im} \leq \delta_{max} + \tfrac{1}{k}\delta_{in}$$

$$\forall n \in \{1, \ldots, n_t\} \backslash \{m\}$$

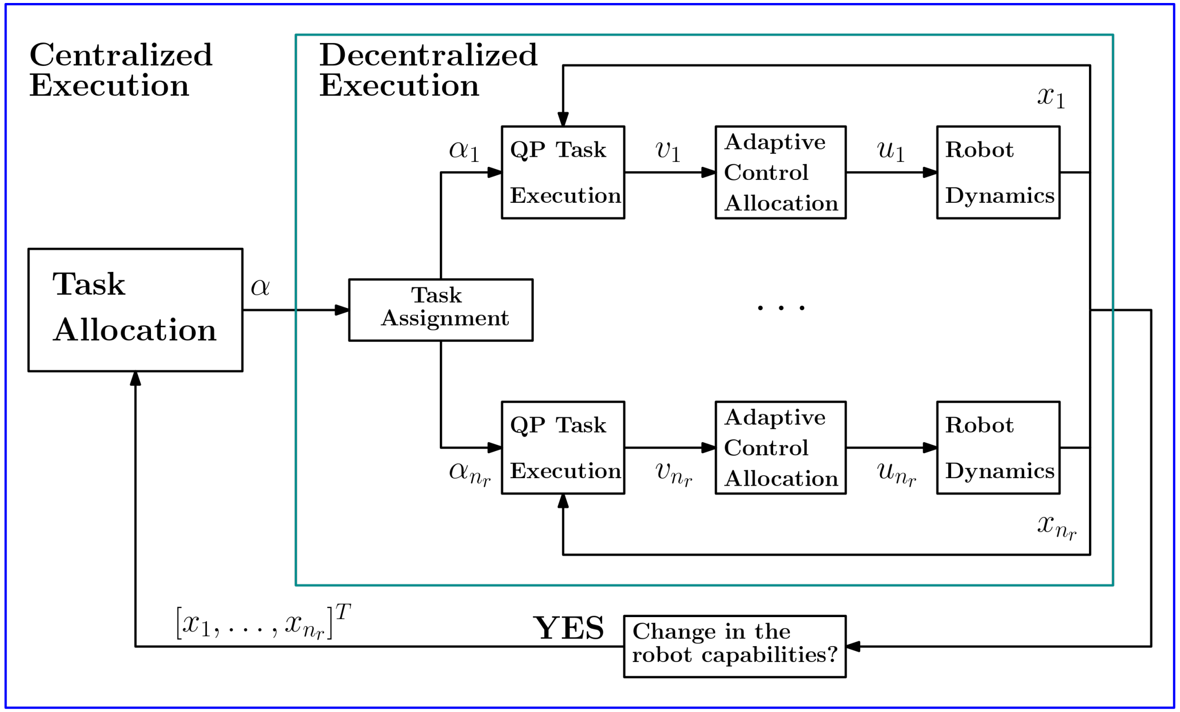
$$\begin{aligned}
\min_{\delta,\alpha} \sum_{i=1}^{n_r} & \left(C||\Pi_i \alpha_{-,i}||^2 + l||\delta_i||^2_{S_i}\right) \\
\text{subject to } & \Theta\delta_i + \Phi\alpha_{-,i} \leq \Psi \\
& \mathbf{1}^T_{n_t}\alpha_{-,i} \leq 1 \\
& F\alpha^T_{m,-} \geq T^T_{m,-} \\
& n_{r,m,min} \leq \mathbf{1}^T_{n_r}\alpha^T_{m,-} \leq n_{r,m,max} \\
& ||\delta_i||_\infty \leq \delta_{max} \\
& \alpha \in \{0,1\}^{n_t \times n_r}
\end{aligned}$$

Fault-tolerant Task Allocation Framework

- Decentralized task execution

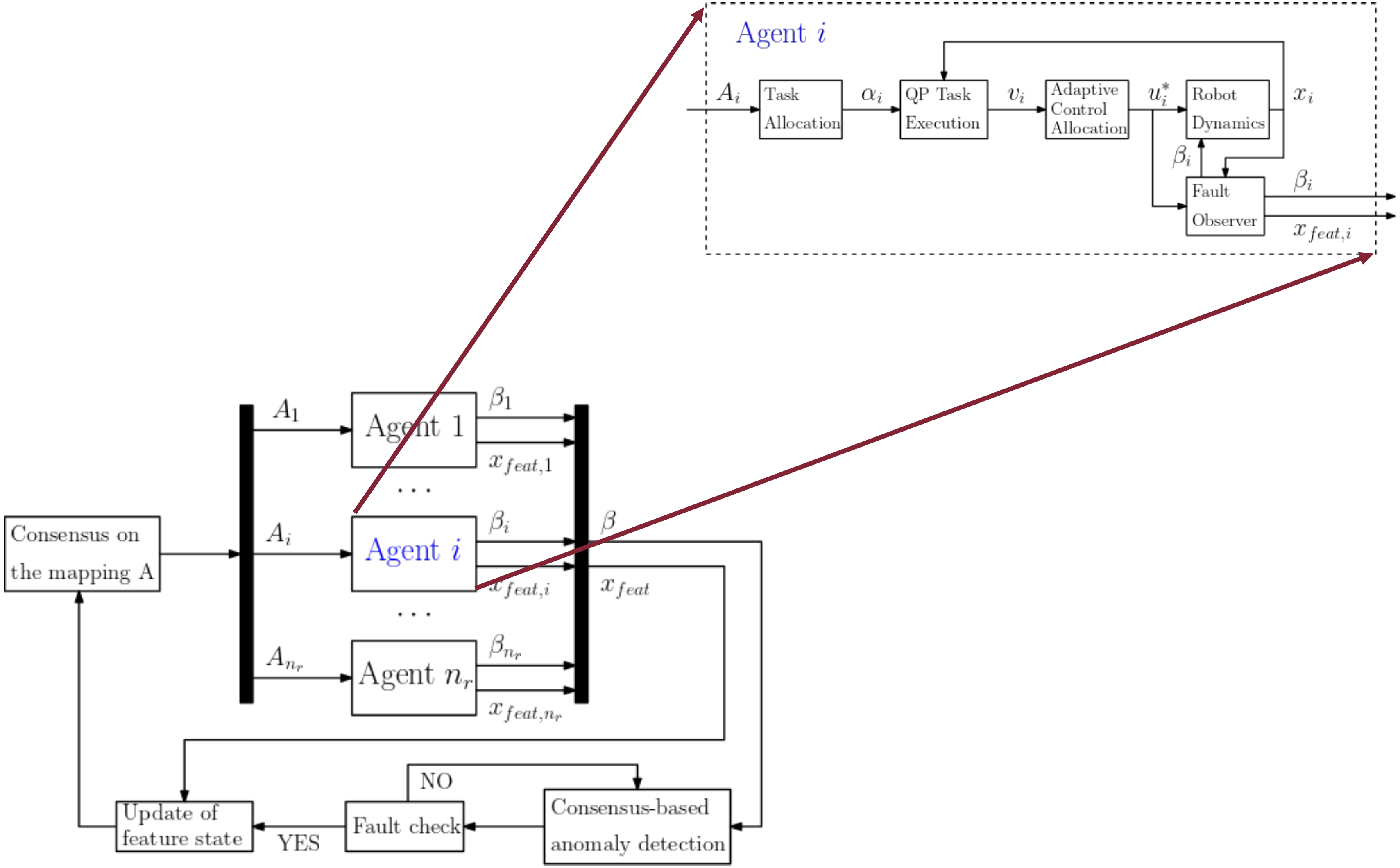
$$\min_{u_i,\delta_i} ||u_i||^2 + l||\delta_i||^2_{S_i}$$
$$\text{subject to} \quad L_f h_m(x) + L_g h_m(x) u_i \geq -\gamma(h_m(x)) - \delta_{im}$$
$$\Theta\delta_i + \Phi\alpha_{-,i} \leq \Psi$$
$$||\delta_i||_\infty \leq \delta_{max}$$

- Control architecture



Decentralized Fault-tolerant Task Allocation Framework

- Control architecture



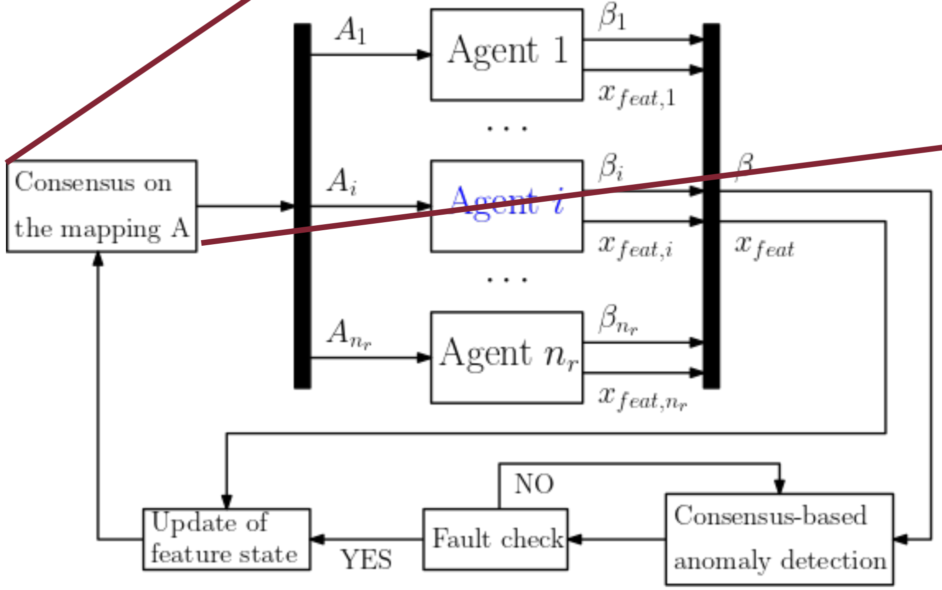
Decentralized Fault-tolerant Task Allocation Framework

- Control architecture

Algorithm 1 Consensus on the mapping A

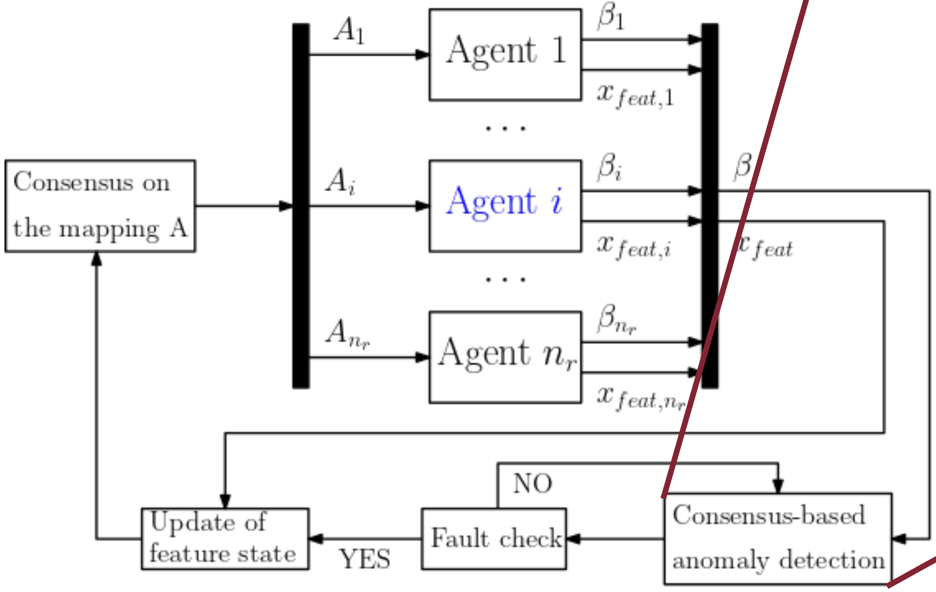
Inputs:
Adjacency matrix $\mathcal{A}$
Number of robots n_r
Number of features n_f
Vector state $x_{feat,i}$ of dimension $n_f \cdot n_r$, $i = \{1, \ldots, n_r\}$

for each robot i **do**
    Initialize $x_{feat,i}$ with local information
    Do the consensus on $x_{feat,i}$
    Normalize the non-zero component of $x_{feat,i}$
    Construct mapping A_i
end for



Decentralized Fault-tolerant Task Allocation Framework

- Control architecture



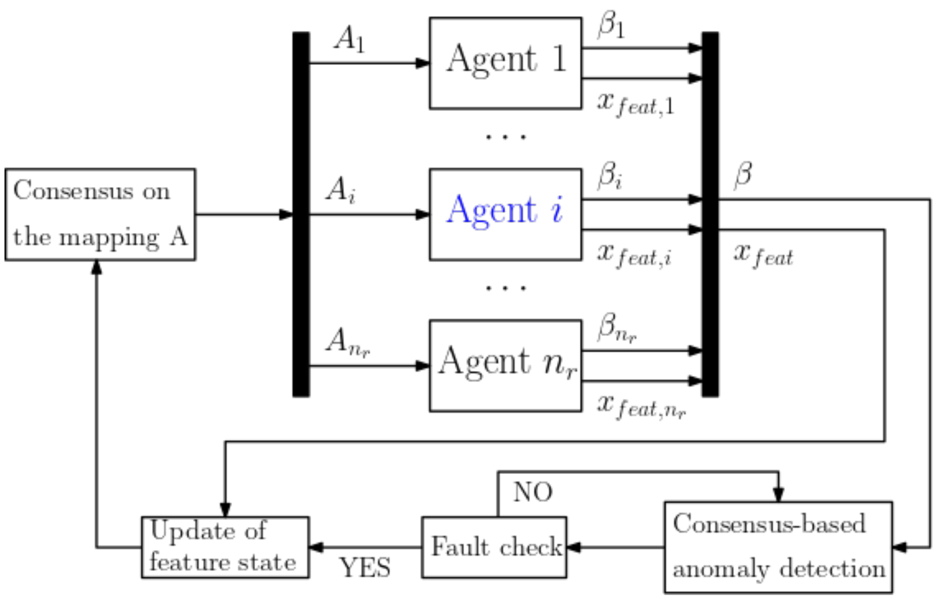
Algorithm 2 Consensus-based fault detection

Inputs:
Adjacency matrix $\mathcal{A}$
Number of robots n_r
Binary vector state β_i of dimension n_r, $i = \{1, \dots, n_r\}$

```
for each robot i do
    Initialize β_i with its own information
    Do the consensus on β_i
    Normalize β_i
    if β_ij = 1 for j = {1, ..., n_r} then
        Fault detected
    end if
end for
```

Decentralized Fault-tolerant Task Allocation Framework

- Control architecture



Algorithm 3 Resilient Task Allocation

Inputs:
Adjacency matrix $\mathcal{A}$
Number of robots n_r and number of features n_f
Tasks $h_m, m \in \{1, \ldots, n_t\}$
Mappings H_k, T
Parameters $n_{r,m,min}, n_{r,m,max}, \delta_{max}, C, l$

```
1:  for each robot i do
2:      Consensus on the mapping A                ▷ Algorithm 1
3:      Construction of the matrices F and S
4:      Evaluation of the matrix α                 ▷ (6)
5:      while true do
6:          Evaluation of the control v_i          ▷ (7)
7:          Fault check                            ▷ Algorithm 2
8:          if there is a fault then
9:              go to step 2
10:         end if
11:         Evaluation of the mapping for v_i to u_i
12:         Execution of u_i for completing the task
13:     end while
14: end for
```

Some references

- Govoni, L., and Cristofaro, A. "A fault-tolerant task allocation framework for overactuated multi-robot systems." 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2023.

- Govoni, L., and Cristofaro, A. "Decentralized task allocation for redundant multi-robot systems: an iterative consensus approach." 2024 IEEE 18th International Conference on Control & Automation (ICCA). IEEE, 2024.

Some videos

- https://drive.google.com/file/d/19qoPOU6GaEonNCaDCQm0NPHrEilhg6CN/view?usp=sharing

- https://youtu.be/9v35xHyCyQk?si=hNmqW4kDaqN6fJ0d