Chapter 4
Incentive Mechanisms for Federated Learning

Abstract In this chapter, we discuss various components of federated learning that must be given some incentive in terms of monetary cost or other benefits. We design incentive mechanism design for federated learning using game theory and auction theory. Finally, we present extensive numerical results to show the validity of our proposed incentive mechanisms.

4.1 Introduction

Federated learning can be trained mainly in two different ways, such as (a) centralized server aggregation-based training, and (b) blockchain-based training, as shown in Figs. 4.1 and 4.2. Federated learning using centralized server aggregation involves continuous, iterative interaction between the end-devices and aggregation server. End-devices use their resources (i.e., computation resource and energy) to train their local learning models. The locally trained model updates will be sent via a wireless channel to the aggregation server for global aggregation. Similar to end-devices, the aggregation server will use its resources (i.e., computation resource and energy) to perform aggregation. To enable successful interaction among end-devices and aggregation servers for federated learning requires an attractive incentive mechanism. End-devices must be provided with benefits in response to their participation in the federated learning process. On the other hand, blockchain-based federated learning involves the computation of local models at the end-devices. The end-devices send their local learning models to their corresponding miners. The miners perform sharing and cross-verification of learning models to avoid injection of wrong models. Then, all the miners start computing their consensus algorithms (e.g., Proof-of-Work). The winning miner that solves the consensus algorithm first, broadcasts its block to all the miners in the network for updating their blocks. In blockchain-based federated learning, there is a need to provide an attractive incentive to both end-devices and miners for their jobs. Therefore, the incentive mechanism for blockchain-based federated learning will be different than the one for federated learning based on a centralized aggregation server.

C. S. Hong et al., *Federated Learning for Wireless Networks*, Wireless Networks,
https://doi.org/10.1007/978-981-16-4963-9_4

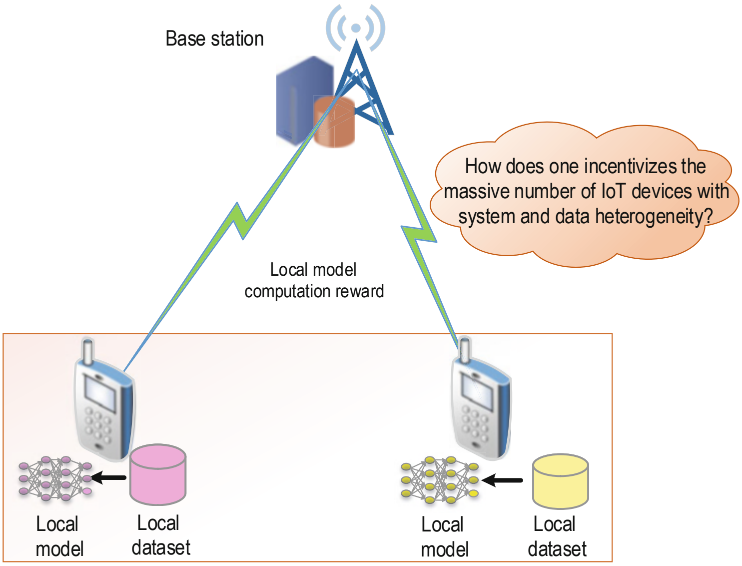

Fig. 4.1 Overview of rewards in centralized aggregation-based federated learning

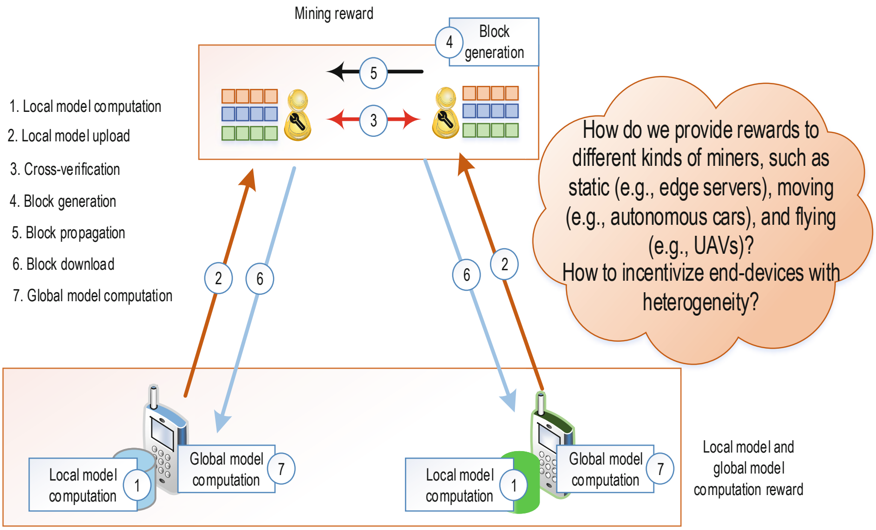

Fig. 4.2 Overview of rewards in blockchain-based federated learning

Generally, we can categorize incentives into two main types: monetary and non-monetary [100]. Monetary incentives are based on providing end-devices with payments as per their participation, whereas non-monetary incentives generally involve providing end-devices with benefits other than payments. Non-monetary incentives in the case of federated learning can be the well-trained global federated learning model for a large number of end-devices. Unless stated otherwise, the keyword incentive in this chapter refers to monetary incentive. Next, we present incentive mechanisms based on game theory and auction theory for federated learning over wireless networks.

4.2 Game Theory-Enabled Incentive Mechanism

Game theory has proven to be one of the successful tools in enabling/optimizing various functions/design aspects in wireless networks, such as wireless resource allocation, computational offloading in edge computing, URLLC/eMBB coexistence, and incentive mechanism design, among others [101–105]. Generally, games can be divided into (a) cooperative and (b) non-cooperative games. Cooperative games are based on achieving the equilibrium state for optimizing the overall benefit via joint decision-making by the various players. On the other hand, the players in non-cooperative games choose their strategies selfishly without coordination with other players (Fig. 4.3). A summary of cooperative and non-cooperative games used for incentive mechanism design in wireless networks is given in Table 4.1.

In federated learning, local computations at the devices and their communication with the centralized coordinating server are interleaved in a complex manner to build a global learning model. Therefore, a communication-efficient federated learning framework [18, 58] requires solving several challenges. Furthermore, because of limited data per device to train a high-quality learning model, the difficulty is to incentivize a large number of mobile users to ensure cooperation. This important aspect in federated learning has been overlooked so far, where the question is *how can we motivate a number of participating clients, collectively providing a large number of data samples to enable federated learning without sharing their private data?* Note that both participating clients and the server can benefit from training a global model. However, to fully reap the benefits

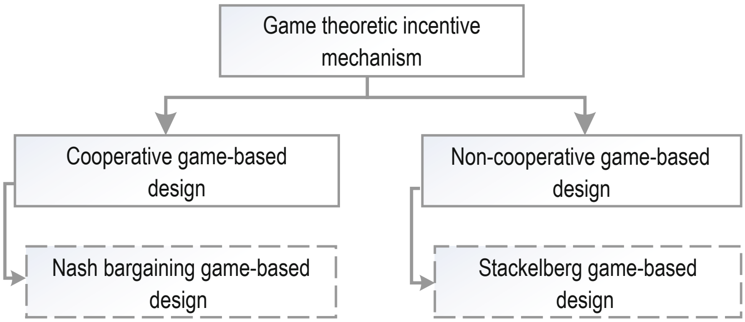

Fig. 4.3 Classification of game theoretic incentive mechanisms

Table 4.1 Overview of game theoretic incentive mechanisms [106]

Reference	Category	Game	Primary focus
Ho et al. [105]	Non-cooperative game	Stackelberg game	Macrocell base station traffic is admitted by the small cell base stations for monetary benefits
Liu et al. [107]	Non-cooperative	Stackelberg game	Incentives are given to femto cells for sharing their resource to macrocell base stations
Poularakis et al. [108]	Non-cooperative game	Stackelberg game	Cache-enabled access points are incentivized for caching by macrocell base stations
Gao et al. [109]	Cooperative game	Nash bargaining game	A cooperative game is proposed to carryout the transactions between the single mobile virtual network operator and multiple access points.
Yu et al. [110]	Cooperative game	Nash bargaining game	A bargaining framework in which mobile network operator bargains with venue owners sequentially for determining the deployment locations of Wi-Fi and how much to pay.

of high-quality updates, the multi-access edge computing (MEC) server has to incentivize clients for participation. In particular, under heterogeneous scenarios, such as an adaptive and cognitive-communication network, the client's participation in federated learning can spur collaboration and provide benefits for operators to accelerate and deliver network-wide services [111]. Similarly, clients, in general, are not concerned with the reliability and scalability issues of federated learning [112]. Therefore, to incentivize users to participate in the collaborative training, we require a marketplace. For this purpose, we present a value-based compensation mechanism to the participating clients, such as a bounty (e.g., data discount package), as per their level of participation in the crowdsourcing framework. This is reflected in terms of local accuracy level, i.e., *quality of solution to the local subproblem*, in which the framework will protect the model from imperfect updates by restricting the clients trying to compromise the model (for instance, with skewed data because of its i.i.d nature or data poisoning) [113]. Moreover, we cast the global loss minimization problem as a primal-dual optimization problem, instead of adopting a traditional gradient descent learning algorithm in the federated learning setting (e.g., FedAvg [18]). This enables (a) proper assessment of the quality of the local solution to improve personalization and fairness amongst the participating clients while training a global model, (b) effective decoupling of the local solvers, thereby balancing communication and computation in the distributed setting.

The goal of this section is two-fold: First, we formalize an incentive mechanism to develop a participatory framework for mobile clients to perform federated learning for improving the global model. Second, we address the challenge of maintaining communication efficiency while exchanging the model parameters with a number of participating clients during aggregation. Specifically, communication efficiency in this scenario accounts for communications per iteration with an arbitrary algorithm to maintain an acceptable accuracy level for the global model. In this work, we design and analyze a novel crowdsourcing framework to realize the federated learning vision. Specifically, our contributions are summarized as follows:

- **A crowdsourcing framework to enable communication -efficient federated learning**. We design a crowdsourcing framework, in which federated learning participating clients iteratively solve the local learning subproblems for an accuracy level subject to an offered incentive. We then establish a communication-efficient cost model for the participating clients. We then formulate an incentive mechanism to induce the necessary interaction between the MEC server and the participating clients for the federated learning in Sect. 4.2.2.
- **Solution approach using Stackelberg game**. With the offered incentive, the participating clients independently choose their strategies to solve the local subproblem for a certain accuracy level in order to minimize their participation costs. Correspondingly, the MEC server builds a high-quality centralized model characterized by its utility function, with the data distributed over the participating clients by offering the reward. We exploit these tightly coupled motives of the participating clients and the MEC server as a two-stage Stackelberg game. The equivalent optimization problem is characterized as mixed-Boolean programming which requires an exponential complexity effort for finding the solution. We analyze the game's equilibria and propose a linear complexity algorithm to obtain the optimal solution.
- **Participant's response analysis and case study**. We next analyze the response behavior of the participating clients via the solutions of the Stackelberg game and establish the efficacy of our proposed framework via case studies. We show that the linear-complexity solution approach attains the same performance as the mixed-Boolean programming problem. Furthermore, we show that our mechanism design can achieve the optimal solution while outperforming a heuristic approach for attaining the maximal utility with up to 22% of gain in the offered reward.
- **Admission control strategy**. Finally, we show that it is significant to have certain participating clients to guarantee the communication efficiency for an accuracy level in federated learning. We formulate a probabilistic model for threshold accuracy estimation and find the corresponding number of participants required to build a high-quality learning model. We analyze the impact of the number of participants in federated learning while determining the threshold accuracy level with closed-form solutions. Finally, with numerical results, we demonstrate the structure of the admission control model for different configurations.

4.2.1 System Model

Figure 4.4 illustrates our proposed system model for the crowdsourcing framework to enable federated learning. The model consists of a number of mobile clients associated with a base station having a central coordinating server (MEC server), acting as a central entity. The server facilitates the computation of the parameters aggregation, and feedback the global model updates in each global iteration. We consider a set of participating clients $\mathcal{K} = \{1, 2, \ldots, K\}$ in the crowdsourcing framework. The crowdsourcer (platform) can interact with mobile clients via an application interface and aims at leveraging federated learning to build a global ML model. As an example, consider a case where the crowdsourcer (referred to as MEC server hereafter, to avoid any confusion) wants to build a ML model. Instead of just relying on available local data to train the global model at the MEC server, the global model is constructed utilizing the local training data available across several distributed mobile clients. Here, the global model parameter is first shared by the MEC server to train the local models in each participating client. The local model's parameters minimizing local loss functions are then sent back as feedback and are aggregated to update the global model parameter. The process continues iteratively, until convergence.

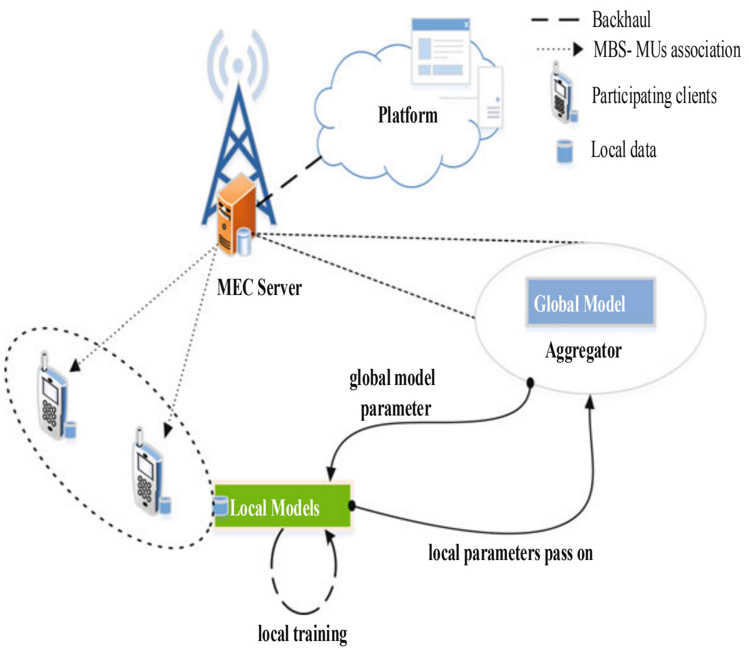

Fig. 4.4 Crowdsourcing framework for decentralized machine learning

Algorithm 4 Federated learning framework

1: **Input:** Initialize dual variable $\alpha^0 \in \mathbb{R}^D$, $D_k, \forall k \in K$.
2: **for** Each aggregation round **do**
3:     **for** k $\in \mathcal{K}$ **do**
4:         Solve local subproblems (4.5) in parallel.
5:         Update local variables as in (4.7).
6:     **end for**
7:     Aggregate to update global parameter as in (4.8).
8: **end for**

Federated Learning Background

For federated learning, we consider unevenly partitioned training data over a large number of participating clients to train the local models under any arbitrary learning algorithm. Each client k stores its local dataset $\mathcal{D}_k$ of size D_k respectively. Then, we define the training data size $D = \sum_{k=1}^{K} D_k$. In a typical supervised learning setting, $\mathcal{D}_k$ defines the collection of data samples given as a set of input-output pairs $\{x_i, y_i\}_{i=1}^{D_k}$, where $x_i \in \mathbb{R}^d$ is an input sample vector with d features, and $y_i \in \mathbb{R}$ is the labeled output value for the sample x_i. The learning problem, for an input sample vector x_i (e.g., the pixels of an image) is to find the *model parameter vector* $w \in \mathbb{R}^d$ that characterizes the output y_i (e.g., the labeled output of the image, such as the corresponding product names in a store) with the loss function $f_i(w)$. Some examples of loss functions include $f_i(w) = \frac{1}{2}(x_i^T w - y_i)^2, y_i \in \mathbb{R}$ for a linear regression problem and $f_i(w) = \max\{0, 1 - y_i x_i^T w\}, y_i \in \{-1, 1\}$ for support vector machines. The term $x_i^T w$ is often called a *linear mapping function*. Therefore, the loss function based on the local data of client k, termed local subproblem is formulated as

$$J_k(w) = \frac{1}{D_k} \sum_{i=1}^{D_k} f_i(w) + \lambda g(w), \tag{4.1}$$

where $w \in \mathbb{R}^d$ is the local model parameter, and $g(\cdot)$ is a regularizer function, commonly expressed as $g(\cdot) = \frac{1}{2} \|\cdot\|^2$; $\forall \lambda \in [0, 1]$. This characterizes the *local model* in the federated learning setting.

Global Problem At the MEC server, the global problem can be represented as the finite-sum objective of the form

$$\min_{w \in \mathbb{R}^d} J(w) \text{ where } J(w) \equiv \frac{\sum_{k=1}^{K} D_k J_k(w)}{D}. \tag{4.2}$$

Problems of such structure as in (4.2) where we aim to minimize an average of K local objectives are well-known as *distributed consensus problems* [114].

Solution Framework under Federated Learning We recast the regularized global problem in (4.2) as

$$\min_{w\in\mathbb{R}^d} J(w) := \frac{1}{D}\sum\nolimits_{i=1}^{D} f_i(w) + \lambda g(w), \tag{4.3}$$

and decompose it as a dual optimization problem[1] in a distributed scenario [115] amongst K participating clients. For this, at first, we define $X \in \mathbb{R}^{d\times D_k}$ as a matrix with columns having data points for $i \in \mathcal{D}_k, \forall k$. Then, the corresponding dual optimization problem of (4.3) for a convex loss function f is

$$\max_{\alpha\in\mathbb{R}^D} \mathcal{G}(\alpha) := \frac{1}{D}\sum\nolimits_{i=1}^{D} -f_i^*(-\alpha_i) - \lambda g^*(\phi(\alpha)), \tag{4.4}$$

where $\alpha \in \mathbb{R}^D$ is the dual variable mapping to the primal candidate vector, f_i^* and g^* are the convex conjugates of f_i and g respectively [116]; $\phi(\alpha) = \frac{1}{\lambda D}X\alpha$. With the optimal value of dual variable α^* in (4.4), we have $w(\alpha^*) = \nabla g^*(\phi(\alpha^*))$ as the optimal solution of (4.3) [115]. For the ease of representation, we will use $\phi \in \mathbb{R}^d$ for $\phi(\alpha)$ hereafter. We consider that g is a strongly convex function, i.e., $g^*(\cdot)$ is continuous differentiable. Then, the solution is obtained following an iterative approach to attain a global accuracy $0 \le \epsilon \le 1$ (i.e., $\mathbb{E}\left[\mathcal{G}(\alpha) - \mathcal{G}(\alpha^*)\right] < \epsilon$).

Under the distributed setting, we further define data partitioning notations for clients $k \in \mathcal{K}$ to represent the working principle of the framework. Let us define a weight vector $\varrho_{[k]} \in \mathbb{R}^D$ at the local subproblem k with its elements zero for the unavailable data points. Following the assumption of having f_i as $(1/\gamma)$-smooth and 1-strongly convex of g to ensure convergence, its consequences is the approximate solution to the local problem k defined by the dual variables $\alpha_{[k]}$, $\varrho_{[k]}$, characterized as

$$\max_{\varrho_{[k]}\in\mathbb{R}^D} \mathcal{G}_k(\varrho_{[k]}; \phi, \alpha_{[k]}), \tag{4.5}$$

where $\mathcal{G}_k(\varrho_{[k]}; \phi, \alpha_{[k]}) = -\frac{1}{K} - \langle\nabla(\lambda g^*(\phi(\alpha))), \varrho_{[k]}\rangle - \frac{\lambda}{2}\|\frac{1}{\lambda D}X_{[k]}\varrho_{[k]}\|^2$ is defined with a matrix $X_{[k]}$ columns having data points for $i \in \mathcal{D}_k$, and zero padded otherwise. Each participating client $k \in \mathcal{K}$ iterates over its computational resources using any arbitrary solver to solve its local problem (4.5) with a local relative θ_k accuracy that characterizes the quality of the local solution, and produces a random output $\varrho_{[k]}$ satisfying

$$\mathbb{E}\left[\mathcal{G}_k(\varrho^*_{[k]}) - \mathcal{G}_k(\varrho_{[k]})\right] \le \theta_k\left[\mathcal{G}_k(\varrho^*_{[k]}) - \mathcal{G}_k(0)\right]. \tag{4.6}$$

[1] The duality gap provides a certificate to the quality of local solutions and facilitates distributed training.

Note that, with local (relative) accuracy $\theta_k \in [0, 1]$, the value of $\theta_k = 1$ suggests that no improvement was made by the local solvers during successive local iterations. Then, the local dual variable is updated as follows:

$$\alpha_{[k]}^{t+1} := \alpha_{[k]}^{t} + \varrho_{[k]}^{t}, \forall k \in \mathcal{K}. \tag{4.7}$$

Correspondingly, each participating client will broadcast the local parameter defined as $\Delta\phi_{[k]}^{t} := \frac{1}{\lambda D} X_{[k]} \varrho_{[k]}^{t}$, during each round of communication to the MEC server. The MEC server aggregates the local parameter (averaging) with the following rule:

$$\phi^{t+1} := \phi^{t} + \frac{1}{K} \sum\nolimits_{k=1}^{K} \Delta\phi_{[k]}^{t}, \tag{4.8}$$

and distributes the global change in ϕ to the participating clients, which is used to solve (4.5) in the next round of local iterations. This way we observe the decoupling of global model parameter from the need of local clients' data[2] for training a global model.

Algorithm 4 briefly summarizes the federated learning framework as an iterative process to solve the global problem characterized in (4.3) for a global accuracy level. The iterative process (S2)–(S8) of Algorithm 4 terminates when the global accuracy ϵ is reached. A participating client k strategically[3] iterates over its local training data $\mathcal{D}_k$ to solve the local subproblem (4.5) up to an accuracy θ_k. In each communication round with the MEC server, the participating clients *synchronously* pass on their parameters $\Delta\phi_{[k]}$ using a shared wireless channel. The MEC server then aggregates the local model parameters ϕ as in (4.8), and broadcasts the global parameters required for the participating clients to solve their local subproblems for the next communication round. Within the framework, consider that each participating client uses any arbitrary optimization algorithm (such as *Stochastic Gradient Descent (SGD)*, *Stochastic Average Gradient (SAG), Stochastic Variance Reduced Gradient (SVRG)*) to attain a relative θ accuracy per local subproblem. Then, for strongly convex objectives, the number of iterations is dependent on local relative θ accuracy of the local subproblem and the global model's accuracy ϵ as [58]:

$$I^{g}(\epsilon, \theta) = \frac{\zeta \cdot \log(\frac{1}{\epsilon})}{1-\theta}, \tag{4.9}$$

where the local relative accuracy measures the quality of the local solution as defined in the earlier paragraphs. Further, in this formulation, we have replaced the term $\mathcal{O}(\log(\frac{1}{\epsilon}))$ in the numerator with $\zeta \cdot \log(\frac{1}{\epsilon})$, for a constant $\zeta > 0$. For fixed iterations I^{g} at the MEC server to solve the global problem, we observe in (4.9) that

[2] Note that we consider the availability of quality of data with each participating client for solving a corresponding local subproblem. Further related demonstration on dependency of the normalized data size and accuracy can be found in [117].

[3] Fewer iterations might not be sufficient to have an optimal local solution [111].

a very high local accuracy (small θ) can significantly improve the global accuracy ϵ. However, each client k has to spend excessive resources in terms of local iterations, I_k^l to attain a small θ_k accuracy as

$$I_k^l(\theta_k) = \gamma_k \log\left(\frac{1}{\theta_k}\right), \tag{4.10}$$

where $\gamma_k > 0$ is a parameter choice of client k that depends on the data size and condition number of the local subproblem [58]. Therefore, to address this trade-off, MEC server can setup an economic interaction environment (a crowdsourcing framework) to motivate the participating clients for improving the local relative θ_k accuracy. Correspondingly, with the increased reward, the participating clients are motivated to attain better local θ_k accuracy, which as observed in (4.9) can improve the global ϵ accuracy for a fixed number of iterations I^g of the MEC server to solve the global problem. In this scenario, to capture the statistical and system-level heterogeneity, the corresponding performance bound in (9) for heterogeneous responses θ_k can be modified considering the worst-case response of the participating client as

$$I^g(\epsilon, \theta_k) = \frac{\zeta \cdot \log(\frac{1}{\epsilon})}{1 - \max_k \theta_k}, \forall k \in \mathcal{K}. \tag{4.11}$$

Figure 4.5 describes an interaction environment incorporating a crowdsourcing framework and federated learning setting. In the following section, we will further discuss in detail the proposed incentive mechanism and present the interaction between the MEC server and participating clients as a two-stage Stackelberg game.

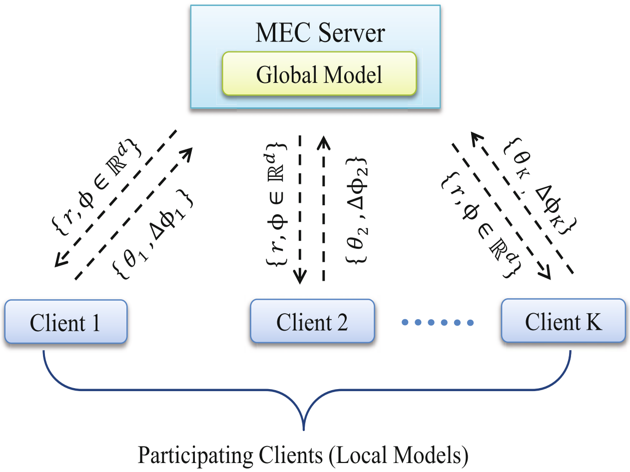

Fig. 4.5 Interaction environment of federated learning setting under crowdsourcing framework

Cost Model

Training on local data for a defined accuracy level incurs a cost for the participating clients. We discuss its significance with two typical costs: the computing cost and the communication cost.

Computing Cost This cost is related to the number of iterations performed by client k on its local data to train the local model for attaining a relative accuracy of θ_k in a single round of communication. With (4.10), we define the computing cost for client k when it performs computation on its local data $\mathcal{D}_k$.

Communication Cost This cost is incurred when client k interacts with MEC server for parameter updates to maintain θ_k accuracy. During a round of communication with the MEC server, let e_k be the size (in bits) of local parameters $\Delta\phi_{[k]}, k \in \mathcal{K}$ in a floating-point representation produced by the participating client k after processing a mini-batch [118]. While e_k is the same for all the participating clients under a specified learning setting of the global problem, each participating client k can invest resources to attain specific θ_k as defined in (4.10). Although the best choice would be to choose θ_k such that the local solution time is comparable with the time expense in a single communication round, larger θ_k will induce more rounds of interaction between clients until global convergence, as formalized in (4.9).

With the inverse relation of global iteration upon local relative accuracy in (4.9), we can characterize the total communication expenditure as

$$T(\theta_k) = \frac{T_k}{(1-\theta_k)}, \tag{4.12}$$

where T_k as the time required for the client k to communicate with MEC server in each round of model's parameter exchanges. Here, we normalize $\zeta > 0$ in (4.9) to 1 as the constant can be absorbed into T_k for each round of model's parameter exchanges when we characterize the communication expenditure in (4.12). Using first-order Taylor's approximation,[4] we can approximate the total communication cost as $T(\theta_k) = T_k \cdot (1 + \theta_k)$. We assume that clients are allocated orthogonal sub-channels so that there is no interference between them.[5] Therefore, the instantaneous data rate for client k can be expressed as

$$R_k = B \log_2\left(1 + \frac{p_k|G_k|^2}{\mathcal{N}_k}\right), \forall k \in \mathcal{K}, \tag{4.13}$$

[4] First-order Taylor's approximation for $f(\theta) = \frac{1}{1-\theta}$ is $f(\theta) \mid_{\theta=a} = f(a) + f'(a)(\theta - a)$. For small θ, the approximation results $f(\theta) \mid_{\theta=0} = 1 + \theta$.

[5] Note that the scenario of possible delay introduced with interference on poor wireless uplink channel can affect the local model update time. This can be mitigated by adjusting maximum waiting time as in [112] at MEC.

where B is the total bandwidth allocated to the client k, p_k is the transmission power of the client k, $|G_k|^2$ is the channel gain between participating client k and the base station, and $\mathcal{N}_k$ is the Gaussian noise power at client k. Then for client k, using (4.13), we can characterize T_k for each round of communication with the MEC server to upload the required updates as

$$T_k = \frac{e_k}{B \log_2\left(1 + \frac{p_k |G_k|^2}{\mathcal{N}_k}\right)}, \forall k \in \mathcal{K}. \tag{4.14}$$

(4.14) provides the dependency of T_k on wireless conditions and network connectivity.

Assimilating the rationale behind our earlier discussions, for a participating client with evaluated T_k, the increase in value of θ_k (poor local accuracy) will contribute to a larger communication expenditure. This is because the participating client has to interact more frequently with the MEC server (increased number of global iterations) to update its local model parameter for attaining relative θ_k accuracy. Further, the authors in [119] have provided the convergence analysis to justify this relationship and the communication cost model, though with a different technique.

Therefore, the participating client k's cost for the relative accuracy level θ_k on the local subproblem is

$$C_k(\theta_k) = (1 + \theta_k) \cdot \left(\nu_k \cdot T_k + (1 - \nu_k) \cdot \gamma_k \log\left(\frac{1}{\theta_k}\right)\right), \tag{4.15}$$

where $0 \leq \nu_k \leq 1$ is the normalized monetary weight for communication and computing costs (i.e., \$/ rounds of iteration). A **smaller value of relative accuracy θ_k indicates a high local accuracy**. Thus, there exists a trade-off between the communication and the computing cost (4.15). A participating client can adjust its preference on each of these costs with the weight metric ν_k. The higher value of ν_k emphasizes the larger rounds of interaction with the MEC server to adjust its local model parameters for the relative θ_k accuracy. On the other hand, the higher value of $(1-\nu_k)$ reflects the increased number of iterations at the local subproblem to achieve the relative θ_k accuracy. This will also significantly reduce the overall contribution of communication expenditure in the total cost formulation for the client. Note that the client cost over iterations could not be the same. However, to make the problem more tractable, according to (9) we consider minimizing the upper-bound of the cost instead of the actual cost, similar to approach in [111].

4.2.2 Stackelberg Game-Based Solution

In this section, firstly, we present our motivation to realize the concept of federated learning by employing a crowdsourcing framework. We next advocate an incentive mechanism required to realize this setting of decentralized learning model with our proposed solution approach.

Incentive Mechanism: A Two-Stage Stackelberg Game Approach

The MEC server will allocate rewards to the participating clients to achieve optimal local accuracy in consideration for improving the communication efficiency of the system. That means the MEC server will plan to incentivize clients for maximizing its own benefit, i.e., an improved global model. Consequently, upon receiving the announced reward, any rational client will individually maximize their own profit. Such an interaction scenario can be realized with a Stackelberg game approach.

Specifically, we formulate our problem as a two-stage Stackelberg game between the MEC server (leader) and participating clients (followers). Under the crowd-sourcing framework, the MEC server designs an incentive mechanism for participating clients to attain a local consensus accuracy level[6] on the local models while improving the performance of a centralized model. The MEC server cannot directly control the participating clients to maintain a local consensus accuracy level and requires an effective incentive plan to enroll clients for this setting.

Clients (Stage II) The MEC server has an advantage, being a leader with the first-move advantage influencing the followers for participation with a local consensus accuracy. It will at first announce a uniform reward rate[7] (e.g., a fair data package discount as $/accuracy level) $r > 0$ for the participating clients. Given r, at Stage II, a rational client k will try to improve the local model's accuracy for maximizing its net utility by training over the local data with global parameters. The proposed utility framework incorporates the cost involved while a client tries to maximize its own individual utility.

Client Utility Model We use a valuation function $v_k(\theta_k)$ to denote the model's effectiveness that explains the valuation of the client k when relative θ_k accuracy is attained for the local subproblem.

Assumption 1 The valuation function $v_k(\theta_k)$ is a linear, decreasing function with $\theta_k > 0$, i.e., $v_k(\theta_k) = (1-\theta_k)$. Intuitively, for a smaller relative accuracy at the local subproblem, there will be an increase in the reward for the participating clients.

---

[6] It signifies the agreement among the participating clients on the quality of solution at the local subproblems for building a high-quality centralized learning model.

[7] Prominently, two kinds of pricing schemes exist at present following different design goals: uniform pricing and discriminatory or differentiated pricing [120]. The differentiated pricing scheme is more efficient, but also requires more information and higher complexity than the uniform pricing [121, 122]. Therefore, based upon offered motivations and benefits, our proposed crowdsourcing framework follows a platform-centric model to train a high-quality global model with low complexity, less information exchange by using the uniform pricing scheme.

Given $r > 0$, each participating client k's strategy is to maximize its own utility as follows:

$$\max_{0 \leq \theta_k \leq 1} \quad u_k(r, \theta_k) = r(1 - \theta_k) - C_k(\theta_k), \tag{4.16}$$

given cost $C_k(\theta_k)$ as (4.15). The feasible solution is always restricted to the value less than 1 (i.e., without loss of generality, for $\theta_k > 1$, it violates the participation assumption for the crowdsourcing framework). Therefore, problem (4.16) can be represented as

$$\max_{\theta_k > 0} \quad u_k(r, \theta_k) = r(1 - \theta_k) - C_k(\theta_k), \forall k \in \mathcal{K}. \tag{4.17}$$

Also, we have $C_k^{''}(\theta_k) > 0$, which means $C_k(\theta_k)$ is a strictly convex function. Thus, there exists a unique solution $\theta_k^*(r), \forall k$.

MEC Server (Stage I) Knowing the response (strategy) of the participating clients, the MEC can evaluate an optimal reward rate r^* to maximize its utility. The utility $U(\cdot)$ of the MEC server can be defined in relation to the satisfaction measure achieved with the local consensus accuracy level.

MEC Server Utility Model We define $x(\epsilon)$ as the number of iterations required for an arbitrary algorithm to converge to some ϵ accuracy. We similarly define $I^g(\epsilon, \theta)$ as global iterations of the framework to reach a relative θ accuracy on the local subproblems.

From this perspective, we require an appropriate utility function $U(\cdot)$ as the satisfaction measure of the framework with respect to the number of iterations for achieving ϵ accuracy. In this regard, use the definition of the number of iterations for ϵ accuracy as

$$x(\epsilon) = \zeta \cdot \log\left(\frac{1}{\epsilon}\right).$$

Due to large values of iterations, we approximate $x(\epsilon)$ as a continuous value, and with the aforementioned relation, we choose $U(\cdot)$ as a strictly concave function of $x(\epsilon)$ for $\epsilon \in [0, 1]$, i.e., with the increase in $x(\epsilon)$, $U(\cdot)$ also increases. Thus, we propose $U(x(\epsilon))$ as the normalized utility function bounded within [0, 1] as

$$U(x(\epsilon)) = 1 - 10^{-(ax(\epsilon)+b)}, \quad a \geq 0, b \leq 0, \tag{4.18}$$

which is strictly increasing with $x(\epsilon)$, and represents the satisfaction of MEC increase with respect to accuracy ϵ.

As for the global model, there exists an acceptable value of threshold accuracy measure correspondingly reflected by $x_{\min}(\epsilon)$. This suggests the possibility of near-zero utility for the MEC server for failing to attain such value.

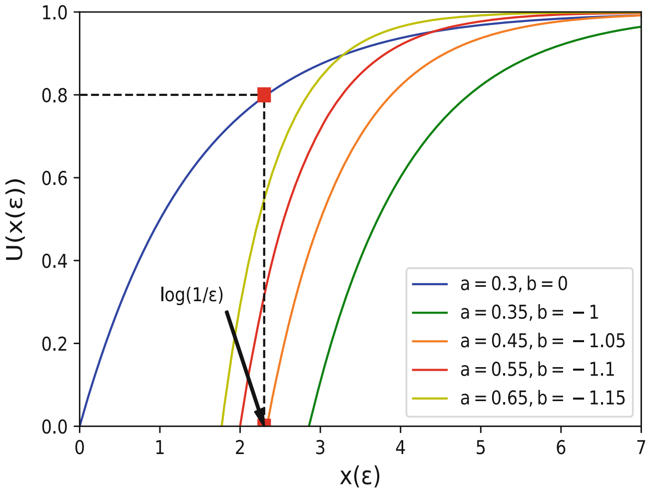

Fig. 4.6 MEC utility $U(\cdot)$ as a function of ϵ with different parameter values of a, b

Figure 4.6 depicts our proposed utility function, a concave function of $x(\epsilon)$ with parameters a and b that reflect the required behavior of the utility function defined in (4.18). In Fig. 4.6, we can observe that a larger value of a means smaller iterations requirement and larger values of b introduces flat curves suggesting more flexibility in accuracy. So we can analyze the impact of parameters a and b in (4.18), and set them to model the utility function for the MEC server as per the design requirements of the learning framework. Furthermore, in our setting, $I^{g}(\epsilon, \theta)$ can be written as

$$I^{g}(\epsilon, \theta) = \frac{x(\epsilon)}{1-\theta} \leq \delta. \tag{4.19}$$

(4.19) explains the efficiency paradigm of the proposed framework in terms of time required for the convergence to some accuracy ϵ. If $\tau^{1}(\theta)$ is the time per iteration to reach a relative θ accuracy at a local subproblem and $T(\theta)$ is the communication time required during a single iteration for any arbitrary algorithm, then we can analyze the result in (4.19) with the efficiency of the global model as

$$I^{g}(\epsilon, \theta) \cdot (T(\theta) + \tau^{1}(\theta)). \tag{4.20}$$

Because the cost of communication is proportional to the speed and energy consumption in a distributed scenario [123], the bound defined in (4.19) explains the efficiency in terms of the MEC server's resource restriction for attaining ϵ accuracy. In this regard, the corresponding analysis of (4.20) is presented in the upcoming sub-section with several case studies.

The utility of the MEC server can therefore be defined for the set of measured best responses $\boldsymbol{\theta}^*$ as

$$\mathcal{U}(x(\epsilon), r|\boldsymbol{\theta}^*) = \beta\left(1 - 10^{-(ax(\epsilon)+b)}\right) - r\sum_{k\in\mathcal{K}}(1 - \theta_k^*(r)),$$

where $\beta > 0$ is the system parameter,[8] and $r\sum_{k\in\mathcal{K}}(1 - \theta_k^*(r))$ is the cost spent for incentivizing participating clients in the crowdsourcing framework for federated learning. So, for the measured $\boldsymbol{\theta}^*$ from the participating clients at the MEC server, the utility maximization problem can be formulated as follows:

$$\max_{r\geq 0, x(\epsilon)} \quad \mathcal{U}(x(\epsilon), r|\boldsymbol{\theta}^*), \tag{4.21}$$

$$\text{s.t.} \quad \frac{x(\epsilon)}{1 - \max_k \theta_k^*(r)} \leq \delta. \tag{4.22}$$

In constraint (4.22), $\max_k \theta_k^*(r)$ characterizes the worst-case response for the server-side utility maximization problem with the bound on permissible global iterations. Note that MEC adapts admission control strategy (discussed in Sect. 4.2.3) to improve the number of participants for maximizing its utility. In fact, MEC has to increase the reward rate to maintain the minimum number of participation (at least two) to realize the distributed optimization setting in federated learning. In addition to this, the framework may suffer from slower convergence due to less participation. Thus, MEC will avoid deliberately dropping the clients to achieve a faster consensus with (4.22).

Furthermore, using the relationship defined in (4.19) between $x(\epsilon)$ and relative θ accuracy for the subproblem, we can analyze the impact of responses $\boldsymbol{\theta}$ on MEC server's utility in a federated learning setting with the constraint (4.11). To be more specific about this relation, we can observe that with the increased value of $(1-\theta)$, i.e., lower relative accuracy (high local accuracy), the MEC server can attain better utility due to the corresponding increment in the value of $x(\epsilon)$. Note that in the client cost problem, $x(\epsilon)$ is treated as a constant provided by the MEC problem, and can be ignored for solving (4.16).

Lemma 1 *The optimal solution* $x^*(\epsilon)$ *for (4.21) can be derived as* $\delta(1 - \max_k \theta_k^*(r))$.

Proof See Appendix.

[8] Note that $\beta > 0$ characterizes a linear scaling metric to the utility function which can be set arbitrarily and will not alter our evaluation. Equivalently, it can be understood as the MEC server's physical resource consignments for the federated learning that reflects the satisfaction measure of the framework.

Therefore, for the given $\boldsymbol{\theta}^*(r)$, we can formalize (4.21) as

$$\max_{r\geq 0} \quad \beta\left(1-10^{-(ax^*(\epsilon)+b)}\right) - r\sum_{k\in\mathcal{K}}(1-\theta_k^*(r)). \tag{4.23}$$

Stackelberg Equilibrium With a solution to MEC server's utility maximization problem, r^* we have the following definition.

Definition 1 For any values of r, and $\boldsymbol{\theta}$, $(r^*, \boldsymbol{\theta}^*)$ is a Stackelberg equilibrium if it satisfies the following conditions:

$$\mathcal{U}(r^*, \boldsymbol{\theta}^*) \geq \mathcal{U}(r, \boldsymbol{\theta}^*), \tag{4.24}$$

$$u_k(\theta_k^*, r^*) \geq u_k(\theta_k, r^*), \ \forall k. \tag{4.25}$$

Next, we employ the backward-induction method to analyze the Stackelberg equilibria: the Stage-II problem is solved at first to obtain $\boldsymbol{\theta}^*$, which is then used for solving the Stage-I problem to obtain r^*.

Stackelberg Equilibrium: Algorithm and Solution Approach

Intuitively, from (4.19), we see that the server can evaluate the maximum value of $x(\epsilon)$ required for attaining accuracy ϵ for the centralized model while maintaining relative accuracy θ_{th} amongst the participating clients. Here, θ_{th} is a consensus on a maximum local accuracy level amongst participating clients, i.e., the local subproblems will maintain at least θ_{th} relative accuracy. So, with the measured responses $\boldsymbol{\theta}$ from the participating clients, the server can design a proper incentive plan to improve the global model while maintaining the worst-case relative accuracy $\max_k \theta_k^*$ as θ_{th} for the local model.

Since the threshold accuracy θ_{th} can be adjusted by the MEC server for each round of solution, each participating client will maintain a response towards the maximum local consensus accuracy θ_{th}. This formalizes the client's selection criteria [see Remark 1.] which is sufficient enough for the MEC server to maintain the accuracy ϵ. We also have the lower bound related with the value of $x_{\min}(\epsilon)$ for equivalent accuracy $\epsilon_{\max}$ while dealing with the client's responses $\boldsymbol{\theta}$, i.e.,

$$\log\left(\frac{1}{\epsilon_{\max}}\right) \leq \frac{x(\epsilon)}{(1-\theta_{\text{th}})} \leq \delta_{\max}. \tag{4.26}$$

where $\delta_{\max}$ is the maximum permissible upper bound to the global iterations.

As explained before and with (4.26), the value of θ_{th} can be varied (lowered) by MEC server to improve the overall performance of the system. For a worst case scenario, where the offered reward r for the client k is insufficient to motivate it for participation with improved local relative accuracy, we might have $\max_k \theta_k^*(r) = 1$, i.e., $\theta_{\text{th}} = 1$, no participation.

Lemma 2 *For a given reward rate r, and T_k which is determined based upon the channel conditions (4.14), we have the unique solution $\theta_k^*(r)$ for the participating client satisfying following relation:*

$$g_k(r) = \log(e^{1/\theta_k^*(r)}\theta_k^*(r)), \forall k \in \mathcal{K}, \tag{4.27}$$

for $g_k(r) \geq 1$, where,

$$g_k(r) = \left[\frac{r + \nu_k T_k}{(1-\nu_k)\gamma_k} - 1\right].$$

Proof Because $C_k^{''}(\theta_k) > 0$ for $\theta_k > 0$, (4.17) is a strictly convex function resulted as a linear plus convex structure. Therefore, by the first-order condition, (4.17) can be deduced as

$$\begin{aligned} &\frac{\partial u_k(r, \theta_k)}{\partial \theta_k} = 0 \\ &\qquad \Leftrightarrow \frac{1}{\theta_k} - \log\left(\frac{1}{\theta_k}\right) = \left[\frac{r + \nu_k T_k}{(1-\nu_k)\gamma_k} - 1\right], \\ &\qquad \Leftrightarrow \log(e^{1/\theta_k}\theta_k) = g_k(r). \end{aligned} \tag{4.28}$$

We observe that Lemma 2 is a direct consequence of the solution structure derived in (4.28). Hence, we conclude the proof.

From Lemma 2, we have some observations with the definition of $g_k(r)$ for the response of the participating clients. First, we can show that θ_k^* is larger for the poor channel condition on a given reward rate. Second, in such scenario, with the increase in reward rate, say for $g_k(r) > 2$ the participating clients will iterate more during their computation phase resulting in lower θ_k^*. This will reduce the number of global iterations to attain an accuracy level for the global problem.

We can therefore characterize the participating client k's best response under the proposed framework as

$$\theta_k^*(r) = \min\left\{\hat{\theta}_k(r) \mid_{g_k(r)=\log(e^{1/\hat{\theta}_k(r)}\hat{\theta}_k(r))}, \theta_{\text{th}}\right\}, \forall k. \tag{4.29}$$

(4.29) represents the best response strategy for the participating client k under our proposed framework. Intuitively, exploring the logarithmic structure in (4.27), we observe that the increase in incentive r will motivate participating clients to increase their efforts for local iteration in one global iteration. This is reflected by a better response, i.e., a lower relative accuracy (high local accuracy) during each round of communication with the MEC server.

Figure 4.7 illustrates such strategic responses of the participating clients over an offered reward for a given configuration. In this scenario, to elaborate the best

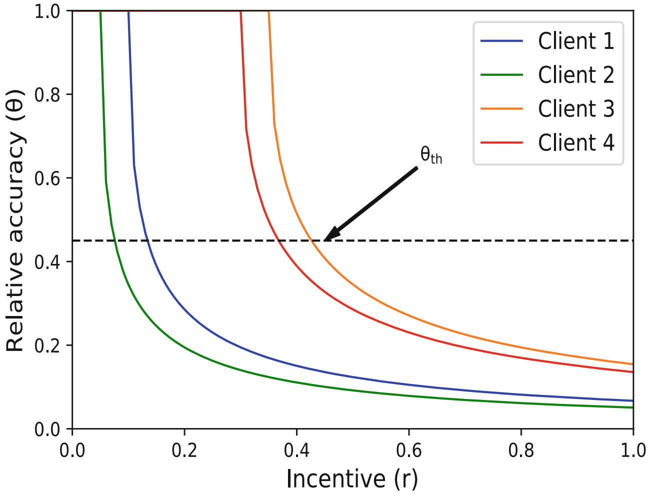

Fig. 4.7 An illustration showing participating clients response over the offered reward rate

response strategy as characterized in (4.29), we have considered four participating clients with different preferences (e.g., Client 3 being the most reluctant participant). We observe that Client 3 seeks more incentive r to maintain a comparable accuracy level as Client 1. Further, we consider the tradeoff between communication cost and the computation cost as discussed with the relation in (4.15). These costs are complementary in relation by ν_k, and for each client k their preferences upon these costs are also different. For instance, the higher value of ν_k for client k emphasizes the increased number of communication with the MEC server to improve the local relative accuracy θ_k.

In Figs. 4.8, 4.9, and 4.10, we briefly present the solution analysis to (4.27) with the impact of channel condition (we define it as communication adversity) on the local relative accuracy for a constant reward. For this, in Fig. 4.8 we consider a participating client with the fixed offered reward setting r from uniformly distributed values of 0.1–5. We use normalized T_k parameter for a client k to illustrate the response analysis scenario. In Figs. 4.9 and 4.10, T_k is uniformly distributed on [0.1, 1], and ν_k is set at 0.6. Intuitively, as in Fig. 4.8, the increase in communication time T_k for a fixed reward r will influence participating clients to iterate more locally for improving local accuracy than to rely upon the global model, which will minimize their total cost. Under this scenario, we observe the increase in communication cost with the increase in communication time T_k. Thus, the clients will iterate more locally. However, the trend is significantly affected by normalized weights ν_k, as observed in Figs. 4.9 and 4.10. For a larger value of T_k (poor channel condition) as in the case of Fig. 4.10, increasing the value of ν_k, i.e., clients with more preference on the communication cost in the total cost model results to higher local iterations for solving local subproblems, as reflected by the better local accuracy, unlike in Fig. 4.9. In both cases we observe the decrease in

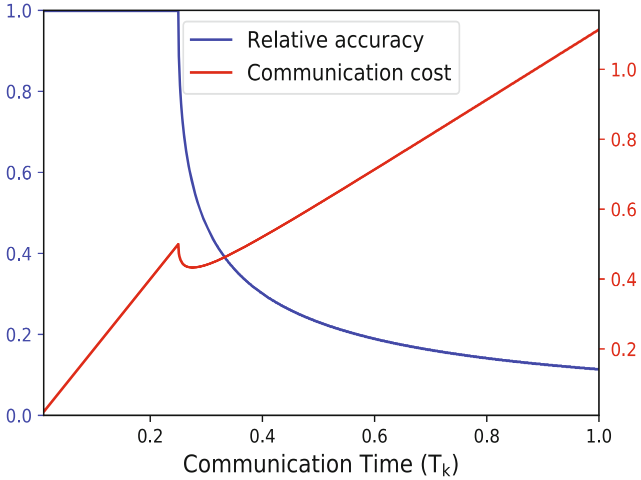

Fig. 4.8 Solution Analysis (4.27) (Left Y-axis: Relative accuracy, Right Y-axis: Communication cost): impact of communication adversity on local relative accuracy for a constant reward

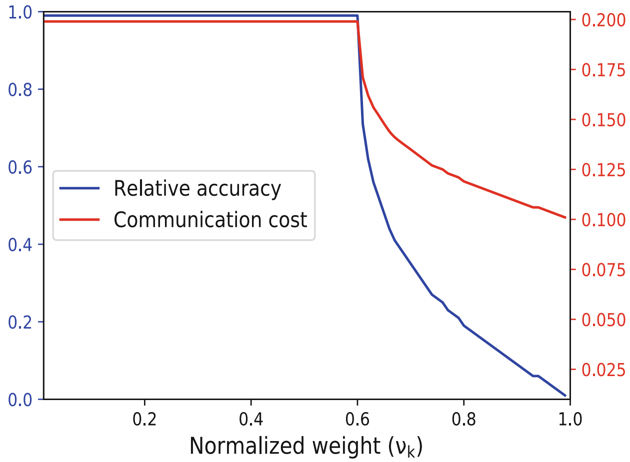

Fig. 4.9 Solution Analysis (4.27) (Left Y-axis: Relative accuracy, Right Y-axis: Communication cost): normalized weight versus relative accuracy for a fair data rate (quality communication channel)

communication cost upon participation. However, in Fig. 4.10 the communication cost is higher because of an expensive data rate. Therefore, for a given r, client k can adjust its weight metrics accordingly to improve the response θ_k.

In Figs. 4.11, 4.12, and 4.13, we explore such behaviors of the participating clients through the heatmap plot. To explain better, we define three categories of participating clients based upon the value of normalized weights $\nu_k, \forall k$, which are their individual preferences upon the computation cost and the communication

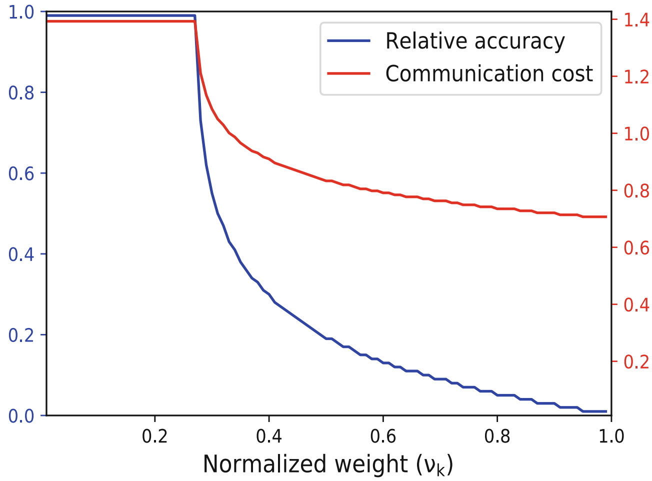

Fig. 4.10 Solution Analysis (4.27) (Left Y-axis: Relative accuracy, Right Y-axis: Communication cost): normalized weight versus relative accuracy for an expensive data rate

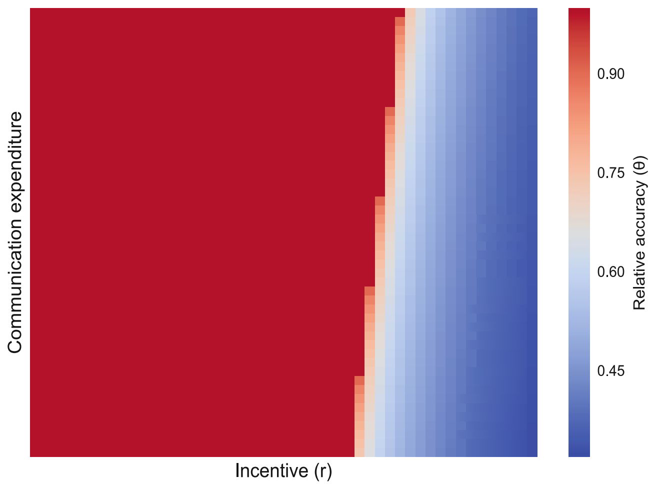

Fig. 4.11 Case Study: impact of communication cost and offered reward rate r for normalized weight (preferences), reluctant clients i.e., $\nu_k = 0.1$. X-axis shows the increase in incentive (r) value from left-to-right, and the y-axis defines the increase in value of communication expenditure (top-to-bottom)

cost for the convergence of the learning framework. (i) *Reluctant* clients with a lower ν_k consume more reward to improve local accuracy, even though the value of T_k is larger (expensive), as observed in Fig. 4.11. (ii) *Sensitive* clients are more susceptible towards the channel quality with larger ν_k, and iterates more locally within a round of communication to the MEC server for improving local accuracy,

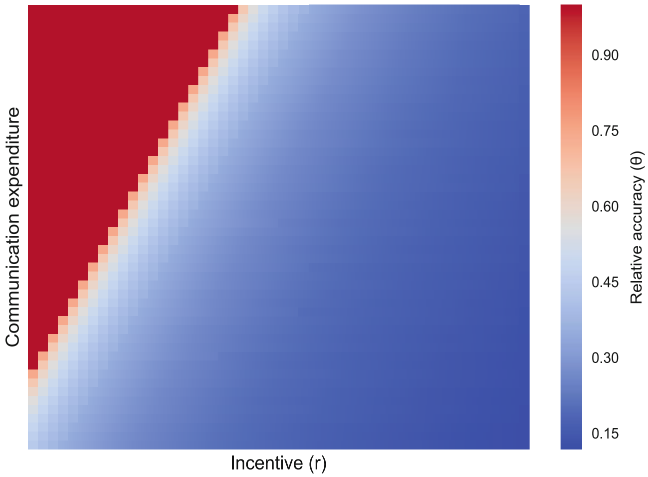

Fig. 4.12 Case Study: impact of communication cost and offered reward rate r for normalized weight (preferences), rational clients, $\nu_k = 0.7$. X-axis shows the increase in incentive (r) value from left-to-right, and the y-axis defines the increase in value of communication expenditure (top-to-bottom)

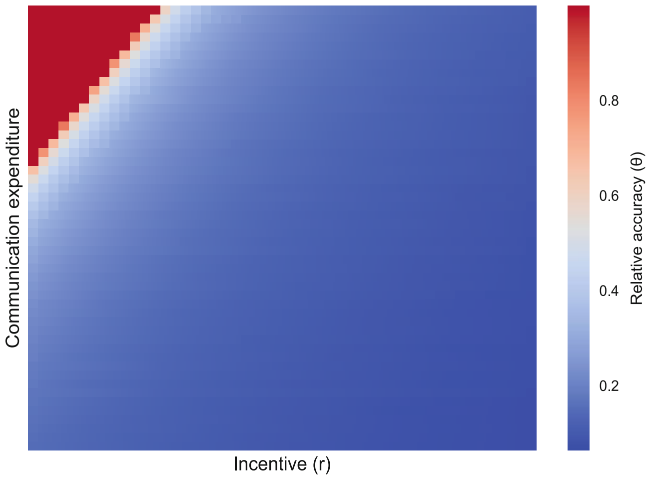

Fig. 4.13 Case Study: impact of communication cost and offered reward rate r for normalized weight (preferences), sensitive clients, $\nu_k = 0.7$. X-axis shows the increase in incentive (r) value from left-to-right, and the y-axis defines the increase in value of communication expenditure (top-to-bottom)

as observed in Fig. 4.13. (iii) *Rational* clients, as referred in Fig. 4.12 tend to balance these extreme preferences (say $\nu_k = 0.5$ for client k), which in fact would be unrealistic to expect all the time due to heterogeneity in participating client's resources.

Algorithm 5 MEC server's utility maximization

Sort clients as with $\hat{r}_1 < \hat{r}_2 < \ldots < \hat{r}_K$
$\mathcal{R} = \{\}, \mathcal{A} = \mathcal{K}, j = K$
while $j > 0$ **do**
    Obtain the solutions r_j to the following problem:

$$\max_{r \geq \hat{r}_1} \beta \left(1 - 10^{-(ax^*(\epsilon)+b)}\right) - r \sum\nolimits_{k \in \mathcal{A}} (1 - \theta_k^*(r))$$

    if $r_j > \hat{r}_j$, **then** $\mathcal{R} = \mathcal{R} \cup \{r_j\}$;
    end if
    $\mathcal{A} = \mathcal{A} \backslash j$;
    $j = j - 1$;
end while
Return $r_j \in \mathcal{R}$ with highest optimal values in problem (4).

To solve (4.23) efficiently, with (4.29) $\theta_k^*(r) = \min \left\{ \hat{\theta}_k(r) \mid_{g_k(r) = \log(e^{1/\hat{\theta}_k(r)} \hat{\theta}_k(r))}, \theta_{\text{th}} \right\}$, $\forall k$, we introduce a new variable z_k in relation with consensus on local relative accuracy θ_{th},

$$z_k = \begin{cases} 1, & \text{if } r > \hat{r}_k; \\ 0, & \text{otherwise}, \end{cases} \tag{4.30}$$

where

$$\hat{r}_k = \left[g_k^{-1}(\log(e^{1/\theta_{\text{th}}} \theta_{\text{th}})) \right]$$

is the minimum incentive value required obtained from (4.29) to attain the local consensus accuracy θ_{th} at client k for the defined parameters ν_k and T_k.

This means, $\theta_k(r) < \theta_{\text{th}}$ when $z_k = 1$, and $\theta_{\text{th}} \leq \theta_k(r) < 1$ when $z_k = 0$. MEC server can use this setting to drop the participants with poor accuracy. As discussed before, for the worst case scenario we consider $\theta_{\text{th}} = 1$.

Therefore, the utility maximization problem can be equivalently written as

$$\max_{r, \{z_k\}_{k \in \mathcal{K}}} \quad \beta \left(1 - 10^{-(ax^*(\epsilon)+b)}\right) - r \sum_{k \in K} z_k \cdot (1 - \theta_k^*(r)), \tag{4.31}$$

$$\text{s.t.} \quad r \geq 0, \tag{4.32}$$

$$z_k \in \{0, 1\}, \forall k. \tag{4.33}$$

The problem (4.31) is a mixed-Boolean programming, which may require exponential-complexity effort (i.e., 2^K configuration of $\{z_k\}_{k \in \mathcal{K}}$) to solve by the exhaustive search. To solve this problem with linear complexity, we refer to the solution approach as in Algorithm 5.

The utility maximization problem at MEC server can be reformulated as a constraint optimization problem (4.34–4.35) assuming a fixed configuration of $\{z_k = 1\}_{k \in \mathcal{K}}$ as

$$\max_{r \geq 0} \quad \beta \left(1 - 10^{-(ax^*(\epsilon)+b)}\right), \tag{4.34}$$

$$\text{s.t.} \quad r \sum_{k \in K} (1 - \theta_k^*(r)) \leq B, \tag{4.35}$$

where (4.35) is budget constraint for the problem. The second-order derivative of function $r(1 - \theta_k^*(r))$ in (4.35) is $\frac{2\gamma_k(1-\nu_k)\nu_k T_k}{(r+\nu_k T_k)^3} > 0$, i.e., the problem (4.34) is a convex problem and can be solved similarly with Algorithm 5 (line 4–5).

Proposition 1 *Algorithm 2 can solve the Stage-I equivalent problem (4.23) with linear complexity.*

Proof As the clients are sorted in the order of increasing $\hat{r}_k$ (line 1), for the sufficient condition $r > \hat{r}_k$ resulting $z_k = 1$, the MEC's utility maximization problem reduces to a single-variable problem that can be solved using popular numerical methods.

Remark 1 Algorithm 2 can maintain consensus accuracy by formalizing the clients selection criteria. This is because from (4.30), $z_k = 1$ for $\theta_k(r) < \theta_{\text{th}}$, and $z_k = 0$ for $\theta_{\text{th}} \leq \theta_k(r) < 1$. Thus, MEC server uses this setting to drop the participants with $\theta_k(r) > \theta_k^*(r) = \theta_{\text{th}}$.

Theorem 1 *The Stackelberg equilibria of the crowdsourcing framework are the set of pairs* $\{r^*, \boldsymbol{\theta}^*\}$.

Proof For any given $\boldsymbol{\theta}$, it is obvious that $\mathcal{U}(r^*, \boldsymbol{\theta}) \geq \mathcal{U}(r, \boldsymbol{\theta}), \forall r$ since r^* is the solution to the Stage-I problem. Thus, we have $\mathcal{U}(r^*, \boldsymbol{\theta}^*) \geq \mathcal{U}(r, \boldsymbol{\theta}^*)$. In the similar way, for any given value of r and $\forall k$, we have $u_k(r, \theta_k^*) \geq u_k(r, \theta_k), \forall \theta_k$. Hence, $u_k(r^*, \theta_k^*) \geq u_k(r^*, \theta_k)$. Combining these facts, we conclude the proof being based upon the definitions of (4.24) and (4.25).

4.2.3 Simulations

In this section, we present numerical simulations to illustrate our results. We consider the learning setting for a strongly convex model such as logistic regression, as discussed in Sect. 4.2.1, to characterize and demonstrate the efficacy of the proposed framework. First, we will show the optimal solution of Algorithm 5 (Algorithm 5) and conduct a comparison of its performance with two baselines. The first one, named OPT, is the optimal solution of problem (4.23) with an exhaustive search for the optimal response $\boldsymbol{\theta}^*$. The second one is called Baseline that considers the worst response amongst the participating clients to attain local consensus θ_{th}

accuracy with an offered price. This is an inefficient scheme but still enables us to attain feasible solutions. Finally, we analyze the system performance by varying different parameters and conduct a comparison of the incentive mechanism with the baseline and their corresponding utilities. In our analysis, the smaller values of local consensus are of specific interest as they reflect the effectiveness of federated learning.

1. *Settings:* For an illustrative scenario, we fix the number of participating clients to 4. We consider the system parameter $\beta = 10$, and the upper bound to the number of global iterations $\delta = 10$, which characterizes the permissible rounds of communication to ensure global ϵ accuracy. The MEC's utility $U(x(\epsilon)) = 1 - 10^{-(ax(\epsilon)+b)}$ model is defined with parameters $a = 0.3$, and $b = 0$. For each client k, we consider normalized weight ν_k is uniformly distributed on [0.1,0.5], which can provide an insight on the system's efficacy as presented in Figs. 4.11, 4.12, and 4.13. We characterize the interaction between the MEC server and the participating clients under homogeneous channel condition, and use the normalized value of T_k for all participating clients.
2. *Reward rate:* In Fig. 4.14 we increase the value of local consensus accuracy θ_{th} from 0.2 to 0.6. When the accuracy level is improved (from 0.4 to 0.2), we observe a significant increase in the reward rate. These results are consistent with the analysis in section "Stackelberg Equilibrium: Algorithm and Solution Approach". The reason is that cost for attaining a higher local accuracy level

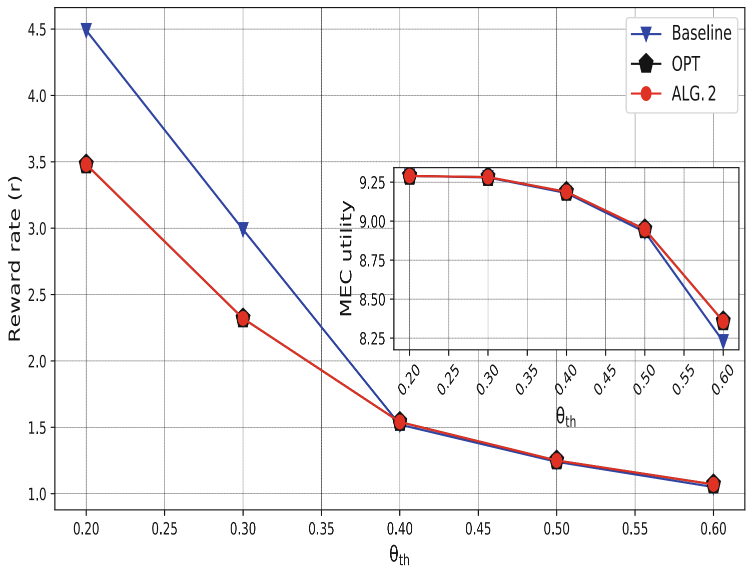

Fig. 4.14 Comparison of (**a**) Reward rate and (**b**) MEC utility under three schemes for different values of threshold θ_{th} accuracy

requires more local iterations, and thus the participating clients exert more incentive to compensate for their costs.

We also show that the reward variation is prominent for lower values of θ_{th}, and observe that scheme Algorithm 2 and OPT achieve the same performance, while Baseline is not as efficient as others. Here, we can observe up to 22% gain in the offered reward against the Baseline by other two schemes. In Fig. 4.14b, we see the corresponding MEC utilities for the offered reward that complements the competence of the proposed Algorithm 2. We see, the trend of utility against the offered reward goes along with our analysis.

3. *Parametric choice:* In Figs. 4.15 and 4.16 we show the impact of parametric choice adopted by the participating client k to solve the local subproblem [124], which is characterized by γ_k. In Fig. 4.15, we see a lower offered reward for the improved local accuracy level for the participating clients adapting same

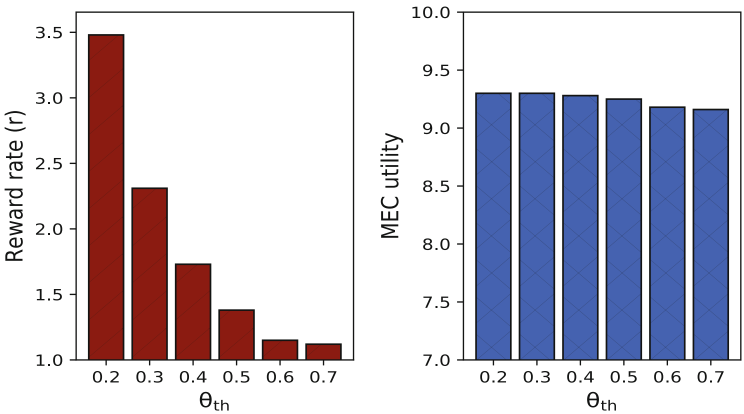

Fig. 4.15 For $|\mathcal{K}| = 4$, $a = 0.3$, $b = 0$, $\gamma_k = 1$, $\forall k$

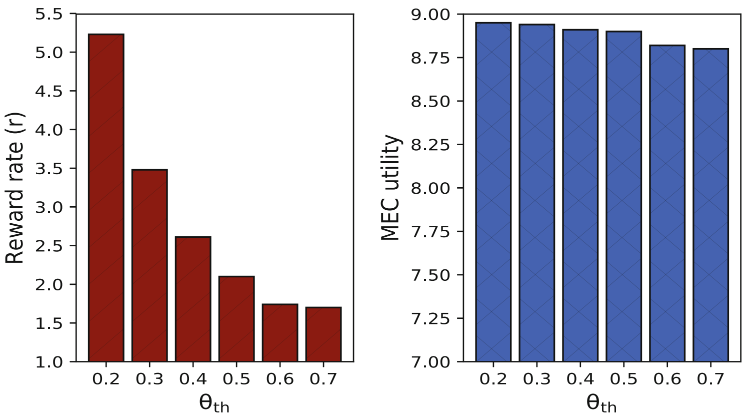

Fig. 4.16 For $|\mathcal{K}| = 4$, $a = 0.3$, $b = 0$, and $\gamma_k \sim U[1, 5]$

Table 4.2 Offered reward rate comparison with randomized γ effect for different (a, b) setting

Threshold accuracy	Baseline	Algorithm 2	Algorithm 2	Algorithm 2
θ_{th}	r	$(0.3, -1)$	$(0.35, -1)$	$0.65, -1)$
0.2	18	5.22	5.22	5.22
0.3	12	3.48	3.48	3.48
0.4	8.99	2.602	2.6	2.61
0.5	7.19	2.79	4.3	2.2
0.6	5.99	2.38	2.87	2.1
0.7	5.13	2.84	3.17	1.9

Table 4.3 Utility comparison with randomized γ effect for different (a, b) setting

Threshold accuracy	Algorithm 2	Algorithm 2	Algorithm 2
θ_{th}	$(0.3, -1)$	$(0.35, -1)$	$(0.65, -1)$
0.2	8.55	8.79	8.96
0.3	8.41	8.60	8.95
0.4	8.33	8.58	8.94
0.5	8.2	8.73	8.91
0.6	8.18	8.4	8.91
0.7	7.8	8.51	8.86

parameters (algorithms) for solving the local subproblem, in contrast to Fig. 4.16 with the uniformly distributed γ_k on [1,5] to achieve the competitive utility.

4. *Comparisons:* In Tables 4.2 and 4.3, we see the effect of randomized parameter γ_k for different configuration of MEC utility model $U(\cdot)$ defined by (a, b). For the smaller values of θ_{th}, which captures the competence of the proposed mechanism, we observe that the choice of (a, b) provides a consistent offered reward for improved utility from $(0.35, -1)$ to $(0.65, -1)$, which follows our analysis in section "Incentive Mechanism: A Two-Stage Stackelberg Game Approach". For larger values of θ_{th}, we also see the similar trend in MEC utility. For a randomized setting, we observe up to 71% gain in offered reward against the Baseline, which validates our proposal's efficacy aiding federated learning.

Our earlier discussion in Sect. 4.2.2 and simulation results explain the significance of choosing a local θ_{th} accuracy to build a global model that maximizes the utility of the MEC server. In this regard, at first, the MEC server evokes admission control to determine θ_{th} and the final model is learned later. This means, with the number of expected clients, it is crucial to appropriately select a proper prior value of θ_{th} that corresponds to the participating client's selection criteria for training a specific learning model. Note that, in each communication round of synchronous aggregation at the MEC server, the quality of local solution benefits to evaluate the performance at the local subproblem. In this section, we will discuss about the probabilistic model employed by the MEC server to determine the value of the consensus θ_{th} accuracy.

We consider the local $\boldsymbol{\theta}$ accuracy for the participating clients is an i.i.d and uniformly distributed random variable over the range $[\theta_{\min}, \theta_{\max}]$, then the PDF of the responses can be defined as $f_{\boldsymbol{\theta}}(\theta) = \frac{1}{\theta_{\max}-\theta_{\min}}$. Let us consider a sequence of discrete time slots $t \in \{1, 2, \ldots\}$, where the MEC server updates its configuration for improving the accuracy of the system. Following our earlier definitions, at time slot t, the number of participating clients in the crowdsourcing framework for federated learning is $|\mathcal{K}(t)|$, or simply K. We restrict the clients with the accuracy measure $\boldsymbol{\theta}(t) \geq \theta_{\max}$. For K number of participation requests, the total number of accepted responses $N(t)$ is defined as $N(t) = K \cdot F_{\boldsymbol{\theta}(t)}(\theta) = K \cdot P[\boldsymbol{\theta}(t) \leq \theta]$. We have $N(t) = K \cdot \left[\frac{\theta(t)-\theta_{\min}}{\theta_{\max}-\theta_{\min}}\right]$. At each time t, the MEC server chooses $\boldsymbol{\theta}(t)$ as the threshold accuracy θ_{th} that maximizes the sum of its utility as defined in (4.18) for the defined parameters $a \geq 0, b \leq 0$ and the total participation, $\beta\left(1 - 10^{-(ax(\epsilon)+b)}\right) + (1-\theta) \cdot N(t)$, subject to the constraint that the response lies between the minimum and maximum accuracy measure ($\theta_{\min} \leq \boldsymbol{\theta}(t) \leq \theta_{\max}$). Using the definitions in (4.19), for $\beta > 0$, the MEC server maximizes its utility for the number of participation with θ accuracy as

$$\begin{aligned} \max_{\theta(t)} \quad & \beta\left(1 - 10^{-(a\cdot\delta(1-\theta(t))+b)}\right) + (1-\theta(t)) \cdot N(t), \\ \text{s.t.} \quad & \theta_{\min} \leq \theta(t) \leq \theta_{\max}. \end{aligned} \tag{4.36}$$

The Lagrangian of the problem (4.36) is as follows:

$$\begin{aligned} \mathcal{L}(\theta(t), \lambda, \mu) = \beta\left(1 - 10^{-(a\cdot\delta(1-\theta(t))+b)}\right) + (1-\theta(t))\cdot \\ \left[\frac{\theta(t)-\theta_{\min}}{\theta_{\max}-\theta_{\min}}\right] + \lambda(\theta(t)-\theta_{\min}) \\ +\mu(\theta_{\max}-\theta(t)), \end{aligned} \tag{4.37}$$

where $\lambda \geq 0$ and $\mu \geq 0$ are dual variables. Problem (4.36) is a convex problem whose optimal primal and dual variables can be characterized using the Karush-Khun-Tucker (KKT) conditions [116] as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta(t)} = \ln(10)\cdot(\beta\delta a)\cdot 10^{-(a\cdot\delta(1-\theta^*(t))+b)} \\ -K \cdot \left[\frac{2\theta(t)-\theta_{\min}}{\theta_{\max}-\theta_{\min}}\right] + \lambda - \mu = 0, \end{aligned} \tag{4.38}$$

$$\lambda(\theta(t)-\theta_{\min}) = 0, \tag{4.39}$$

$$\nu(\theta_{\max}-\theta(t)) = 0. \tag{4.40}$$

Following the complementary slackness criterion, we have

$$\lambda^*(\theta^*(t) - \theta_{\min}) = 0, \mu^*(\theta_{\max} - \theta^*(t)) = 0, \lambda^* \geq 0, \mu^* \geq 0. \tag{4.41}$$

Therefore, from (4.41), we solve (4.36) with the KKT conditions assuming that $\theta^*(t) < \theta_{\max}$ as an admission control strategy, and find the optimal $\theta^*(t)$ that satisfies the following relation

$$K = \frac{\ln(10) \cdot (\beta\delta a) \cdot 10^{-(a\cdot\delta(1-\theta^*(t))+b)} \cdot (\theta_{\min} - \theta_{\max})}{1 - 2\theta^*(t) + \theta_{\min}}. \tag{4.42}$$

(4.42) can be rearranged as

$$f(\theta^*(t)) = \ln(10) \cdot (\beta\delta a) \cdot 10^{-(a\cdot\delta(1-\theta^*(t))+b)} + K \cdot \left[\frac{1 - 2\theta^*(t) + \theta_{\min}}{\theta_{\max} - \theta_{\min}}\right] = 0. \tag{4.43}$$

To obtain the value of $\theta^*(t)$ we will use *Netwon-Raphson method* [125] employing an appropriate initial guess that manifests the quadratic convergence of the solution. We choose $\theta_0^*(t) = E(\boldsymbol{\theta}(t)) = \frac{\theta_{\max}+\theta_{\min}}{2}$ as an initial guess for finding $\theta^*(t)$ which follows the PDF $f_{\boldsymbol{\theta}}(\theta) \sim U[\theta_{\min}, \theta_{\max}]$. Then the solution method is an iterative approach as follows:

$$\theta_{i+1}^*(t) = \theta_i^*(t) - \frac{f(\theta_i^*(t))}{\beta\delta^2 a^2 \cdot \ln^2(10) \cdot 10^{-(a\cdot\delta(1-\theta_i^*(t))+b)}}. \tag{4.44}$$

Numerical Analysis: In Figs. 4.17 and 4.18, we vary the number of participating clients up to 50 with different values of δ. The response of the clients is set to follow a uniform distribution on [0.1, 0.9] for the ease of representation. In Fig. 4.17, for the model parameters (a,b) as (0.35, −1), we see θ_{th} increases with the increase in the number of participating clients for all values of δ. It is intuitive and goes along with our earlier analysis that for the small number of participating clients, the smaller θ_{th} captures the efficacy of our proposed framework. Because it is an iterative process, the evolution of θ_{th} over the rounds of communication will be reflected in the framework design. Subsequently, the larger upper bound δ exhibits the similar impact on setting θ_{th}, where smaller δ imposes strict local accuracy level to attain high-quality centralized model. Also due to the same reason, in Fig. 4.18, we see θ_{th} is increasing for the increase in the number of participating clients, however, with the lower value. It is because of the choice of parameters (a, b) as explained in section "Incentive Mechanism: A Two-Stage Stackelberg Game Approach". So the value of θ_{th} is lower in Fig. 4.18.

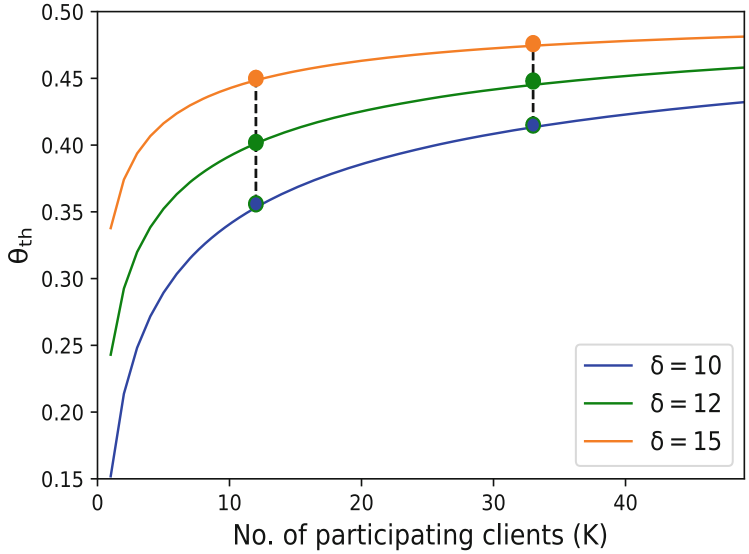

Fig. 4.17 Variation of local θ_{th} accuracy for different values of δ given the density function, $f_{\boldsymbol{\theta}}(\theta) \sim U[0.1, 0.9]$, $|\mathcal{K}| = [0, 50]$, for $a = 0.35$, $b = -1$

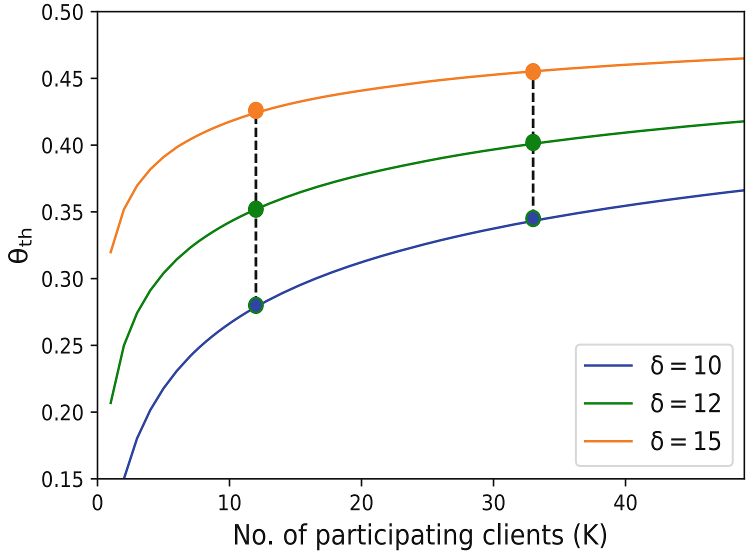

Fig. 4.18 Variation of local θ_{th} accuracy for different values of δ given the density function, $f_{\boldsymbol{\theta}}(\theta) \sim U[0.1, 0.9]$, $|\mathcal{K}| = [0, 50]$, for $a = 0.45$, $b = -1.05$

4.3 Auction Theory-Enabled Incentive Mechanism

In a typical wireless system, there are a variety of players, such as end-users, mobile network operators, and cloud server providers, among others. These players interact with each other to maximize their own benefits [126]. The benefits can be high data rate, load balancing, latency minimization, overall system utility maximization, energy efficiency, and profit maximization. However, enabling interaction among various players of wireless systems requires some effective business models. One can use auction theory to enable efficient interaction among the players of wireless systems. Auction theory enables people how to act in an auction market and systematically investigate the auction markets [127]. For instance, a mobile network operator can sell the spectrum resources to end-users for maximizing its profit. On the other hand, end-users want to increase their own benefits (e.g., data rate). To model this interaction between a mobile network operator and a set of users, one can use auction theory where users act as buyers (i.e, bidders) and mobile network operator as a seller. The description of three main players used in auction theory are as follows [128]:

- **Bidder:** In auction theory, a bidder is an entity that wants to buy commodities from the seller. An example of a bidder in a wireless system is the end-user.
- **Seller:** A seller denotes the owner of all commodities (e.g., radio resources) of a wireless system.
- **Auctioneer:** It refers to the intermediate agent that helps the bidder and seller in performing auctions. For instance, a base station in a wireless system can be considered as an auctioneer between the mobile network operator and end-users. Furthermore, a mobile network operator itself can also act as an auctioneer. Therefore, depending on the scenario one can choose an auctioneer.
- **Commodity:** Commodities of a wireless system represent the resources (e.g., spectrum, edge computing resource) that are traded between the sellers and bidders.

In this section, we provide an incentive mechanism design for federated learning using auction theory. In spite of the many benefits of federated learning, there are remaining two key challenges of having an efficient federated learning framework. The first challenge is the economic challenge. Data samples per mobile device are small to train a high-quality learning model so a large number of mobile users are needed to ensure cooperation. In addition, the mobile users who join the learning process are independent and uncontrollable. Here, mobile users may not be willing to participate in the learning due to the energy cost incurred by model training. In other words, the base station (BS), which generates the global model, has to stimulate the mobile users for participation. The second challenge is the technical challenge. On the one hand, we need users to collectively provide a large number of data samples to enable federated learning without sharing their private data. On the other hand, we need to protect the model from imperfect updates. The global loss minimization problem should enable (a) proper assessment of the quality of the local solution to improve personalization and fairness amongst the participating clients while training a global model, (b) effective decoupling of the local solvers, thereby balancing communication and computation in the distributed setting. Moreover, we need to consider wireless resource limitations (such as time, antenna number, and bandwidth)affecting the performance of federated learning. Besides, the limited energy of wireless devices is a crucial challenge for deploying federated learning. Indeed, it is necessary to optimize the energy efficiency for federated learning implementation because of these resource constraints.

To deal with the above challenges, we model the federated learning service between the BS and mobile users as an auction game in which the BS is a buyer and mobile users are sellers. In particular, the BS first initiates and announces a federated learning task. When each mobile user receives the federated learning task information, they decide the amount of resources required to participate in the model training. After that, each mobile user submits a bid, which includes the required amount of resource, local accuracy, and the corresponding energy cost, to the BS. Moreover, the BS plays the role of the auctioneer to decide the winners among mobile users as well as clear payment for the winning mobile users. In addition, the auction used in this work is a type of combinational auction [56, 129] since each mobile user can bid for combinations of resources. However, the proposed auction mechanism allows mobile users sharing the resources at the BS, which is different from the conventional combinatorial auction. The proposed mechanism directly determines the trading rules between the buyer (BS) and sellers (mobile users) and motivates the mobile users to participate in the model training. Compared with other incentive mechanism approaches (e.g., contract theory) in which the service market is a monopoly market, where mobile users can only decide whether or not to accept the contracts, the proposed auction enables mobile users to bids on any combinations of resources. Moreover, the proposed auction mechanism can simultaneously provide truthfulness and individual rationality. An auction mechanism is truthful if a bidder's utility does not increase when that bidder makes other bidding strategies, rather than the true value. Revealing the true value is a dominant strategy for each participating user regardless of what strategies other users use[130]. An absent-truthfulness auction mechanism could leave the door to possible market manipulation and produce inferior results [131]. Additionally, if the value of any bidder is non-negative, an auction process will ensure individual rationality. The contributions of this section are summarized as follows:

- We propose an auction framework for the wireless federated learning services market. *Then, we present the bidding cost in every user's bid submitted to the BS.* From the mobile users' perspective, each mobile user makes optimal decisions on the amount of resources and local accuracy to minimize weighted sum of completion time and energy costs while the delay requirement for federated learning is satisfied. To solve the cost decision problem, a low-complexity iterative algorithm is proposed.
- From the perspective of the BS, we formulate the winner selection problem in the auction game as the social welfare maximization problem, which is an NP-hard problem considering the limitation of the wireless resource. We propose a primal-dual greedy algorithm to deal with the NP-hard problem of selecting the winning users and critical value-based payment. We also proved that the proposed auction mechanism is truthful, individually rational, and computationally efficient.

- Finally, we carry out the numerical study to show that a proposed auction mechanism can guarantee the approximation factor of the integrality to the maximal welfare that is derived by the optimal solution and outperforms compared with baseline.

4.3.1 System Model

Preliminary of Federated Learning

Consider a cellular network in which one BS and a set $\mathcal{N}$ of N users cooperatively perform a federated learning algorithm for model learning, as shown in Fig. 4.19. A summary of all notations used is given in Table 4.4. Each user n has $\boldsymbol{s}_n$ local data samples. Each data set $\boldsymbol{s}_n = \{a_{nk}, b_{nk,1 \le k \le s_n}\}$ where a_{nk} is an input and b_{nk}

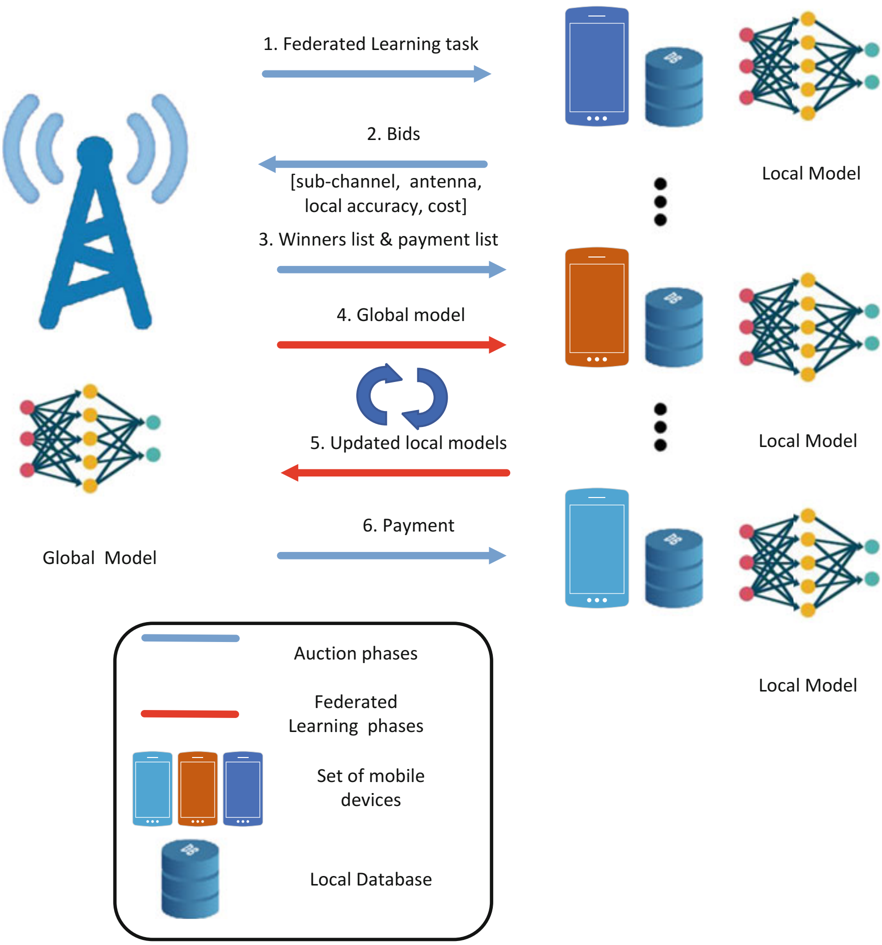

Fig. 4.19 System model

Table 4.4 Table of key notations

Description	Notation
Global loss function	$L(\omega)$
Computing energy consumption of user n	E_n^{comp}
Computing time of one local iteration of user n	T_n^{comp}
Transmission time of user n	T_n^{com}
Transmission energy of user n	E_n^{com}
Total transmission time of user n	T_n^{tol}
Total energy consumption of user n	E_n^{tol}
Local accuracy of user n	ε_n
Computing resource of user n	f_n
Transmission power of user n	p_n
Antenna number desired by user n	A_n
The real cost of the ith bid of user n	V_{ni}
The claimed cost of the ith bid of user n	v_{ni}
The satisfaction level of the ith bid of user n	χ_{ni}
The winner indicator of the ith bid of user n	x_{ni}

is its corresponding output. The federated learning model trained by the dataset of each user is called the local federated learning model, while the federated learning model at the BS aggregates the local model from all users as the global federated learning model. We define a vector ω as the model parameter. We also introduce the loss function $l_n(\omega, a_{nk}, b_{nk})$ that captures the federated learning performance over input vector a_{nk} and output b_{nk}. The loss function may be different, depending on the different learning tasks. The total loss function of user n will be

$$L_n(\omega) = \frac{1}{s_n} \sum_{k=1}^{s_n} l_n(\omega, a_{nk}, b_{nk}). \tag{4.45}$$

Then, the learning model is the minimizer of the following global loss function minimization problem

$$\min_{\omega} \quad L(\omega) = \frac{1}{S} \sum_{n=1}^{N} \sum_{k=1}^{s_n} l_n(\omega, a_{nk}, b_{nk}), \tag{4.46}$$

where $S = \sum_{n=1}^{N} s_n$ is the total data samples of all users.

To solve the problem in (4.46), we adopt the federated learning algorithm of [72]. The algorithm uses an iterative approach that requires a number of global iterations (i.e., communication rounds) to achieve a global accuracy level. In each global iteration, there are interactions between the users and BS. Specifically, at a given global iteration t, users receive the global parameter ω^t, users computes $\nabla L_n(\omega^t), \forall n$ and send it to the BS. The BS computes [70]

$$\nabla L(\omega^t) = \frac{1}{N} \sum_{n=1}^{N} \nabla L_n(\omega^t), \tag{4.47}$$

and then broadcasts the value of $\nabla L(\omega^t)$ to all participating users. Each participating user n will use local training data s_n to solve the local federated learning problem is defined as

$$\min_{\phi_n} \quad \mathcal{G}_n(\omega^t, \phi_n) \\ = L_n(\omega^t + \phi_n) - \left(\nabla L_n(\omega^t) - \varpi \nabla L(\omega^t)\right)^T \phi_n, \tag{4.48}$$

where ϕ_n represents the difference between global federated learning parameter and local federated learning parameter for user n. Each participating user n uses the gradient method to solve (4.48) with local accuracy ε_n that characterizes the quality of the local solution, and produces the output ϕ_n that satisfies

$$\mathcal{G}_n(\omega^t, \phi_n) - \mathcal{G}_n(\omega^t, \phi_n^*) < \varepsilon_n(\mathcal{G}_n(\omega^t, 0) - \mathcal{G}_n(\omega^t, \phi_n^*)). \tag{4.49}$$

Solving (4.48) also takes multiple local iterations to achieve a particular local accuracy. Then each user n sends the local parameter ϕ_n to the BS. Next, the BS aggregates the local parameters from the users and computes

$$\omega^{t+1} = \omega^t + \frac{1}{N}\sum_{n=1}^{N} \phi_n^t, \tag{4.50}$$

and broadcasts the value to all users, which is used for next iteration $t + 1$. This process is repeated until the global accuracy γ of (4.46) is obtained.

Assume that $L_n(\omega)$ is H-Lipschitz continuous and π-strongly convex, i.e.,

$$\pi \boldsymbol{I} \preceq \nabla^2 L_n(\omega) \preceq H\boldsymbol{I}, \forall n \in \mathcal{N},$$

the general lower bound on the number of global iterations is depends on local accuracy ε and the global accuracy γ as [70]:

$$I^g(\gamma, \varepsilon) = \frac{C_1 \log(1/\gamma)}{1 - \varepsilon}, \tag{4.51}$$

where the local accuracy measures the quality of the local solution as described in the preceding paragraphs.

In (4.51), we observe that a very high local accuracy (small ε) can significantly boost the global accuracy γ for a fixed number of global iterations I^g at the BS to solve the global problem. However, each user n has to spend excessive resources in terms of local iterations, I_n^l to attain a small value of ε_n. The lower bound on the number of local iterations needed to achieve local accuracy ε_n is derived as [70]

$$I_n^l(\varepsilon_n) = \vartheta_n \log\left(\frac{1}{\varepsilon_n}\right), \tag{4.52}$$

where $\vartheta_n > 0$ is a parameter choice of user n that depends on parameters of $L_n(\omega)$ [70]. In this section, we normalize $\vartheta_n = 1$. Therefore, to address this trade-off, the BS can set up an economic interaction environment to motivate the participating users to enhance local accuracy ε_n. Correspondingly, with the increased payment, the participating users are motivated to attain better local accuracy ε_n (i.e., smaller values), which as noted in (4.51) can improve the global accuracy γ for a fixed number of iterations I^g of the BS to solve the global problem. In this case, the corresponding performance bound in (4.51) for the heterogeneous responses ϵ_n can be updated to catch the statistical and system-level heterogeneity regarding the worst case of the participating users' responses as:

$$I^g(\gamma, \varepsilon_n) = \frac{\varpi \log(1/\gamma)}{1 - \max_n \varepsilon_n}, \forall n. \tag{4.53}$$

Computation and Communication Models for Federated Learning

The contributed computation resource that user n contributes for local model training is denoted as f_n. Then, c_n denotes the number of CPU cycles needed for the user n to perform one sample of data in local training. Thus, energy consumption of the user for one local iteration is presented as

$$E_n^{com}(f_n) = \zeta c_n s_n f_n^2, \tag{4.54}$$

where ζ is the effective capacitance parameter of computing chipset for user n. The computing time of a local iteration at the user n is denoted by

$$T_n^{comp} = \frac{c_n s_n}{f_n}. \tag{4.55}$$

It is noted that the uplink from the users to the BS is used to transmit the parameters of the local federated learning model while the downlink is used for transmitting the parameters of the global federated learning model. In this section, we just consider the uplink bandwidth allocation due to the relation of the uplink bandwidth and the cost that user experiences during learning a global model. We consider the uplink transmission of an OFDMA-based cellular system. A set of $\mathcal{B} = \{1, 2, ..., B\}$ subchannels each with bandwidth W. Moreover, the BS is equipped with A antennas and each user equipment has a single antenna (i.e., multi-user MIMO). We assume A to be large (e.g., several hundreds) to achieve massive MIMO effect which scales up traditional MIMO by orders of magnitude. Massive MIMO uses spatial-division multiplexing. In this section, we assume that the BS has perfect channel state information (CSI) and the channel gain is perfectly estimated, similar to [132, 133].

Then, the achievable uplink data rate of mobile user n is expressed as [133, 134]:

$$r_n = b_n W \log_2\left(1 + \frac{(A_n - 1)p_n h_n}{b_n W N_0}\right), \tag{4.56}$$

where p_n is the transmission power of user n, h_n is the channel gain of peer to peer link between user and the BS, N_0 is the background noise, A_n is the number of antennas the BS assigns to user n, and b_n is the number of sub-channels that user n uses to transmit the local model update to the BS.

We denote σ as the data size of a local model update and it is the same for all users. Therefore, the transmission time of a local model update is

$$T_n^{com}(p_n, A_n, b_n) = \frac{\sigma}{r_n}. \tag{4.57}$$

To transmit local model updates in a global iteration, the user n uses the amount of energy given as

$$E^{com}(p_n, f_n, A_n, b_n) = T^{com} p_n = \frac{\sigma p_n}{r_n}. \tag{4.58}$$

Hence, the total time of one global iteration for user n is denoted as

$$\begin{aligned} &T_n^{tol}(p_n, f_n, A_n, b_n, \varepsilon_n) \\ &= \log\left(\frac{1}{\varepsilon_n}\right) T_n^{comp}(f_n) + T_n^{com}(p_n, A_n, b_n). \end{aligned} \tag{4.59}$$

Therefore, the total energy consumption of a user n in one global iteration is denoted as follows

$$\begin{aligned} &E_n^{tol}(p_n, f_n, A_n, b_n, \varepsilon_n) \\ &= \log\left(\frac{1}{\varepsilon_n}\right) E_n^{comp}(f_n) + E_n^{com}(p_n, A_n, b_n). \end{aligned} \tag{4.60}$$

Auction Model

As described in Fig. 4.19, the BS first initializes the global network model. Then, the BS announces the auction rule and advertises the federated learning task to the mobile users. The mobile users then report their bids. Here, mobile user n submits a set of I_n of bids to the BS. A bid Δ_{ni} denotes the ith bid submitted by the mobile user n. Bid b_{ni} consists of the resource (sub-channel number b_{ni}, antenna number A_{ni}, local accuracy level ε_{ni}) and the claimed cost v_{ni} for the model training. Each mobile user n has its own discretion to determine its true cost V_{ni}, which will be presented in Sect. 4.3.1. Let x_{ni} be a binary variable indicating the bid Δ_{ni} wins or not. After receiving all the bids from mobile users, the BS decides winners and then allocates the resource to the winning mobile users. The winning mobile users join the federated learning and receive the payment after finishing the training model.

Remark In each bid, the bidder declares the requested resources, the local accuracy, and the corresponding cost. And the cost is calculated before submitting bids. Therefore, the cost corresponding to the requesting resources can be included in the bid during the bidding process.

Following we discuss one practical usage of our proposed auction scheme in federated learning. Let's consider a concrete example of a mobile phone keyboard such as Gboard (Google Keyboard). A large amount of local data will be generated when users interact with the keyboard app on their mobile devices. Suppose that Google server wants to train a next-word prediction model based on users' data. The server can announce the learning project to users through the app and encourage their participation. If a user wants to know more about this project, the app will display an interface to submit the bids and calculate the expected cost. If the user is interested in learning, he/she will download apps, calculate cost and submit the bids through the interface. Once the BS receive all bids in certain time, the BS will start the training process by broadcasting an initial global model to all the winning users. On behalf of the user, the app will download this global model and upload the model updates generated by the training on the user's local data. After finishing the model training project, the BS will give users rewards (e.g., money) based on the bid it wins.

Deciding Mobile Users's Bid

To transmit the local model update to the BS, mobile users need sub-channels and antenna resources. However, given the maximum tolerable time of federated learning, there is a correlation between resource and corresponding energy cost. In this section, we present the way mobile users decide bids. Specially, for bid Δ_{ni}, mobile user n calculates transmission power p_{ni}, computation resource f_{ni} and cost v_{ni} corresponding to a given sub-channel number b_{ni} and antenna number A_{ni}. However, for simplicity, the process to decide mobile users' bid is the same for every submitted bid. Thus, we remove the bid index i in this section. The energy cost of mobile user n is defined after user n solve the weighted sum of completion time and total energy consumption in the submitted bid, which is given as

$$\mathbf{P1}: \quad \min_{f_n, p_n, A_n, b_n, \varepsilon_n} \quad I_0^n \left(E_n^{tol}(p_n, f_n, \varepsilon_n) + \rho T_n^{tol}(p_n, f_n, \varepsilon_n) \right) \tag{4.61a}$$

$$s.t. \quad I_0^n T_n^{tol}(p_n, f_n, A_n, b_n, \varepsilon_n) \le T_{max}, \tag{4.61b}$$

$$f_n \in [f_n^{\min}, f_n^{\max}], \tag{4.61c}$$

$$p_n \in (0, p_n^{\max}], \tag{4.61d}$$

$$\varepsilon_n \leq (0, 1], \tag{4.61e}$$

$$A_n \in (0, A_n^{\max}], \tag{4.61f}$$

$$b_n \in (0, b_n^{\max}], \tag{4.61g}$$

where $f_n^{\max}$ and $p_n^{\max}$ are the maximum local computation capacity and maximum transmit power of mobile user n, respectively. $A_n^{\max}$ and $b_n^{\max}$ are the maximum antenna and maximum sub-channel that mobile user n can request in each bid, respectively. $A_n^{\max}$ and $b_n^{\max}$ are chosen by mobile user n. $I_0^n = \frac{C_1 \log(1/\gamma)}{1-\varepsilon_n}$ is the lower bound of the number global iterations corresponding to local accuracy ε_n. Note that the cost to the mobile user cannot be the same over iterations. ρ is the weight. However, to make the problem more tractable, we consider minimizing the approximated cost rather than the actual cost, similar to approach in [111]. Constraint (4.61b) indicates delay requirement of federated learning task.

According to **P1**, the maximum number of antennas and sub-channels are always energy efficient, i.e., the optimal antenna is $A_n = A_n^{max}$, $b_n = b_n^{max}$ and $\varepsilon_n^*, p_n^*, f_n^*$ are the optimal solution to:

$$\begin{aligned}
\mathbf{P2}: \quad \min_{f_n, p_n, \varepsilon_n} \quad & I_0^n \left(E_n^{tol}(p_n, f_n, \varepsilon_n) + \rho T_n^{tol}(p_n, f_n, \varepsilon_n) \right) \\
s.t. \quad & I_0^n T_n^{tol}(p_n, f_n, \varepsilon_n) \leq T_{max}, \\
& f_n \in [f_n^{\min}, f_n^{\max}], \\
& \varepsilon_n \in (0, 1], \\
& p_n \in (0, p_n^{\max}].
\end{aligned} \tag{4.62}$$

Because of the non-convexity of **P2**, it is challenging to obtain the global optimal solution. To overcome the challenge, an iterative algorithm with low complexity is proposed in the following subsection.

Iterative Algorithm

The proposed iterative algorithm basically involves two steps in each iteration. To obtain the optimal, we first solve (**P2**) with fixed ε_n, and then ε_n is updated based on the obtained f_n, p_n in the previous step. In the first step, we consider the first case

Algorithm 6 Optimal uplink power transmission

Calculate $\phi(p_n^{max})$
Calculate p_n^{min} so that $I_0^n T_n^{tol}(p_n^{min}) = T_{max}$
if $\phi(p_n^{max} < 0)$ **then**
    $p_n^* = p_n^{max}$
else
    $p_1 = \max(0, p_n^{min})$ and $p_2 = p_n^{max}$
    while $(p_2 - p_1 \leq \epsilon)$ **do**
        $p_u = (p_1 + p_1)/2$
        if $\phi(p_u) \leq 0$ **then**
            $p_1 = p_u$
        else
            $p_2 = p_u$
        end if
    end while
    $p_n^* = (p_1 + p_2)/2$
end if

when ε_n is fixed, and **P2** becomes

$$\begin{aligned} \mathbf{P3}: \quad & \min_{f_n, p_n, \varepsilon_n} \quad I_0^n \left(E_n^{tol}(p_n, f_n, \varepsilon_n) + \rho T_n^{tol}(p_n, f_n, \varepsilon_n) \right) \\ & s.t. \quad I_0^n T_n^{tol}(p_n, f_n, \varepsilon_n) \leq T_{max}, \\ & \qquad f_n \in [f_n^{\min}, f_n^{\max}], \\ & \qquad p_n \in (0, p_n^{\max}]. \end{aligned} \tag{4.63}$$

P3 can be decomposed into two sub-problems as follows.

Optimization of Uplink Transmission Power

Each mobile user assigns its transmission power by solving the following problem:

$$\begin{aligned} \mathbf{P3a}: \quad & \min_{p_n} \quad f(p_n) \\ & s.t. \quad I_0^n T_n^{tol}(p_n, f_n, \varepsilon_n) \leq T_{max}, \\ & \qquad p_n \in (0, p_n^{\max}], \\ & \qquad f_n, \varepsilon_n \quad \text{are given.} \end{aligned} \tag{4.64}$$

where $f(p_n) = \frac{\sigma(1+\rho)p_n}{b_n W \log_2(1+\frac{(A_n-1)p_n h_n}{b_n W N_0})}$.

Algorithm 7 Optimal local accuracy

1: Initialize $\varepsilon_n = \varepsilon_n^{(0)}$, set $j = 0$
2: **repeat**
3: Calculate $\varepsilon_n^* = \frac{\alpha_1}{(\ln 2)\xi^j}$
4: Update $\xi^{(j+1)} = \frac{\gamma_1 \log_2(1/\varepsilon_n)+\gamma_2}{1-\varepsilon_n}$
5: Set $j = j + 1$
6: **until** $|H(\xi^{(n+1)})|/|H(\xi^{(n)})| < \epsilon_2$

Note that $f(p_n)$ is quasiconvex in the domain [135]. A general approach to the quasiconvex optimization problem is the bisection method, which solves a convex feasibility problem each time [116]. However, solving convex feasibility problems by an interior cutting-plane method requires $O(\kappa^2/\alpha^2)$ iterations, where κ is the dimension of the problem [135]. On the other hand, we have

$$f'(p_n) = \frac{\sigma \log_2(1 + \theta_n p_n h_n) + \frac{\sigma(1+\rho)p_n\theta_n h_n}{\ln 2(1+\theta_n p_n h_n)}}{b_n W(\log(1 + \theta_n p_n h_n))^2}, \tag{4.65}$$

where $\theta_n = \frac{(A_n-1)}{WN_0}$. Then, we have

$$\phi(p_n) = \sigma \log_2(1 + \theta_n p_n h_n) + \frac{\sigma(1 + \rho)p_n\theta_n h_n}{\ln 2(1 + \theta_n p_n h_n)} \tag{4.66}$$

is a monotonically increasing transcendental function and negative at the starting point $p_n = 0$ [135]. Therefore, in order to obtain the optimal power allocation p_n as shown in Algorithm 6, we follow a low-complexity bisection method by calculating $\phi(p_n)$ rather than solving a convex feasibility problem each time.

Optimization of CPU Cycle Frequency and Number of Antennas

$$\begin{aligned} \textbf{P3b}: \quad & \min_{f_n} \quad I_0^n \log\left(\frac{1}{\varepsilon_n}\right) c_n s_n \left(\zeta f_n^2 + \rho/f_n\right) \\ & s.t. \quad I_0^n \left(\log\left(\frac{1}{\varepsilon_n}\right)\frac{c_n s_n}{f_n} + T_n^{com}\right) \le T_{max}, \\ & \qquad f_n \in [f_n^{\min}, f_n^{\max}], \\ & \qquad p_n, \varepsilon_n \quad \text{are given.} \end{aligned} \tag{4.67}$$

P3b is the convex problem, so we can solve it by any convex optimization tool.

Algorithm 8 Iterative algorithm

1: Initialize a feasible solution p_n, f_n, ε_n and set $j = 0$.
2: **repeat**
3: With $\varepsilon_n^{(j)}$ obtain the optimal $p_n^{(j+1)}$, $f_n^{(j+1)}$ of problem **P2** :
4: With $p_n^{(j+1)}$, $f_n^{(j+1)}$ obtain the optimal $\varepsilon_n^{(j+1)}$ of problem **P2** :
5: Set $j = j + 1$
6: **until** Objective value of **P2** converges

In the second step, **P2** can be simplified by using f_n and p_n calculated in the first step as:

$$\mathbf{P4}: \quad \min_{\varepsilon_n} \quad \frac{\gamma_1 \log_2(1/\varepsilon_n) + \gamma_2}{1 - \varepsilon_n} \tag{4.68a}$$

$$s.t. \quad T_n^{tol} \leq T_{max}, \tag{4.68b}$$

where $\gamma_1 = a\left(E_n^{comp} + T_n^{comp}\right)$ and $\gamma_2 = a\left(E_n^{com} + T_n^{com}\right)$. The constraint (4.68b) is equivalent to $T_n^{com} \leq \vartheta(\varepsilon_n)$, where $\vartheta(\varepsilon_n) = \frac{1-\varepsilon_n}{m} T_{max} + \frac{c_n s_n \log_2 \varepsilon_n}{f_n}$. We have $\vartheta(\varepsilon_n)'' < 0$, and therefore, $\vartheta(\varepsilon_n)$ is a concave function. Thus, constraint (4.68b) can be equivalent transformed to $\varepsilon_n^{min} \leq \varepsilon_n \leq \varepsilon_n^{max}$, where $\vartheta(\varepsilon_n^{min}) = \vartheta(\varepsilon_n^{max}) = T_n^{com}$. Therefore, ε_n is the optimal solution to

$$\begin{aligned} \mathbf{P5}: \quad & \min_{\varepsilon_n} \quad \frac{\gamma_1 \log_2(1/\varepsilon_n) + \gamma_2}{1 - \varepsilon_n} \\ & s.t. \quad \varepsilon_n^{min} \leq \varepsilon_n \leq \varepsilon_n^{max}. \end{aligned} \tag{4.69}$$

Obviously, the objective function of **P5** has a fractional in nature, which is generally difficult to solve. According to [70, 136], solving **P5** is equivalent to finding the root of the nonlinear function $H(\xi)$ defined as follows

$$H(\xi) = \min_{\varepsilon_n^{min} \leq \varepsilon_n \leq \varepsilon_n^{max}} \gamma_1 \log_2(1/\varepsilon_n) + \gamma_2 - \xi(1 - \varepsilon_n) \tag{4.70}$$

Function $H(\xi)$ with fixed ξ is convex. Therefore, the optimal solution ε_n can be obtained by setting the first-order derivative of $H(\xi)$ to zero, which leads to the optimal solution is $\varepsilon_n^* = \frac{\gamma_1}{(\ln 2\xi)}$. Thus, similar to [70], problem **P5** can be solved by using the Dinkelbach method in [136] (shown as Algorithm 7).

Convergence Analysis

The algorithm that solves problems **P2** is given in Algorithm 4, which iteratively solves problems **P3** and **P4**. Since the optimal solution of problem **P3** and **P4** is obtained in each step, the objective value of problem **P2** is non-increasing in each step. Moreover, the objective value of problem **P2** is lower bounded by zero. Thus, Algorithm 4 always converges to a local optimal solution.

Complexity Analysis

Because of the non-convexity of **P2**, it is challenging to obtain the global optimal solution. To overcome the challenge, an iterative algorithm with low-complexity is proposed in the following subsection. In particular, to solve the general energy-efficient resource allocation problem **P2** using Algorithm 3, the major complexity in each step lies in solving problems **P3** and **P4**. To solve problem **P3**, the complexity is $\mathcal{O}(L_e \log_2(1/\epsilon_1))$, where ϵ_1 is the accuracy of solving **P3** with the bisection method and L_e is the number of iterations for optimizing f_n and p_n. To solve problem **P4**, the complexity is $\mathcal{O}(\log_2(1/\epsilon_2))$ with accuracy ϵ_2 by using the Dinkelbach method [136]. As a result, the total complexity of the proposed Algorithm 4 is $H_e S$, where H_e is the number of iterations for problems **P3** and **P4** and S is equal to $\mathcal{O}(L_e \log_2(1/\epsilon_1)) + \mathcal{O}(\log_2(1/\epsilon_2))$.

After deciding the bids, the mobile users submit bids to the BS. The following section describes the auction mechanism between the BS and mobile users for selecting winners, allocating bandwidth and deciding on payment.

4.3.2 Auction Mechanism Between BS and Mobile Users

After receiving all bids submitted by mobile users, the BS decides a set of winners by solving the problem (**P6**), aiming to maximize social welfare. The BS's aim is to achieve social welfare because the BS needs to incentive mobile users to participate in learning. Here, the BS's freedom in designing the incentive mechanism is the payment determination, which can force participant mobile users to be truthful. Moreover, if the BS wants to select winners to maximize its utility, the BS needs to know the distribution of mobile users' private information in advance[129], which is assumed to be unavailable in our work. In case the prior distribution of mobile users' private information is not available, worst-case analysis can be applied, but that method could lead to overly pessimistic results [129].

Problem Formulation

In bid Δ_{ni} that mobile user n submits to the BS includes the number of subchannels b_{ni}, the number of antennas A_{ni}, local accuracy ϵ_{ni}, and claimed cost v_{ni}. The utility of one bid is the difference between the payment g_{ni} and the real cost V_{ni}.

$$U_{ni} = \begin{cases} g_{ni} - V_{ni}, & \text{if bid } \Delta_{ni} \text{ wins,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.71}$$

The payment that the BS pays for winning bids is $\sum_{n,i} g_{ni}$. As we described in Sect. 4.3.1, high local accuracy will significantly improve the global accuracy for a fixed number of global iterations. The utility of the BS is the difference between the BS's satisfaction level and the payment for mobile users. The satisfaction level of the BS to bid Δ_{ni} is measured based on the local accuracy that mobile user n can provide in the ith bid and is defined as follows

$$\chi_{ni} = \frac{\tau}{\varepsilon_{ni}}. \tag{4.72}$$

Thus, the total utilities of the system or the social welfare is

$$\sum_{n,i} (\chi_{ni} - v_{ni}) x_{ni}. \tag{4.73}$$

If mobile users truthfully submit their cost, $V_{ni} = v_{ni}$, we have the social welfare maximization problem defined as follows:

$$\mathbf{P6}: \quad \max_{x} \quad \sum_{n,i} (\chi_{ni} - v_{ni}) x_{ni} \tag{4.74a}$$

$$s.t. \quad \sum_{n} x_{ni} b_{ni} \le B_{max}, \tag{4.74b}$$

$$\sum_{n} x_{ni} A_{ni} \le A_{max}, \tag{4.74c}$$

$$\sum_{i} x_{ni} \le 1, \forall n, \tag{4.74d}$$

$$x_{ni} = \{0, 1\}, \tag{4.74e}$$

where (4.74b) and (4.74c) indicate the bandwidth resource (i.e., sub-channels) and the antennas limitation constraints of the BS, respectively. Then, (4.74d) shows that a mobile user can win at most one bid and (4.74e) is the binary constraint that presents whether bid Δ_{ni} wins or not.

Problem **P6** is a minimization knapsack problem, which is known to be NP-hard. This implies that no algorithm is able to find out the optimal solution of **P6** in polynomial time. It is also known that a mechanism with Vickrey-Clarke-Groves (VCG) payment rule is truthful only when the resource allocation is optimal. Hence, using VCG payment directly is unsuitable due to the problem **P6** is computationally intractable. To deal with the NP-hard problem, we proposed the primal-dual based greedy algorithm. The following economic properties are desired.

Truthfulness An auction mechanism is truthful if and only if for every bidder n can get the highest utility when it reports true value.

Individual Rational If each mobile user reports its true information (i.e., cost and local accuracy), the utility for each bid is nonnegative, i.e., $U_{ni} \geq 0$.

Computation Efficiency The problem can be solved in polynomial time.

Among these three properties, truthfulness is the most challenging one to achieve. In order to design a truthful auction mechanism, we introduce the following definitions.

Definition 1 (Monotonicity) If mobile user n wins with the bid $\Delta_{ni} = \{v_{ni}, 1/\varepsilon_{ni}, b_{ni}, A_{ni}\}$, then mobile user n can win the bid with $\Delta_{nj} = \{v_{nj}, 1/\varepsilon_{nj}, b_{nj}, A_{nj}\} \succ \Delta_{ni} = \{v_{ni}, 1/\varepsilon_{ni}, b_{ni}, A_{ni}\}$.

The notation $\succ$ denotes the preference over bid pairs. Specifically, $\Delta_{nj} = \{v_{nj}, 1/\varepsilon_{nj}, b_{nj}, A_{nj}\} \succ \Delta_{ni} = \{v_{ni}, 1/\varepsilon_{ni}, b_{ni}, A_{ni}\}$ if $\varepsilon_{nj} < \varepsilon_{ni}$ for $v_{nj} = v_{ni}, b_{nj} = b_{ni}, A_{nj} = A_{ni}$ or $v_{nj} < v_{ni}, b_{nj} < b_{ni}, A_{nj} < A_{ni}$ for $\varepsilon_{nj} = \varepsilon_{ni}$. The monotonicity implies that the chance to obtain a required bundle of resources can only be enhanced by either increasing the local accuracy or decreasing the amount of resources required or decreasing the cost.

Definition 2 (Critical Value) For a given monotone allocation scheme, there exists a critical value c_{ni} of each bid Δ_{ni} such that $\forall n, i (\chi_{ni} - v_{ni}) \geq c_{ni}$ will be a winning bid, while $\forall n, i (\chi_{ni} - v_{ni}) < c_{ni}$ is a losing bid.

In our proposed mechanism, the difference between the satisfaction based on local accuracy and cost of one bid can be considered as the value of that bid. Therefore, the critical value can be seen as the minimum value that one bidder has to bid to obtain the requested bundle of resources. With the concepts of monotonicity and critical value, we have the following lemma.

Lemma 4.1 *An auction mechanism is truthful if the allocation scheme is monotone and each winning mobile user is paid the amount that equals to the difference between the satisfaction based on the local accuracy and the critical value.*

Proof Similar Lemma 1 and Theorem 1 in [130].

In the next subsection, we propose a primal-dual greedy approximation algorithm for solving problem **P6**. The algorithm iteratively updates both primal and dual variables and the approximation analysis is based on duality property. As the result, we firstly relax $1 \geq x_{ni} \geq 0$ of **P6** to have the linear programming relaxation (LPR) of **P6**. Then, we introduce the dual variable vectors $\boldsymbol{y}$, $\boldsymbol{z}$ and $\boldsymbol{t}$ corresponding to constraints (4.74b), (4.74c) and (4.74d) and we have the dual of problem LPR of **P6** can be written as

$$\mathbf{P7}: \max_{\mathbf{y},\mathbf{z},\mathbf{t}} \sum_{n \in \mathcal{N}} y_n + zB_{max} + tA_{max} \tag{4.75a}$$

$$s.t. \quad y_n + zA_{ni} + tB_{ni} \geq q_{ni}, \forall n, i, \tag{4.75b}$$

$$y_n \geq 0, \forall n, \tag{4.75c}$$

$$z, t \geq 0. \tag{4.75d}$$

In Sect. 4.3.2, we devise an greedy approximation algorithm and Sect. 4.3.2, a theoretical bound is achieved for the approximation ratio of the proposed algorithm.

Approximation Algorithm Design

In this section, we use a greedy algorithm to solve problem **P6** The main idea of the greedy algorithm is to allocate the resource to bidders with the larger normalized value. The winner selection process is described in the Algorithm 4. The process consists 3 steps:

Step 1: Based on the bid's value and the weighted sum of requested resources, each bid Δ_{ni} calculates the normalized value. The bid's value is defined as the difference between the satisfaction level of the BS and the cost declared in this bid, $q_{ni} = \chi_{ni} - v_{ni}$. The weighted sum of different types of resources declared in this bid is defined as $s_n^i = \eta_b B_n + \eta_a A_n$, where η_b, η_a are the weights. The normalized value of the bid is defined as the ration between the value of this bid and the weighted sum of requested resources, and is denoted as

$$\bar{q}_{ni} = \frac{q_{ni}}{s_{ni}}.$$

Step 2: The bid with maximum $\bar{q}_{ni}$ wins the bidding.
Step 3: Delete user n from the list of bidders. Then go back to Step 2 until either one of the following termination conditions is satisfied:

(i) The BS has not enough resource to satisfy the demand;
(ii) All the mobile users win one bid.

Approximation Ratio Analysis

In this subsection, we analyze approximation ratio of Algorithm 10. Our approach is to use the duality property to derive a bound for approximation algorithm. We denote the optimal solution and the optimal value of LPR of **P6** as x_{ni}^* and OP_f. Furthermore, let OP and φ as the optimal value of **P6** and the primal value of **P6** obtained by Algorithm 10. Our analysis consists of two steps. First, Theorem 4.1 shows that Algorithm 10 generates a feasible solution to **P7**, and Proposition 1 provides approximation factor.

Theorem 4.1 *Algorithm 10 provides a feasible solution to* ***P7****.*

Proof We discuss the following three cases:

- Case 1: mobile user μ wins, i.e., $\mu \in \mathcal{U}$ and $b_{\mu i_\mu} = \max_{i' \in \mathcal{I}_\mu}\{q_{\mu i'}\}$. Then we have $y_\mu = q_{\mu i_\mu} \geq q_{\mu i'}, \forall i' \in \mathcal{I}_\mu$. Thus, constraint (4.75b) is satisfied for all mobile users in $\mathcal{U}$.

Algorithm 9 The Greedy approximation algorithm

Input: $(B, A, \chi, v, B_{max}, A_{max})$
Output: solution $\mathbf{x}$
$\mathcal{U} = \varnothing$, $\mathbf{x} = \mathbf{0}$
$\forall n : y_n = 0$, $\psi = 0$;
$\varphi = 0$, $B = 0$, $A = 0$;
$s_n^i = \eta_b B_{ni} + \eta_a A_{ni}$;
$q_{kj} = \chi_{ni} - v_{ni}$;
for $n \in \mathcal{N}$ **do**
    $i_n = \arg\max_i \{q_{ni}\}$;
end for
$\kappa = \max \frac{s_{ni}}{s_{ni'}}$;
while $\mathcal{N} \neq \varnothing$ **do**
    $\mu = \arg\max_{n \in \mathcal{N}} \frac{q_{ni}}{s_{ni_n}}$;
    if $B + b_{\mu i_\mu} <= B_{max}$ and $A + a_{\mu i_\mu} <= A_{max}$ **then**
        $x_{\mu i_\mu} = 1$; $y_\mu = q_{\mu i_\mu}$;
        $\varphi = \varphi + q_{\mu i_\mu}$;
        $\psi = \frac{\sum_{n \in \mathcal{U}} q_{ni_n}}{\sum_{n \in \mathcal{U}} s_{ni_n}}$;
        $\mathcal{U} = \mathcal{U} \cup \{\mu\}$ and $\mathcal{N} = \mathcal{N} \setminus \{\mu\}$
    else
        break;
    end if
end while
$\bar{\psi} = \kappa \psi$;
$z = \eta_b \bar{\psi}$, $t = \eta_a \bar{\psi}$

- Case 2: mobile user μ loses the auction, i.e., $\mu \in \mathcal{N} \setminus \mathcal{U}$. According to the while loop, it is evident that

$$\frac{q_{ni_n}}{s_{ni_n}} > \frac{q_{\mu i_\mu}}{s_{\mu i_\mu}}, \forall n \in \mathcal{U}.$$

Therefore, $\psi > \frac{q_{\mu i_\mu}}{s_{\mu i_\mu}}$. Thus,

$$\bar{\psi} \geq \kappa \frac{q_{\mu i_\mu}}{s_{\mu i_\mu}} \geq \frac{q_{\mu i_\mu}}{s_{\mu i_\mu}}.$$

In addition, we have

$$q_{\mu i_\mu} \geq q_{\mu i'} \quad \text{and} \quad \kappa > \frac{s_{\mu i_\mu}}{s_{\mu i'}}, \forall i' \neq i_\mu.$$

Therefore,

$$\bar{\psi} \geq \frac{q_{\mu i'}}{n_{\mu i'}}, \forall i' \neq i_\mu.$$

Therefore, we have

$$\eta_b \bar{\psi} B_{in} + \eta_a \bar{\psi} A_{in} \geq q_{in}, \forall i' \neq i_\mu.$$

or

$$zC_{in} + tA_{in} \geq q_{in}, \forall i' \neq i_\mu.$$

Therefore, constraint (4.75b) is also satisfied for all mobile users in $\mathcal{N} \setminus \mathcal{U}$.

Proposition 1 *The upper bound of integrality gap* α *between* ***P6*** *and its relaxation and the approximation ratio of Algorithm 10 are* $1 + \frac{\kappa \Upsilon}{\Upsilon - S}$, *where* $\Upsilon = \eta_b B_{max} + \eta_a A_{max}$, $S = \max_{n,i} s_{ni}$.

Proof Let OP and OP_f be the optimal solution for **P6** and LPR of **P6**. We can obtain the following:

$$\begin{aligned}
OP \leq OP_f &\leq \sum_{n=1}^{N} y_n + zB_{max} + tA_{max} \\
&\leq \sum_{n=1}^{N} y_n + \bar{\psi}(\eta_b B_{max} + \eta_a A_{max}) \\
&\leq \sum_{n \in \mathcal{N}} q_{ni_n} + \bar{\psi}(\eta_b B_{max} + \eta_a A_{max}) \\
&\leq \left(\sum_{n \in \mathcal{N}} q_{ni_n} \right) \left(1 + \frac{(\eta_b B_{max} + \eta_a A_{max})\kappa}{\eta_b B_{max} + \eta_a A_{max} - S} \right) \\
&\leq \varphi \left(1 + \frac{\Upsilon \kappa}{\Upsilon - S} \right),
\end{aligned}$$

Therefore, the integrality α is given as

$$\begin{aligned}
& OP_f / OP \\
\leq \quad & OP_f / \varphi \\
\leq \quad & \left(1 + \frac{\kappa \Upsilon}{\Upsilon - S} \right).
\end{aligned}$$

The approximation ratio is

$$OP/\varphi \leq OP_f/\varphi \leq \left(1 + \frac{\kappa \Upsilon}{\Upsilon - S} \right).$$

Payment

Then we will find the critical value which is the minimum value a bidder has to bid to win the requested bundle of resources. In this section, we consider the bid combinations submitted by mobile user n as the combinations of bids submitted by virtual bidders, in which each virtual bidder can submit one bid. Therefore, the number of virtual bidders corresponding to mobile user n is equal to the number of bids I_n that mobile user n submits. Denote by m the losing mobile user with the highest normalized value if mobile user n is not participating in the auction. Accordingly, the minimum value mobile user n needs to place is $\frac{q_{mi_m}}{s_{mi_m}} s_{ni_n}$, where i_m and i_n are the indexes of highest normalized value bids of mobile user m and n, respectively. Thus, the payment of winning mobile user n in the pricing scheme is $g_{ni_n} = \chi_{ni_n} - \frac{q_{mi_m}}{s_{mi_m}} s_{ni_n}$.

Properties

Now, we show that the winner determination algorithm is monotone and the payment determined for a winner mobile user is the difference between the local accuracy-based satisfaction and the critical value of its bid. From line 13 of the Algorithm 10, it is clear that a mobile user can increase its chance of winning by increasing its bid. Also, a mobile user can increase its chance to win by decreasing the weighted sum of the resources. Therefore, the winner determination algorithm is monotone with respect to mobile user's bids. Moreover, the value of a winning bidder is equals to the minimum value it has to bid to win its bundle, i.e., its critical value. This is done by finding the losing bidder m who would win if bidder n would not participate in the auction. Thus, the proposed mechanism has a monotone allocation algorithm and payment for the winning bidder equals the difference between the local accuracy-based satisfaction and the critical value of its bid. We conclude that the proposed mechanism is a truthful mechanism according to Lemma 4.1.

Next, we prove that the proposed auction mechanism is individually rational. For any mobile user n bidding its true value, we consider two possible cases:

- If mobile user n is a winner with its bid ith, its payment is

$$\begin{aligned} U_{ni} &= g_{ni} - v_{ni} \\ &= (\chi_{ni} - \frac{q_{mi_m}}{s_{mi_m}} s_{ni} - v_{ni} \\ &= \left(\frac{\chi_{ni} - v_{ni}}{s_{ni}} - \frac{q_{mi_m}}{s_{mi_m}} \right) s_{ni} \\ &= \left(\frac{q_{ni}}{s_{ni}} - \frac{q_{mi_m}}{s_{mi_m}} \right) s_{ni} \geq 0 \end{aligned}$$

where m the losing bidder with the highest normalized valuation if n does not participate in the auction and the last inequality follows from Algorithm 10.

- If mobile user n is not a winner. Its utility is 0.

Therefore, the proposed auction mechanism is individually rational.

Finally, we show that the proposed auction mechanism is computationally efficient. We can see that in Algorithm 10, the while-loop (lines 12–22) takes at most N times, linear to input. Calculating the payment takes at most $N(N-1)$ times. Therefore, the proposed auction mechanism is computationally efficient. Therefore, the time complexity of Algorithm 10 is $\mathcal{O}(N^2)$.

4.3.3 Simulations

In this section, we provide some simulation results to evaluate the proposed mechanism. The parameters for the simulation are set the following. The required CPU cycles for performing a data sample c_n is uniformly distributed between $[10, 50]$ cycles/bit [70]. The size of data samples of each mobile user is $s_n = 80 \times 10^6$. The effective switched capacitance in local computation is $\xi = 10^{-28}$ [70]. We assume that the noise power spectral density level N_0 is -174dBm/Hz, the sub-channel bandwidth is $W = 15$ kHz and the channel gain is uniformly distributed between $[-90, -95]$ dB [132]. In addition, the maximum and minimum transmit power of each mobile user is uniformly distributed between $[6, 10]$ mW and between $[0, 2]$ mW, respectively. The maximum and minimum computation capacity is uniformly distributed between $[3, 5]$ GHz and between $[10, 20]$ Hz, respectively. We also assume that the total number of sub-channels and antennas of the BS are 100 and 100, respectively.

Firstly, we use the iterative Algorithm 8 to perform the characteristic of evaluating bids when $\rho = 1$. The maximum number of sub-channels B_n^{max} and antennas A_n^{max} for mobile user n to request in each bid vary from 10 to 50. Figure 4.20 shows the accuracy level that mobile user n requires to provide increases when the maximum number of sub-channels B_n^{max} and antennas A_n^{max} increase. In particular, when the sub-channels and antennas are both 50, the local accuracy 0.92 while when the sub-channels and antennas are both 10, the local accuracy 0.81. This is because the transmission time and transmission cost decrease when wireless resources increase. It requires less global round to satisfy the learning task performance. Therefore, the local accuracy increases or . As shown in Fig. 4.21, the energy cost decreases when the number of sub-channels and antennas increases. This is because mobile user n can keep low contributing CPU cycle frequency and transmission rate while guaranteeing the delay constraint.

Figures 4.22 and 4.23 present the cost of one bid of the mobile user and local accuracy, respectively, when the weight ρ varies from 1 to 9. As shown in Figs. 4.22 and 4.23, when ρ increases, the local accuracy decreases and the energy cost increases. This is because when ρ increases, the objective focuses more on

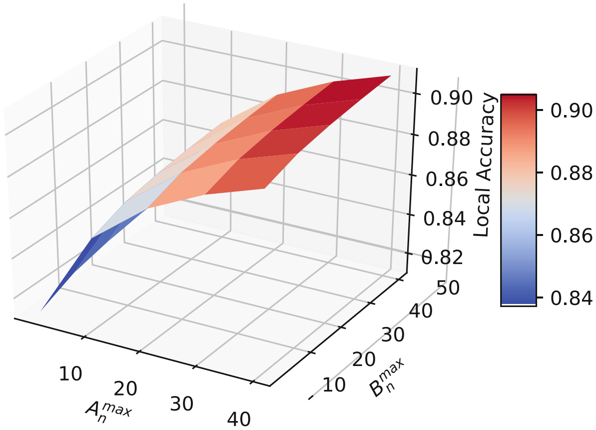

Fig. 4.20 Changes local accuracy when the maximum number sub-channels and antennas in one bid vary

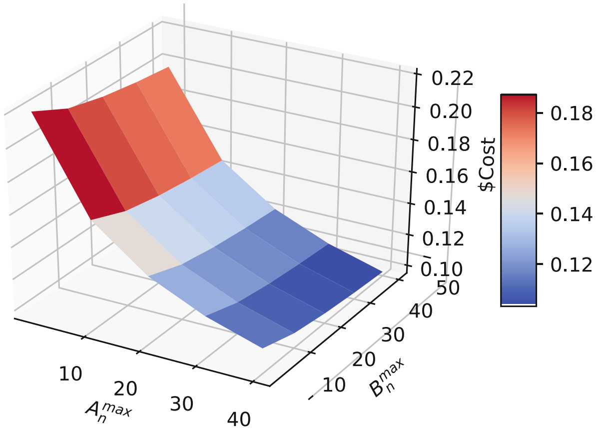

Fig. 4.21 Changes in energy cost when the maximum number sub-channels and antennas in one bid vary

minimizing the time completion of one global round. It requires more computation resources as well as better quality of data (low local accuracy).

In the following, we evaluate the performance of the proposed auction algorithm. To compare with the proposed algorithm, we use four baselines:

- Optimal Solution: **P6** is solved optimally.
- Fixed Price Scheme [137]: In this scheme, price vector $f = \{f_b, f_a\}$ is the price mobile users need to pay for the resource. In this scheme, the mobile users are served in a first-come, first-served basic until the resources are exhausted. The mobile user can get the resource when the valuation of mobile user's bid is at least $F_{ni} = B_{ni} f_b + A_{ni} f_a$ which is the sum of the fixed price of each resource in their bid. We consider three kinds of price vector: linear price ($f_i = f_o \times \eta_i, i = a, b$), sub-linear price vector ($f_i = f_o \times \eta_i^{0.85}$, $i = a, b$) [137], and a super-linear price vector ($f_i = f_o \times \eta_i^{1.15}, i = a, b$) [137]. Here, we call f_o as the basic price. Unless specified otherwise, we choose $f_o = 0.01$.

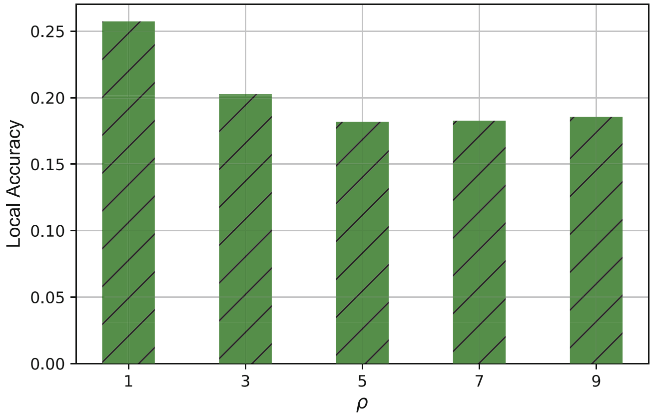

Fig. 4.22 Local accuracy v.s. ρ

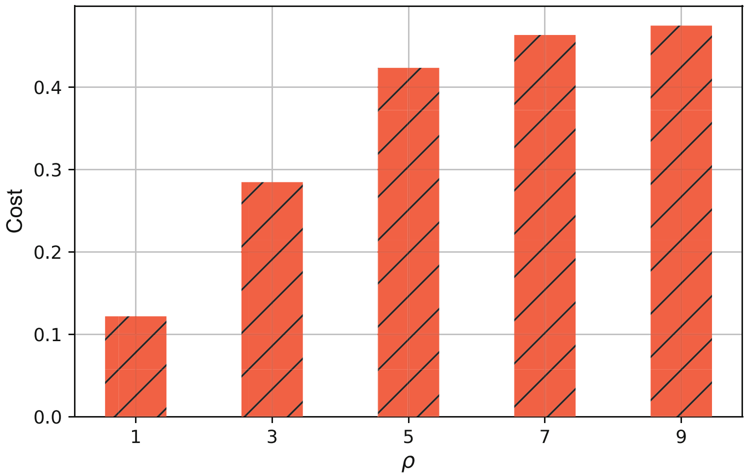

Fig. 4.23 Energy cost v.s. ρ

- Reward-based greedy auction [138].
- Maximum utility of the BS.

Figure 4.24 reports the performance of the optimal solution, the lower bound, and the proposed greedy scheme. The lower bound is determined by the fractional optimal solution divided by gap when the number of mobile users varies from 20 to 100 with a step size of 20. We note that with the number of mobile users increasing, all schemes produce higher social welfare. This is because there are more chances to choose winning bids with a higher value. Although the social welfare obtained through the proposed greedy scheme is lower than through optimal solution and much higher than the lower bound.

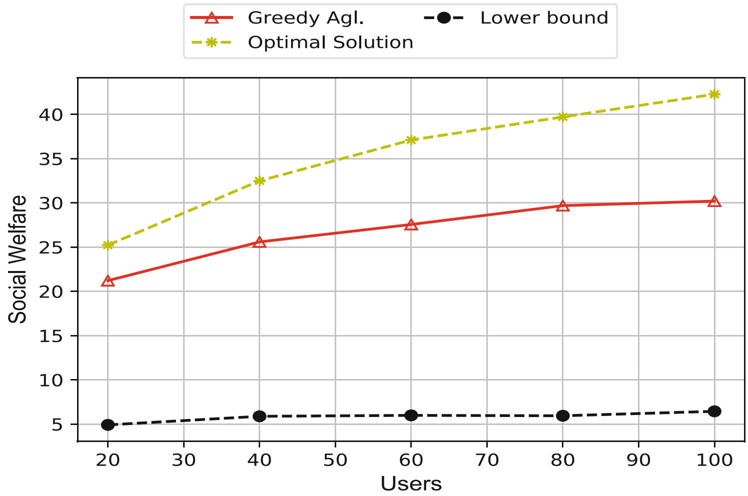

Fig. 4.24 Social welfare vs. users

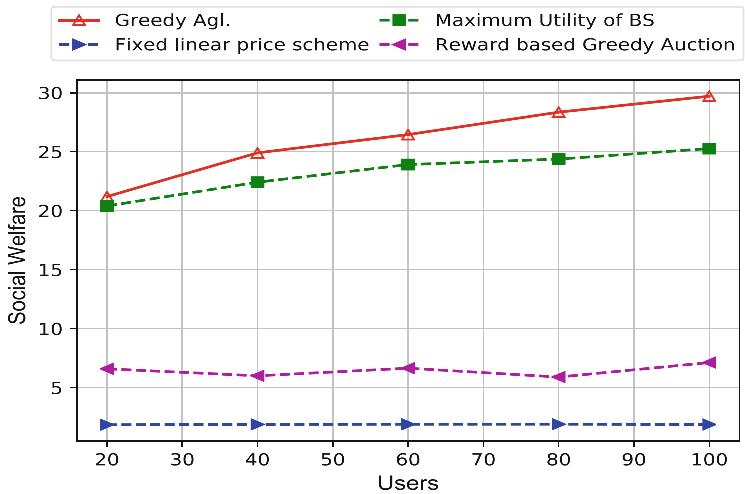

Fig. 4.25 Social welfare vs. users

Figure 4.25 shows the social cost achieved by the proposed greedy scheme, the maximum utility of the BS, reward-based greedy auction, and fixed linear scheme when the number of mobile users varies 20–100 with the step size of 20. We can see that the proposed greedy scheme can provide much higher social welfare than the baseline. When the number of users is 100, the social welfare obtained by a proposed greedy algorithm is approximately 16% higher than the one obtained by the maximum utility of BS. The result is that our proposed algorithm focuses on maximizing social welfare. In addition, when the number of users is 100, the social welfare obtained by the proposed greedy algorithm is approximately 4 times and 15 times higher than the one obtained by the reward-based Greedy Auction and fixed linear price scheme, which ignores the wireless resource limitation when deciding the winning bids.

Since the fixed price scheme heavily depends on the prices of resources, the next experiment helps us to decide whether the fixed-price vector or the performance of the proposed mechanisms is better when we change the basic price f_o between $[0.01, 0.31]$ with the step is 0.03. Figures 4.26, 4.27, and 4.28 show that the social

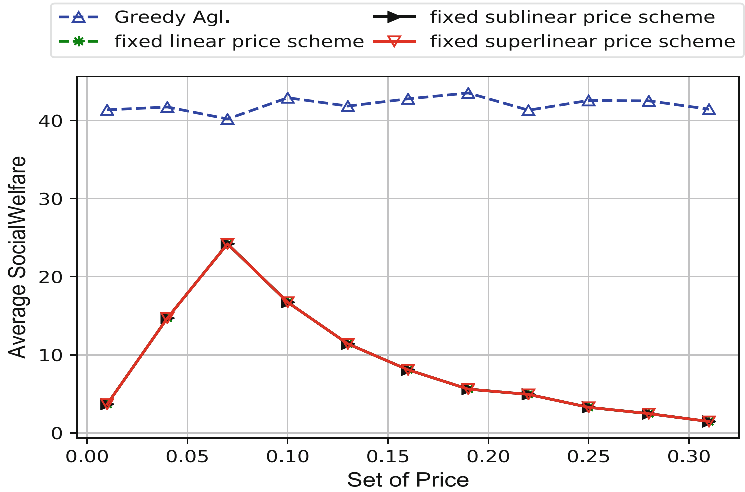

Fig. 4.26 Social welfare for $\eta_a = 1, \eta_b = 0.5$

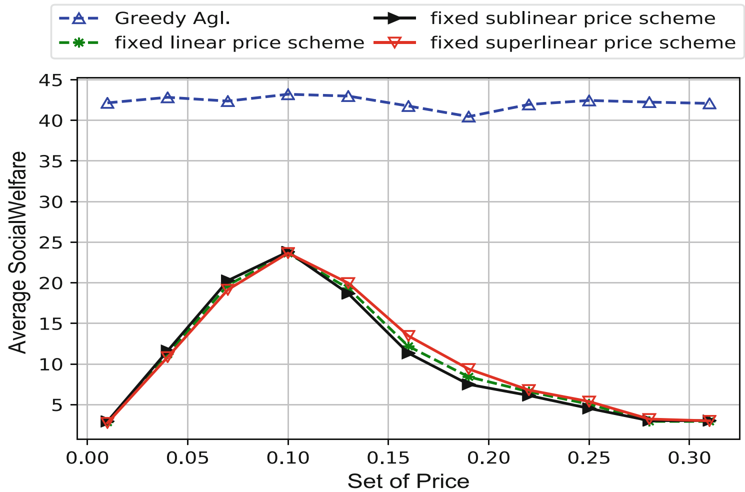

Fig. 4.27 Social welfare for $\eta_a = 1, \eta_b = 1$

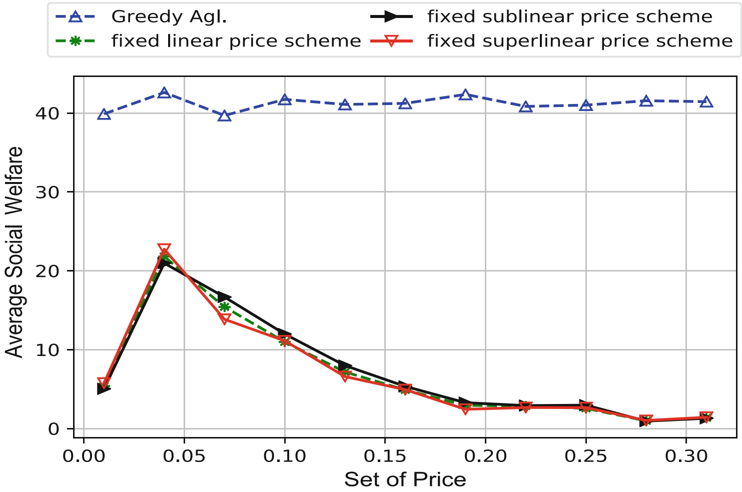

Fig. 4.28 Social welfare for $\eta_a = 1, \eta_b = 2$

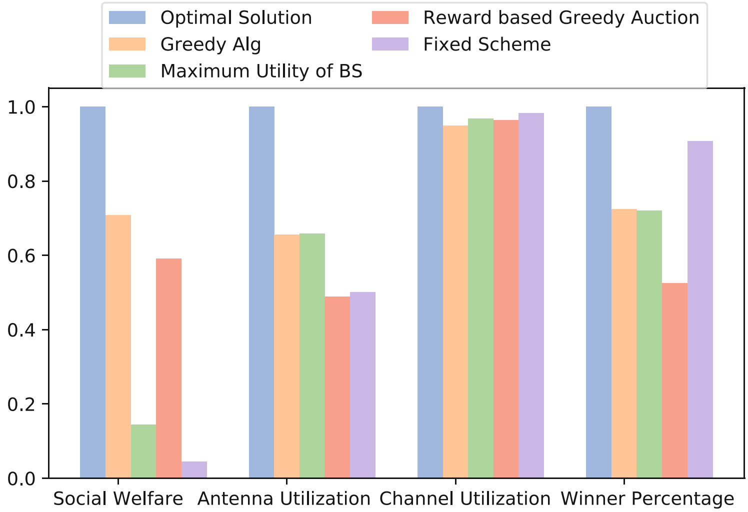

Fig. 4.29 Normalized ratio for $\eta_a = 1, \eta_b = 0.5$

welfare of fixed price firstly increases and then decreases and equal to 0 when the initial price increases. This is because when the basic price becomes too high, the sum of the price is higher than the valuation of the resources claimed in a bid. Moreover, the social welfare achieved by linear, sublinear, and superlinear price schemes is lower than by the proposed greedy scheme. This proves our proposed auction scheme outperforms the fixed price scheme.

In Figs. 4.29, 4.30, and 4.31, we observe the metrics: social welfare, resource utilization and percentage of five schemes: greedy proposed scheme and fixed price

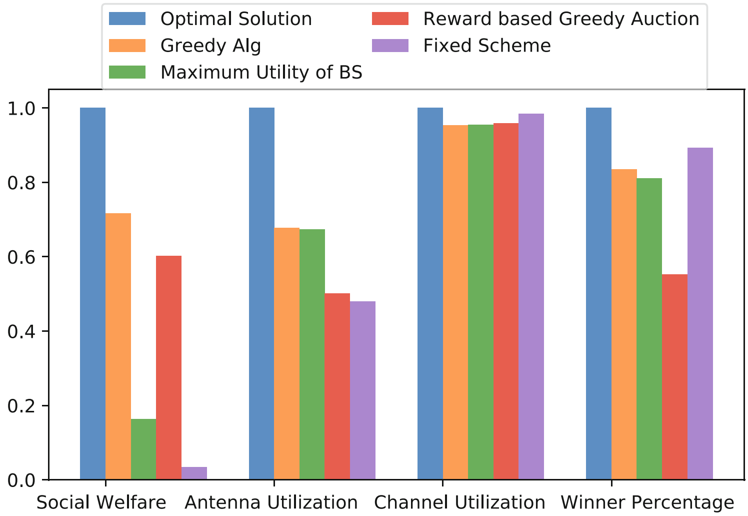

Fig. 4.30 Normalized ratio for $\eta_a = 1, \eta_b = 1$

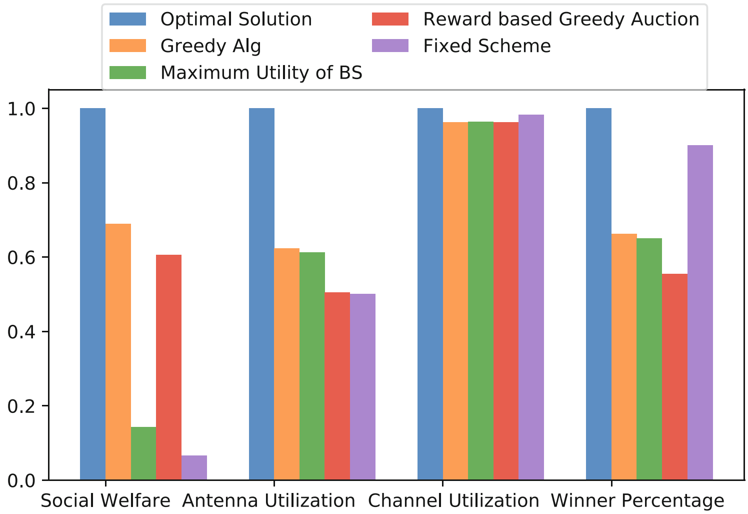

Fig. 4.31 Normalized ratio for $\eta_a = 1, \eta_b = 2$

schemes with other baselines. We perform in terms of the ratio with the proposed greedy scheme. Among these schemes, the optimal solution is the highest in terms of all metrics. Compared with the proposed scheme, the fixed price can utilize more resources and more mobile users but provides less social welfare. This is due to the fact that the fixed price mechanism heavily depends on the prices of the resources. In addition, the resource utilization of our proposed scheme is competitive to the one of the maximum utility of the BS scheme and the reward-based greedy auction scheme. Furthermore, the proposed scheme provides more social welfare.

4.4 Summary

In this chapter, we have proposed two incentive mechanisms, such as Stackelberg game-based incentive mechanism and the auction theory-based incentive mechanism, for federated learning. In the first part, we have designed and analyzed a novel crowdsourcing framework to enable federated learning. An incentive mechanism has been established to enable the participation of several devices in federated learning. In particular, we have adopted a two-stage Stackelberg game model to jointly study the utility maximization of the participating clients and edge computing server interacting via an application platform for building a high-quality learning model. We have incorporated the challenge of maintaining communication efficiency for exchanging the model parameters among participating clients during aggregation. Further, we have derived the best response solution and proved the existence of Stackelberg equilibrium. We have examined the characteristics of participating clients for different parametric configurations. Additionally, we have conducted numerical simulations and presented several case studies to evaluate the framework's efficacy. Through a probabilistic model, we have designed and presented numerical results on an admission control strategy for the number of client's participation to attain the corresponding local consensus accuracy. In the second part, we formulated the incentive problem between the BS and mobile users in the federated learning service market as the auction game with the objective of maximizing social welfare. Then, we presented the method for mobile users to decide the bids submitted to the BS so that mobile users can minimize the energy cost. We also proposed the iterative algorithm with low complexity. In addition, we proposed a primal-dual greedy algorithm to tackle the NP-hard winner selection problem. Finally, we showed that the proposed auction mechanism guarantees truthfulness, individual rationality, and computation efficiency. Simulation results demonstrated the effectiveness of the proposed mechanism where social welfare obtained by our proposed mechanism is 400% larger than by the fixed price scheme. The model in our work can be extended to multi BS when users are one a large area. One BS can not cover the whole area. In that case, one BS performs edge aggregations of local models which are transmitted from devices in proximity. When each BS achieves a given learning accuracy, updated models at the edge are transmitted to the cloud or macro base station for global aggregation. Intuitively, this hierarchical model can help to reduce significant communication overhead between device users and the cloud via edge model aggregations and reduce the latency. In addition, through the coordination by the edge servers in proximity, more efficient communication and computation resource allocation among device users can be achieved. Moreover, we can consider the hierarchical auction mechanism consisting of two hierarchical auction models. i.e. a single-seller multiple-buyer model where the lower stage is between BS and mobile users and the higher stage is between the cloud and base stations. Another direction is the case which there are many base stations from different organizers who are interested in using the data from the set of users to train similar types of machine learning models. In that situation, there may be a competition of base stations. This will make base stations' decision-making different from our work. Therefore, we can also consider it as future work.

Appendix

A.1 KKT Solution

The utility maximization problem in (4.21) is a convex optimization problem whose optimal solution can be obtained by using Lagrangian duality. The Lagrangian of (4.21) is

$$\begin{aligned} \mathcal{L}(r, x(\epsilon), \lambda) = \beta \left(1 - 10^{-(ax(\epsilon)+b)}\right) - r \sum_{k \in K} (1 - \theta_k^*(r)) \\ + \lambda \left[\delta(1 - \max_k \theta_k^*(r) - x(\epsilon)\right] \end{aligned} \tag{A.1}$$

where $\lambda \geq 0$ is the Lagrangian multiplier for constraint (4.22).
By taking the first-order derivative of (A.1) with respect to $x(\epsilon)$ and λ, KKT conditions are expressed as follows:

$$\frac{\partial \mathcal{L}}{\partial x(\epsilon)} = a\beta e^{-(a(x(\epsilon))+b)} - \lambda \leq 0, \text{ if } x(\epsilon) \geq 0. \tag{A.2}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \left[\delta(1 - \max_k \theta_k^*(r)) - x(\epsilon)\right] \geq 0, \text{ if } \lambda \geq 0. \tag{A.3}$$

By solving (A.2), the solution to the utility maximization problem (4.21) is

$$x^*(\epsilon) = \frac{-\ln(\lambda/a\beta) - b}{a}. \tag{A.4}$$

From (A.3), the Lagrangian multiplier λ is as

$$\lambda^* = a\beta e^{[a\delta(1-\max_k \theta_k^*(r))+b]}. \tag{A.5}$$

Thus, from (A.4) and (A.5) the optimal solution to the utility maximization problem (4.21) is

$$x^*(\epsilon) = \delta(1 - \max_k \theta_k^*(r)). \tag{A.6}$$