Multi-Agent Deep Reinforcement Learning for Cell-free MIMO Systems: From Distributed Power Allocation to Auction-Based RIS Access

11th Annual European Future of Wireless Technology Workshop

Associate Prof. Stefan Schwarz

in collaboration with: Charmae F. Mendoza, Prof. Markus Rupp and Prof. Megumi Kaneko

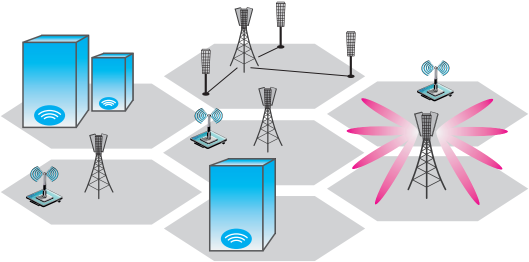
September 2025, stefan.schwarz@tuwien.ac.at

Technische Universität Wien

Institute of Telecommunications

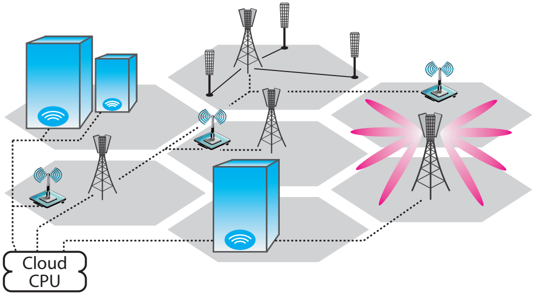
Contents

Cell-free Massive MIMO



- Main issue of dense heterogeneous 4G/5G networks: inter-cell-interfence

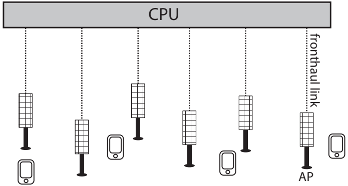
Cell-free Massive MIMO



- Main issue of dense heterogeneous 4G/5G networks: inter-cell-interfence
- Cell-free: independently operating cells are replaced by joint cloud-processing

  $\Rightarrow$ Interfering signals become useful signals

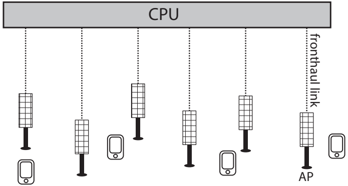
Cell-free Massive MIMO Uplink System Model



- Consider a canonical cell-free system with M access points (APs) serving K users in uplink
- At a given time t, a subset $\mathcal{K}_{\text{on}}^{(t)} \subset \{1, \ldots, K\}$ of users is active (slowly varying)

Enhancing the Uplink of Cell-Free Massive MIMO Through Prioritized Sampling and Personalized Federated Deep Reinforcement Learning, C. F. Mendoza et al., IEEE Transactions on Cognitive Communications and Networking, early access, 2025

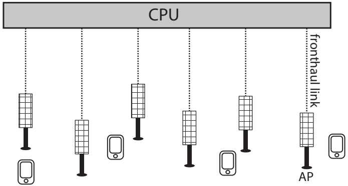
Cell-free Massive MIMO Uplink System Model



- Consider a canonical cell-free system with M access points (APs) serving K users in uplink
- At a given time t, a subset $\mathcal{K}_{\text{on}}^{(t)} \subset \{1, \ldots, K\}$ of users is active (slowly varying)
- Depending on its SINR_k, an active user k achieves **user utility** $u_k = f(\text{SINR}_k)$

Enhancing the Uplink of Cell-Free Massive MIMO Through Prioritized Sampling and Personalized Federated Deep Reinforcement Learning, C. F. Mendoza et al., IEEE Transactions on Cognitive Communications and Networking, early access, 2025

Cell-free Massive MIMO Uplink System Model



- Consider a canonical cell-free system with M access points (APs) serving K users in uplink
- At a given time t, a subset $\mathcal{K}_{\text{on}}^{(t)} \subset \{1, \ldots, K\}$ of users is active (slowly varying)
- Depending on its SINR_k, an active user k achieves **user utility** $u_k = f(\text{SINR}_k)$
- The SINR depends on the users' power allocations

  $\Rightarrow$ Increasing the power ρ_k of user k will improve its utility, but may decrease other users' utilities

  $\Rightarrow$ Goal: **learn to allocate power optimally**

Enhancing the Uplink of Cell-Free Massive MIMO Through Prioritized Sampling and Personalized Federated Deep Reinforcement Learning, C. F. Mendoza et al., IEEE Transactions on Cognitive Communications and Networking, early access, 2025

Model-Based versus Model-Free Optimization

- **Model-based** optimization: user utility is available in analytical form

  $\Rightarrow$ **Classical optimization** methods can be applied

- **Model-free** optimization: relies on observed data rather than (potentially inaccurate) models

  $\Rightarrow$ **Data-driven machine learning** techniques

Model-Based versus Model-Free Optimization

- **Model-based** optimization: user utility is available in analytical form

  $\Rightarrow$ **Classical optimization** methods can be applied

- **Model-free** optimization: relies on observed data rather than (potentially inaccurate) models

  $\Rightarrow$ **Data-driven machine learning** techniques

- Deep reinforcement learning (DRL): often combines both approaches

  $\Rightarrow$ Initial model-based training in simulations (digital twins), followed by real-world fine-tuning

  $\Rightarrow$ Keeps real-world training duration reasonable

Model-Based versus Model-Free Optimization

- **Model-based** optimization: user utility is available in analytical form

  $\Rightarrow$ **Classical optimization** methods can be applied

- **Model-free** optimization: relies on observed data rather than (potentially inaccurate) models

  $\Rightarrow$ **Data-driven machine learning** techniques

- Deep reinforcement learning (DRL): often combines both approaches

  $\Rightarrow$ Initial model-based training in simulations (digital twins), followed by real-world fine-tuning

  $\Rightarrow$ Keeps real-world training duration reasonable

- In our simulations, we train based on the Shannon rate

$$u_k = B\left(1 - \frac{\tau_p}{\tau_c}\right)\log_2\left(1 + \mathrm{SINR}_k\right)$$

- SINR under MMSE detection considering pilot contamination and CSI imperfections

Model-Based versus Model-Free Optimization

- **Model-based** optimization: user utility is available in analytical form

  $\Rightarrow$ **Classical optimization** methods can be applied

- **Model-free** optimization: relies on observed data rather than (potentially inaccurate) models

  $\Rightarrow$ **Data-driven machine learning** techniques

- Deep reinforcement learning (DRL): often combines both approaches

  $\Rightarrow$ Initial model-based training in simulations (digital twins), followed by real-world fine-tuning

  $\Rightarrow$ Keeps real-world training duration reasonable

- In our simulations, we train based on the Shannon rate

$$u_k = B\left(1 - \frac{\tau_p}{\tau_c}\right)\log_2\left(1 + \mathrm{SINR}_k\right)$$

- SINR under MMSE detection considering pilot contamination and CSI imperfections
- In practice, u_k could, for example, also be obtained from user feedback (CQI)

Scalable vs. Non-Scalable Optimization

- We want to **maximize a global utility**:

$$
\begin{aligned}
\max_{\rho_1,\ldots,\rho_K} \quad & U\left(u_1, \ldots, u_K\right) \\
\text{subject to:} \quad & 0 \le \rho_k \le \rho_{\max}, \ \forall k
\end{aligned}
$$

Scalable vs. Non-Scalable Optimization

- We want to **maximize a global utility**:

$$\max_{\rho_1,\ldots,\rho_K} \quad U\left(u_1,\ldots,u_K\right)$$
$$\text{subject to:} \quad 0 \le \rho_k \le \rho_{\max}, \ \forall k$$

$\Rightarrow$ Solving this problem **centrally is not scalable** as the network size grows

Scalable vs. Non-Scalable Optimization

- We want to **maximize a global utility**:

$$\max_{\rho_1,\ldots,\rho_K} \quad U\left(u_1,\ldots,u_K\right)$$
$$\text{subject to:} \quad 0 \leq \rho_k \leq \rho_{\max}, \; \forall k$$

$\Rightarrow$ Solving this problem **centrally is not scalable** as the network size grows

- We need a **decentralized approach** $\Rightarrow$ **multi-agent DRL**
- Scalability could be achieved via AP-clustering $\Rightarrow$ each cluster handled by a DRL agent
- Here, we consider the extreme case: **one agent per user**

Scalable vs. Non-Scalable Optimization

- We want to **maximize a global utility**:

$$\max_{\rho_1,\ldots,\rho_K} \quad U\left(u_1,\ldots,u_K\right)$$

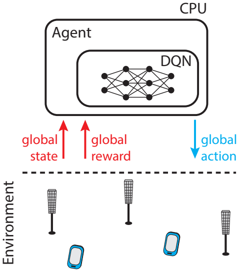
$$\text{subject to:} \quad 0 \leq \rho_k \leq \rho_{\max}, \ \forall k$$

$\Rightarrow$ Solving this problem **centrally is not scalable** as the network size grows

- We need a **decentralized approach** $\Rightarrow$ **multi-agent DRL**
- Scalability could be achieved via AP-clustering $\Rightarrow$ each cluster handled by a DRL agent
- Here, we consider the extreme case: **one agent per user**
- As an example, we consider the **guaranteed user rate** as the utility function

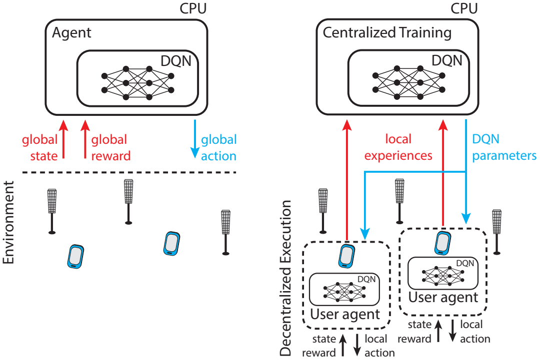
$$U\left(u_1,\ldots,u_K\right) = \min_{k\in\mathcal{K}_{\text{on}}^{(t)}} u_k$$

Three DRL Frameworks



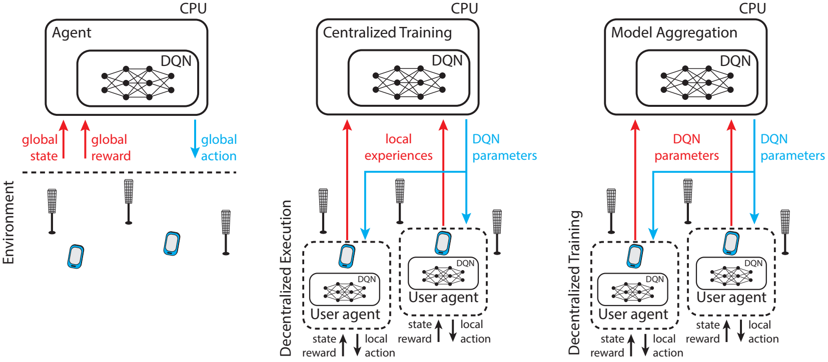
- **Single-agent RL (SARL)**: CPU handles power allocation for all users

Three DRL Frameworks



- **Single-agent RL (SARL)**: CPU handles power allocation for all users
- **Multi-agent RL (MARL)**: users make power allocation decisions
  - **Centralized training, decentralized execution (CTDE)**: same agent model shared across users

Three DRL Frameworks



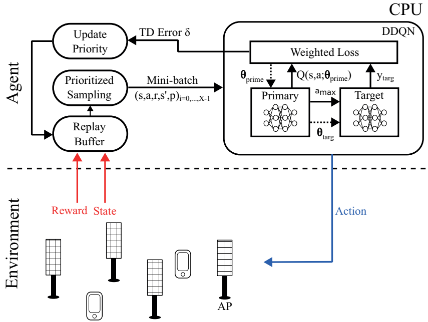
- **Single-agent RL (SARL)**: CPU handles power allocation for all users
- **Multi-agent RL (MARL)**: users make power allocation decisions
  - **Centralized training, decentralized execution (CTDE)**: same agent model shared across users
  - **Personalized federated learning (FPer)**: model parameters partially federated

SARL – Details

- **States** of the single-agent environment

$$\mathbf{s}^{(t)} = \left[d_1^{(t)}, \ldots, d_K^{(t)}, v_1^{(t-1)}, \ldots, v_K^{(t-1)}, u_1^{(t-1)}, \ldots, u_K^{(t-1)}\right]$$

$$v_k^{(t-1)} = \begin{cases} 1, & \text{if } \rho_k^{(t-1)} > 0 \text{ and } d_k^{(t-1)} = 0 \\ 0, & \text{else} \end{cases}$$



- **States** of the single-agent environment

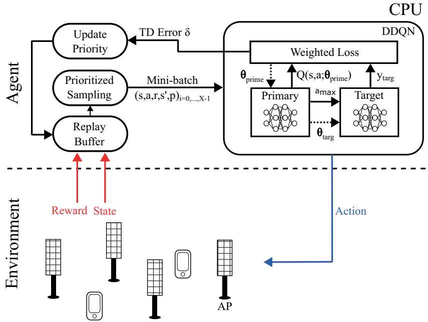
$$\mathbf{s}^{(t)} = \left[d_1^{(t)}, \dots, d_K^{(t)}, v_1^{(t-1)}, \dots, v_K^{(t-1)}, u_1^{(t-1)}, \dots, u_K^{(t-1)}\right]$$

$$v_k^{(t-1)} = \begin{cases} 1, & \text{if } \rho_k^{(t-1)} > 0 \text{ and } d_k^{(t-1)} = 0 \\ 0, & \text{else} \end{cases}$$

- **Actions** taken by the CPU

$$\mathbf{a}^{(t)} = \left[\rho_1^{(t)}, \dots, \rho_K^{(t)}\right], \quad \rho_k \in \{0, \Delta_\rho, 2\Delta_\rho, \dots, \rho_{\max}\}$$

N_{pow} possible power levels $\Rightarrow$ action space of size N_{pow}^K



SARL – Details

- **States** of the single-agent environment

$$\mathbf{s}^{(t)} = \left[d_1^{(t)}, \dots, d_K^{(t)}, v_1^{(t-1)}, \dots, v_K^{(t-1)}, u_1^{(t-1)}, \dots, u_K^{(t-1)}\right]$$

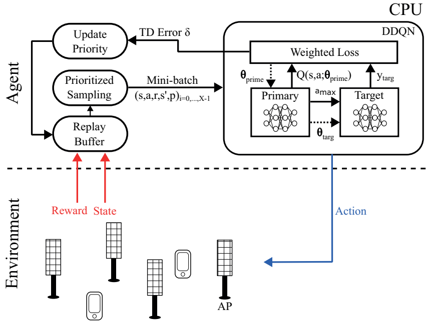
$$v_k^{(t-1)} = \begin{cases} 1, & \text{if } \rho_k^{(t-1)} > 0 \text{ and } d_k^{(t-1)} = 0 \\ 0, & \text{else} \end{cases}$$

- **Actions** taken by the CPU

$$\mathbf{a}^{(t)} = \left[\rho_1^{(t)}, \dots, \rho_K^{(t)}\right], \quad \rho_k \in \{0, \Delta_\rho, 2\Delta_\rho, \dots, \rho_{\max}\}$$

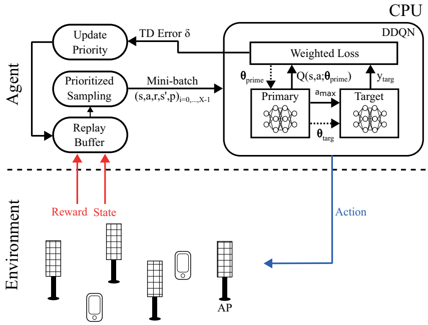
N_{pow} possible power levels $\Rightarrow$ action space of size N_{pow}^K

- **Rewards**: $r^{(t+1)} = \min_{k \in \mathcal{K}_{\text{on}}^{(t)}} u_k^{(t)} - \gamma \sum_{k=1}^{K} v_k^{(t)}$



SARL – Details (II)

- **Double deep-Q networks** (DDQN)
  - Stabilizes training and reduces overestimation bias
  - More robust in non-stationary environments
- **Prioritized sampling**
  - Prioritizes experiences with high temporal-difference (TD) error for replay
  - Speeds up learning and improves sample efficiency
  - Can introduce bias; requires importance-sampling correction

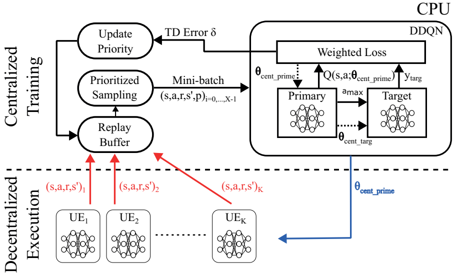


MARL CTDE – Details

- **User-specific** states and actions

$$\mathbf{s}_k^{(t)} = \left[u_k^{(t-1)}, u_{j\in\mathcal{N}_k}^{(t-1)}\right], \quad a_k^{(t)} = \rho_k^{(t)}$$

- **Sharing of utilities** at least in a neighborhood $\mathcal{N}_k \subseteq \mathcal{K}$
- No violation variables; only active users allocate power



MARL CTDE – Details

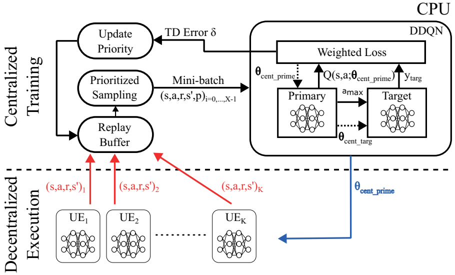
- **User-specific** states and actions

$$\mathbf{s}_k^{(t)} = \left[u_k^{(t-1)}, u_{j\in\mathcal{N}_k}^{(t-1)}\right], \quad a_k^{(t)} = \rho_k^{(t)}$$

- **Sharing of utilities** at least in a neighborhood $\mathcal{N}_k \subseteq \mathcal{K}$
- No violation variables; only active users allocate power
- **Global reward** can be calculated by CPU

$$r^{(t+1)} = \min_{k\in\mathcal{K}_{\text{on}}^{(t)}} u_k^{(t)}$$

- No need at users since training happens on CPU



- **User-specific** states and actions

$$\mathbf{s}_k^{(t)} = \left[u_k^{(t-1)}, u_{j\in\mathcal{N}_k}^{(t-1)} \right], \quad a_k^{(t)} = \rho_k^{(t)}$$

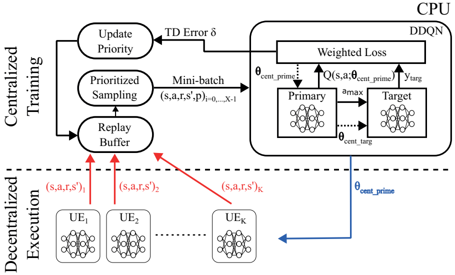
- **Sharing of utilities** at least in a neighborhood $\mathcal{N}_k \subseteq \mathcal{K}$
- No violation variables; only active users allocate power
- **Global reward** can be calculated by CPU

$$r^{(t+1)} = \min_{k\in\mathcal{K}_{\text{on}}^{(t)}} u_k^{(t)}$$

- No need at users since training happens on CPU
- Training at CPU based on **users' experiences**

$$\left(\mathbf{s}_k^{(t)}, a_k^{(t)}, r^{(t+1)}, \mathbf{s}_k^{(t+1)}\right)$$

- Reporting of action $a_k^{(t)}$ is sufficient (could be estimated)

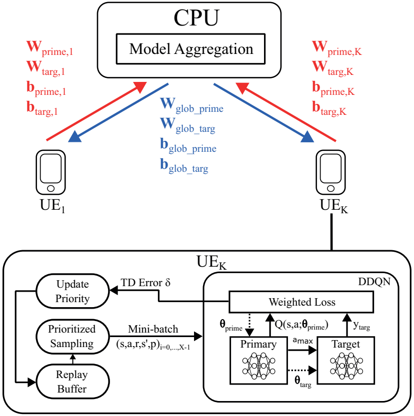


- Users train local models based on their **local experiences**

$$\left(\mathbf{s}_k^{(t)}, a_k^{(t)}, r_k^{(t+1)}, \mathbf{s}_k^{(t+1)}\right),$$

$$r_k^{(t+1)} = \min_{j \in \mathcal{N}_k} u_j^{(t)}$$

- States/rewards are determined over the **neighborhood** $\mathcal{N}_k$
- Interference is negligible if users are sufficiently separated



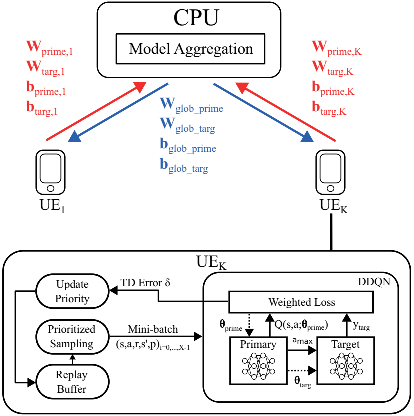
MARL FPer – Details

- Users train local models based on their **local experiences**

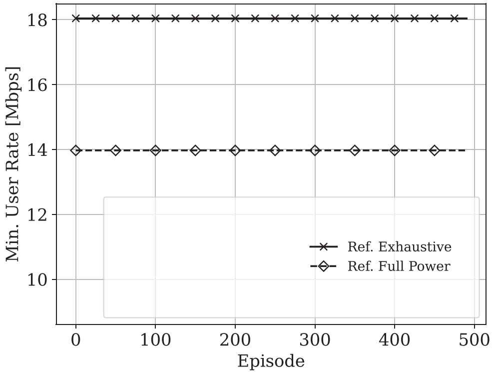
$$\left(\mathbf{s}_k^{(t)}, a_k^{(t)}, r_k^{(t+1)}, \mathbf{s}_k^{(t+1)}\right),$$

$$r_k^{(t+1)} = \min_{j \in \mathcal{N}_k} u_j^{(t)}$$

- States/rewards are determined over the **neighborhood** $\mathcal{N}_k$
- Interference is negligible if users are sufficiently separated
- Early DDQN layers are periodically shared with the CPU
- CPU aggregates users' layers and returns a federated model

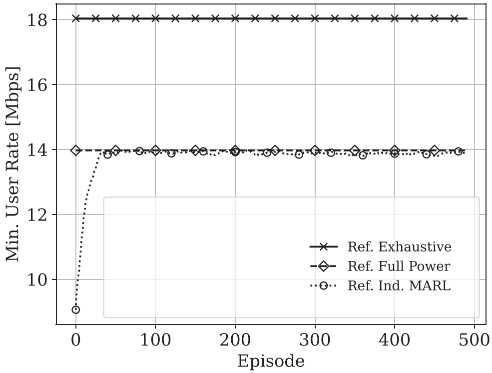


Comparison of DRL Frameworks – Static Scenario



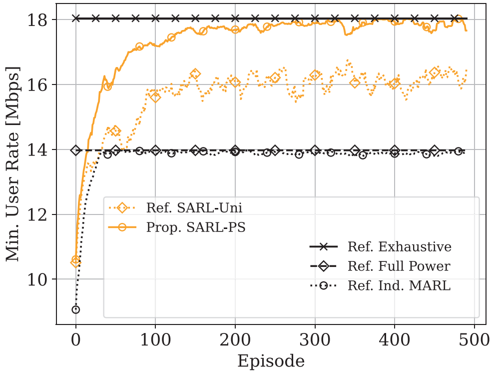
- Small-scale scenario to allow for exhaustive search (best case upper bound)
- Selfish behavior (full power transmission) leads to reduced guaranteed rate

Comparison of DRL Frameworks – Static Scenario



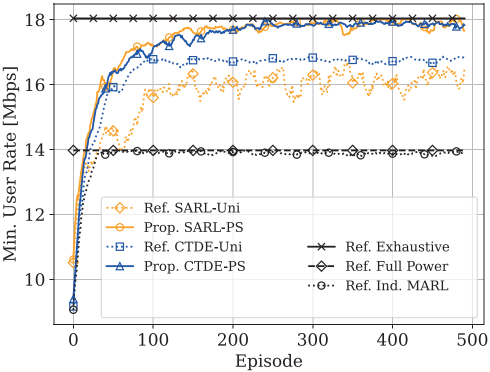
- Small-scale scenario to allow for exhaustive search (best case upper bound)
- Selfish behavior (full power transmission) leads to reduced guaranteed rate

Comparison of DRL Frameworks – Static Scenario



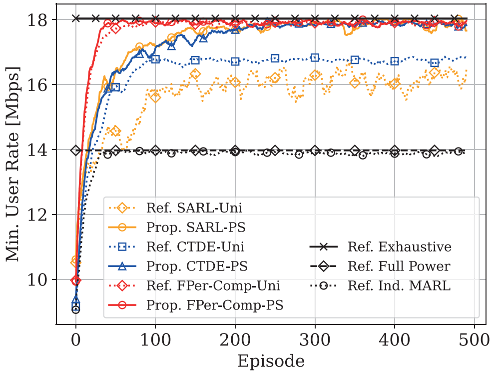
- Small-scale scenario to allow for exhaustive search (best case upper bound)
- Selfish behavior (full power transmission) leads to reduced guaranteed rate

Comparison of DRL Frameworks – Static Scenario



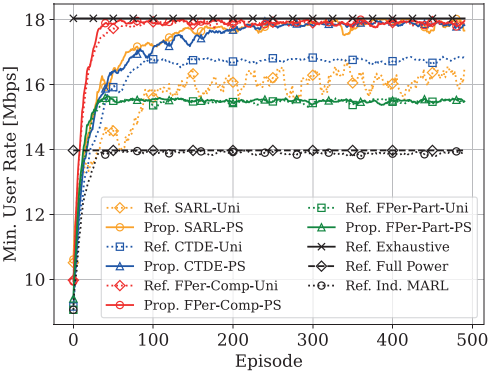
- Small-scale scenario to allow for exhaustive search (best case upper bound)
- Selfish behavior (full power transmission) leads to reduced guaranteed rate

Comparison of DRL Frameworks – Static Scenario



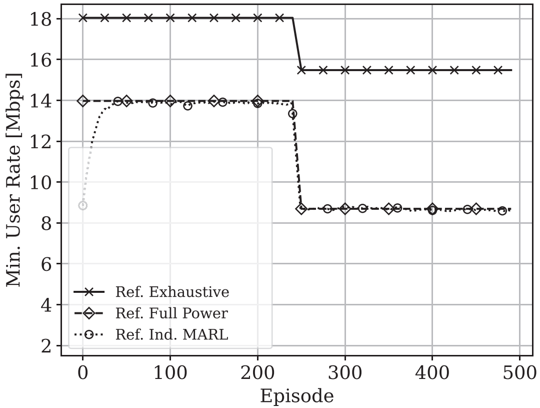
- Small-scale scenario to allow for exhaustive search (best case upper bound)
- Selfish behavior (full power transmission) leads to reduced guaranteed rate

Comparison of DRL Frameworks – Static Scenario



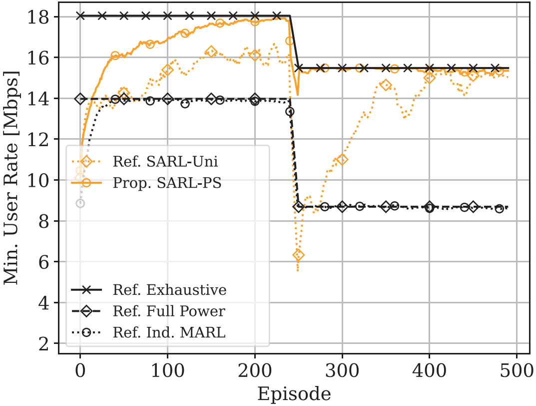
- Small-scale scenario to allow for exhaustive search (best case upper bound)
- Selfish behavior (full power transmission) leads to reduced guaranteed rate
- Considering a neighborhood of only 40% of closest users is here not sufficient (small scenario)

Comparison of DRL Frameworks – Transient Scenario



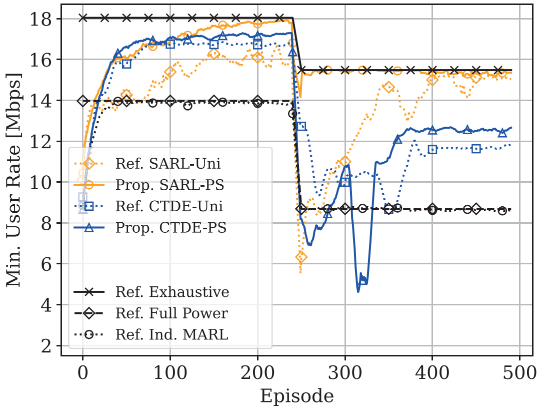
- Toggling of activation state of 20% of users after 250 episodes
- Personalized federated learning provides robust and fast adaptation capabilities

Comparison of DRL Frameworks – Transient Scenario



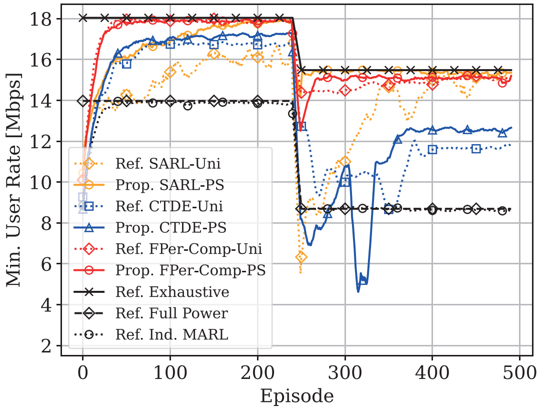
- Toggling of activation state of 20% of users after 250 episodes
- Personalized federated learning provides robust and fast adaptation capabilities

Comparison of DRL Frameworks – Transient Scenario



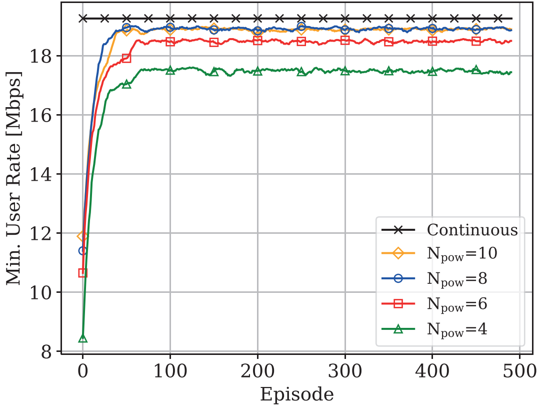
- Toggling of activation state of 20% of users after 250 episodes
- Personalized federated learning provides robust and fast adaptation capabilities

Comparison of DRL Frameworks – Transient Scenario



- Toggling of activation state of 20% of users after 250 episodes
- Personalized federated learning provides robust and fast adaptation capabilities

Impact of Number of Power Levels



- Performance close to continuous power allocation with modest number of discrete power levels

Remarks and Future Work

- The interference landscape is currently inferred from rate observations

  $\Rightarrow$ Makes it relatively difficult for the DNN to disentangle mutual inter-dependencies

  $\Rightarrow$ Acceptable when training in a DT, but too slow to adapt in direct real-world deployment

Remarks and Future Work

- The interference landscape is currently inferred from rate observations

  $\Rightarrow$ Makes it relatively difficult for the DNN to disentangle mutual inter-dependencies

  $\Rightarrow$ Acceptable when training in a DT, but too slow to adapt in direct real-world deployment

- Extend the state-space to provide additional information about mutual interference (path gains)
- Incorporate network structure into the DQN – graph neural networks (GNNs)

Remarks and Future Work

- The interference landscape is currently inferred from rate observations

  $\Rightarrow$ Makes it relatively difficult for the DNN to disentangle mutual inter-dependencies

  $\Rightarrow$ Acceptable when training in a DT, but too slow to adapt in direct real-world deployment
- Extend the state-space to provide additional information about mutual interference (path gains)
- Incorporate network structure into the DQN – graph neural networks (GNNs)
- Generalization and transferability across environments, user numbers, ...

RIS in Multi-Operator Environments

- RIS may be **integrated into various objects**

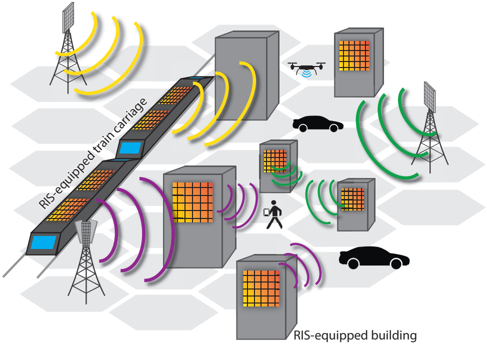
  ⇒ Network operators are unlikely to have a monopoly on their deployment

- RIS technology can potentially **support multiple frequency bands**

  ⇒ Not restricted to a single operator

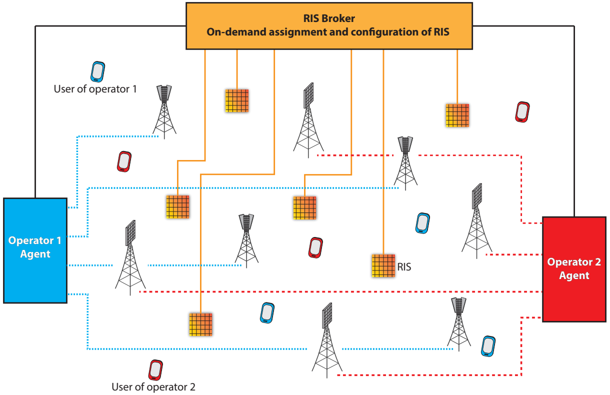
- **Who should** be allowed to **control the RIS response** configuration?

⇒ We propose a **competitive free-market** setup



Gambling on Reconfigurable Intelligent Surfaces, S. Schwarz, IEEE Communications Letters, vol. 28, no. 4, 2024

RIS Broking in Cell-free MIMO Setups



- RIS control is dynamically assigned to operators by a RIS broker
- RIS-to-operator assignment is achieved through an auction
- The auction is repeated whenever there are significant changes in demand or user positions

RIS Auction

- Simple auction format: **simultaneously ascending forward auction**
  - In auction-round t, RIS broker sets a uniform price $p_t > p_{t-1}$ for available RISs
  - Operators bid on RISs for which they are willing to pay the current price p_t
  - If only one operator bids on an RIS, it is assigned to this operator for payment p_t
  - If RISs are remaining, proceed to next round $t + 1$

RIS Auction

- Simple auction format: **simultaneously ascending forward auction**
  - In auction-round t, RIS broker sets a uniform price $p_t > p_{t-1}$ for available RISs
  - Operators bid on RISs for which they are willing to pay the current price p_t
  - If only one operator bids on an RIS, it is assigned to this operator for payment p_t
  - If RISs are remaining, proceed to next round $t+1$
  - Auctioneer enforces an **activity rule** – bidders cannot enter late

RIS Auction

- Simple auction format: **simultaneously ascending forward auction**
  - In auction-round t, RIS broker sets a uniform price $p_t > p_{t-1}$ for available RISs
  - Operators bid on RISs for which they are willing to pay the current price p_t
  - If only one operator bids on an RIS, it is assigned to this operator for payment p_t
  - If RISs are remaining, proceed to next round $t+1$
  - Auctioneer enforces an **activity rule** – bidders cannot enter late
- Challenges for operators:
  - How to **estimate the value of a RIS** and decide whether or not to pay price p_t?

    $\Rightarrow$ The value of a RIS depends on which other RISs can be secured (combinatorial)
  - How to design an **efficient bidding strategy**?

RIS Allocation – Utility and Value Estimation

- We employ the **α-fair function family** to quantify the utility of a RIS allocation $\mathcal{R}$

$$U^{(o)}(\mathcal{R}) = \frac{\sum_{u=1}^{N_{\mathrm{U}}^{(o)}} \left(\bar{r}_u^{(o)}(\mathcal{R})\right)^{1/\alpha}}{\sum_{u=1}^{N_{\mathrm{U}}^{(o)}} \left(\bar{r}_u^{(o)}(\emptyset)\right)^{1/\alpha}} - 1$$

$\ldots\ \bar{r}_u^{(o)}(\mathcal{R})$ estimate of achievable rate of user u

RIS Allocation – Utility and Value Estimation

- We employ the **α-fair function family** to quantify the utility of a RIS allocation $\mathcal{R}$

$$U^{(o)}(\mathcal{R}) = \frac{\sum_{u=1}^{N_{\mathrm{U}}^{(o)}} \left(\bar{r}_u^{(o)}(\mathcal{R})\right)^{1/\alpha}}{\sum_{u=1}^{N_{\mathrm{U}}^{(o)}} \left(\bar{r}_u^{(o)}(\emptyset)\right)^{1/\alpha}} - 1$$

  ... $\bar{r}_u^{(o)}(\mathcal{R})$ estimate of achievable rate of user u

  Calculated from macroscopic channel parameters (path gains, number of antennas and RIS elements), because microscopic fading channel is not known prior to RIS assignment

RIS Allocation – Utility and Value Estimation

- We employ the **α-fair function family** to quantify the utility of a RIS allocation $\mathcal{R}$

$$U^{(o)}(\mathcal{R}) = \frac{\sum_{u=1}^{N_{\mathrm{U}}^{(o)}} \left(\bar{r}_u^{(o)}(\mathcal{R})\right)^{1/\alpha}}{\sum_{u=1}^{N_{\mathrm{U}}^{(o)}} \left(\bar{r}_u^{(o)}(\emptyset)\right)^{1/\alpha}} - 1$$

  … $\bar{r}_u^{(o)}(\mathcal{R})$ estimate of achievable rate of user u

  Calculated from macroscopic channel parameters (path gains, number of antennas and RIS elements), because microscopic fading channel is not known prior to RIS assignment

- **Value of acquiring RIS r** in auction-round t

$$V_t^{(o)}(r) = U^{(o)}\left(\mathcal{R}_{t-1}^{(o)} \cup r\right) - U^{(o)}\left(\mathcal{R}_{t-1}^{(o)}\right)$$

  … assuming r is the sole secured RIS in round t – breaking combinatorial complexity

DRL-based Bidding

- **Observations** available to operators/agents

$$\mathcal{O}_t^{(o)} = \left(p_t, B_t^{(o)}, \left\{V_t^{(o)}(r) \middle| \forall r\right\}\right)$$

... only partial information; not the full state of the environment

- **Observations** available to operators/agents

$$\mathcal{O}_t^{(o)} = \left(p_t, B_t^{(o)}, \left\{V_t^{(o)}(r) \middle| \forall r\right\}\right)$$

... only partial information; not the full state of the environment

- **Reward** achieved when winning RISs $w_t^{(o)}$

$$r^{(o)} = c_V^{(o)} V_t^{(o)}\left(w_t^{(o)}\right) - p_t \left|w_t^{(o)}\right|.$$

DRL-based Bidding

- **Observations** available to operators/agents

$$\mathcal{O}_t^{(o)} = \left(p_t, B_t^{(o)}, \left\{V_t^{(o)}(r) \middle| \forall r\right\}\right)$$

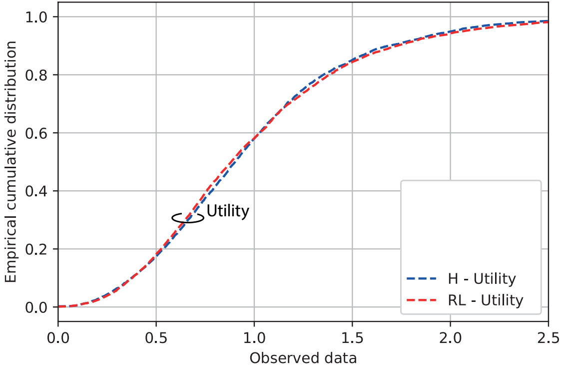
... only partial information; not the full state of the environment

- **Reward** achieved when winning RISs $w_t^{(o)}$

$$r^{(o)} = c_V^{(o)} V_t^{(o)}\left(w_t^{(o)}\right) - p_t \left|w_t^{(o)}\right|.$$

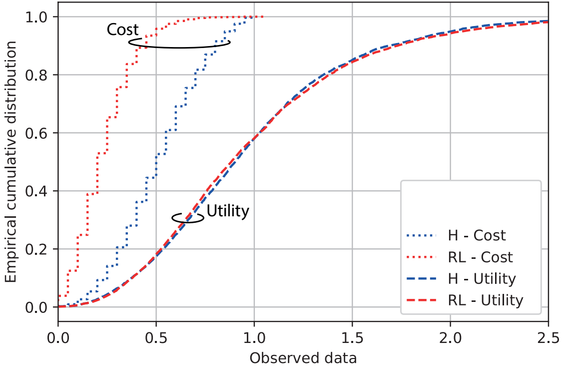
Penalty terms when bidding on already assigned RISs and when overshooting the budget

Investigation of Utility, Costs and Reward



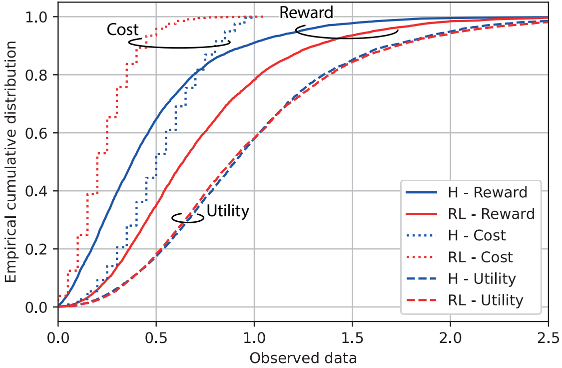
- Simple greedy bidding is a dominant strategy in terms of utility for each operator

Investigation of Utility, Costs and Reward



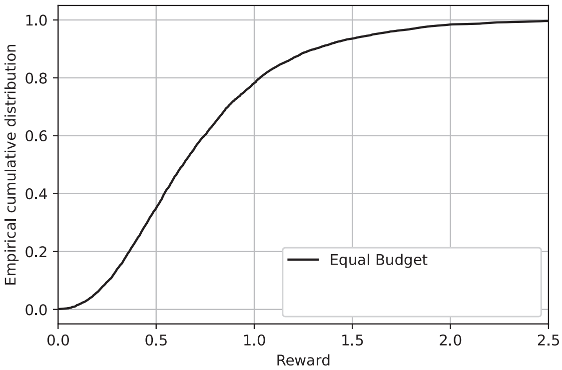
- Simple greedy bidding is a dominant strategy in terms of utility for each operator
- However, it is much more costly than DRL-based bidding

Investigation of Utility, Costs and Reward



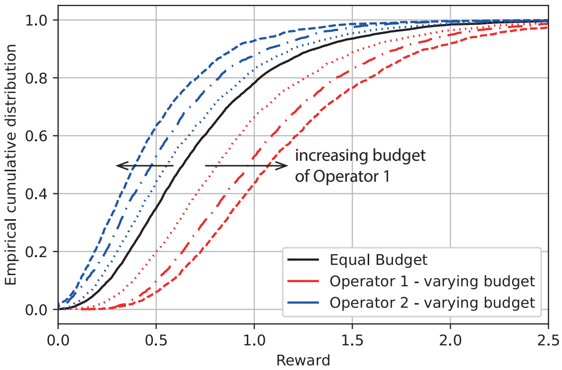
- Simple greedy bidding is a dominant strategy in terms of utility for each operator
- However, it is much more costly than DRL-based bidding
- Thus, DRL-based bidding achieves higher reward than greedy bidding

Investigation of Operators' Budgets



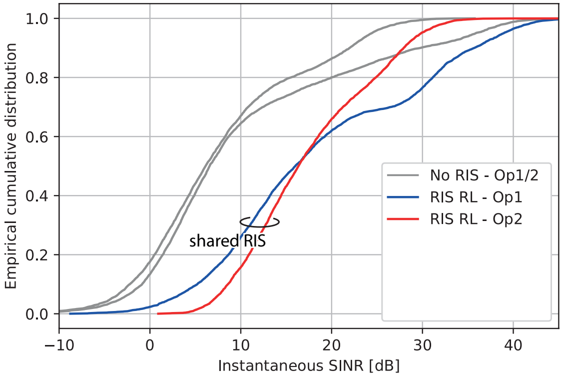
- With equal budgets both operators achieve the same performance for reasons of symmetry

Investigation of Operators' Budgets



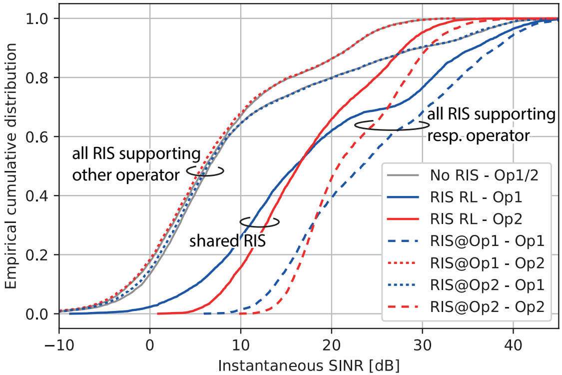
- With equal budgets both operators achieve the same performance for reasons of symmetry
- If one operator is willing to spend more, it can secure more RISs and therefore boost its performance

Investigation of Users' SINRs



- Single snapshot of positions of network elements; distribution over users and microscopic fading

Investigation of Users' SINRs



- Single snapshot of positions of network elements; distribution over users and microscopic fading
- Sharing RISs can significantly improve the performance of both operators
- If all RISs are assigned to one operator, the performance of the other remains virtually unaffected

Conclusions

- Multi-agent RL enables efficient decentralized, model-free optimization
- Real-world deployment can be improved through model-based pre-training or training within a digital twin
- Approaches to coordinating multiple agents include:
  - CTDE or FPer in cooperative scenarios with common goals
  - Game-theoretic mechanisms such as auctions in competitive scenarios

Multi-Agent Deep Reinforcement Learning for Cell-free MIMO Systems: From Distributed Power Allocation to Auction-Based RIS Access

11th Annual European Future of Wireless Technology Workshop

Associate Prof. Stefan Schwarz
in collaboration with: Charmae F. Mendoza, Prof. Markus Rupp and Prof. Megumi Kaneko

September 2025, stefan.schwarz@tuwien.ac.at

Technische Universität Wien

Institute of Telecommunications