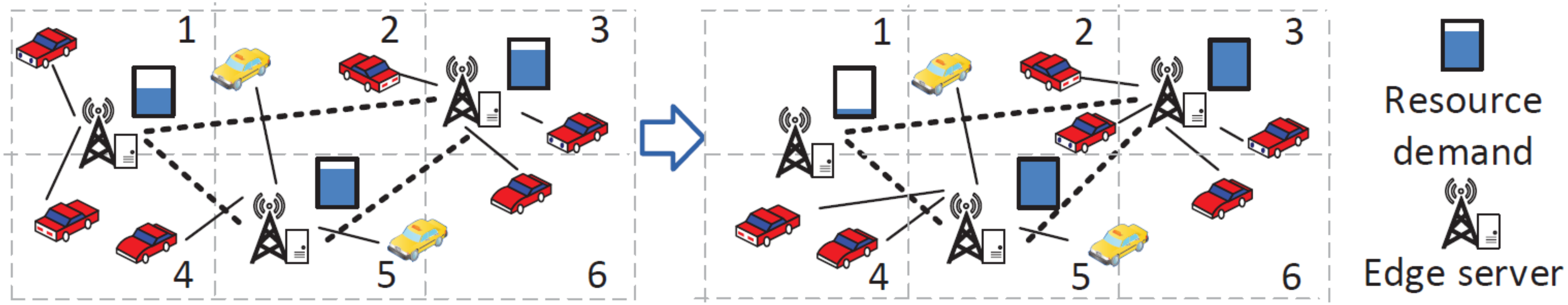
JOINT INSTITUTE
交大密西根学院

DeepReserve: dynamic edge server reservation for connected vehicles with deep reinforcement learning

Jiawei Zhang
Wireless Networking and Artificial Intelligence Lab
Advisor: Prof. Xudong Wang

- **Problem statement**
  - Computational resource demands change according to vehicle mobility



  - Edge servers should be dynamically reserved (rent or start) for providing service
    - Reduce cost for idle server
    - Ensure available server for nearby users
- **Challenges**
  - Reservation according to statistical demands is infeasible
    - Participants are **unwilling** to share routes
    - Statistical information **cannot be aligned** with real-time demands

Physical placement of edge servers

- With given demand information
  - Determine the positions of edge servers that are geographically close to users [1,2]
  - Determine locations of edge servers based on the number of user requests aggregated in nearby BSs [3]
  - Choose suitable edge servers to hold multiple interrelated services [4]
  - Determine the placement of multiple services into edge servers with heterogeneous capacities [5]

Difference: they determine **fixed placement** based on **given statistical demands**, while the edge-server reservation problem requires an **online solution** adaptive to the real-time demands

- Without demand information
  - Leverages the collected contexts of connected users (e.g., equipment types and external environment factors) to predict the demands [6]

Difference: we do not assume any context information of users

[1] H. Yin, et. al, "Edge provisioning with flexible server placement," *IEEE Trans. Parallel Distrib. Syst (**TPDS**), 2016*.
[2] Z. Xu , et. al, "Efficient algorithms for capacitated cloudlet placements," *IEEE Trans. Parallel Distrib. Syst (**TPDS**)*, 2015.
[3] S. Wang , et. al, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Computing*, 2019.
[4] I. Lera , et. al, "Availability-aware service placement policy in fog computing based on graph partitions," *IEEE Internet of Things J*, 2018.
[5] S. Pasteris , et. al, "Service placement with provable guarantees in heterogeneous edge computing systems," in *Proc. **INFOCOM***, 2019.
[6] L. Chen , et. al, "Spatio–temporal edge service placement: A bandit learning approach," IEEE Trans. Wireless Commun. ***(TWC)***, 2018.

Problem formulation

- ## Optimization problem
  - **Task**: dynamically reserve edge servers $x_{i,t}$
  - **Target:** maximize the system utility
    - ➢ (Reward of providing service) – (server cost) – (punishment of connection failure)

$$\boxed{\max \quad \sum_{i \in \mathcal{E}} (-\alpha x_{i,t} + \beta u_{i,t} - \gamma q_{i,t})}$$

  - **Constraints**
    - ➢ Latency $d_{i,j,t} y_{i,j,t} \leq D x_{i,t}$
    - ➢ Connect to at most one server $\sum y_{i,j,t} \leq 1$
    - ➢ Edge server capacity $\sum_{j \in \mathcal{V}_t} y_{i,j,t} \leq U x_{i,t}$
    - ➢ Connect to reserved server $y_{i,j,t} \leq x_{i,t}$
- ## Infeasibility of optimization method
  - This problem can be reduced to the **K-median** problem which is proved to be **NP-hard** [1]
  - Lack of parameters
    - ➢ Workload $u_{i,t}$ cannot be obtained before reservation
    - ➢ Latency $d_{i,j,t}$ cannot be measured, due to unknown CV location and network status

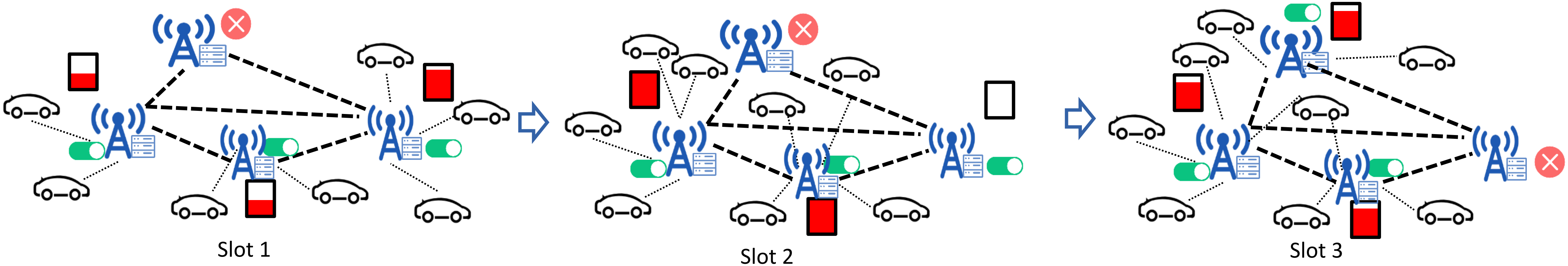
[1] M. Charikar, S. Guha, ´ E. Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the k-median problem," *J. Comput. And Syst. Sciences*, vol. 65, no. 1, pp. 129–149, 2002.

Facts observed from the system

- Vehicle distribution **reflects on workload** of edge servers
- Vehicles show **spatio-temporal** relation

Basic idea

- **Learn** from the spatio-temporal correlated workload information of edge servers to guide reservation



Basic tool

- Deep reinforcement learning

RL model design

- **RL model**
  - States: $\mathbf{s}_t = [u_{1,t} + q_{1,t}, \cdots, \underline{u_{E,t} + q_{E,t}}]$ → The number of **connected** and **denied** vehicles
  - Actions: $\mathbf{a}_t = [x_{1,t}, \cdots, x_{E,t}]$ → The **reservation** decision of edge server
  - Reward: $r_t = \sum_{i=1}^{E} (\underbrace{-\alpha x_{i,t}}_{\text{Cost of deploying edge servers}} + \underbrace{\beta u_{i,t}}_{\text{Reward from connected vehicles}} \underbrace{- \gamma q_{i,t}}_{\text{Punishment of connection failure}})$

- **Challenge in such an RL problem**
  - The system has **large state space** (e.g., workload of 1910 edge servers) and **action space** (up to 2^1910)
  - Traditional exploration process is hard to obtain sufficient "good" experience for a DRL agent [1]

[1] Xu, Zhiyuan, et al. "Experience-driven networking: A deep reinforcement learning based approach." in Proc. *IEEE INFOCOM*, 2018.

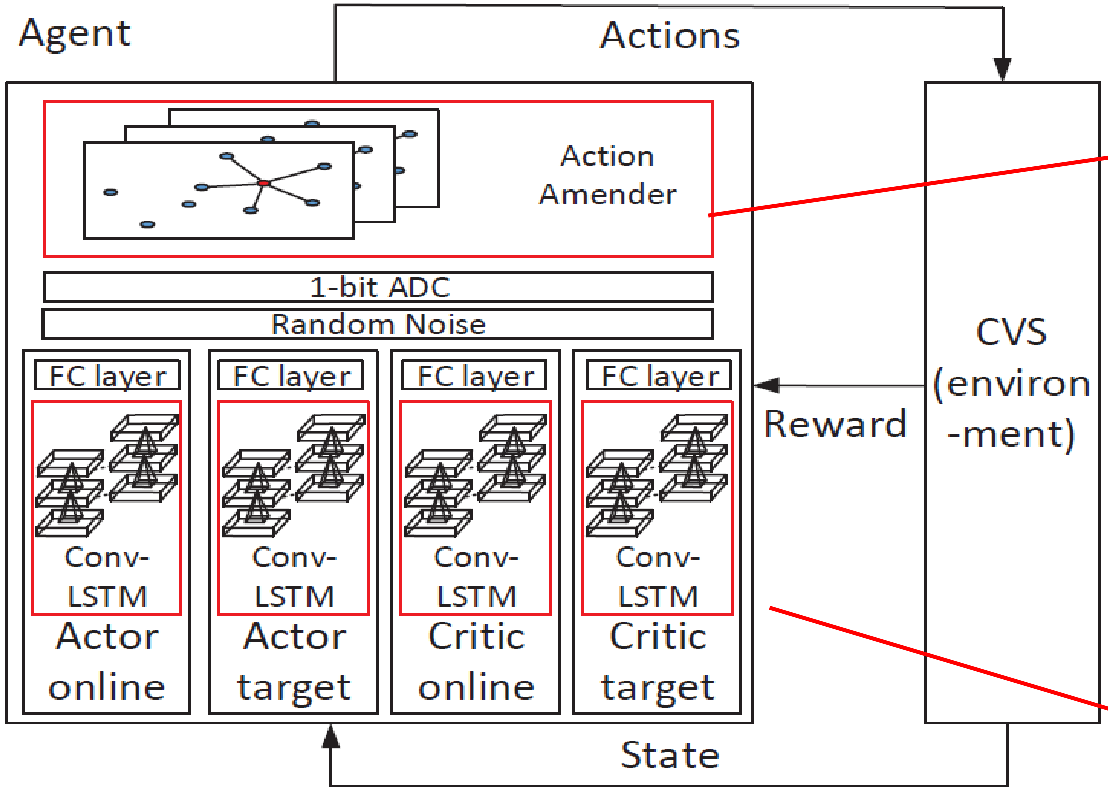
RL framework selection and remaining challenges

- **Framework: DDPG [1]**
  - Large action space → **policy gradient** → continuous action space
  - Large state space → **deep neural network** → replace large Q tables
  - Speedup training → **Actor Critic**

- **Remaining issues**
  - Fully-connected layers in DNN adopted by DDPG **do not encode spatial and temporal features**
    - output of DNN is not an accurate prediction of future states
  - **Random exploration** from the huge action space is unlikely to gather enough high-reward experiences
    - low system utility during exploration

[1] Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." in Proc. ICLR, 2015.

- **Two designs to improve DDPG**



Framework of DDPG based DRL

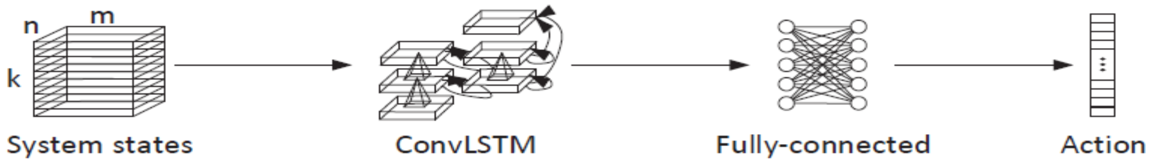
No vehicles in nearby edge servers

No vehicles recently

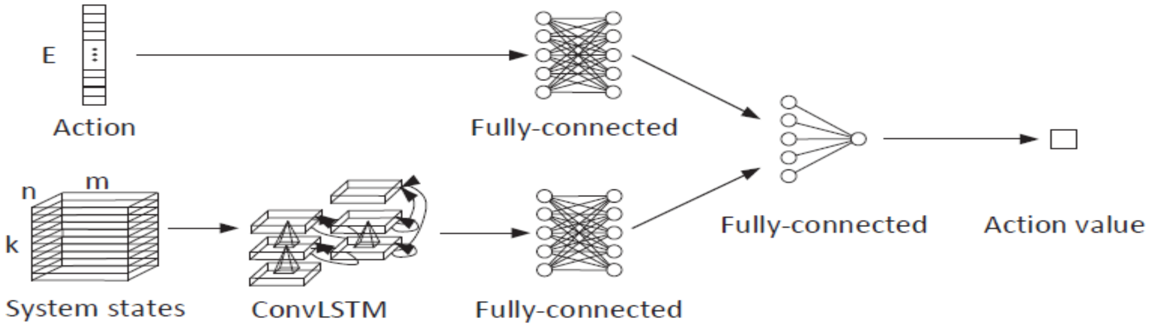
$$X_a:\quad x_{i,t} = x_{i,t}^{(\lambda)} H\Big[\sum_{j=0}^{l}(u_{i,t-j}+q_{i,t-j}) + \sum_{\delta\in\mathcal{E}_g}(u_{\delta,t}+q_{\delta,t})\Big]$$

Network-selected action

Technique 1: **action amender** to modify “bad” actions



(a) The actor network



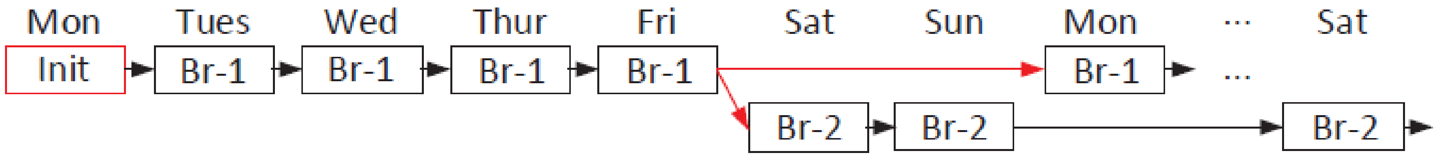
(b) The critic network

Technique 2: **ConvLSTM** to capture spatio-temporal information

[1] Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." in Proc. **NeurIPS**. 2015.

DR-Train to accelerate training

- **Two key designs**
  - Training initializer
    - Apply the greedy algorithm to initialize experience pools
    - Avoid “bad” experiences to deteriorate training
  - Forking model branches
    - To adapt to different traffic pattern



Algorithm 1 DR-Train

```
Randomly initialize critic online network Q_w(·) and actor online network μ_w(·) for weekdays with parameters θ_w^Q and θ_w^μ, respectively;
Initialize target networks Q'_w(·) and μ'_w(·) with parameters θ_w^{Q'} ← θ_w^Q and θ_w^{μ'} ← θ_w^μ, respectively;
Initialize a random process N and experience pools R_w and R_h for weekdays and weekends, respectively;
Receive initial observation state s_1;
for z in Z do
  if z is weekday then
    for t = 1 to T do
      if z is the first weekday then
        Select a_t according to the greedy algorithm;
```

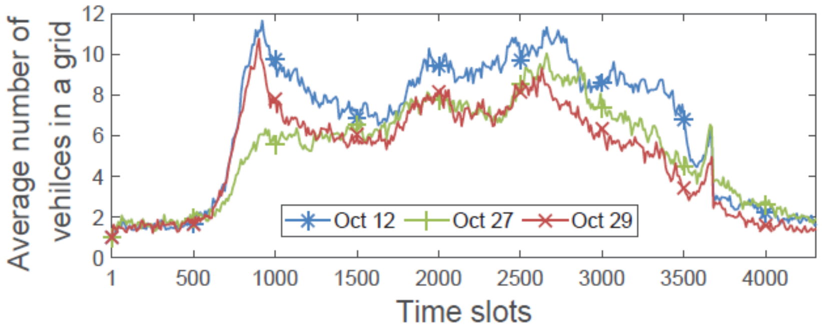
……

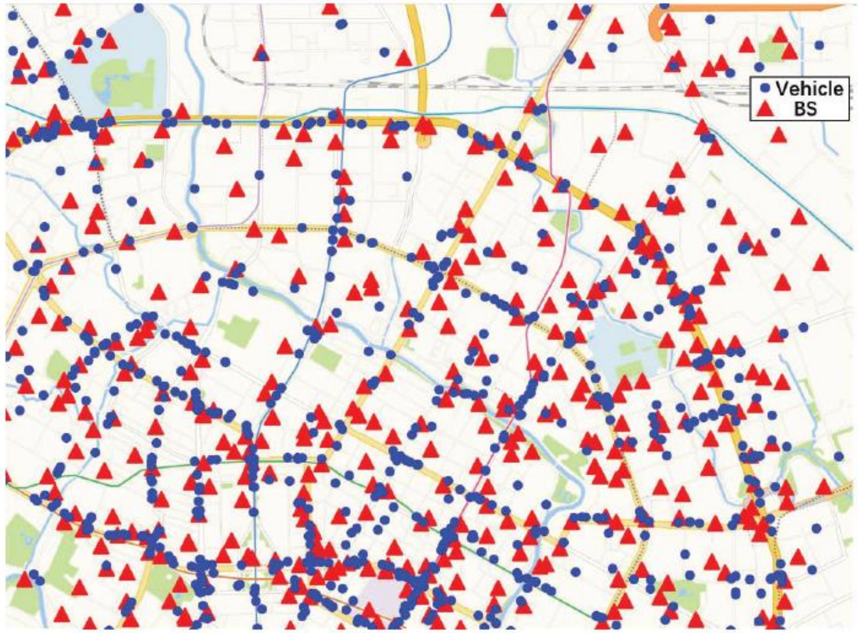
```
  else
    if z is the first weekend then
      Initiate the critic networks Q_h(·) and Q'_h(·) and
```

……

Experimental settings

- **Data sets**
  - Didi express in Chendu in 2018.10.8-11.30
  - Base station positions in Chendu

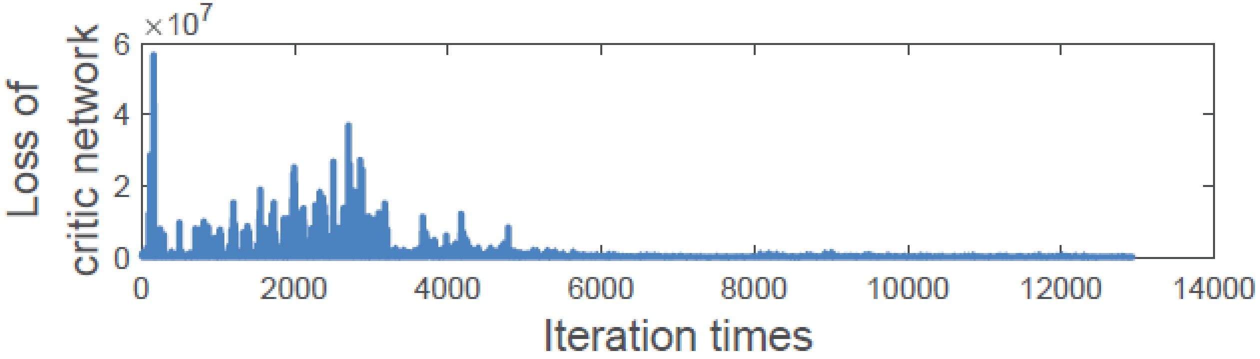




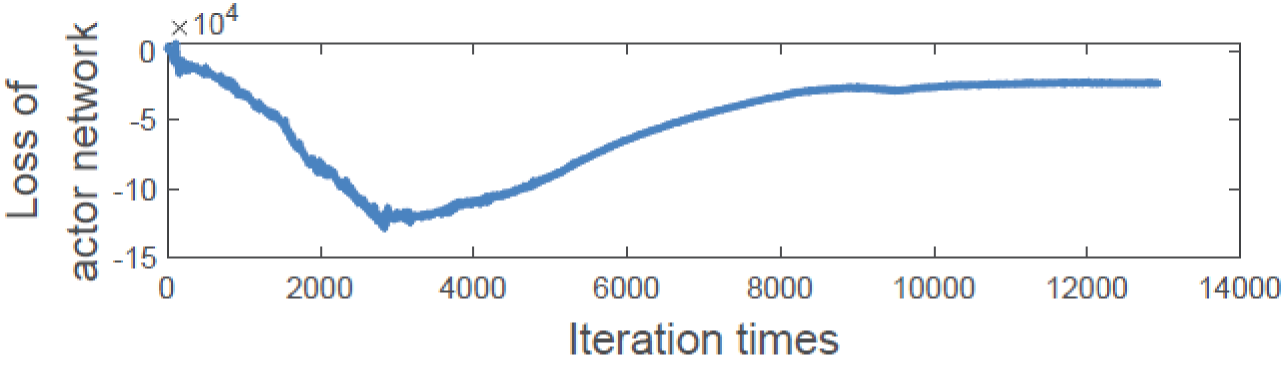
- **Metrics**
  - training loss of both the critic network and actor network
  - system utility
  - average resource utilization of reserved MEC servers
  - the probability of successful connections among all CVs
- **Benchmarks**
  - Variants: 1) DC (remove ConvLSTM), 2) DA (remove action amender), 3) DDPG
  - Server placement with demand information: 1) UC [1], 2) HAF [2], 3) GSP [3]
  - Others: 1) optimal, 2) random

[1] H. Yin, X. Zhang, H. H. Liu, Y. Luo, C. Tian, S. Zhao, and F. Li, "Edge provisioning with flexible server placement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 1031–1045, 2016.
[2] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," IEEE Trans. Cloud Computing, vol. 5, no. 4, pp. 725–737, 2015.
[3] T. He, H. Khamfroush, S. Wang, T. La Porta, and S. Stein, "It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources," in Proc. Int. Conf. Distributed Computing Systems (ICDCS). IEEE, 2018, pp. 365–375.

Experimental results: DR-Train

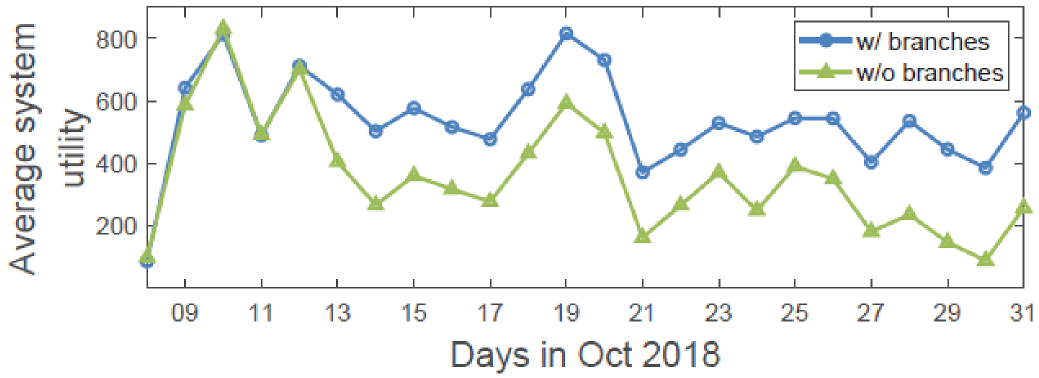


(a) Convergence of the critic network

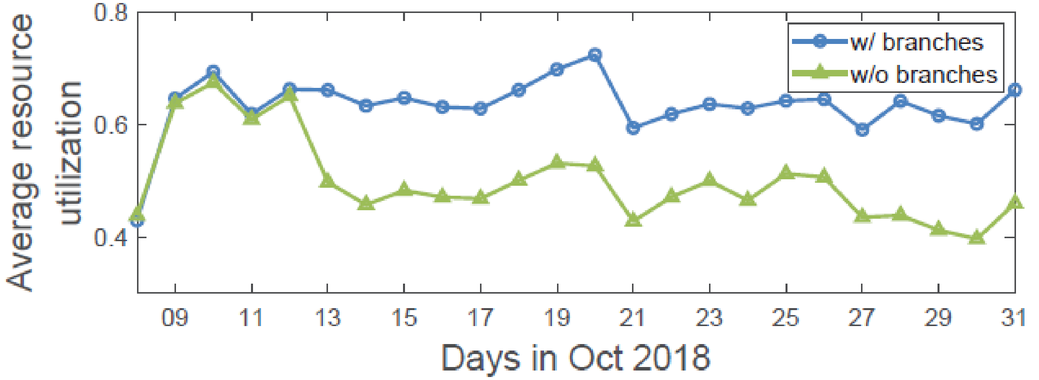


(b) Convergence of the actor network

Fig. 7. The training loss with experience-pool initialization.



(a) Average system utility



(b) Average resource utilization

Fig. 8. The effectiveness of dividing model branches.

Quickly converge to reduce inefficient tries

Improve performance with forking branches

Experimental results: variants of DeepReserve

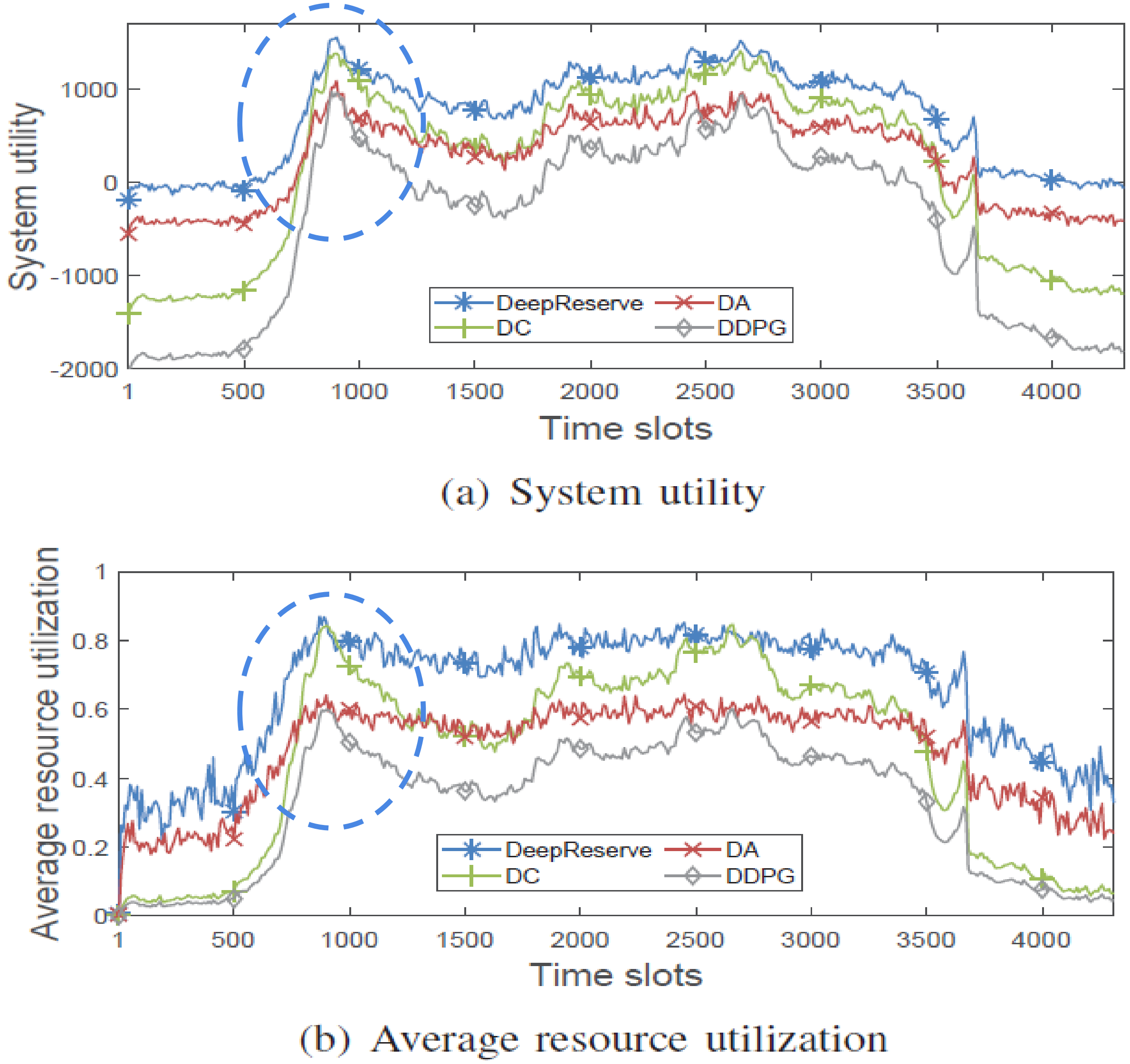
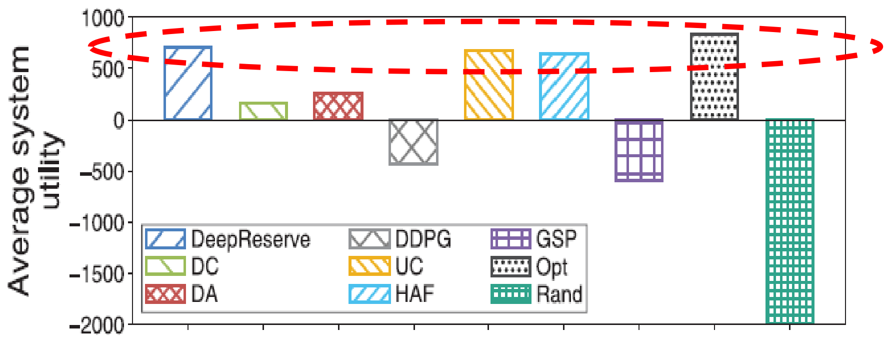


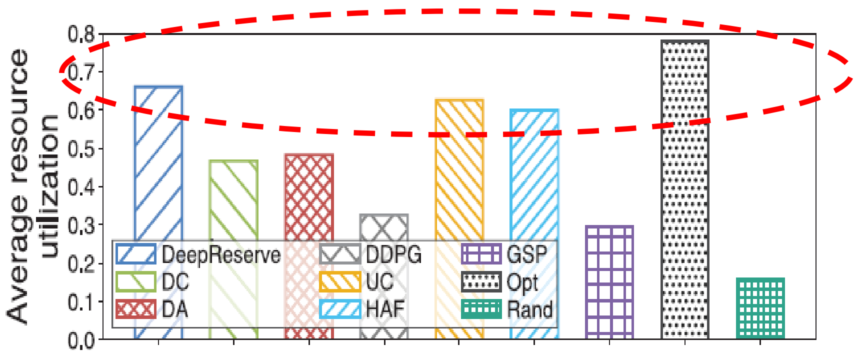
Fig. 11. The performance compared with different variants of DeepReserve.

Observation: action amender and ConvLSTM contribute differently

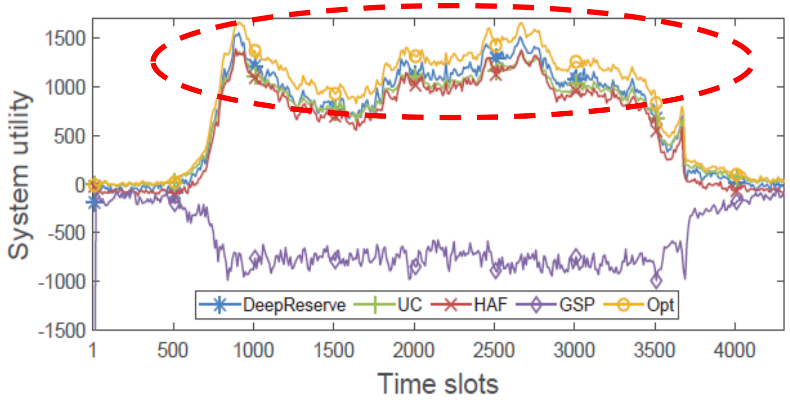
Experimental results: compared to benchmark approaches



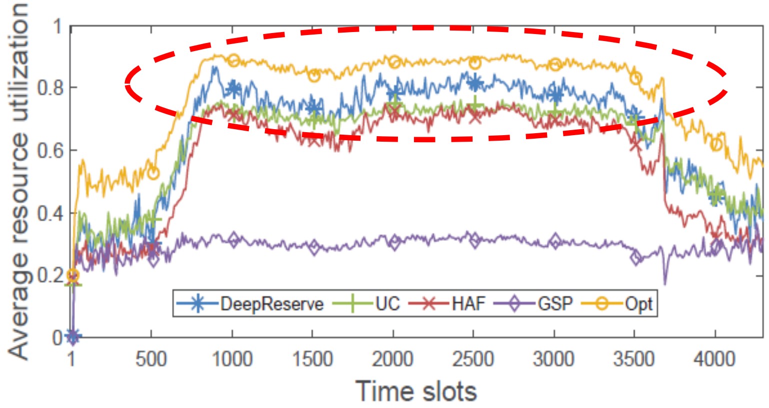
(a) Average system utility



(b) Average resource utilization



(a) System utility



(b) Average resource utilization

Fig. 10. The performance compared with benchmark approaches. Fig. 12. The performance compared with state-of-the-art approaches.

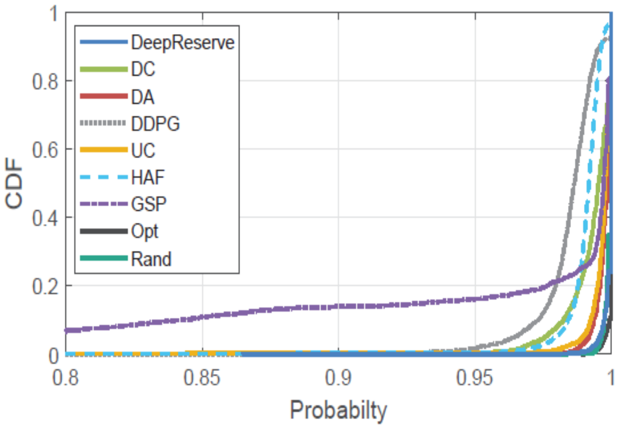


Fig. 13. The CDF plot of the probability of successful connection.

Near performance with optimal and the approaches with full demand information

Contributions

- The **system model** of edge computing based CV system is built and the edge-server reservation problem is **formulated**, which is proved to be NP-hard.
- A DRL based scheme called **DeepReserve** is developed, which is adapted from DDPG with two improvements, i.e., adopting ConvLSTM and the action amender. DeepReserve can efficiently learn to dynamically reserve edge servers without accurate demand information
- A training method called **DR-Train** is designed. Featured with two techniques, i.e., experience-pool initialization and model branches, DR-Train can stably train models for different vehicle traffic patterns.

Thank you !