On Mobile Edge Computing: Game Theory, Edge AI and Other New Ideas

Hai-Liang Zhao
hliangzhao97@gmail.com

January 8, 2019

© ***Please do not distribute this slide without permission.***

Outline

1. Game Theory and Its Applications
   - Theoretical Basis
   - Applications in MEC
   - My Contributions

Outline

Outline

1. Game Theory and Its Applications
   - Theoretical Basis
   - Applications in MEC
   - My Contributions

2. Edge AI
   - Existing frameworks on Edge Intelligence
   - Distributed Large-Scale Machine Learning

What we will talk in *Theoretical Basis*

1. What is a Congestion Game?
2. What sort of interactions do they model?
3. What good theoretical properties do they have?
4. What are Potential Games, and how are they related to congestion games?
5. How much time will be consumed to find a Nash Equilibrium†?
6. How to evaluate the inefficiency of `MyopicBestResponse` (the approach to obtain Nash Equilibrium)?

Definition

Each player chooses some subset from a set of resources, and *the cost of each resource* depends on the number of players who select it.

Definition of Congestion Game

A congestion game is a tuple $(\mathcal{N}, \mathcal{R}, \mathcal{A}, \mathcal{C})$, where

1. $\mathcal{N}$ is a set of n players;
2. $\mathcal{R}$ is a set of r resources;
3. $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times ... \times \mathcal{A}_n$, where $\mathcal{A}_i \subseteq 2^{\mathcal{R}} \backslash \varnothing$ is the set of actions / choices / strategies of player i (symmetric game);
4. $\mathcal{C} = (c_1, ..., c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in \mathcal{R}$ (nondecreasing with #? need to be monotonic?).

Utility function of every player:

1. Define $\# : \mathcal{R} \times \mathcal{A} \to \mathbb{N}$ as a function that counts the number of players who took any action that involves resource r under action profile $\boldsymbol{a}$.
2. Given an action profile $\boldsymbol{a} = (a_i, \boldsymbol{a}_{-i}), a_i \in \mathcal{A}_i$:

$$u_i(\boldsymbol{a}) = -\sum_{r \in \mathcal{A}_i} c_r(\#(r, \boldsymbol{a})).$$

Why we care about congestion games?

Theorem 1 (How to prove it?)

Every congestion game has at least one pure-strategy *Nash Equilibrium (NE)*.

Theorem 2 (Proof. with or without a potential function: 2 ways)

A sample procedure `MyopicBestResponse` is guaranteed to find a pure-strategy NE of a congestion game with finite steps.

PROCEDURE:

1. Start with an arbitrary action profile $\boldsymbol{a}$
2. **While** there exists an player i for whom a_i is not a best response to $\boldsymbol{a}_{-i}$
    1. $a'_i \leftarrow$ some best response by i to $\boldsymbol{a}_{-i}$
    2. $\boldsymbol{a} \leftarrow (a'_i, \boldsymbol{a}_{-i})$
3. **Return** $\boldsymbol{a}$

`MyopicBestResponse` returns a pure-strategy NE when terminates. (What about *general games?*)

Potential Games

Definition of (Exact) Potential Games

A game $G = (\mathcal{N}, \mathcal{A}, \mathcal{U})$ is a potential game if there exists a function $P : \mathcal{A} \to \mathbb{R}$ such that, $\forall i \in \mathcal{N}, \forall \boldsymbol{a}_{-i} \in \mathcal{A}_{-i}$ and $a_i, a_i' \in \mathcal{A}_i$,

$$u_i(a_i, \boldsymbol{a}_{-i}) - u_i(a_i', \boldsymbol{a}_{-i}) = P(a_i, \boldsymbol{a}_{-i}) - P(a_i', \boldsymbol{a}_{-i}).$$

Theorem 3

Every potential game has at least one pure-strategy NE.

Proof.

Let $\boldsymbol{a}^\star = \text{argmax}_{\boldsymbol{a} \in \mathcal{A}} P(\boldsymbol{a})$. Clearly $\forall \boldsymbol{a}' \in \mathcal{A} \backslash \{\boldsymbol{a}\}$, $P(\boldsymbol{a}^\star) \geq P(\boldsymbol{a}')$. Thus for any player i who can change action profile from $\boldsymbol{a}^\star$ to $\boldsymbol{a}'$ by changing his own action, $u_i(\boldsymbol{a}^\star) \geq u_i(\boldsymbol{a}')$.

Relationship between *CGs* and *PGs*

Theorem 4 (Theorem 1 established)

Every congestion game is a potential game.

1. Every congestion game has the potential function
$$P(\boldsymbol{a}) = \sum_{r \in \mathcal{R}} \sum_{j=1}^{\#(r,\boldsymbol{a})} c_r(j).$$
2. *Main Intuition*: Considering that player i changes action profile from $\boldsymbol{a}^\star$ to $\boldsymbol{a}'$ by changing his own action, most of the terms are canceled out when we take the difference, thus we have
$$\Delta u_i = \Delta P.$$
3. Actually, every potential game is 'isomorphic' a congestion game (Detailed proof based on constructing *Coordination and Dummy Games*† can be found at link, or define *pseudodelay* on each resource‡.).

`MyopicBestResponse` of Congestion Games

Theorem 2 (also can be proved by *Contradiction*: *link*, p_{114})

The `MyopicBestResponse` procedure is guaranteed to find a pure-strategy NE of a congestion game with finite steps.

Proof. with the properties of PGs

As we have proved, for a single player's strategy change we get $\Delta P = \Delta u_i$. Thus we can start from an arbitrary deterministic strategy $\boldsymbol{a}$ and at each step one player reduces his cost. Since P can accept a finite amount of values, it will eventually reach a local minima. At this point, no player can achieve any improvement, and we reach a NE. Thus, the `MyopicBestResponse` procedure is guaranteed to find a pure-strategy NE of a potential game. Combining with Theorem 4, q.e.d.

Conclusions:

1. Congestion game is a compact and intuitive way of representing interactions where players care about the number of others who choose a given resource, and their utility decomposes additively across these resources.
2. Potential game is a less-intuitive but analytically useful characterization equivalent to congestion game. (potential function P)

Conclusions on CGs (PGs)

`MyopicBestResponse` converges for CGs *regardless of*:

1. the cost functions (they do not need to be monotonic),
2. the action profile $\boldsymbol{a}$ with which the algorithm is initial,
3. which player's best responds to choose,
4. and even if we change *best response* to *better response*.

Complexity considerations:

1. The problem of finding a pure NE in a congestion game is PLS-complete (polynomial-time local search) (as hard as finding a local minimum in TSP using local search, PLS lies somewhere between P and NP)
2. It's resonable to expect `MyopicBestResponse` to be inefficient in the worst case
3. How to analysis the inefficient? (Price of Anarchy (PoA))

Extensions on *Nonsymmetric* Congestion Games

Theorem 5 (Finite Improvement Property (FIP))

Nonsymmetric congestion games involving only two strategies, i.e., $|\mathcal{A}| = 2$, is guaranteed to find the pure-strategy NE by the `MyopicBestResponse` procedure.

Theorem 6 (Conclusion on Player-specific Congestion Game)

For Nonsymmetric Congestion Games, if *each player only chooses only one responds*†, and *the cost received actually increases (not necessary strictly so) with the number of other players selecting the same resource*‡, there always exists a pure-strategy NE, while not generally admitting a potential function.

The detailed proof. of Theorem 6 can be found at *link*.

Extensions on Potential Games

Definition of Weighted Potential Games

$G = (\mathcal{N}, \mathcal{A}, \mathcal{U})$ is a weighted potential game if there exists a function $P : \mathcal{A} \to \mathbb{R}$ such that, $\forall i \in \mathcal{N}, \forall \boldsymbol{a}_{-i} \in \mathcal{A}_{-i}$ and $a_i, a_i' \in \mathcal{A}_i$,

$$w_i(u_i(a_i, \boldsymbol{a}_{-i}) - u_i(a_i', \boldsymbol{a}_{-i})) = P(a_i, \boldsymbol{a}_{-i}) - P(a_i', \boldsymbol{a}_{-i}),$$

where $\boldsymbol{w} = (w_i)_{i \in \mathcal{N}}$ is a vector of positive numbers.

Definition of Ordinal Potential Games

$G = (\mathcal{N}, \mathcal{A}, \mathcal{U})$ is a weighted potential game if there exists a function $P : \mathcal{A} \to \mathbb{R}$ such that, $\forall i \in \mathcal{N}, \forall \boldsymbol{a}_{-i} \in \mathcal{A}_{-i}$ and $\boldsymbol{a}_{-i} \in \mathcal{A}_i$,

$$u_i(a_i', \boldsymbol{a}_{-i}) - u_i(a_i, \boldsymbol{a}_{-i}) > 0 \Rightarrow P(a_i', \boldsymbol{a}_{-i}) - P(a_i, \boldsymbol{a}_{-i}) > 0,$$

where the opposite takes place for a minimum game.

It can be seen that *Exact Potential Games* and *Weighted Potential Games* are private cases of *Ordinal Potential Games*. ***Every finite ordinal potential game has a pure-strategy NE.***

Computing Equilibrium in Congestion Games

Definition of Symmetric Network's Game (NG)

Given a graph $G = (V, E)$ with source and destination vertices (S, T) (can be different for each palyer), the players have to choose a route on G leading from S to T. Each edge has a delay value which is a function of number of players using it.

Reamrk 1 For NG, the potential function $P(\boldsymbol{a}) = \sum_{e=1}^{M} \sum_{k=1}^{\#(e,\boldsymbol{a})} c_e(k)$ is exact.

Question How hard it is to find the equilibrium? Polynomial or exponential time?

Theorem 7 (Computation Complexity, already mentioned)

A general congestion game, symmetric congestion game, and asymmetric network game are all PLS-complete, even every $c_e(\cdot)$ is *linear*.

Detailed info. about *PLS class*†, *PLS-complete problems*‡ and the proof. of Theorem 7 can be found at link.

$\varepsilon - Nash$ of Congestion Game

Complexity Analysis of $\varepsilon - Nash$

For a general congestion game, it may be easier to find $\varepsilon - Nash$ instaed of deterministic Nash Equilibrium.

For a congestion game with the potential function being $\sum_{e=1}^{M} \sum_{k=1}^{\#(e,\boldsymbol{a})} c_e(k)$, apparently we have

$$P(\boldsymbol{a}) \leq \sum_{e=1}^{M} \#(e,\boldsymbol{a}) c_e(\#(e,\boldsymbol{a})) \leq n \cdot M \cdot c_{max}.$$

We start from an arbitrary deterministic strategy vector $\boldsymbol{a}$. At each step we decrease P at least ε, and if we can't, we reach a $\varepsilon - Nash$. Thus the number of steps is at most $\frac{P(\boldsymbol{a})}{\varepsilon}$, which is limited by $\frac{n \cdot M \cdot c_{max}}{\varepsilon}$.

This part can be used as ***Complexity Analysis***.

Price of Anarchy (PoA)

NE are inefficient because they generally do not minimize the social cost (maximize the social utility). In order to analyze the worst case, PoA is defined as

$$\text{PoA} = \min_{\boldsymbol{a}^{\dagger} \in set(NE)} \frac{u(\text{Equilibrium} : \boldsymbol{a}^{\dagger})}{u(\text{optimal} : \boldsymbol{a}^{\star})},$$

which is the ratio of the worst social cost of a Nash equilibrium to the cost of an optimal solution.

PoA analysis on *Nonatomic Congestion Games with Separable Cost*

Nonatomic $\equiv$ infiniteimal players, continuous flows

General settings In Nonatomic congestion games, all k players (indexed by $\mathcal{K}$) are infiniteimal, and the continuum of players of type i is represented by the interval $[0, n_i]$.

$\triangleright$ $\eta_{r,a_i} \geq 0$: the rate of consumption of a resource r by strategy $a_i \in \mathcal{A}_i$

$\triangleright$ $\boldsymbol{x} = (x_{a_i})_{a_i \in \mathcal{A}_i, i \in \mathcal{K}}$, $n_i = \sum_{a_i \in \mathcal{A}_i} x_{a_i}$: the strategy distribution

$\triangleright$ $x_r = \sum_{i=1}^{k} \sum_{a_i \in \mathcal{A}_i : r \in a_i} \eta_{r,a_i} x_{a_i}$: the utility rate of resource r

$\triangleright$ $c_r : \mathbb{R}_+^A \to \mathbb{R}_+$: the nondecreasing and continuous cost function of resource r

$\triangleright$ $c_{a_i}(\boldsymbol{x}) = \sum_{r \in a_i} \eta_{r,a_i} c_r(\boldsymbol{x})$: the total cost of using strategy a_i of player i

$\triangleright$ $C(\boldsymbol{x}) = \sum_{i=1}^{k} \sum_{a_i \in \mathcal{A}_i} c_{a_i}(\boldsymbol{x}) x_{a_i} = \sum_{r \in \mathcal{R}} c_r(\boldsymbol{x}) x_r$: the social cost

Definition A social optimal $\boldsymbol{x}^{\text{opt}}$ is a strategy distribution of minimum social cost;

Definition A strategy destination $\boldsymbol{x}^{\text{NE}}$ is a NE of the game.

PoA analysis on *Nonatomic Congestion Games with Separable Cost*

Theorem 8 (NE of nonatomic congestion games)

A strategy destination $\boldsymbol{x}_{\text{NE}}$ is a NE *iff* it satisfies

$$\forall \boldsymbol{x}, \sum_{r \in \mathcal{R}} c_r(\boldsymbol{x}^{\text{NE}})(\boldsymbol{x}_r^{\text{NE}} - \boldsymbol{x}_r) \leq 0.$$

Theorem 9 (PoA with *Affine* Cost Functions)

For nonatomic congestion games with separable, affine cost functions, i.e., $\forall r \in \mathcal{R}, c_r(\boldsymbol{x}) = \boldsymbol{h}_r \circ \boldsymbol{x}_r + b_r$, then $C(\boldsymbol{x}^{\text{NE}}) \leq 4/3 C(\boldsymbol{x}^{\text{opt}})$.

For nonatomic congestion games with *general cost functions*†, or *cost functions with limited congestion effects*‡, the results of PoA analysis are complicated. Details be can found at ▸ *link*.

Applications in MEC

▷ ***Potential games: theory and applications in wireless networks***, April 24, 2008.

Power control in cellular networks

Δ Shannon–Hartley theorem:

$$R_i = \omega \log_2(1 + SINR),$$
$$SINR = \frac{g_i p_i}{\varpi + \sum_{j \neq i} g_j p_j}$$

Power control in Cognitive Radio Networks

Δ Utility function:

$$u_i(\boldsymbol{A}) = -|\hat{\gamma} - \frac{g_i a_i}{\frac{1}{K}\{\sum_{j \neq i} g_j a_j + \varpi\}}|$$

Δ Potential function:

$$\begin{aligned} P(\boldsymbol{A}) &= 2\hat{\gamma}/K(\sum_i \sum_{k>i} g_i p_i g_k p_k) \\ &+ \sum_i (-g_i^2 p_i^2 + 2\hat{\gamma} g_i p_i / K) \end{aligned}$$

Applications on MEC

▷ ***Price of Anarchy for Congestion Games in Cognitive Radio Networks***, IEEE Trans. on Wireless Communications, Oct 2012. ▸ *link*
▷ ***Decentralized Computation Offloading Game for Mobile Cloud Computing***, IEEE Trans. on PDS, Apr 2015. ▸ *link* [Xu Chen]

Single-channel CDMA

△ Uplink data rate: $R_n(\boldsymbol{a}) = W\log_2(1 + \frac{P_n H_{n,s}}{\omega_n + \sum_{m\in\mathcal{N}:a_m=1} P_m H_{m,s}})$
△ Potential function:

$$\begin{aligned}\Phi(\boldsymbol{a}) &= \frac{1}{2}\sum_{n\in\mathcal{N}}\sum_{m\neq n} P_n H_{n,s} P_m H_{m,s} I_{\{a_n=1\}} I_{\{a_m=1\}} \\ &+ \sum_{n\in\mathcal{N}} P_n H_{n,s} L_n I_{\{a_n=0\}}\end{aligned}$$

Applications on MEC

▷ ***Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing***, IEEE/ACM Trans. on Networking, Oct 2015. ▸ *link* [Xu Chen]

Multi-channel CDMA

Δ Uplink data rate: $R_n(\boldsymbol{a}) = W \log_2(1 + \frac{P_n H_{n,s}}{\omega_n + \sum_{m \in \mathcal{N}: a_m = a_n} P_m H_{m,s}})$

Δ Potential function:

$$\begin{aligned}\Phi(\boldsymbol{a}) &= \frac{1}{2} \sum_{n \in \mathcal{N}} \sum_{m \neq n} P_n H_{n,s} P_m H_{m,s} I_{\{a_n = 1\}} I_{\{a_m = a_n\}} \\ &+ \sum_{n \in \mathcal{N}} P_n H_{n,s} L_n I_{\{a_n = 0\}}\end{aligned}$$

Applications on MEC

▷ ***A Game-Theoretical Approach for User Allocation in Edge Computing Environment***, IEEE Trans. on PDS, under reviewing. [Qiang He]

EUA Game

Δ Potential function:

$$\begin{aligned}\Phi(\boldsymbol{a}) \quad &= \quad -\frac{1}{2}\sum_{i\in\mathcal{N}}\sum_{j\neq i}\sum_{k\in\mathcal{D}}\lambda_i^k f^k(s_{a_i})\omega_i^k\lambda_j^k f^k(s_{a_j})\omega_j^k I_{\{a_i=a_j\}}I_{\{a_i>0\}} \\ &- \quad \sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{D}}\lambda_i^k f^k(s_{a_i})\omega_i^k \cdot T_i I_{\{a_i=0\}}\end{aligned}$$

Applications on MEC

▷ ***Follow Me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing***, IEEE Journal on Selected Areas in Communications, Oct 2018. ▸ *link* [Xu Chen]

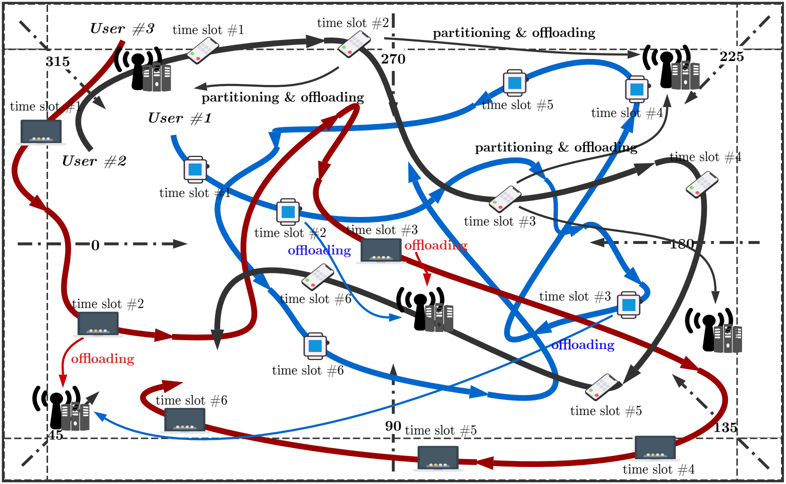
Congestion Game with no potential function provided

$$\mathcal{P}: \min_{c(t)} \sum_{k=1}^{N} \sum_{i=1}^{M} x_i^k(t) \left(\frac{V R^k(t) \sum_{k=1}^{N} x_i^k(t)}{F_i} + V H_i^k(t) + \rho_i^k(t) \right).$$

Δ Methods: Lyapunov Optimization, Markov Approximation, Algorithmic Game Theory.

They are all the same! :-(

My Contribution: *Cross-edge* Computation Offloading Framework



Cross-Edge Computation Offloading

Edge site-selection subproblem

$$\mathcal{P}_2^{es}: \quad \min_{\forall i, \mathbf{I}_i(t)} \left\{ -\sum_{i=1}^{N} \psi_i'(t)\left[\epsilon_i^l + \sum_{j\in\mathcal{M}_i(t)} \epsilon_{i,j}(t) I_{i,j}(t)\right] - V\cdot\sum_{i\in\mathcal{N}} \mathcal{U}_i(t)\right\}$$

$$s.t. \quad \sum_{i\in\mathcal{N}} I_{i,j}(t) \le N_j^{max}, j\in\mathcal{M}_i(t), t\in\mathcal{T},$$

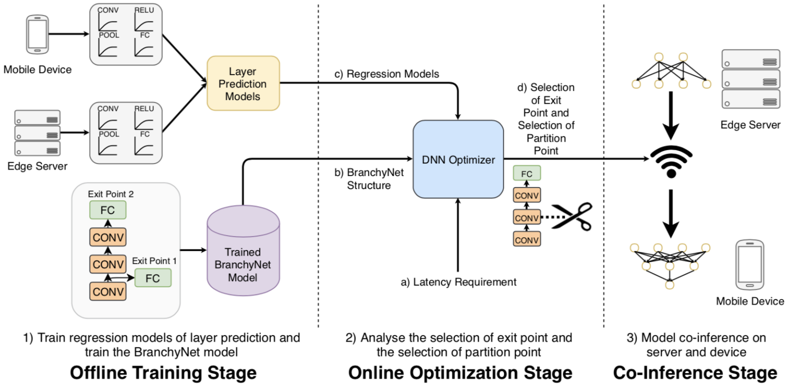
$$\tau_d \ge \max_{j\in\mathcal{M}_i(t)} \left\{\tau_{i,j}^{tx}(t) + \tau_{i,j}^{rc}(t)\right\} + \tau_i^{lc} + \varphi\cdot\sum_{j\in\mathcal{M}_i(t)} I_{i,j}(t).$$

△ What's the category of the game? (Nonsymmetric? Nonatomic?)
△ What's the cost function of each resource (edge site)?
△ What's the potential function of this game?
△ The vanilla version of Network Congestion Games can not be applied directly.

Outline

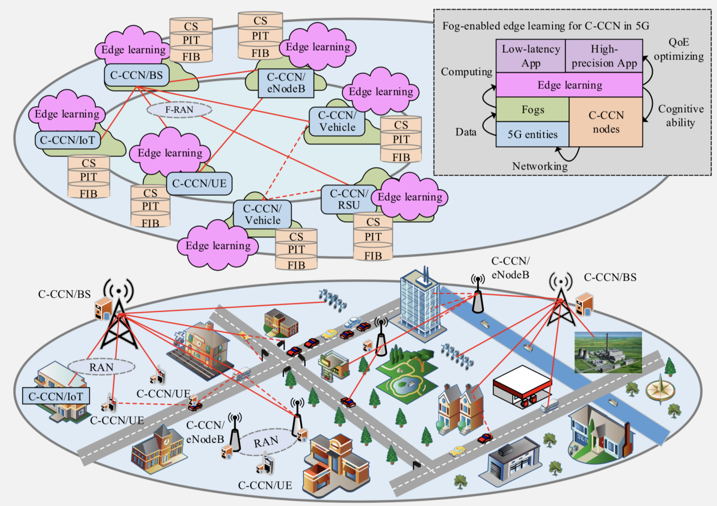
Edge AI - Existing frameworks

▷ ***Edge Intelligence: On-Demand Deep Learning Model Co-Inference with Device-Edge Synergy***, Jun 2018, under reviewing. *link* [Xu Chen]

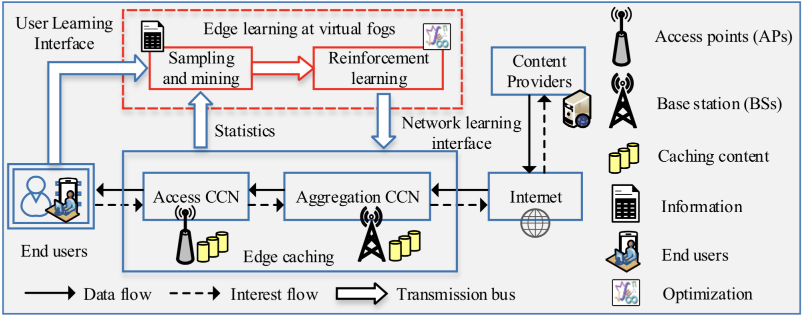


Edge AI - Existing frameworks

▷ ***Fog-enabled Edge Learning for Cognitive Content-Centric Networking in 5G***, Aug 2018, under reviewing. ▸ *link* [SJTU]

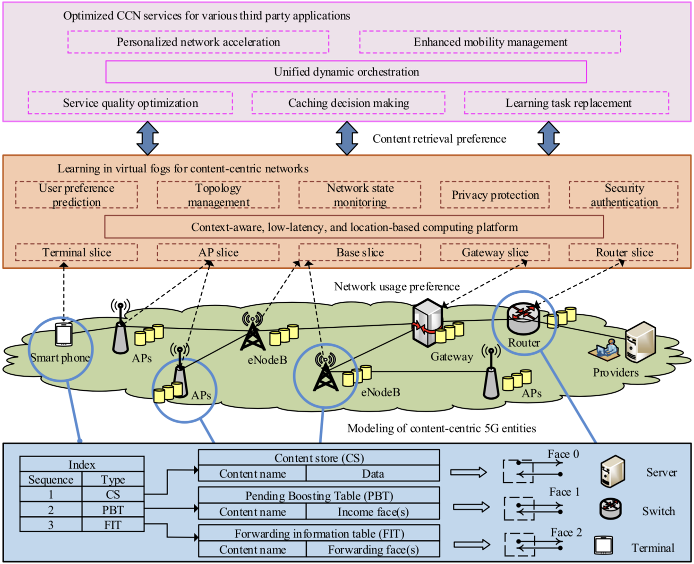


Fog-enabled Edge Learning for Cognitive Content Centric Networking in 5G

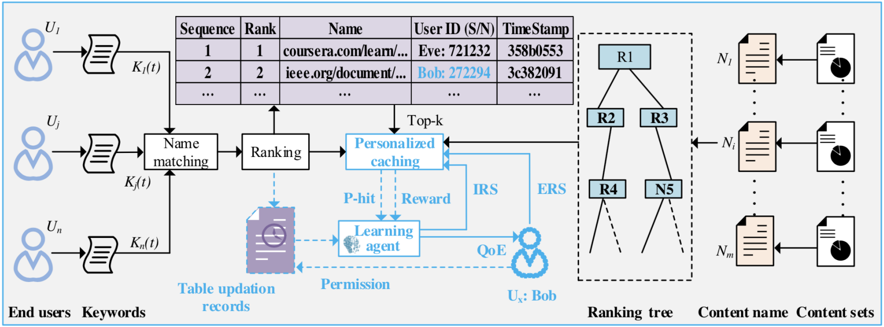


The system model of fog-enabled edge learning for cognitive CCN in 5G

5G FEL Framework

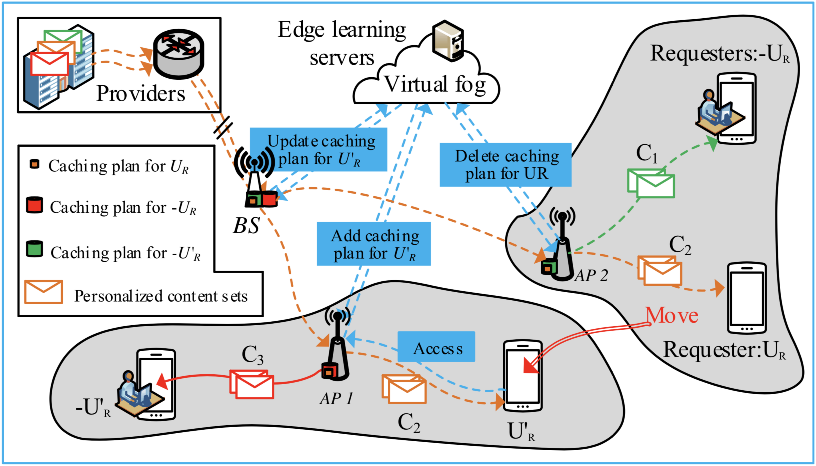


5G FEL Framework



FEL-based personalized CCN acceleration in 5G

5G FEL Framework



Fog-enabled FEL CCN mobility management

Edge AI - Existing frameworks

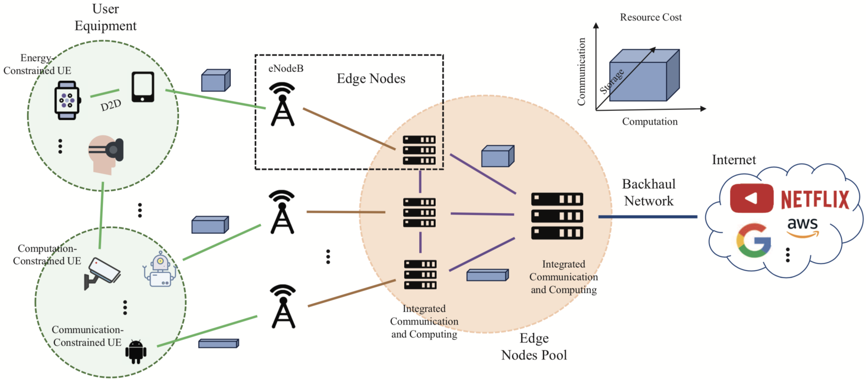
▷ ***Towards an Intelligent Edge: Wireless Communication Meets Machine Learning***, Sep 2018, under reviewing. ▸ *link* [HKU, HKUST]

Commun. Tech.	Item	Conventional Commun.	Learning-Driven Commun.
Multiple access (Section II)	Target	Decoupling messages from users	Computing func. of distributed data
	Case study	OFDMA	Model-update averaging by AirComp
Resource Allocation (Section III)	Target	Maximize sum-rate or reliability	Fast intelligence acquisition
	Case study	Reliability-based retransmission	Importance-aware retransmission
Signal Encoding (Section IV)	Target	Optimal tradeoffs between rate and distortion/reliability	Latency minimization while preserving the learning accuracy
	Case study	Quantization, adaptive modulation and polar code	Grassmann analog encoding

Learning-driven Communication, detailed slide: ▸ *link*

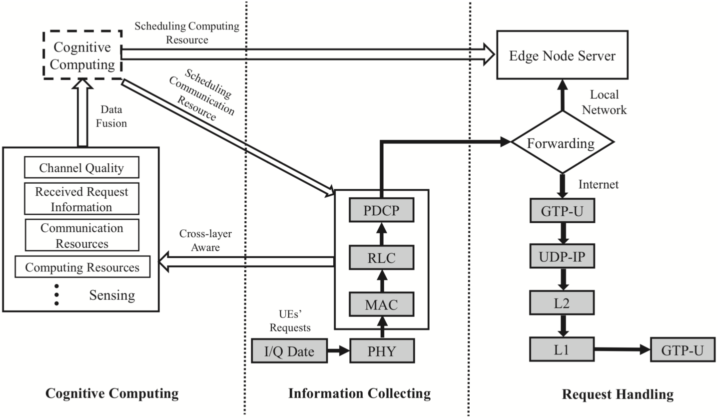
Edge AI - Existing frameworks

▷ ***In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning***, Sep 2018, under reviewing. ▸ *link* [TJU, HUAWEI]



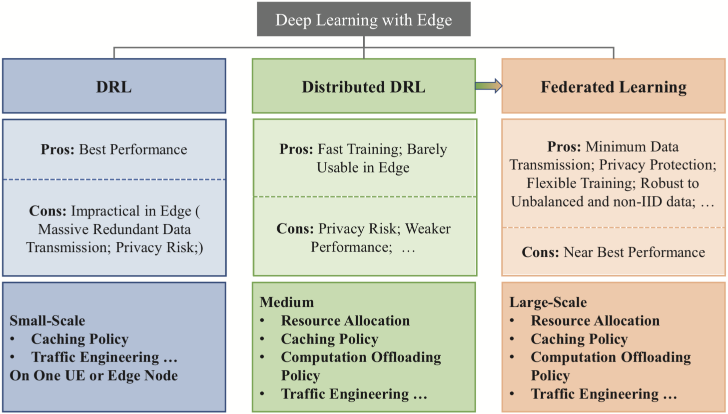
Framework of AI-supported mobile edge system with cognitive ability

In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning



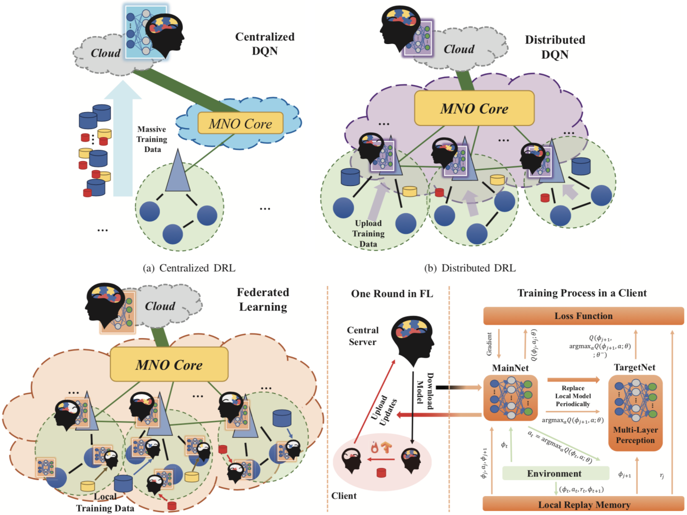
Procedure of utilizing cognitive computing in mobile edge system among protocol stacks

In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning



Taxonomy of applying Deep Reinforcement Learning in mobile edge system

In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning



(a) Centralized DRL

(b) Distributed DRL

(c) DRL with Federated Learning

Distributed Machine Learning

Distributed machine learning refers to multi-node machine learning algorithms and systems that are designed to improve performance, increase accuracy, and scale to larger input data sizes. Increasing the input data size for many algorithms can significantly reduce the learning error and can often be more effective than using more complex methods.

Δ Focus on *Optimization*

$\triangleright$ ***A survey of methods for distributed machine learning***, Nov 2012, Progress in AI. link

Combining with Communication and Networking with low latency and QoE guranteed $\rightarrow$ Edge AI

Distributed Machine Learning

▷ ***Large Scale Distributed Deep Networks***. link [Jeff Dean, Andrew Ng et al, 1724 citations]

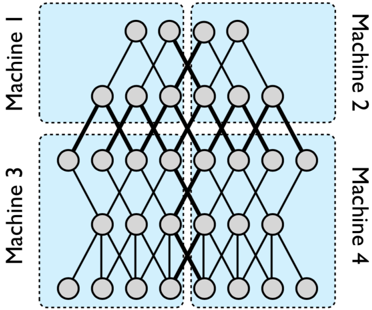


Figure 1: An example of model parallelism in DistBelief. A five layer deep neural network with local connectivity is shown here, partitioned across four machines (blue rectangles). Only those nodes with edges that cross partition boundaries (thick lines) will need to have their state transmitted between machines. Even in cases where a node has multiple edges crossing a partition boundary, its state is only sent to the machine on the other side of that boundary once. Within each partition, computation for individual nodes will the parallelized across all available CPU cores.

Large Scale Distributed Deep Networks

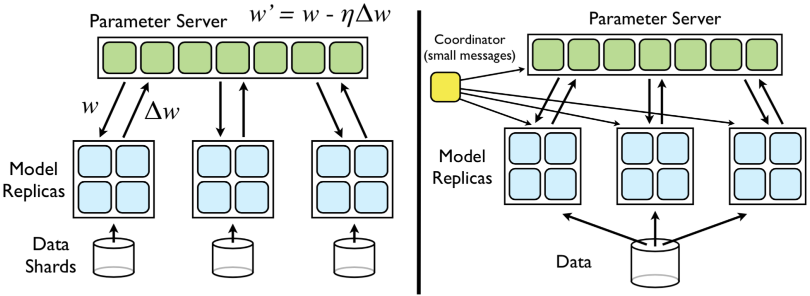


Figure 2: Left: Downpour SGD. Model replicas asynchronously fetch parameters w and push gradients Δw to the parameter server. Right: Sandblaster L-BFGS. A single 'coordinator' sends small messages to replicas and the parameter server to orchestrate batch optimization.