Social-Aware DT-Assisted Service Provisioning in Serverless Edge Computing

Jing Li[1], Jianping Wang[1], Weifa Liang[1], Jie Wu[2], Quan Chen[3] and Zichuan Xu[4]

[1] City University of Hong Kong, Hong Kong, P. R. China
[2] Temple University, Philadelphia, USA
[3] Guangdong University of Technology, Guangzhou, P. R. China
[4] Dalian University of Technology, Dalian, P. R. China

Outline

Digital twin technique and Mobile Edge Computing

Digital Twins (DTs) monitor physical objects and represent them in a virtual world.

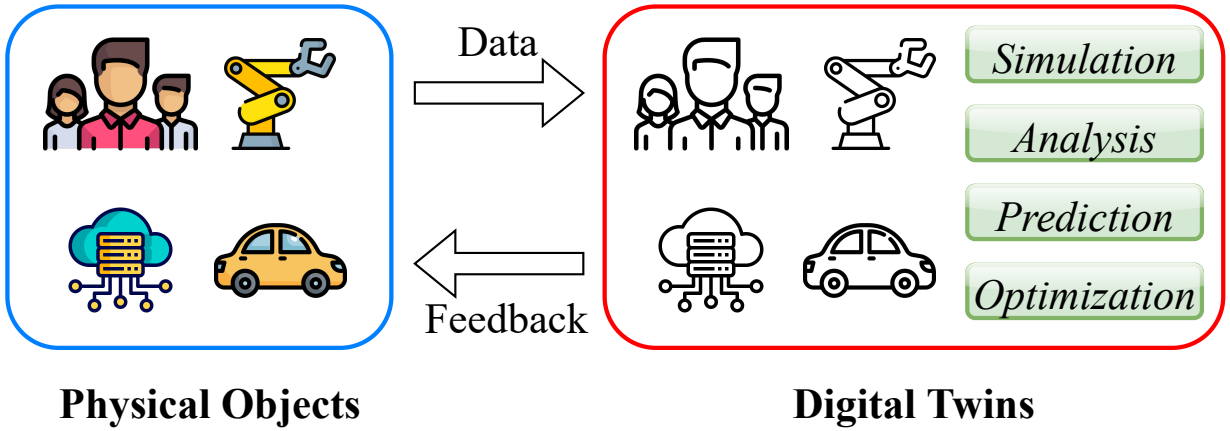


Figure: Concept of the digital twin technique.

Mobile Edge Computing (MEC)

- Physical objects feed DTs in cloudlets in real-time.
- DTs provide users with fresh DT data.

Serverless Edge Computing (SEC)

MEC: suffers from resource scarcity and inefficient resource management.

Serverless Edge Computing (SEC) integrates MEC and serverless computing, and has emerged as a crucial research area and endows DTs with new vigors.

- utilizes **severless functions** based on the **container** technology,
- provides **fine-grained resource allocation** with high elasticity for MEC.

DT network-enabled Serivce

The DT network:

- group a collection of DTs and analyze their global information
- deliver comprehensive DT services to users.
- the rising concerns on privacy and security: distributed learning architectures.

Federated learning framework:

- models are trained locally and the model parameters, rather than raw data, are uploaded to a server for aggregation.
- potential leakages of private data, through examining the differences of model parameters that are uploaded via local devices.

Exsiting works adopt a privacy-preserving method - **differential privacy**, which prevents data leakage through adding artificial noise.

A federated learning framework for constructing a DT network for each service request

Initially, each IoT device has a **Primary DT (P_DT)** placed in a cloudlet, containing all features of the IoT device through monitoring its state continuously.

An IoT device may issue a request for the DT data of other IoT devices, and there is a candidate set of IoT devices as its potential participants through inter-twin communication.

Each selected IoT device then extracts the needed features from its P_DT to build a **Sub-DT (S_DT)** to participate in the execution of the request, where each S_DT is deployed in a container.

The DT network of a request consists of

- the P_DT of the IoT device issuing the request.
- the S_DTs of selected IoT devices for providing DT data

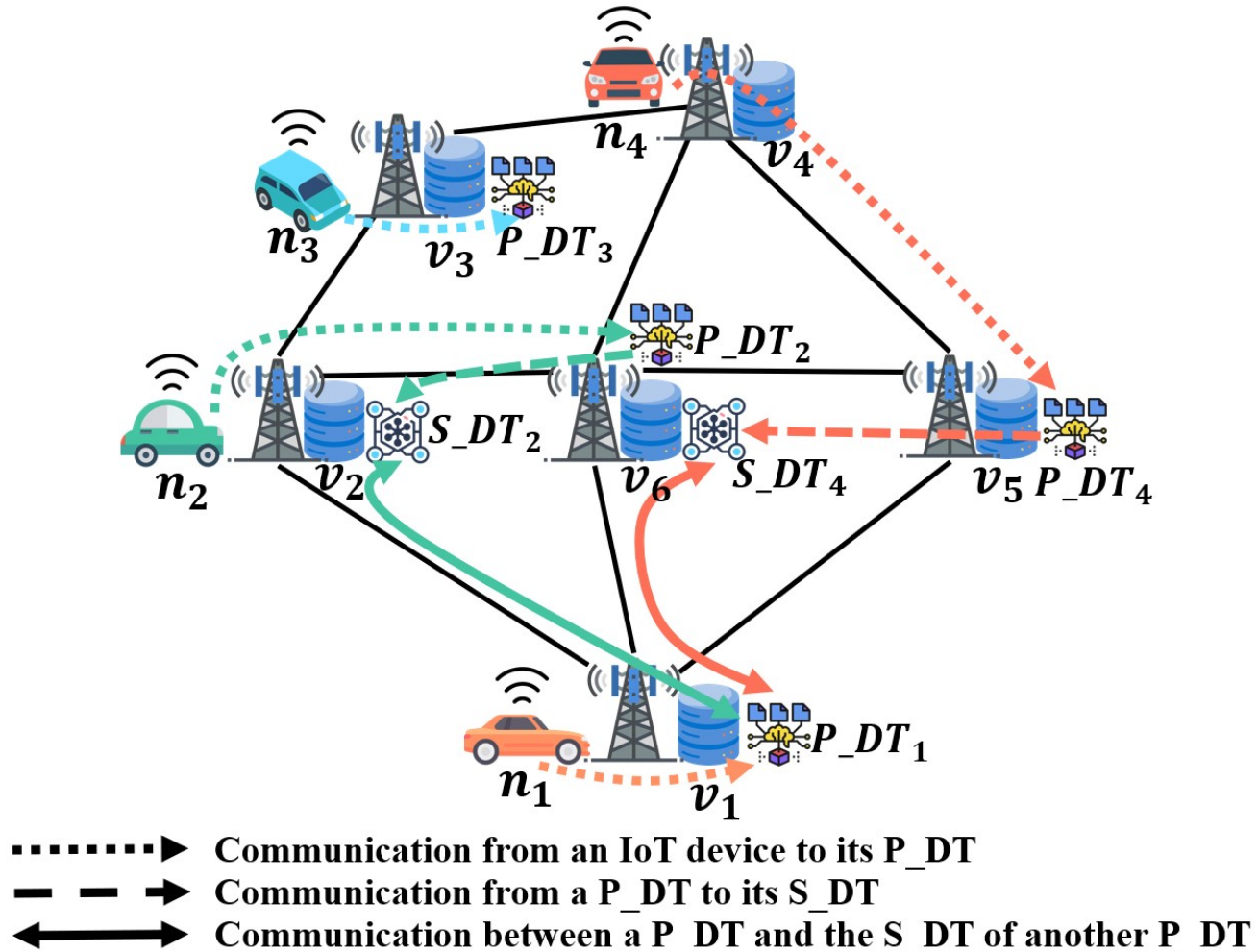
A federated learning framework for constructing a DT network for each request

To protect privacy, each request is executed by adopting a differential privacy-based federated learning framework:

- the P_DT of an IoT device issuing the request first uses itself to train a global model, and then transmits the trained global model to the S_DT of each selected IoT device.
- Each S_DT performs local training and sends the updated model to the P_DT for aggregation with an amount of allocated privacy budget, depending on the social relationships (e.g., trust) among IoT devices.
- Each service request has a **global privacy budget** to bound its privacy leakage.

Example of DT network-enabled services

The driving simulation of a vehicle may need the DT data of other vehicles in the same area (prefers the features, such as locations and driving behaviors of vehicles).



Challenges for providing social-aware DT services in an SEC network

The utility for admitting a request (quality of services)

$\leftarrow$ **Importance of the DT data** of each selected IoT device

$\leftarrow$ **Interaction intensiveness** among IoT devices (**social relationships**)

- How to **select appropriate IoT devices** for providing DT data demanded by each request, subject to its given global privacy budget?
- How to **deploy S_DTs** of the selected IoT devices in cloudlets for each request, considering limited resource capacities on cloudlets?
- How to cope with DT network-enabled request admissions through S_DT deployments to **maximize the total utility of admitted requests**, by exploring the social-aware DT relationships?

Contributions

- Formulate a novel social-aware S_DT placement problem for DT-enabled service provisioning, considering data privacy.
- Propose an Integer Linear Program (ILP) solution for the problem when the problem size is small.
- Devise an approximation algorithm for the problem with a provable approximation ratio.
- Evaluate the algorithm performance via simulations, and simulation results demonstrate that the proposed algorithm is promising.

.

System model

Following existing works, **the allocated privacy budget** for data communication from IoT device n_1 to IoT device n_2 is calculated as follows.

$$\epsilon_{n_1,n_2} = \frac{l_{n_1,n_2}}{l_{n_1,n_2} + \tau} \cdot \kappa, \tag{1}$$

where l_{n_1,n_2} is the value of the trust from IoT device n_1 to device n_2 with $0 \leq l_{n_1,n_2} \leq 1$, τ is a constant with $0 \leq \tau \leq 1$ to avoid the denominator to be zero, and $\kappa > 0$ is a tuning parameter.

The global privacy budget constraint for each request r is as follows.

$$\sum_{n \in N_r} \sum_{v \in V} \epsilon_{n,n_r} \cdot x_{r,n,v} \leq B_r, \qquad \forall r \in R \tag{2}$$

The resource capacity constraint on a cloudlet is

$$\sum_{r \in R} \sum_{n \in N_r} m_{r,n} \cdot x_{r,n,v} \leq M_v, \qquad \forall v \in V \tag{3}$$

The social-aware S_DT placement problem

We define **the utility gain $u_{r,n}$ of selecting an IoT device** n from the candidate set N_r as

$$u_{r,n} = \frac{\lambda_{n_r,n}}{\sum_{n' \in N_r} \lambda_{n_r,n'}}, \tag{4}$$

where $\lambda_{n_r,n}$ is the intensiveness that the selected IoT device n_r interacts with IoT device n issuing the request, with $0 \leq \lambda_{n_r,n} \leq 1$, indicating the importance of the DT data.

Definition

The social-aware S_DT placement problem is to maximize the utility gain of the requests, through deploying S_DTs in cloudlets, subject to a given global privacy budget and resource capacities on cloudlets

Approximation algorithm for the social-aware S_DT placement problem

The core idea:

- We first obtain a potential solution with a set of S_DTs deployed in cloudlets, which allows violations on resource capacities of cloudlets and global privacy budgets on requests.
- We then refine the potential solution to obtain the final solution without any constraint violation.

Approximation algorithm for the social-aware S_DT placement problem

We define **the privacy consumption ratio** $\sigma(\lambda^l)$ of placing the lth S_DT λ^l as follows.

$$\sigma(\lambda^l) = \frac{\epsilon(\lambda^l)}{B(\lambda^l)}, \tag{5}$$

where $\epsilon(\lambda^l)$ is the consumed privacy budget of λ^l.

To guide the deployment of S_DTs, we adopt a metric - **the ratio** $\rho(\lambda^l)$ for deploying the lth S_DT λ^l, with

$$\rho(\lambda^l) = \frac{u(\lambda^l)}{m(\lambda^l) \cdot \sigma(\lambda^l)}, \tag{6}$$

where $u(\lambda^l)$ is the utility of deploying λ^l, and $m(\lambda^l)$ is the resource consumed in a cloudlet by λ^l.

Approximation algorithm for the social-aware S_DT placement problem

To deploy the lth S_DT λ^l, we identify an S_DT with the largest $\rho(\lambda^l)$.

- We put λ^l into set S_1, if its placement results in no violation on the privacy budget of its request.
- Otherwise, we put λ^l into set S_2.
- We will no longer consider the request by removing its rest S_DTs from further consideration, if the utilized privacy budget of a request is no less than its given privacy budget after deploying λ^l.

We then identify a cloudlet with the largest residual resource for deploying each identified S_DT λ^l.

- We put λ^l into set S_3, if its placement causes capacity violations on the cloudlet.
- The cloudlet is removed from further consideration, if the consumed resource of the cloudlet is no less than its capacity after deploying S_DT λ^l.

Approximation algorithm for the social-aware S_DT placement problem

Now we can get a potential $\mathbb{S}$, which has been partitioned into two sets S_1 and S_2, and one of them with the larger utility is identified as S'.

- *The S_DTs in either S_1 or S_2 will not cause violations on the global privacy budget constraint.*

We then update $S_3 = S_3 \cap S'$. Let $S_4 = S' \backslash S_3$, and S' is now partitioned into two disjoint sets S_3 and S_4.

- *The S_DTs in either S_3 or S_4 will not cause violations on the cloudlet capacity constraint.*

We finally choose S_3 or S_4 with the larger utility as the final solution to the social-aware S_DT placement problem.

Algorithm analysis

Theorem

Given an SEC network $G = (V, E)$, a set $\mathcal{N}$ of IoT devices, a set R of requests, each request $r \in R$ has a candidate set of IoT devices N_r for its S_DT deployment. There is an approximation algorithm, `Algorithm` *1, for the social-aware S_DT placement problem with an approximation ratio of* $\frac{1}{4} \cdot \min\{\frac{m_{min}}{m_{max}+m_{min}}, \frac{\theta_{min}}{\theta_{max}}\}$*, where m_{max} and m_{min} are the maximum and minimum amounts of resource consumed by any S_DT, respectively.* $\theta(\lambda^l) = \frac{u(\lambda^l)}{m(\lambda^l)}$*, θ_{max} and θ_{min} are the maximum and minimum values of* $\theta(\lambda^l)$*.*

Benchmarks

Evaluate `Algorithm 1` (`Alg.1`) against the following benchmarks:

- **Gdy_u**:
  - It iteratively identifies an S_DT with the maximum utility.
  - The chosen S_DT then is deployed in a cloudlet with enough residual resource.
- **Gdy_m**:
  - Similar to **Heu.1**,
  - However, it identifies an S_DT with the smallest resource consumption iteratively.
- **LP**:
  - The relaxed Linear Program (LP) solution, where the binary variables are relaxed into real numbers.
  - An upper bound on the optimal solution.

Algorithm performance with network sizes from 50 to 250.

Alg.1 can determine the placement of S_DTs efficiently to maximize the total utility.

Alg.1 outperforms the benchmarks by no less than 21.1%.

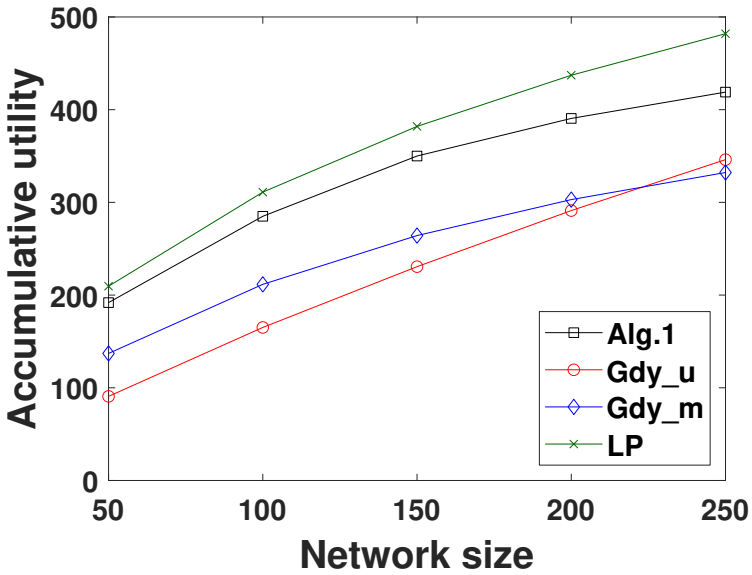


Figure: The performance

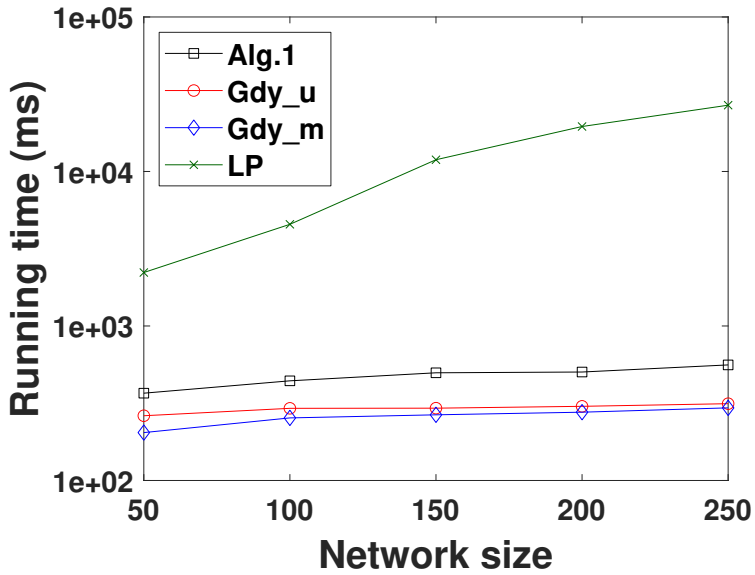


Figure: The running time

The impact of the number $|R|$ of requests on the performance of `Alg.1`.

More utilities can be obtained with a larger number of requests, while taking more time to examine the requests.

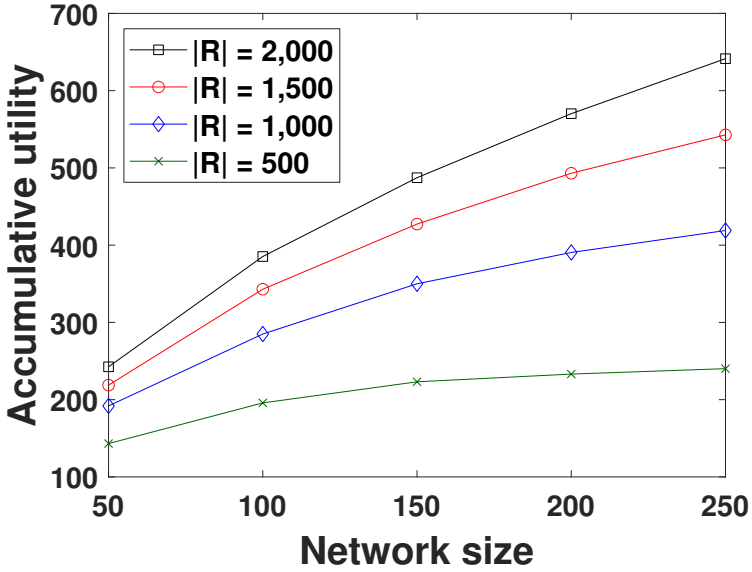


Figure: The performance

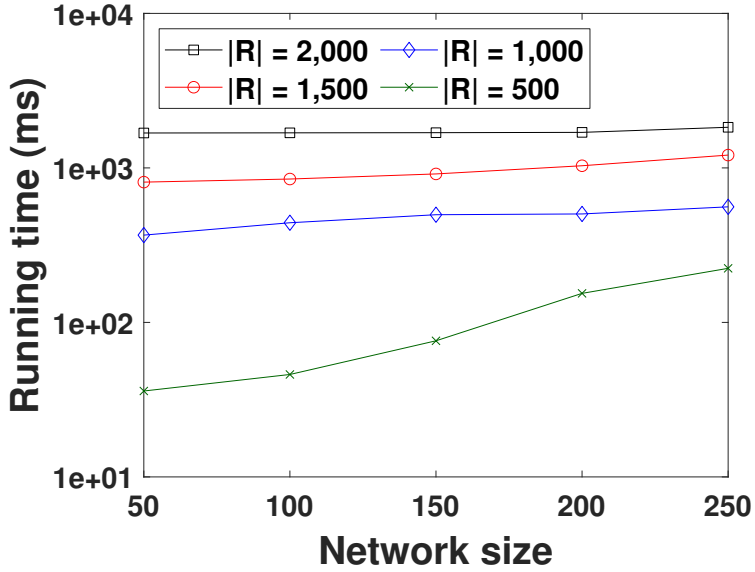


Figure: The running time

The impact of the number $|N_r|$ of candidate IoT devices for each request r on the performance of `Alg.1`.

The utility gain of selecting an IoT device for a request depends on the total interaction intensiveness of all the candidate IoT devices of the request.

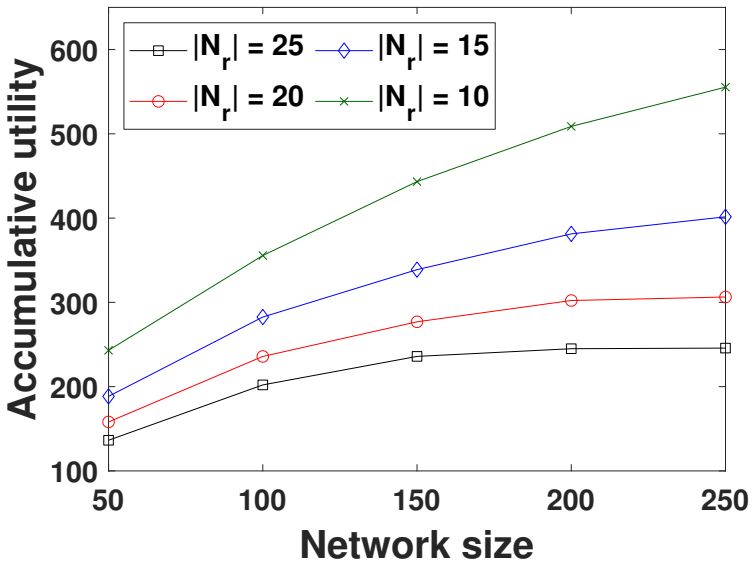


Figure: The performance

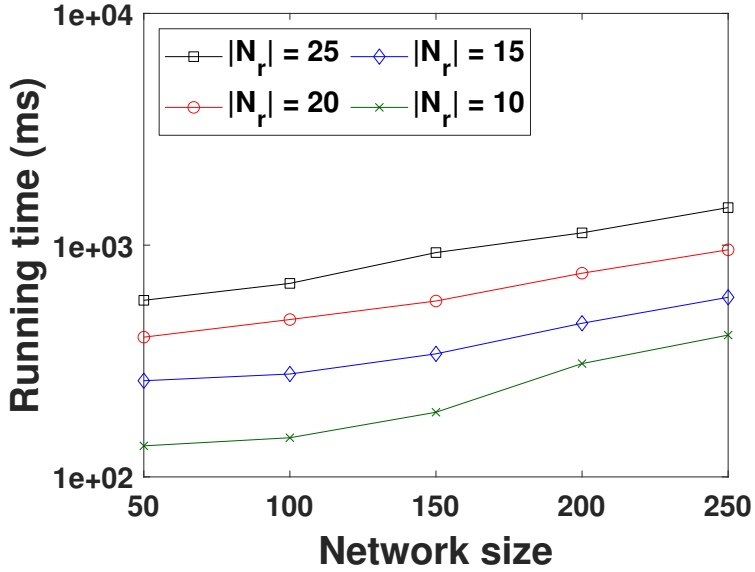


Figure: The running time

The impact of the global privacy budget B_r of each request r on the performance of `Alg.1`.

A larger global privacy budget means more IoT devices can be selected to provide DT data for each request.

When the global privacy budgets of requests are large, the resource capacity constraint on cloudlets is the bottleneck.

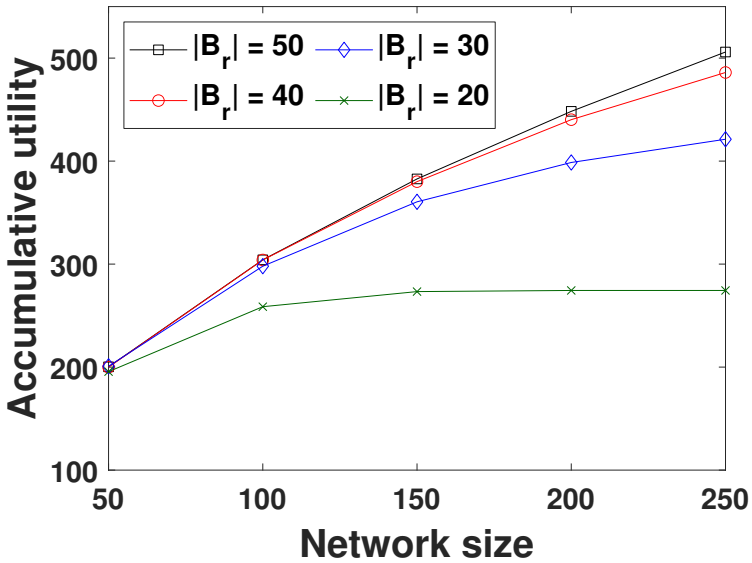


Figure: The performance

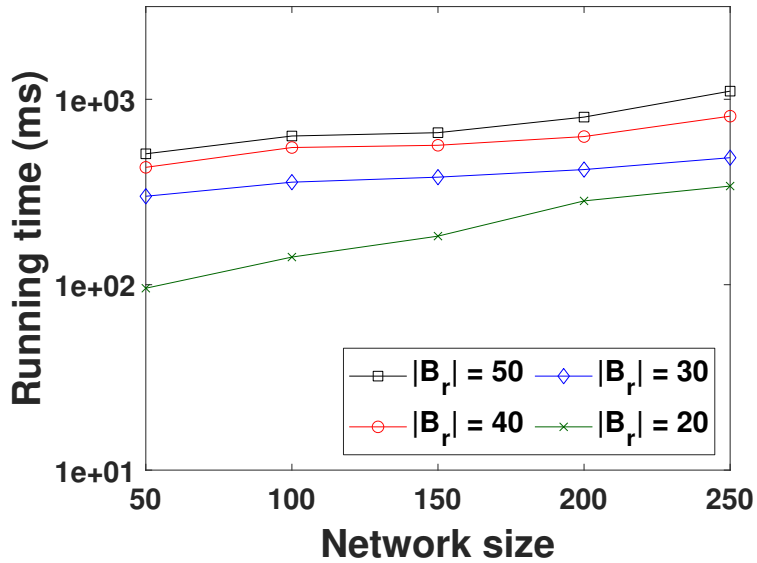


Figure: The running time

Conclusions

- Investigate social-aware service provisioning for DT-assisted SEC environments through S_DT placements.
- Introduce a differential privacy-based federated learning framework for admitting DT network-enabled service requests.
- Formulate a social-aware S_DT placement problem, with the aim to maximize the total utility.
- Devise a performance-guaranteed approximation algorithm.
- Evaluate the algorithm performance via simulations.

Q&A