Cooperative Scenarios For Multi-agent Reinforcement learning In Wireless Edge Caching

Navneet Garg, Tharmalingam Ratnarajah

School of engineering, The University of Edinburgh, UK.

ICASSP 2021

Outline

Outline

Introduction (1): Edge Caching

- Nowadays, digital devices (smartphones, tablets, and other IP connected devices) support high video data rates.
- Bottleneck remains from the network side for its inability to deliver the content in a timely manner [1].
  - Major cause is the **increasingly data traffic** due to an increased number of application services and devices.
  - Assessing traffic patterns reveals that the **repeated content requests** cause backhaul congestion and slower content delivery [2].
- To avoid these issues, **edge caches** at the base stations (BSs) are used in the future wireless networks, where BSs intelligently cache the contents based on the latest **content popularity** profiles [3].

[1]N. Wang et al. "Satellite Support for Enhanced Mobile Broadband Content Delivery in 5G". In: *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. 2018, pp. 1–6. DOI: 10.1109/BMSB.2018.8436902.

[2]L. Li, G. Zhao, and R. S. Blum. "A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies". In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 1710–1732.

[3]N. Garg et al. "Online Content Popularity Prediction and Learning in Wireless Edge Caching". In: *IEEE Transactions on Communications* 68.2 (2020), pp. 1087–1100. DOI: 10.1109/TCOMM.2019.2956041.

Introduction (2): RL

- Modeling popularity profiles as a finite state Markov chain, reinforcement learning (RL) strategies have been investigated:
  - **Function approximation** based RL methods are employed to reduce the storage and computational overhead [4].
  - A **deep RL** method is proposed for dynamic content updation in [5].
  - Towards secure edge caching, a **zero sum game framework** is formulated between an attacker and the content provider in [6], while mobile users acting as game-followers.
  - In [7], **hierarchical DQL** is used to predict demands and learn popularitities, i.e., decoupling the placement problem for BS and users.
  - In [8], under **network virtualization**, a BS is decoupled into a service provider and a virtual network operator, which leases resources to users.

[4]A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis. "Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-Time Popularities". In: *IEEE Journal of Selected Topics in Signal Processing* 12.1 (2018), pp. 180–190, N. Garg et al. "Function Approximation Based Reinforcement Learning for Edge Caching in Massive MIMO Networks". In: *IEEE Transactions on Communications* (2020), pp. 1–1. DOI: 10.1109/TCOMM.2020.3047658.

[5]P. Wu et al. "Dynamic Content Update for Wireless Edge Caching via Deep Reinforcement Learning". In: *IEEE Communications Letters* 23.10 (2019), pp. 1773–1777.

[6]Q. Xu, Z. Su, and R. Lu. "Game Theory and Reinforcement Learning Based Secure Edge Caching in Mobile Social Networks". In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3415–3429.

[7]Y. Qian et al. "Reinforcement Learning-Based Optimal Computing and Caching in Mobile Edge Network". In: *IEEE Journal on Selected Areas in Communications* 38.10 (2020), pp. 2343–2355.

Introduction (3): MARL

- MARL can effectively improve the cache placement performance with some cooperation.
  - [9] proposes a cognitive cache, which collaborates with **neighbors** to learn caching strategy.
  - In [10], a deep RL method is presented to promote **cooperation among edge servers** via a centralized remote server to improve the storage utilization.
  - In [11], considering **collaboration among computing and caching** resources, a double DQL approach is used for resource allocation, while [12] considers non-orthogonal multiple access (NOMA).
  - In [13], considering **trilateral-cooperation (macro-cell, roadside units and vehicles)** in vehicular edge computing, a joint content delivery and placement problem is formulated as a mixed-integer linear program (MILP), and nature-inspired deep deterministic policy gradient (DDPG) is used to obtain the suboptimal solution.

[9] M. Radenkovic and V. S. H. Huynh. "Cognitive Caching at the Edges for Mobile Social Community Networks: A Multi-Agent Deep Reinforcement Learning Approach". In: *IEEE Access* 8 (2020), pp. 179561–179574.

[10] Y. Zhang et al. "Cooperative Edge Caching: A Multi-Agent Deep Learning Based Approach". In: *IEEE Access* 8 (2020), pp. 133212–133224.

[11] J. Ren et al. "Collaborative Edge Computing and Caching With Deep Reinforcement Learning Decision Agents". In: *IEEE Access* 8 (2020), pp. 120604–120612.

[12] S. Li, B. Li, and W. Zhao. "Joint Optimization of Caching and Computation in Multi-Server NOMA-MEC System via Reinforcement Learning". In: *IEEE Access* 8 (2020), pp. 112762–112771.

[13] G. Qiao et al. "Deep Reinforcement Learning for Cooperative Content Caching in Vehicular Edge Computing and Networks". In: *IEEE Internet of Things*

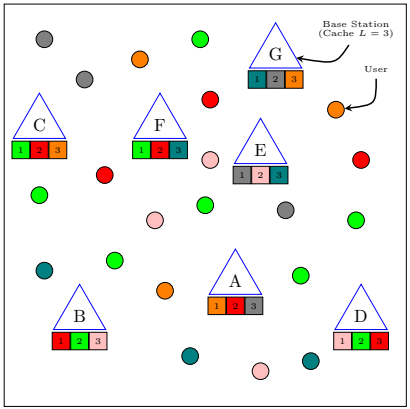
Overview of MARL approaches

- In this work, for a multi-cell system, MARL algorithms are investigated under four scenarios for cooperation.
  - Full cooperation (S1),
  - Episodic cooperation (S2),
  - Distributed cooperation (S3), and
  - Independent operation (no-cooperation, S4).

Outline

System model

- We consider a network of J SBSs in a given region, who fetch the contents cooperatively. Each SBS is equipped with a cache of size L units.
- Contents in the cache are chosen from a content library $\mathcal{F} = \{W_1, \ldots, W_F\}$, where each content is assumed to be of same size.
- In figure, each BS has a cache of $L = 3$ units, where different colors indicates different contents.



Normalized cache hit rate (1)

- Let $p_{jft} \in [0,1]$ denote the normalized demand (or popularity) of the f^{th} content at time t at the j^{th} SBS such that $\sum_{f \in \mathcal{F}} p_{jft} = 1$.
- For the uncoded cache placement, where the content in not splittable, let $c_{jft} \in \{0,1\}$ denote the status of the f^{th} content whether it is cached at j^{th} SBS in time slot t or not.
- Let $\mathbf{c}_{jt}^T = [c_{j1t}, \ldots, c_{jFt}] \in \{0,1\}^{1 \times F}$ such that $\mathbf{c}_{jt}^T \mathbf{1}_F = L$, where $\mathbf{1}_F$ defines an $F \times 1$ vectors of all ones.
- Then, for the popularity vector $\mathbf{p}_{jt}^T = [p_{j1t}, \ldots, p_{jFt}] \in [0,1]^{1 \times F}$, the normalized cache hit is defined as the sum of normalized demands served from the cached content and can be expressed for the j^{th} SBS as

$$\mathcal{H}_{local}(\mathbf{p}_{jt}, \mathbf{c}_{jt}) = \sum_{f \in \mathcal{F}} p_{jft} c_{jft}, \tag{1}$$

which represent the local cache hit for the j^{th} SBS[14].

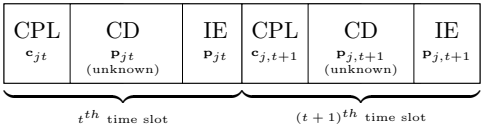
[14] $\mathbf{p}_{jt}$ is not known in advance.

Normalized cache hit rate (2)

- If the content is assumed coded, then $c_{jft} \in [0, 1]$.
- The average cache hit rate across network-wide can be written as $\mathcal{H}(\mathbf{P}_t, \mathbf{C}_t) = \frac{1}{J} \sum_j \mathcal{H}_{local}\left(\mathbf{p}_{jt}, \mathbf{c}_{jt}\right)$, where $\mathbf{C}_t = [\mathbf{c}_{1t}, \ldots, c_{Jt}]$ and $\mathbf{P}_t = [\mathbf{p}_{1t}, \ldots, \mathbf{p}_{Jt}]$.
- To observe the approximate value of $\mathcal{H}$, let $\mathbf{p}_{jt} = \frac{1}{F}\mathbf{1}_F$ (for uniformly popular contents). Then, we have

$$\mathcal{H}_{local}\left(\mathbf{p}_{jt}, \mathbf{c}_{jt}\right) = \frac{L}{F}. \tag{2}$$

Caching phases



- We consider a time slotted model [15].
- In the content placement (CPL) phase, each j^{th} BS updates their caches to the latest content popularity using their synced local RL strategy.
- In the content delivery (CD) phase, contents are delivered from edge caches to users as the requests arrive.
- In the information exchange (IE) phase, local demands are aggregated and communicated to the central BS based on the cooperation scheme selected.

[15]A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis. "Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-Time Popularities". In: *IEEE Journal of Selected Topics in Signal Processing* 12.1 (2018), pp. 180–190.

Objective (1)

- Our objective is to optimize the content placement at SBSs in order to maximize the long-term network-wide normalized cache hit rate as

$$\max_{\mathcal{C}_j \forall j} \sum_{i=t}^{\infty} \gamma^{i-t} \mathcal{H}(\mathbf{P}_t, \mathbf{C}_t) \tag{3a}$$

$$\text{subject to } \mathbf{C}_t^T \mathbf{1}_F = L \mathbf{1}_J, \forall t. \tag{3b}$$

- The above problem cannot be decoupled since the user content requests are correlated across SBSs and across content library [16].
- Also, $\mathbf{C}_t$ is decided based on previous popularity $\mathbf{P}_{t-1}$.

[16]S. Liu et al. "Distributed Caching Based on Matching Game in LEO Satellite Constellation Networks". In: *IEEE Communications Letters* 22.2 (2018), pp. 300–303. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2017.2771434, C. Chau, M. Khonji, and M. Aftab. "Online Algorithms for Information Aggregation From Distributed and Correlated Sources". In: *IEEE/ACM Transactions on Networking* 24.6 (2016), pp. 3714–3725. ISSN: 1063-6692. DOI: 10.1109/TNET.2016.2552083.

Objective (2)

- Further, the problem is difficult to solve as the search space increases exponentially with the content size and the number of SBSs $\binom{F}{L}^{J}$.
- Therefore, to analyze the overhead complexity due to multiple SBSs, the proposed cooperative multi-agent reinforcement learning algorithms are considered under four cooperative scenarios.

Outline

States and actions

- The content popularity at different time slots is assumed to follow Markov process [17] with unknown transition probabilities.
- A state at time t can be defined as the network content popularity at time t,

$$s_{jt} = \mathbf{p}_{j,t-1} \in \mathcal{S} = [0,1]^F .$$

- Similarly, a caching action in the time slot t is defined as the cache placement $a_{jt} = \mathbf{c}_{jt} \in \mathcal{A} = \{0,1\}^{F\times 1}$.

[17]A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis. "Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-Time Popularities". In: *IEEE Journal of Selected Topics in Signal Processing* 12.1 (2018), pp. 180–190.

Rewards

- At the end of time slot t, when popularity is revealed, the reward is computed, which is given for the j^{th} SBS as

$$r\left(s_{jt}, a_{jt}\right) = \mathcal{H}_{local}\left(\mathbf{p}_{j,t}, \mathbf{c}_{jt}\right). \tag{4}$$

- Since the action space cardinality is huge and is difficult to implement in practice, the objective of RL agent is take suboptimal decision steps and reach to the better solution sequentially.
- The cooperation among SBSs is coordinated by a central controller (CC).
- Four cooperation scenarios are as follows.

Scenario 1 (S1): full cooperation at each step

- Each SBS forward the **tuple** of states, actions, rewards and next-states (SARS).
- The CC gather these tuples (SARS) to **update** one global Q-matrix.
- Then, CC **broadcasts** the Q-matrix updates to each SBS.
- Since Q-values are updated at every step, this scenario is the most **resource-inefficient**, though providing the best performance.
- In this algorithm, each SBS benefits from the observations of the others, that is, J scouts are searching the optimum point with full inter-communications.
- The **updation step** of the critic Q-function can be written as for all $(s_*, a_*) \in \{(s_{jt}, a_{jt}), \forall j\}$

$$Q(s_*, a_*) \leftarrow (1 - \beta_t) Q(s_*, a_*) + \beta_t \left[r_*(s_*, a_*) + \gamma \max_b Q(s_*^{'}, b) - Q(s_*, a_*) \right],$$

where $s_*^{'}$ denote the next state.

MARL algorithm for S1

- Initialize $Q(s, a) = 0, \forall (s, a)$.
- For episode $= 1, \ldots, M$
  - receive an initial observation state $\mathbf{s}_t$
  - For step $= 1, \ldots, T$
    - select ϵ-greedy J actions at BSs, $\mathbf{a}_t$
    - observe the rewards $r_{jt}, \forall j$ and the next state $\mathbf{s}_{t+1}$
    - send SARS $(s_{jt}, a_{jt}, r_{jt}, s_{j,t+1}), \forall j$ to the CC
    - update $Q(s, a)$, $\forall (s, a) \in \{(s_{jt}, a_{jt}), \forall j\}$
  - EndFor
- EndFor

Scenario 2 (S2): episodic cooperation

- At each step of an episode, SBSs send SARS information to the CC.
- CC updates the global Q-matrix (Q_{global}) at each step.
- The local Q-matrices at SBSs (Q_{local}) are updated at the end of an episode via the global Q-matrix at the CC Q_{global}.
- As compared to Scenario 1, the advantage is the reduced cooperation messages.
- Since in S1, Q-function was updated at every step, thus, for T steps per episode, the cooperation complexity is reduced by a factor of $1/T$.

MARL algorithm for episodic cooperation (S2)

- Initialize $Q_{local}(s,a) = Q_{global}(s,a) = 0, \forall (s,a)$.
- For episode $= 1, \ldots, M$
  - receive an initial observation state $\mathbf{s}_t$
  - For step $= 1, \ldots, T$
    - select ϵ-greedy J actions $(\mathbf{a}_t)$ at SBSs via Q_{local}
    - observe the rewards $r_{jt}, \forall j$ and the next state $\mathbf{s}_{t+1}$
    - send SARS $(s_{jt}, a_{jt}, r_{jt}, s_{j,t+1}), \forall j$ to the CC
    - update $Q_{global}(s,a)$, $\forall (s,a) \in \{(s_{jt}, a_{jt}), \forall j\}$
  - EndFor
  - update $Q_{local} \leftarrow Q_{global}$.
- EndFor

Scenario 3: distributed cooperation

- This scenario further reduces the cooperation, and it does not require a central controller.
- In a distributed operation, a SBS can cooperate with its nearest SBS only.
- Each SBS has their local Q-matrices.
- At each step, each SBS shares the SARS information to one of the nearest SBSs only.
- Therefore, at each step, two SARS observations are used to update Q-values, (instead of J values from all SBSs in S1).

MARL algorithm for distributed cooperation (S3)

- Initialize $Q_j(s, a) = 0, \forall(s, a), \forall j$.
- For episode $= 1, \ldots, M$
  - receive an initial observation state $\mathbf{s}_t$
  - For step $= 1, \ldots, T$
    - For $j = 1, \ldots J$
      - select ϵ-greedy actions a_{jt} via Q_j
      - observe r_{jt} and $\mathbf{s}_{t+1}$
      - sync SARS $(s_{jt}, a_{jt}, r_{jt}, s_{j,t+1})$ to a neighbor
      - update local $Q_j(s, a)$ with two observations
    - EndFor
  - EndFor
- EndFor

Scenario 4: independent (no-cooperation)

- Here, each SBS operate independently, although the sets of states and actions are same.
- In other words, they do not utilize other experience and search on their own.
- There is no-cooperation or any exchange of SARS information or Q-values.
- Local Q-values at each SBSs are updated independently with their local SARS information.

MARL for independent operation (S4)

- Initialize $Q_j(s,a) = 0, \forall (s,a), \forall j$.
- For episode$= 1, \ldots, M$
  - receive an initial observation state $\mathbf{s}_t$
  - For step $= 1, \ldots, T$
    - For $j = 1, \ldots J$
      - select ϵ-greedy actions a_{jt} via Q_j
      - observe r_{jt} and $\mathbf{s}_{t+1}$
      - update local $Q_j(s,a)$
    - EndFor
  - EndFor
- EndFor

Cooperation comparison

S$*$	Per SBS per episode-	cooperation
S1	$4JT$ UL	$\lvert\mathcal{S}\rvert \lvert\mathcal{A}\rvert^J T$ DL
S2	$4JT$ UL	$\lvert\mathcal{S}\rvert \lvert\mathcal{A}\rvert^J$ DL
S3	$4T$ UL	$4T$ DL
S4	0 UL	0 DL

Table 1: Cooperation comparison (UL: uplink, DL: downlink, T: #steps per episode, J: #SBSs).

Outline

Simulation Settings

- Let $F = 100$ files in the content library,
  $L = 20$ cache size at each SBS.
- MARL algorithms are run for two setups:
  - $J = 3$ SBSs with $|\mathcal{S}| = 8$ and $|\mathcal{A}| = 128$ actions
  - $J = 7$ SBSs with $|\mathcal{S}| = 16$ and $|\mathcal{A}| = 256$ actions.
- Learning parameters (e.g., learning rate, discount factor, etc.) are assumed to be same.
- We have averaged the cache hits across SBSs and steps.
- For uniformly distributed random popularity profiles, the average cache hit rate is approximately $\frac{L}{F} = 0.2$ from eqn. (2).

Averaged rewards progress versus episodes (1): Uniform popularities

The normalized cache hit rate of

- S1 is the best.
- S3 $\approx$ S1, since $J = 3$ is small.
- S4 is the worst.
- S2 is better than S1 due to small $T = 400$.

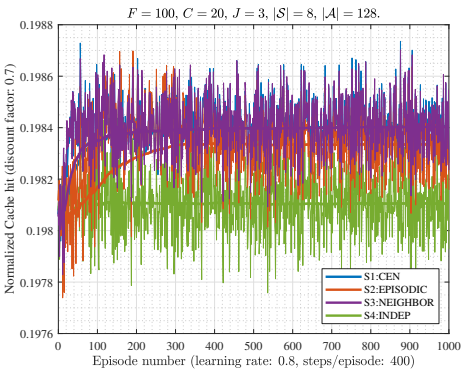


Figure 1: Averaged rewards progress versus episodes for $|\mathcal{S}| = 8$ and $|\mathcal{A}| = 128$ with $J = 3$ SBSs.

Averaged rewards progress versus episodes (2): Uniform popularities

The normalized cache hit rate of

- S1 is the best.
- the gap between S3 & S1 is larger since $J = 7$.
- S2 is worse than S4, since $T = 1500$ is large, and S4 has more frequent updates than S2.

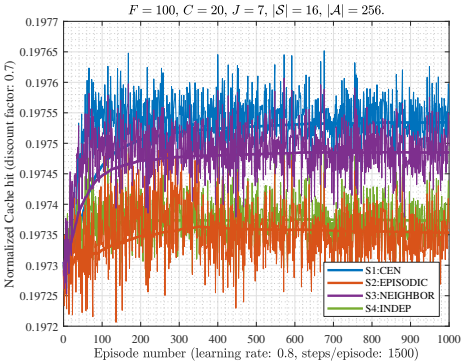


Figure 2: Averaged rewards progress versus episodes for $|\mathcal{S}| = 16$ and $|\mathcal{A}| = 256$ with $J = 7$ SBSs.

Averaged rewards progress versus episodes (3): Zipf popularities

The normalized cache hit rate of

- S1 is the best.
- S2 is worse than S4, since $T = 1500$ is large.
- The gap between S3 and S1 is small, due to Zipf popularities.
- S4 approximates S3, due to small $J = 3$ and Zipf popularities.

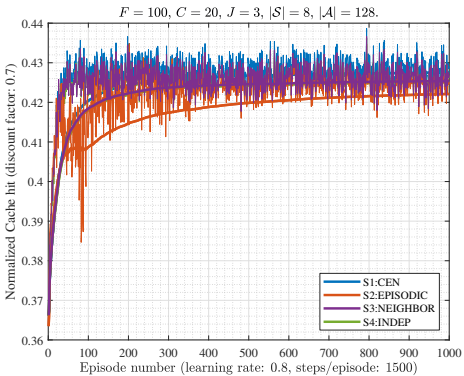


Figure 3: Averaged rewards progress versus episodes for $|\mathcal{S}| = 8$ and $|\mathcal{A}| = 128$ with $J = 3$ SBSs.

Averaged rewards progress versus episodes (4): Zipf popularities

The normalized cache hit rate of

- S1 is the best.
- S2 is worse than S4, since $T = 1500$ is large.
- The gap between S3 and S1 is small, due to Zipf popularities.
- S4 is closer to S3, due to Zipf popularities.

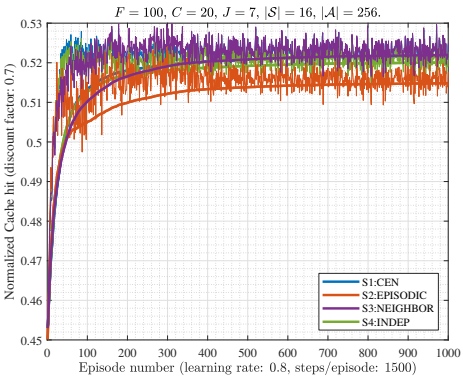


Figure 4: Averaged rewards progress versus episodes for $|\mathcal{S}| = 16$ and $|\mathcal{A}| = 256$ with $J = 7$ SBSs.

Outline

Conclusion

- In this work, we have studied four levels of cooperation for multi-agent Q-learning in wireless edge caching.
- Simulations results show that
  - When #SBSs are low, distributed cooperation with one neighbor (S3) can provide excellent improvements close to the full cooperation (S1).
  - When #SBSs are more, the performance gap between S1 & S3 increases.
  - S4 (no-cooperation) has better hit rates than that of S2 (episodic cooperation), due to only less frequent episodic updates (no-local updates).
  - Performance of S2 depends on the number of steps per episode (T).
    If T is less, more frequent updates lead to better performance of S2 than S4's.

Thank you

- Thank you.[18]

[18]The work was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/P009549/1, in part by the U.K.-India Education and Research Initiative Thematic Partnerships under Grant DST UKIERI-2016- 17-0060, and in part by the SPARC Project 148.