LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

The Sharer's Dilemma in Collective Adaptive Systems of Self-Interested Agents

Martin Wirsing
Ludwig-Maximilians-Universität München

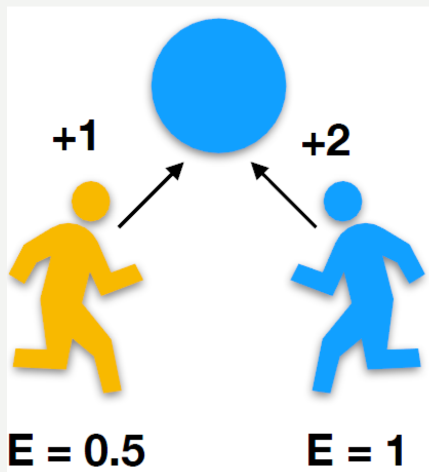
In cooperation with
Lenz Belzner, Kyrill Schmid, Thomy Phan, Thomas Gabor

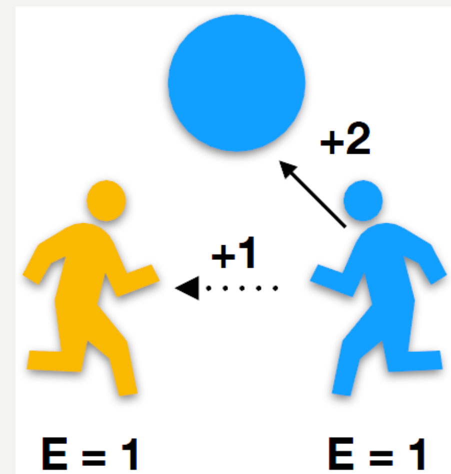
IFIP WG 1.3, Royal Holloway, July 2018

[L. Belzner, K. Schmid, T. Phan, T. Gabor, M. Wirsing: The sharer’s dilemma in collective adaptive systems of self-interested agents.. In: Margaria, T., Steffen, B. (eds.): Leveraging Applications of Formal Methods, Verification and Validation. Distributed Systems (ISoLA 2018, vol. 3). Lecture Notes in Computer Science 11246, Springer, Cham 2018, 241-256]

Coin game

- Yellow (Y) and blue agent (B) compete for a coin, with fifty-fifty chance
- Y gets reward +1, B gets reward +2





Expected value: Individual optimization

Y resists to get the coin
B shares reward +2 by
transfering +1 of reward to Y

➢ **Sharing may increase the individual and the global reward**

Simplified stochastic game (one single state)

- $N = \{0, \ldots, n\}$ is a finite set of agents
- $A = A_1 \times \ldots \times A_n$ is a set of joint actions. A_i is a finite set of actions for agent i
- $R = \{r_i : A \rightarrow \text{Real}\}_{i \in N}$ is a family of reward functions, one for each agent

Utility u_i of agent i for (joint) action a

- Pure self-interest

$$u_i(a) = r_i(a)$$

- Sharing with share s_i

$$u_i(a, s_1, \ldots, s_n) = r_i(a) - s_i + (\Sigma_{j, \neg j=i}\, s_j) / (n-1) \qquad \text{Equation (1)}$$

Policy π_i of an agent i for action a_i

- Probability distribution over actions and shares

Distributed Optimization with Sharing Algorithm DOS

Iterate the following for a predefined number of steps:

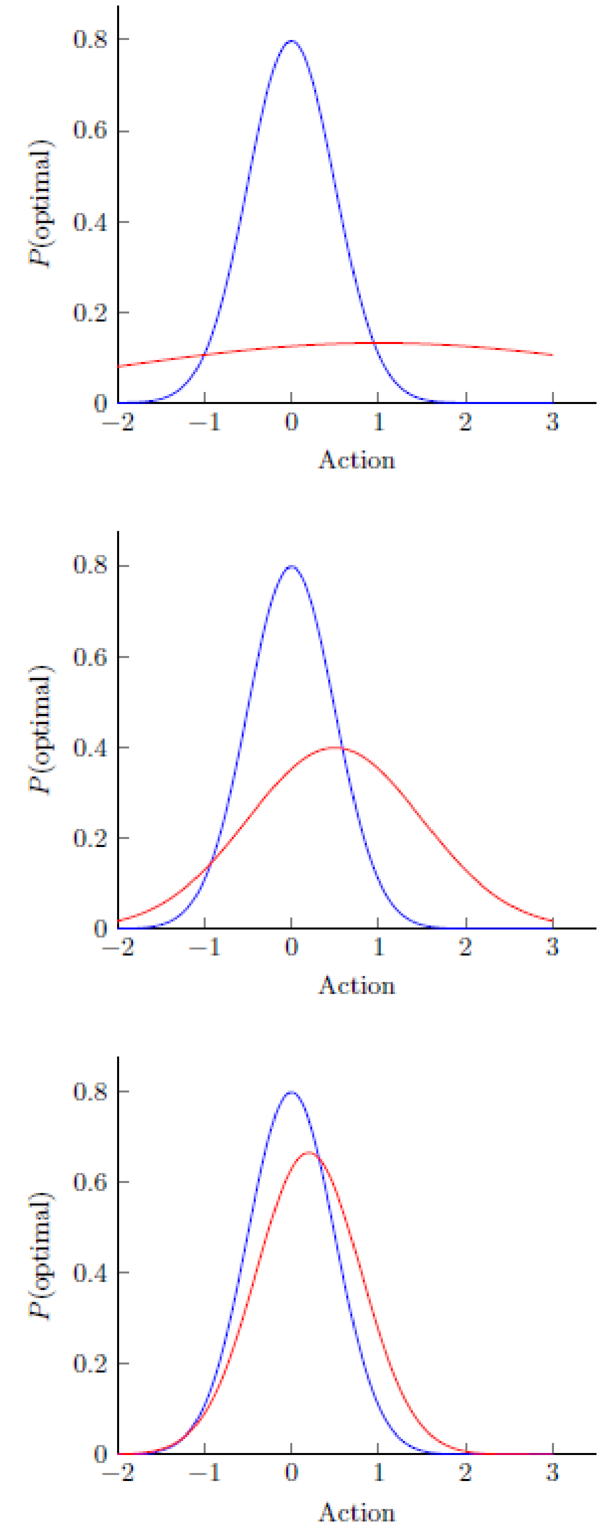
1: initialize policy π_i for each agent i
2: **for** n_{iter} iterations **do**
3:     **for** each agent i **do**
4:         each agent samples a list of n_{sample} actions and shares from i
5:         broadcast sampled actions and shares
6:     **for** each agent i **do**
7:         build joint actions a
8:         determine utility $u_i(a, s_1, \ldots, s_n)$ according to Eq. 1
9:         update policy π_i to increase the likelihood of sampling high-utility actions
10: **for** each agent i **do**
11:     execute a_i with share s_i sampled from π_i

Cross-entropy optimization

Idea

- Model policy π as (isotropic) normal distribution $\mathcal{N}(\mu, \sigma)$ with μ mean, σ standard deviation
- Start with a normal distribution
- In each step

  Update the distribution based on “elite” samples to produce a "better" distribution in the next iteration



Cross-entropy optimization

Notations

- Prior mean μ_0 and standard deviation σ_0 for policies
- Bound $\sigma_{\min}$ on the policy standard deviations
- Fraction $\psi \in (0, 1]$ of elite samples to keep
- Learning rate $\alpha \in (0, 1]$

Cross-Entropy DOS (CE-DOS)

Initialize π_i by $\mathcal{N}(\mu_0, \sigma_0)$ for each agent i
for n_{iter} iterations **do**
    for each agent i **do**
        sample n_{sample} actions and shares s_i from π_i
        clip s_i such that $s_i >= 0$
        broadcast sampled actions and shares
    for each agent i **do**
        build joint actions $a = (a_1, \ldots, a_n)$ and shares $s = (s_1, \ldots, s_n)$
        determine utility $u_i(a, s)$ according to Eq. 1
        keep $\psi \cdot n_{\text{sample}}$ elite samples $a,\ s$ with highest utility
        compute μ_{new} and σ_{new} from a_i, s_i in the elite samples
        $\mu_{t+1} := (1 - \alpha)\, \mu_t + \alpha \mu_{\text{new}}$
        $\sigma_{t+1} := (1 - \alpha)\, \sigma_t + \alpha \sigma_{\text{new}}$
        $\sigma_{t+1} := \max(\sigma_{t+1}, \sigma_{\min})$
        $\pi_i := \mathcal{N}(\mu_{t+1}, \sigma_{t+1})$
for each agent i **do**
    execute a_i with share s_i sampled from π_i

8

Simple market model

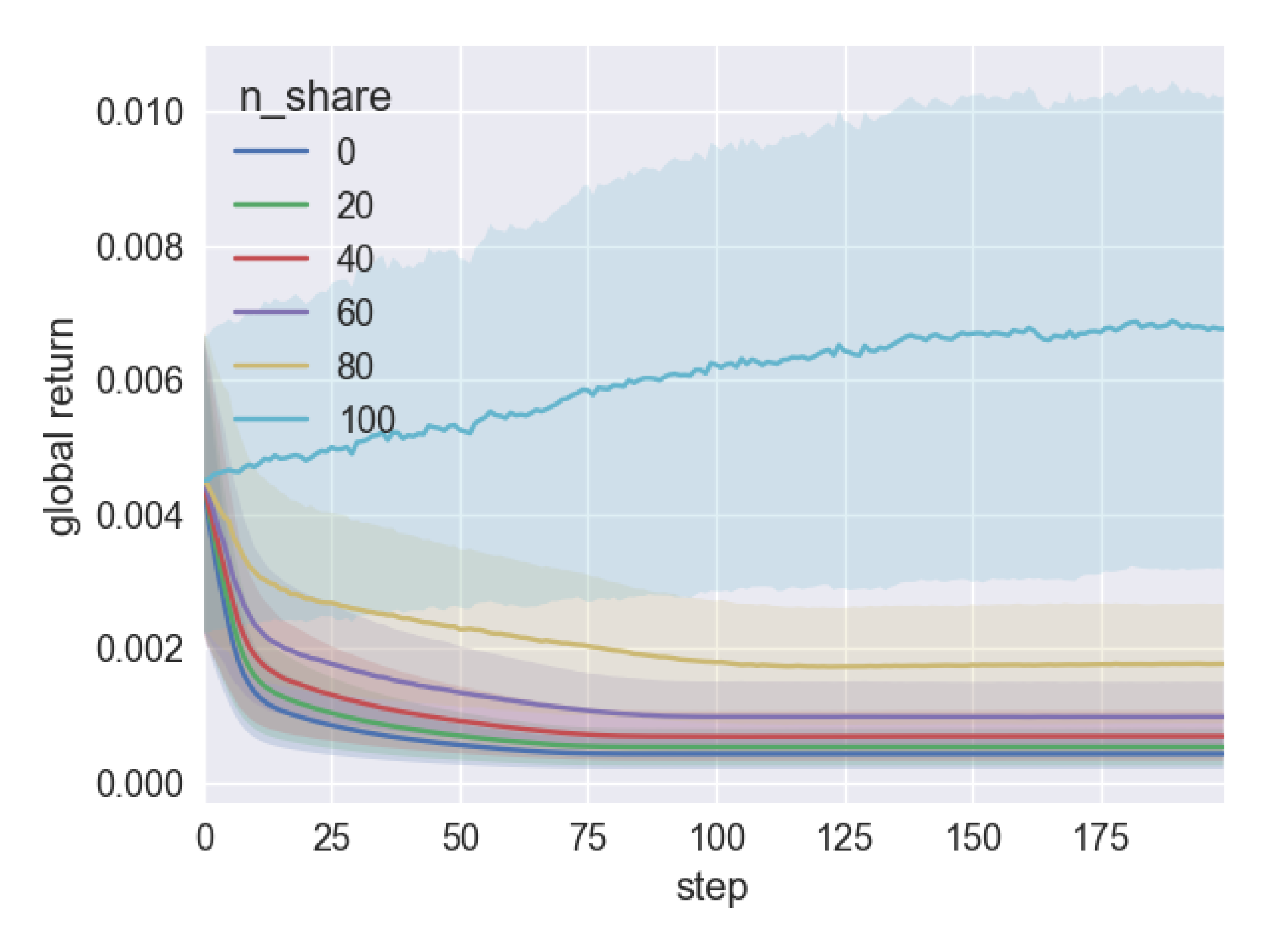
- A_i = Real models (directly) the production amount
- Global production = $\Sigma_{i \in N} a_i$
- Reward $r_i(a) = a_i / (\Sigma_{j \in N} a_j)^{\xi}$ correlates to a_i's market share

Settings

- No. agents n = 10, n = 50 and n = 100
- Individual action spaces A_i = [.1, 4]
- number of iterations n_{iter} = 100
- Number of samples n_{sample} = 100 for each agent
- Prior mean μ_0 = 0 and standard deviation σ_0 = 1
- Fraction of elite samples ψ = 0,25
- Learning rate α = 0,5
- Minimal policy standard deviation σ_{min} = 0,2.

Global Payoff

Results forSimple Market, 100 Agents

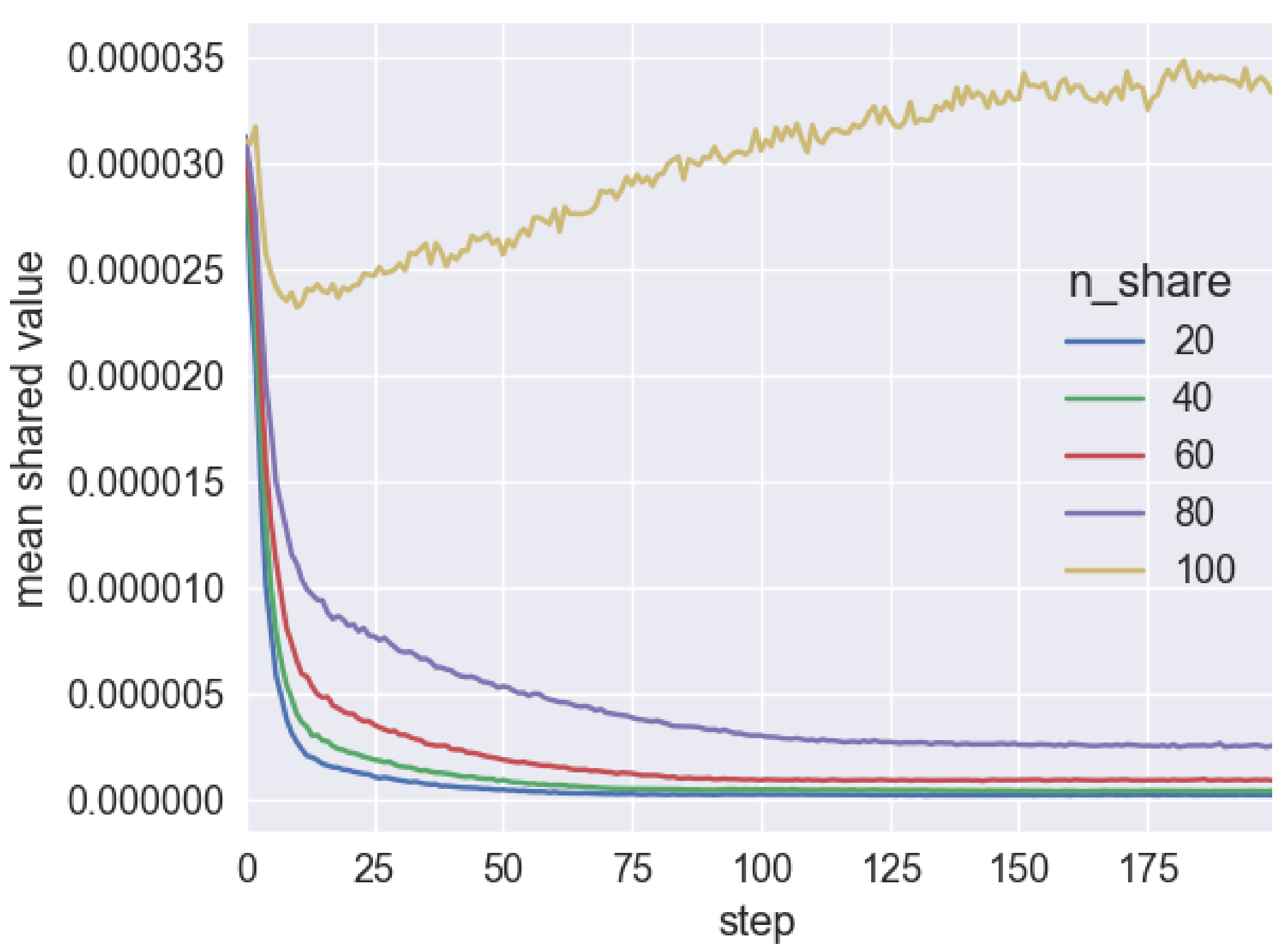


- Similar results hold for the cases of 10 agents and of 50 agents

- **Best global payoff if all agents are sharing**
- **Worst global results if all agents are selfish**

Mean Shared Value

Results forSimple Market, 100 Agents

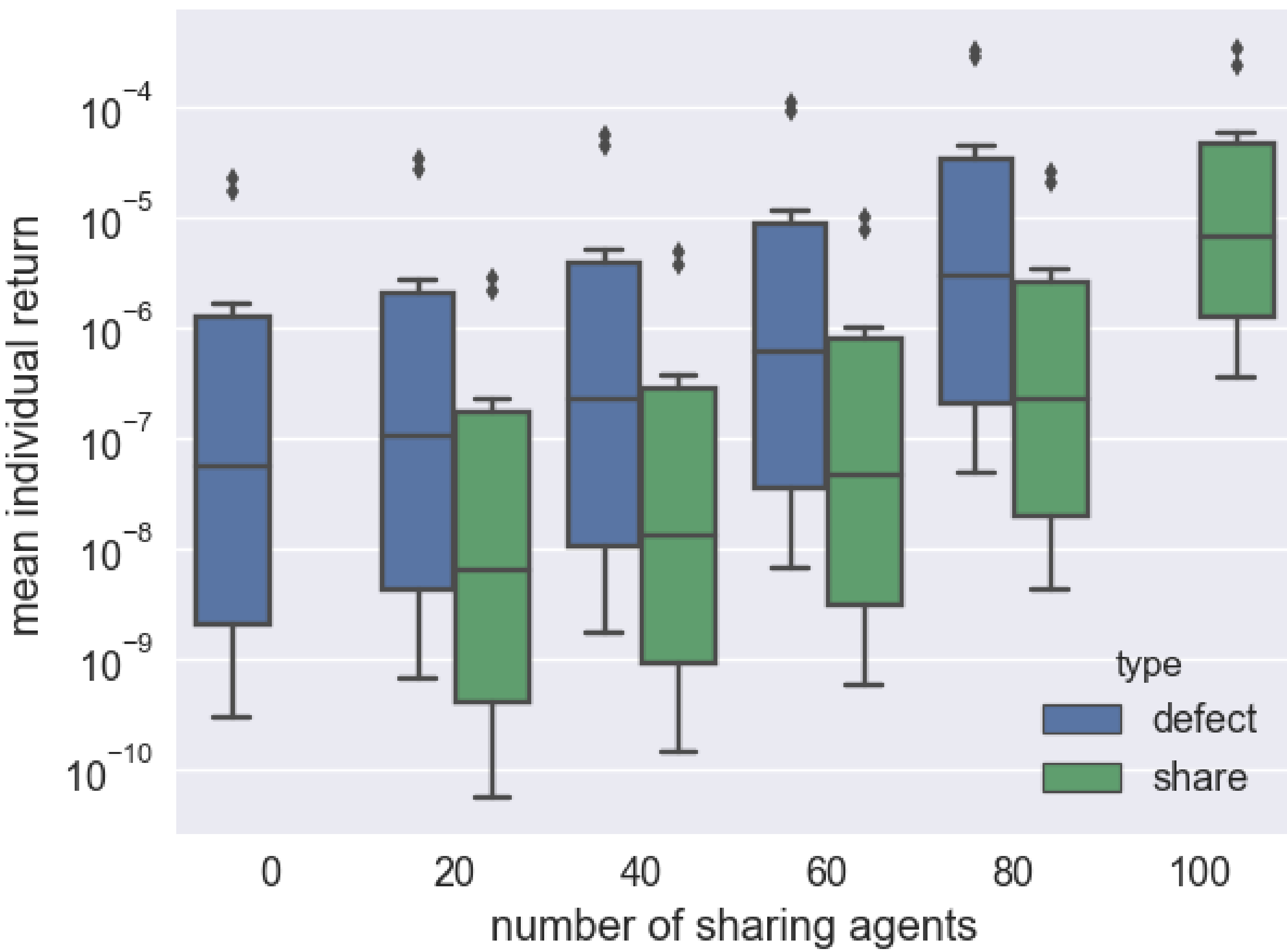


- Similar results hold for the cases of 10 agents and of 50 agents

- **Best mean shared value if all agents are sharing**
- **Worst mean shared value if all agents are selfish**

Mean Individual Return

Results forSimple Market, 100 Agents



- Similar results hold for the cases of 10 agents and of 50 agents
- **But selfish (defecting) agents get higher individual return than sharing agents**
- **However, global payoff is best if all agents are sharing**

Summary and outlook

Summary

- Adaptation implemented by optimization wrt. utility: CE_DOS algorithm
- Agents are self-interested; utilities may depend on other agents choices

Results: Dilemma

- Utility sharing increases expected individual and global payoff
- But defection increases the mean expected individual payoff at the expense of sharing individuals' payoff
- Presence of too many defectors decreases expected individual and global payoff in comparison to optimization with utility sharing

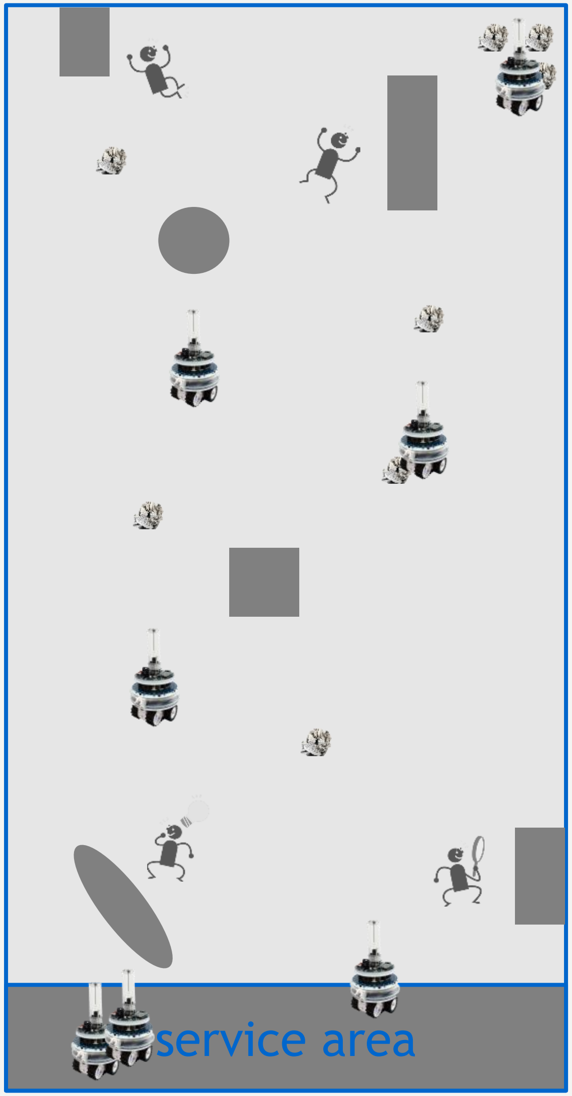
Limitations of the experiment

- CE-DOS is stateless and memoryless, no temporal effect

Future Work

- Temporal domains and multi-agent reinforcement learning with model sharing
- Other (not equally distributed) models of sharing

Herleiten einer geeigneten Reward-Funktion aus einer Anforderungsspezifikation



Beispiel:

Ein/Mehrere Roboter suchen Objekte in einem Raum und müssen auf ausreichende Batterieladung achten

Beginn: ab März 2020