Wireless and Computing Resource Allocation for Selfish Computation Offloading in Edge Computing

Slađana Jošilo and György Dán

Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

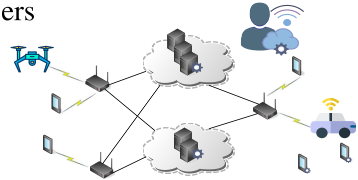
Paris, May 2, 2019

Mobile Edge Computing

Mobile Edge Computing

Enabler of 5G that provides

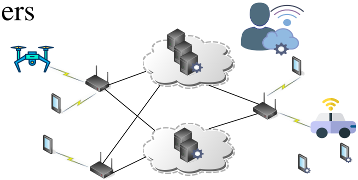
- computing resources close to the end users
- support for autonomous IoT devices
- bandwidth and computing resource orchestration on demand



Mobile Edge Computing

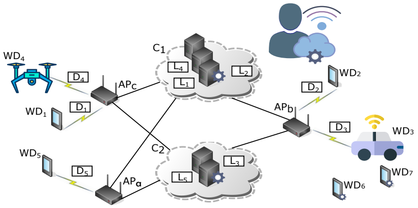
Enabler of 5G that provides

- computing resources close to the end users
- support for autonomous IoT devices
- bandwidth and computing resource orchestration on demand



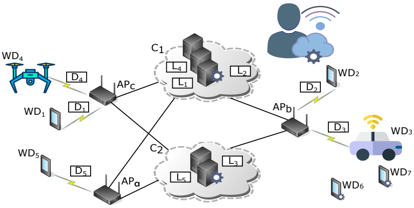
- How do device autonomy and infrastructure resource management interact?

Mobile Edge Computing System



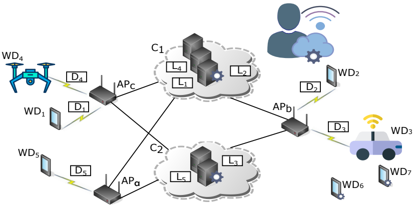
- Wireless devices (WDs) $\mathcal{N}$

Mobile Edge Computing System



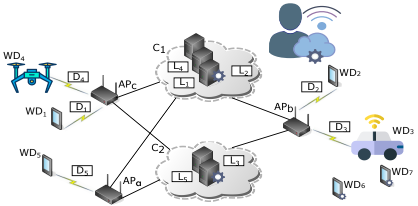
- Wireless devices (WDs) $\mathcal{N}$
- Operator manages
  - Edge clouds (ECs) $\mathcal{C}$
  - APs $\mathcal{A}$

Mobile Edge Computing System



- Wireless devices (WDs) $\mathcal{N}$
- Operator manages
  - Edge clouds (ECs) $\mathcal{C}$
  - APs $\mathcal{A}$
  - WD i can connect to APs $\mathcal{A}_i \subseteq \mathcal{A}$

Mobile Edge Computing System

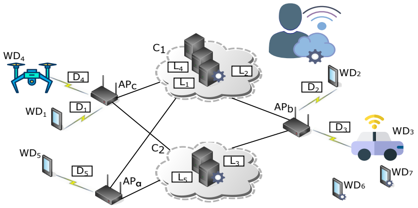


- Wireless devices (WDs) $\mathcal{N}$
- Operator manages
  - Edge clouds (ECs) $\mathcal{C}$
  - APs $\mathcal{A}$
  - WD i can connect to APs $\mathcal{A}_i \subseteq \mathcal{A}$

Computation offloading

- Task of WD i, $< D_i, L_i >$
  - size of the input data D_i
  - computational complexity L_i

Mobile Edge Computing System

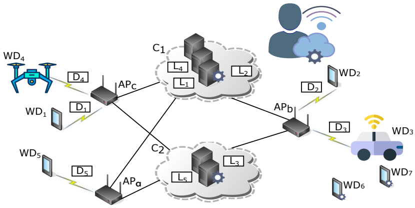


- Wireless devices (WDs) $\mathcal{N}$
- Operator manages
  - Edge clouds (ECs) $\mathcal{C}$
  - APs $\mathcal{A}$
  - WD i can connect to APs $\mathcal{A}_i \subseteq \mathcal{A}$

Computation offloading

- Task of WD i, $< D_i, L_i >$
  - size of the input data D_i
  - computational complexity L_i
- Decision d_i of WD $i \in \mathcal{N}$

Mobile Edge Computing System

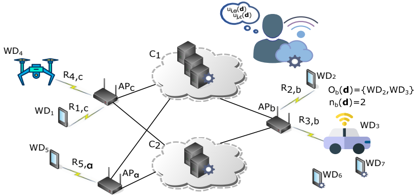


- Wireless devices (WDs) $\mathcal{N}$
- Operator manages
  - Edge clouds (ECs) $\mathcal{C}$
  - APs $\mathcal{A}$
  - WD i can connect to APs $\mathcal{A}_i \subseteq \mathcal{A}$

Computation offloading

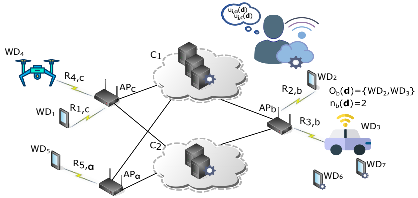
- Task of WD i, $< D_i, L_i >$
  - size of the input data D_i
  - computational complexity L_i
- Decision d_i of WD $i \in \mathcal{N}$
- Set of decisions for all WDs is a *strategy profile* **d**

Communication Model



- $R_{i,a}$: PHY rate of WD i on AP a
- $u_{i,a}$: uplink access provisioning coefficient

Communication Model



- $R_{i,a}$: PHY rate of WD i on AP a
- $u_{i,a}$: uplink access provisioning coefficient

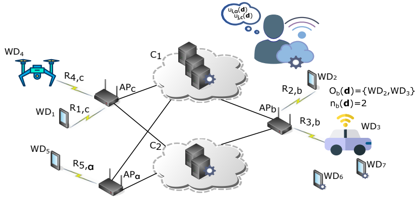
Transmission time

$O_a(\mathbf{d})$: set of offloaders via AP a in strategy profile $\mathbf{d}$

- Uplink rate of WD i via AP a

$$\omega_{i,a}(\mathbf{d}, \mathbf{u}_a) = R_{i,a} \frac{u_{i,a}}{\sum_{j \in O_a(\mathbf{d})} u_{j,a}}$$

Communication Model



- $R_{i,a}$: PHY rate of WD i on AP a
- $u_{i,a}$: uplink access provisioning coefficient

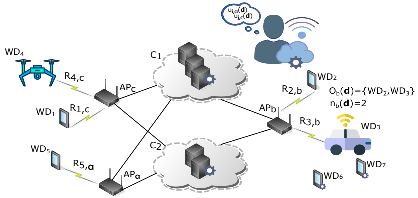
Transmission time

- Uplink rate of WD i via AP a

e.g., $u_{i,a} = 1 \implies$ time-fair policy

$$\omega_{i,a}(\mathbf{d}, \mathbf{u}_a) = R_{i,a} \frac{u_{i,a}}{\sum_{j \in O_a(\mathbf{d})} u_{j,a}}$$

Communication Model



- $R_{i,a}$: PHY rate of WD i on AP a
- $u_{i,a}$: uplink access provisioning coefficient

Transmission time

e.g., $u_{i,a} = 1 \implies$ time-fair policy

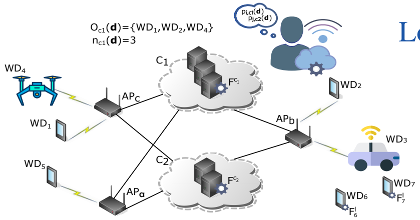
- Uplink rate of WD i via AP a

$$\omega_{i,a}(\mathbf{d}, \mathbf{u}_a) = R_{i,a} \frac{u_{i,a}}{\sum_{j \in O_a(\mathbf{d})} u_{j,a}}$$

- Transmission time of WD i for offloading via AP a

$$T^{off}_{i,a}(\mathbf{d}, \mathbf{u}_a) = \frac{D_i}{\omega_{i,a}(\mathbf{d},\mathbf{u}_a)}$$

Computation Model

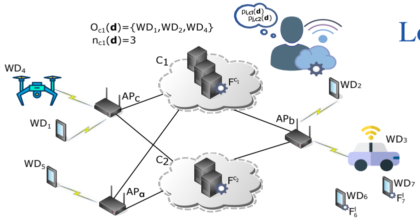


Local computing

- F_i^l computing capability of WD i
- Local execution time of WD i's task

$$T_{i,l}^{exe} = \frac{L_i}{F_i^l}$$

Computation Model



Local computing

- F_i^l computing capability of WD i
- Local execution time of WD i's task

$$T_{i,l}^{exe} = \frac{L_i}{F_i^l}$$

Computation offloading

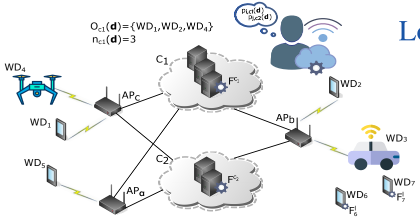
- F^c : computing capability of EC c
- $p_{i,c}$: computing power provisioning coefficient

$O_c(\mathbf{d})$: set offloaders to EC c in strategy profile $\mathbf{d}$

- Computing capability allocated to WD i by EC c

$$F_i^c(\mathbf{d}, \mathbf{p}_c) = F^c \frac{p_{i,c}}{\sum_{j\in O_c(\mathbf{d})} p_{j,c}}$$

Computation Model



Local computing

- F_i^l computing capability of WD i
- Local execution time of WD i's task

$$T_{i,l}^{exe} = \frac{L_i}{F_i^l}$$

Computation offloading

- F^c : computing capability of EC c
- $p_{i,c}$: computing power provisioning coefficient
- Computing capability allocated to WD i by EC c

$$F_i^c(\mathbf{d}, \mathbf{p}_c) = F^c \frac{p_{i,c}}{\sum_{j \in O_c(\mathbf{d})} p_{j,c}}$$

- Execution time of WD i's task in EC c

$$T_{i,c}^{exe}(\mathbf{d}, \mathbf{p}_c) = \frac{L_i}{F_i^c(\mathbf{d}, \mathbf{p}_c)}$$

Cost - Task Completion Time

Cost - Task Completion Time

Computation offloading cost

$$C^{c}_{i,a}(\mathbf{d}, \mathbf{u}_a, \mathbf{p}_c) = T^{off}_{i,a}(\mathbf{d}, \mathbf{u}_a) + T^{exe}_{i,c}(\mathbf{d}, \mathbf{p}_c)$$

Local computing cost

$$C^{l}_{i} = T^{exe}_{i,l}$$

Cost - Task Completion Time

Computation offloading cost

$$C^c_{i,a}(\mathbf{d},\mathbf{u}_a,\mathbf{p}_c) = T^{off}_{i,a}(\mathbf{d},\mathbf{u}_a) + T^{exe}_{i,c}(\mathbf{d},\mathbf{p}_c)$$

Local computing cost

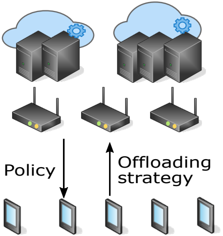
$$C^l_i = T^{exe}_{i,l}$$

System cost

$$C(\mathbf{d},\mathbf{u},\mathbf{p}) = \underbrace{\sum_{i\in\mathcal{N}} \sum_{(a,c)\in\mathcal{A}_i\times\mathcal{C}} I_{d_i,(a,c)} C^c_{i,a}(\mathbf{d},\mathbf{u}_a,\mathbf{p}_c)}_{\text{offloading}} + \underbrace{\sum_{i\in\mathcal{N}} I_{d_i,i} C^l_i}_{\text{local execution}},$$

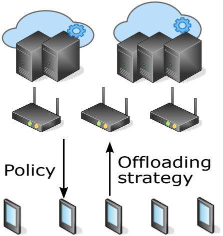
$$I_{d_i,r} = \begin{cases} 1, & \text{if } d_i = r \\ 0, & \text{otherwise} \end{cases}$$

Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game

Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game

Objective of the operator

- Minimization of total cost

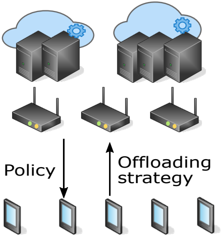
$$\min_{\mathbf{u},\mathbf{p}\succeq 0} C(\mathbf{d},\mathbf{u},\mathbf{p})$$

Objective of WDs

- Minimization of own cost

$$\min_{d_i\in\mathfrak{D}_i} C_i(d_i, d_{-i}, \mathbf{u}_a^*, \mathbf{p}_c^*)$$

Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game

Objective of the operator

- Minimization of total cost

$$\min_{\mathbf{u},\mathbf{p}\succeq 0} C(\mathbf{d},\mathbf{u},\mathbf{p})$$

Objective of WDs

- Minimization of own cost

$$\min_{d_i\in\mathfrak{D}_i} C_i(d_i,d_{-i},\mathbf{u}_a^*,\mathbf{p}_c^*)$$

Strategic game played by WDs

- For any allocation policy of the operator, game $\Gamma = < \mathcal{N}, (\mathfrak{D}_i)_i, (C_i)_i >$ played by WDs is a player-specific weighted congestion game

Questions Addressed

Questions Addressed

1. Does the MEC-OG have a subgame perfect equilibrium (SPE)?
   - How should an operator allocate resources to selfish WDs?
   - Is there an offloading strategy profile in which all WDs are satisfied?

Questions Addressed

1. Does the MEC-OG have a subgame perfect equilibrium (SPE)?
   - How should an operator allocate resources to selfish WDs?
   - Is there an offloading strategy profile in which all WDs are satisfied?
2. Can it be computed using a decentralized algorithm?

Questions Addressed

1. Does the MEC-OG have a subgame perfect equilibrium (SPE)?
    - How should an operator allocate resources to selfish WDs?
    - Is there an offloading strategy profile in which all WDs are satisfied?
2. Can it be computed using a decentralized algorithm?
3. How good is the system performance?

Results

Optimal resource allocation policy of the operator

- Best response of the operator to strategy profile **d** chosen by WDs

$$u^*_{i,a}(\mathbf{d}) = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\mathbf{d})}\sqrt{D_j/R_{j,a}}}, \forall i \in O_a(\mathbf{d}), \forall a \in \mathcal{A}$$

$$p^*_{i,c}(\mathbf{d}) = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\mathbf{d})}\sqrt{L_j/F^c}}, \forall i \in O_c(\mathbf{d}), \forall c \in \mathcal{C}$$

Results

Optimal resource allocation policy of the operator

- Best response of the operator to strategy profile **d** chosen by WDs

$$u^*_{i,a}(\mathbf{d}) = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\mathbf{d})}\sqrt{D_j/R_{j,a}}}, \forall i \in O_a(\mathbf{d}), \forall a \in \mathcal{A}$$

$$p^*_{i,c}(\mathbf{d}) = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\mathbf{d})}\sqrt{L_j/F^c}}, \forall i \in O_c(\mathbf{d}), \forall c \in \mathcal{C}$$

Game Γ under the optimal operator policy

- We transform Γ into a congestion game Γ^* with resource dependent weights

Offloading cost: $C^c_{i,a}(\mathbf{d}) = \omega_{i,a}\sum_{j\in O_a(\mathbf{d})}\omega_{j,a} + \omega_{i,c}\sum_{j\in O_c(\mathbf{d})}\omega_{j,c}$

Weights: $\omega_{i,a} = \sqrt{\frac{D_i}{R_{i,a}}}, \omega_{i,c} = \sqrt{\frac{L_i}{F^c}}$

Results

Optimal resource allocation policy of the operator

- Best response of the operator to strategy profile **d** chosen by WDs

$$u^*_{i,a}(\mathbf{d}) = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\mathbf{d})}\sqrt{D_j/R_{j,a}}}, \forall i \in O_a(\mathbf{d}), \forall a \in \mathcal{A}$$

$$p^*_{i,c}(\mathbf{d}) = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\mathbf{d})}\sqrt{L_j/F^c}}, \forall i \in O_c(\mathbf{d}), \forall c \in \mathcal{C}$$

Game Γ under the optimal operator policy

- We transform Γ into a congestion game Γ^* with resource dependent weights

Offloading cost: $C^c_{i,a}(\mathbf{d}) = \omega_{i,a}\sum_{j\in O_a(\mathbf{d})}\omega_{j,a} + \omega_{i,c}\sum_{j\in O_c(\mathbf{d})}\omega_{j,c}$

Weights: $\omega_{i,a} = \sqrt{\frac{D_i}{R_{i,a}}}, \omega_{i,c} = \sqrt{\frac{L_i}{F^c}}$

- Does strategic game Γ^* have a Nash equilibrium (NE)?

Results

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
  - Proof based on exact potential function

Results

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
  - Proof based on exact potential function

Improve Local Computing (ILC) algorithm

- Starts from a strategy profile in which all WDs perform computation locally
- Allows WDs to start to offload in non-increasing order of their task complexities

Results

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
  - Proof based on exact potential function

Improve Local Computing (ILC) algorithm

- Starts from a strategy profile in which all WDs perform computation locally
- Allows WDs to start to offload in non-increasing order of their task complexities

SPE existence

- The MEC-OG has a SPE $(\mathbf{d}^*, \mathbf{u}^*, \mathbf{p}^*)$
  - Optimal provisioning coefficients $\mathbf{u}^*$ and $\mathbf{p}^*$ have closed form expressions
  - ILC algorithm computes an equilibrium $\mathbf{d}^*$ of offloading decisions

Results

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
  - Proof based on exact potential function

Improve Local Computing (ILC) algorithm

- Starts from a strategy profile in which all WDs perform computation locally
- Allows WDs to start to offload in non-increasing order of their task complexities

SPE existence

- The MEC-OG has a SPE $(\mathbf{d}^*, \mathbf{u}^*, \mathbf{p}^*)$
  - Optimal provisioning coefficients $\mathbf{u}^*$ and $\mathbf{p}^*$ have closed form expressions
  - ILC algorithm computes an equilibrium $\mathbf{d}^*$ of offloading decisions

Price of Anarchy (PoA) bound

- Upper bound on the PoA for the MEC-OG is $\frac{3+\sqrt{5}}{2}$

User Focused Performance Analysis

Evaluation scenario

- $A = 5$ APs, heterogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

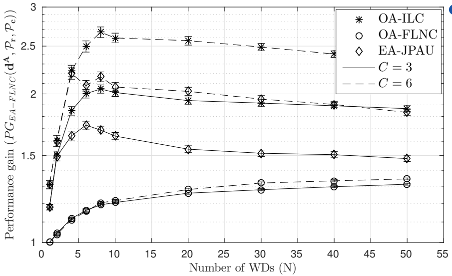
User Focused Performance Analysis

Evaluation scenario

- $A = 5$ APs, heterogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

Performance gain

Defined w.r.t. *equal alocation* (EA) policy and the *fastest-link nearest-cloud* (FLNC) algorithm



- *Performance gain* increases with decreasing marginal gain in N

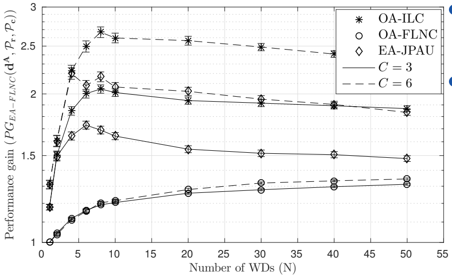
User Focused Performance Analysis

Evaluation scenario

- $A = 5$ APs, heterogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

Performance gain

Defined w.r.t. *equal alocation* (EA) policy and the *fastest-link nearest-cloud* (FLNC) algorithm



- *Performance gain* increases with decreasing marginal gain in N
- *Performance gain* increases with the number of edge ECs

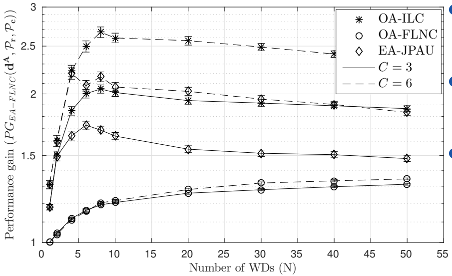
User Focused Performance Analysis

Evaluation scenario

- $A = 5$ APs, heterogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

Performance gain

Defined w.r.t. *equal alocation* (EA) policy and the *fastest-link nearest-cloud* (FLNC) algorithm



- *Performance gain* increases with decreasing marginal gain in N
- *Performance gain* increases with the number of edge ECs
- Largest *performance gain*
  - Operator implements OA policy
  - WDs compute offloading decisions using the ILC algorithm

Cloud Focused Performance Analysis

Evaluation scenario

- $A = 5$ APs, $C = 3$ heterogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

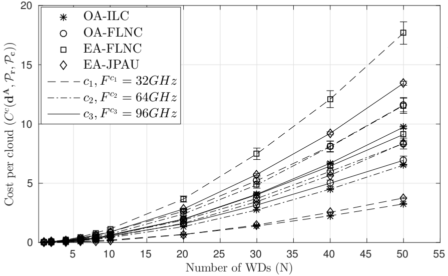
Cloud Focused Performance Analysis

Evaluation scenario

- $A = 5$ APs, $C = 3$ heterogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

Cost per cloud

Defined as $C^c(\mathbf{d}) = \sum_{i \in O_c(\mathbf{d})} C_i(\mathbf{d})$



- *Cost per cloud* increases with the number N of WDs

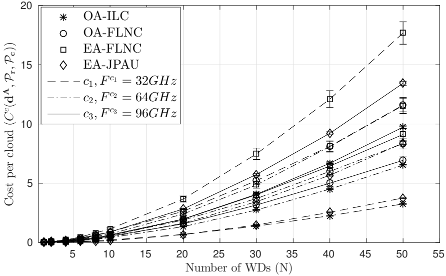
Cloud Focused Performance Analysis

Evaluation scenario

- $A = 5$ APs, $C = 3$ heterogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

Cost per cloud

Defined as $C^c(\mathbf{d}) = \sum_{i \in O_c(\mathbf{d})} C_i(\mathbf{d})$



- *Cost per cloud* increases with the number N of WDs
- *Cost per cloud* is proportional to the EC's computing capability in case of equilibria under OA and EA policies

Computational Complexity

Evaluation scenario

- $A = 5$ APs, homogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

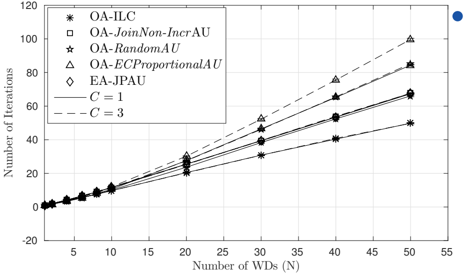
Computational Complexity

Evaluation scenario

- $A = 5$ APs, homogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

Number of iterations

- Randomly chosen strategy profile
- All WDs offload-congestion per EC proportional to its computing capability
- Empty system-WDs added in non-increasing order of their task complexities



- Number of iterations scales approximately linearly with the number N of WDs

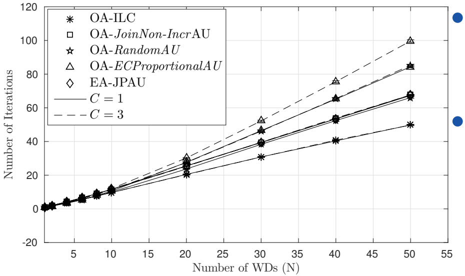
Computational Complexity

Evaluation scenario

- $A = 5$ APs, homogeneous ECs $F^{c,tot} = 192$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.2, 4)$ *Mb* , $L_i = D_i X$ *Gcycles* , $X \sim \Gamma(0.5, 1.6)$ *Gcycles/b*

Number of iterations

- Randomly chosen strategy profile
- All WDs offload-congestion per EC proportional to its computing capability
- Empty system-WDs added in non-increasing order of their task complexities



- Number of iterations scales approximately linearly with the number N of WDs
- Number of iterations is sensitive to the starting strategy profile
  - Smallest in the case of the ILC algorithm

Summary and Future Work

- Provided game theoretical analysis of the interaction between
  - an edge operator that jointly manages wireless and computing resources
  - autonomous WDs that aim at minimizing their own tasks completion times

Summary and Future Work

- Provided game theoretical analysis of the interaction between
  - an edge operator that jointly manages wireless and computing resources
  - autonomous WDs that aim at minimizing their own tasks completion times

- Interesting extensions
  - energy consumption minimization problem
  - stochastic model of task arrivals
  - learning in congestion games

Wireless and Computing Resource Allocation for Selfish Computation Offloading in Edge Computing

Slađana Jošilo and György Dán

Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Paris, May 2, 2019