Lifting DecPOMDPs for Nanoscale Systems – A Work in Progress

Tanya Braun, Stefan Fischer, Florian Lau, Ralf Möller

Link to paper (QR code): https://arxiv.org/abs/2110.09152

UNIVERSITÄT ZU LÜBECK

Decentralised Partially Observable Markov Decision Processes (DecPOMDPs)

- *Set of agents* working collaboratively towards a common goal with a *joint reward function*
- Environment describable by a *Markov-1 stochastic process*, only *partially observable*
- Problem: memory requirements ***exponential*** in the agent set size
    - Especially a problem for applications with agent set sizes $\gg 10{,}000$

Observation: Symmetries in Agent Set

- *Partitions* in agent set
    - In each partition: the same set of actions, observations, behaviour w.r.t. transition and sensor model, i.e., *indistinguishable* agents

→ *Joint actions*: Only important how many agents in a partition take a particular action (count in histograms, e.g., $[n, m]$)

→ *Joint observation*: Only important how many agents in a partition observe a particular value (count in histograms)

→ Symmetries in transition / sensor model

⇩

Liftable DecPOMDP

- Agent set with partitions such that the descriptions above hold

Solution Idea: *Lifted DecPOMDPs*

$\left(\{\mathfrak{I}_k\}_k, S, \{\#_{X_k}[A_k(X_k)]\}_k, T, R, \{\#_{X_k}[O_k(X_k)]\}_k, \Omega\right)$, $k = 1, \dots, K$, with

- $\{\mathfrak{I}_k\}_k$ a partitioning of an agent set $\boldsymbol{I}, |\boldsymbol{I}| = N$,
- S a random variable for the state space,
- $\#_{X_k}[A_k(X_k)]$ a counting random variable; $\boldsymbol{A} = \times_k ran(\#_{X_k}[A_k(X_k)])$ set of joint actions
- $T(S', S, \boldsymbol{A}) = P(S' \mid S, \boldsymbol{A})$ a transition model,
- $R(S)$ a reward function,
- $\#_{X_k}[O_k(X_k)]$ a counting random variable; $\boldsymbol{O} = \times_k ran(\#_{X_k}[O_k(X_k)])$ set of joint observations,
- $\Omega(\boldsymbol{O}, S) = P(\boldsymbol{O} \mid S)$ a sensor model.

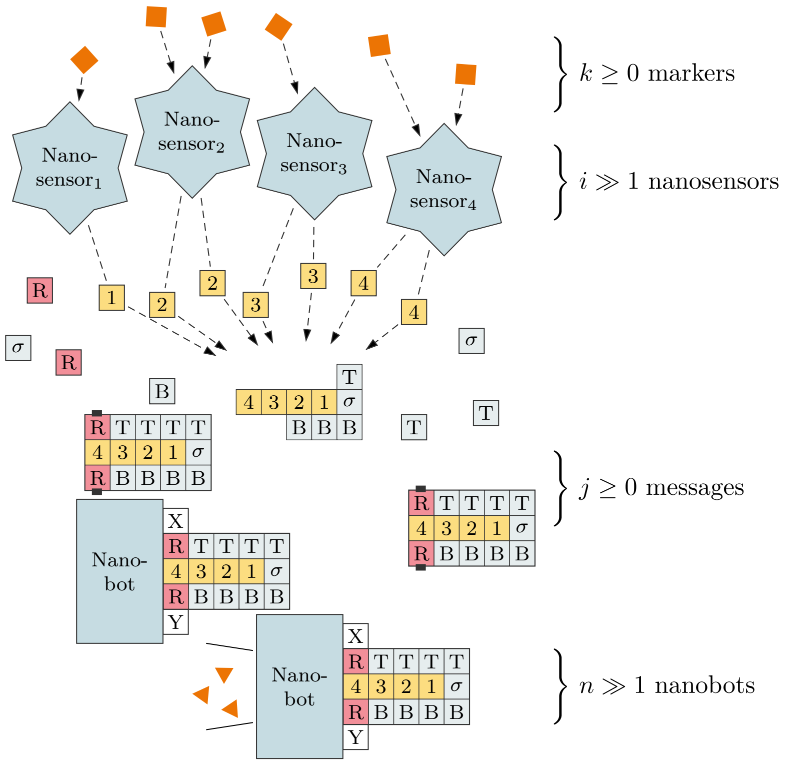
Each partition $\mathfrak{I}_k$ has a local policy π_k.

The solution is a joint policy $\boldsymbol{\pi} = (\pi_k)_k$.

⇩

- Memory ***polynomial*** in the number of agents

Application: Nanoscale Medical Systems



- Agent set: $\kappa = 4$ different types of markers/sensors, $\iota = 1$ different types of messages/nanobots; $\kappa + \iota = K$, partition size ~64,000 (preliminary experiments)
- Per partition: 2 actions (output vs. no action), 2 observations (sense/receive vs. no obs.)