IMP-MARL: a Suite of Environments for Large-scale Infrastructure Management Planning via MARL

Pascal Leroy, Pablo G. Morato, Jonathan Pisane, Athanasios Kolios, Damien Ernst

LIEGE université | DTU

moratodpg/imp_marl

arXiv
2306.11551

IMP-MARL is a novel open-source suite with **real-world** environments. **Infrastructure management planning** (IMP) coordinates inspections and repairs, minimising system failure risks and maintenance costs.
We benchmark SOTA **cooperative multi-agent RL** (MARL) methods with **up to 100 agents!** They perform better than IMP baselines but important challenges must still be resolved: **Are cooperative MARL methods scalable?**

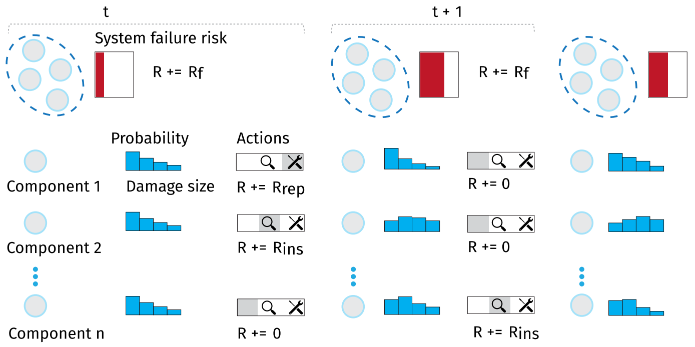
IMP

- ✓ **Real-world** application.
- ✓ Can be learned via **MARL**.
- ✗ Not **open-sourced**.
- ✗ Not compared against **SOTA** MARL algorithms.

Cooperative MARL

- ✓ Common benchmarks are **games or simulators**.
- ✓ **Open-source** methods.
- ✗ Few **real-world** environments.
- ✗ Few **large-scale** environments.

Infrastructure management planning (IMP)



- Inspect or repair based on components' damage probability.
- System failure risk depends on the components' failure probability.
- **Goal**: Minimise maintenance costs and avoid system failure.
- **Challenge**: Joint action space exponentially growing with n.
- Damage probabilities d and deterioration rate τ evolve over time:

$$p(d_{t+1}) = \sum_{\tau_{t+1}} \sum_{d_t} p(d_{t+1}|d_t, \tau_{t+1}) p(d_t) p(\tau_{t+1})$$

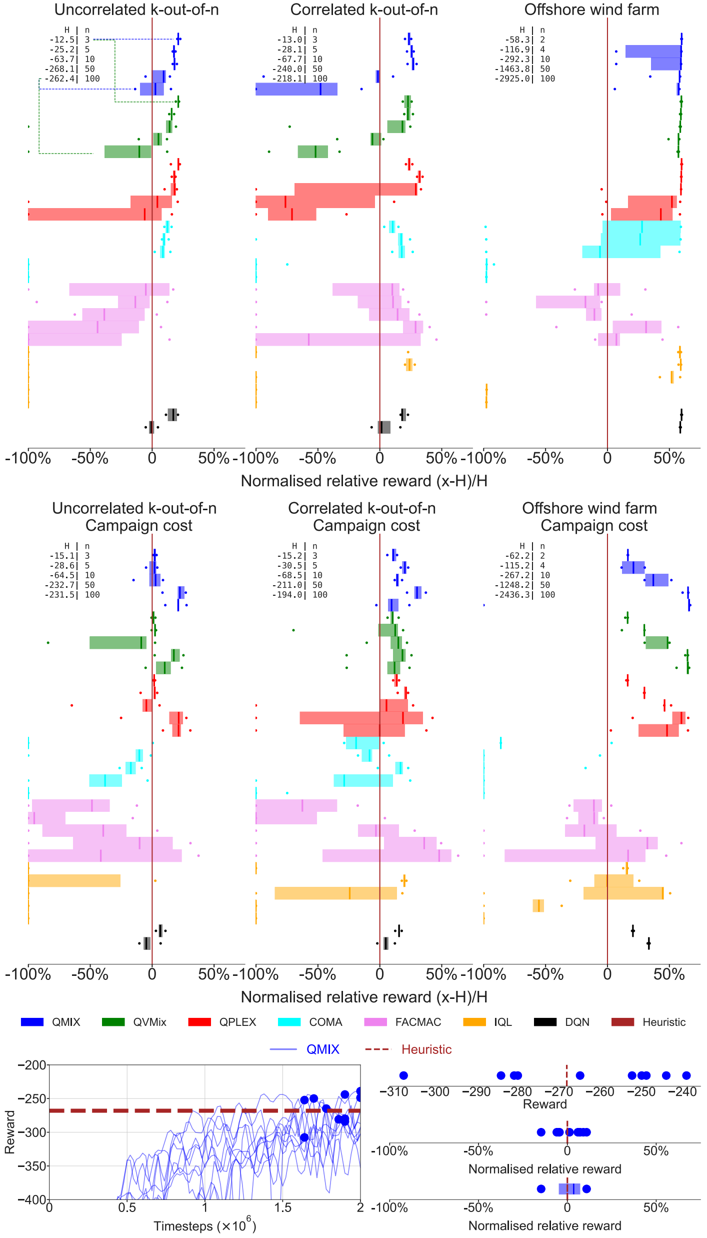
- Inspections i_d update damage probabilities:

$$p\left(d_{t+1}|i_{d_{t+1}}\right) \propto p\left(i_{d_{t+1}}|d_{t+1}\right) p(d_{t+1})$$

- Repairs reset damage to its initial condition: d_0, τ_0.

IMP as a Decentralised-POMDP

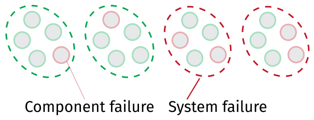
- Component = agent.
- Observation = damage probability.
- State = all damage probabilities and environment's info.
- Actions = **inspect** or **repair** or **do-nothing**.
- Common reward = $R_f + \sum_{a=1}^{n} (R^a_{\text{ins}} + R^a_{\text{rep}}) + R_{\text{camp}}$.
- Finite time horizon.



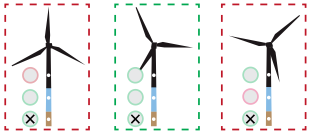
Benchmark: MARL vs IMP heuristic

- Centralised training with decentralised execution (CTDE): **QMIX**, **QVMIX**, **QPLEX**, **COMA**, **FACMAC**.
- Decentralised: **IQL** (DQN for each agent).
- Centralised: **DQN**.
- Baseline: inspection and maintenance planning **heuristic**.

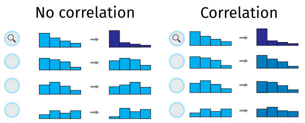
IMP-MARL environments



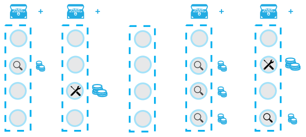
Generic category: System fails if more than n-k components fail.



Realistic category: 3 representative components per wind turbine.



Challenging scenarios: inspecting a component provides information to uninspected ones.



Practical scenarios: Campaign costs can be activated in all IMP-MARL environments.

Conclusions and future work

- CTDE methods generally outperform heuristics.
- Centralised RL methods do not scale well with the number of agents.
- IMP demands cooperation among agents: CTDE >> decentralised.
- Remaining challenges: Correlation and group campaign costs.

What we have:

- Compatibility with CleanRL, MARLLib, BenchMARL, Epymarl,...

What we need:

- New IMP environments and additional challenges.
- Contribute to the repository!