Applications of Bipartite Graphs and Matchings in Federated Learning

Sheng Yun
Graduate School of Arts and Sciences
Fordham University
New York, USA
syun13@fordham.edu

***Abstract*—Bipartite graph and bipartite matching can play a significant role in Federated Learning, and it can be a good help for improving conventional federated learning algorithms. Federated Learning has gained much attention, leading to a trend in Data Science. It can be used in many scenarios, such as Security Systems, Recommendation Systems, Smart City, and Medical Record Distribution Systems due to its superior ability to alleviate central server computation and privacy preservation. However, Federated Learning has challenges, such as data heterogeneity and malicious data poisoning. One approach to handling the data heterogeneity is to utilize the bipartite graph in part of the federated learning to mitigate the impacts of the non-iid data. Two kinds of approaches have been proposed; one is bipartite mating the entity resolution, which implies the datasets matching, and the other way is the client and sub-server matching. We surveyed five pieces of literature to describe the existing approaches and how bipartite graphs and matching are involved in federated learning. In those approaches, most of the methods apply the usage of the Hungarian method(KM) to solve the bipartite matching. We will demonstrate the problem setups, use, and effectiveness, and then we will conclude the concepts and necessities of those applications.**

***Index Terms*—Bipartite Matching, Federated Learning, Greedy Algorithm, Heterogeneity**

I. Introduction

With the development of IoT devices, the computing ability and storage of devices are growing dramatically. This trend is giving researchers and engineers more ways of dealing with modeling problems. One of the trends, Federated Learning, has been getting so much attention in the recent five years due to its decentralization, privacy preservation, and low requirements on the server side[3]. Federated Learning is a novel framework that distributes the modeling and machine learning tasks to the edge IoT devices and aggregates the result in the center server. This framework enables the centralized server to combine the model weights and perform the machine-learning tasks without accessing the clients' raw data by mapping the edge IoT devices to the centralized server(Fig. 1).

One of the challenges in Federated Learning is the slow/inability of convergence in heterogeneous data when the server is aggregating the model[4]. Many methods have been proposed since state of the art to handle the heterogeneity. One primary solution for handling unmatched data and clients is to use bipartite matching to find the matching between one data and another. Using graph theory and bipartite matching evolved from the graph and distributed system[5].

Bipartite graphs are one of the most famous types of graphs in graph theory. A bipartite graph can be divided into two groups of nodes connected to the nodes in another group, but the nodes in the same group will not have a direct edge. The most significant feature of the bipartite graph is that it does not contain an odd-number vertices circle[7]. Bipartite can be applied to many fields, like stable marriage, IoT networks, and supply chain networks, since many problems appear bipartisan.

The methodology of solving the bipartite matching largely involves the greedy algorithm since it is a method of seeking the local optimal with practical effectiveness[8]. The greedy algorithm, however, receives much criticism for its inability to avoid a trap, deadlock[6], and short-sightedness[9]. In this paper, we will ignore the negativity the greedy algorithm may bring and focus on the applications of bipartite matching in Federated Learning.

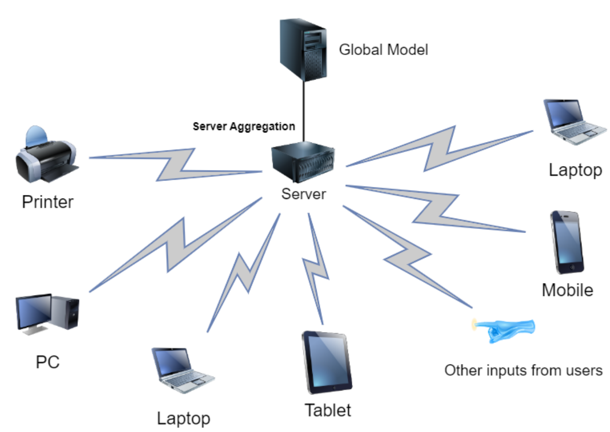

Fig. 1. Federated Learning Framework

The rest of the paper will be structured as follows:

- In section (II), we will introduce the problem background.
- In section (III), we will analyze the usage of bipartite matching in federated learning and the problem formulation.
- In section(IV), we will discuss the future challenges and outlook of utilizing bipartite matching in other parts of

federated learning.

The use of graph theory is still at an entry level, and we need more deep work in this direction since federated learning is an early-stage framework. We hope this paper can lead to more attention in this area and encourage more inspirational ideas in the future.

II. Backgound

A. Bipartite Graphs

In graph theory, we will have the vertices and edges(directed and undirected) to form a complete graph. A bipartite graph is one of the most famous and fundamental types of graphs in graph theory[7]. Like the name Bipartite, a Bipartite graph can be separated into two paralleled sets such that none of the vertices in that two sub-graphs within the same subgraph are adjacent. Weighted bipartite is a bipartite graph with a value assigned to each edge. The bipartite graph is widely used in many fields, such as graph coloring, job scheduling, pebbling in hypergrid, job matching, stable marriage, IoT devices matching, and more.

B. Bipartite Matchings

Bipartite matching is the process of looking for the matching between the two subgraphs' vertices[7]. To better illustrate the definition, we will make the metaphor with the dating matching in a group of people (this is also one of the famous and conventional applications of bipartite graphs). Assigning a set of Females F to a set of Males M. Each female has associated matching conditions, the so-called constraint, which is captured by the edge set $E \subseteq F \times M$: the edge $(f, m) \in E$ if and only if people in F can be matched to dating with male d. Each man d has a corresponding constraint on the max dating capacity $Cm \in N$, indicating the maximum number of females that can be matched. We can formulate the above metaphor as follows:

$$G = (J, D, \epsilon), \text{ and} C \in N^D$$

$$\textbf{Matching:} M \subseteq J \times D$$

A Matching is **maximal** if and only if for all for All $e \in \epsilon$ M, $M \cup$ {e} is not feasible.

C. Greedy Algorithms on Bipartite Graphs

The greedy algorithm is a methodology that always chooses the local optimal at each step. The greedy algorithm is also widely used worldwide, a methodology that has been used almost since human history[9]. However, this methodology has been criticized and proved to be short-sighted and sometimes unable to reach the global optimal in the system and unable to get the optimal solution in certain circumstances[9]. However, the greedy algorithm gains its position because of its fast and straightforward mechanism. In the idea of "greedy" in the weighted bipartite graph, the matching will always find the local optimal in each step. However, this mechanism was proved by mathematicians that it could not guarantee a perfect matching in bipartite matching even if a perfect matching solution exists in the system[10].

D. Federated Learning Overview

Federated Learning is a framework that distributes computing tasks to edge IoT devices. Each device will update and learn from the data individually and then upload the learned model parameters to the central server. The central server will then aggregate the parameters and update the global model. In the central server, the server will generally do an average for all the collected parameters from the devices and make the modifications(FedAvg[11], state-of-the-art algorithm). This works ideally under the assumption that each dataset the devices have is independent and identically distributed(iid). However, this assumption is way too optimistic and rare in real-world scenarios.

E. Data Distribution in Federated Learning

There are different types of distribution of data generated from the devices on a circling basis, so it is important to regularize the data and transform it to iid data if possible. In order to personalize the capability of regularizing the data, researchers normally categorize the data into two types of distributions: vertically distributed, horizontally distributed by their way of aggregating(Fig. 2). Non-iid data can lead to an update shift when the model has been aggregated in many rounds of updates.

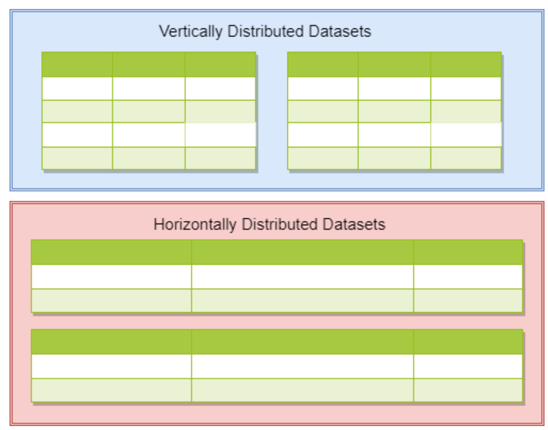

Fig. 2. Vertically and Horizontally distributed datasets

F. Heterogeneity Challenges in Federated Learning

One of the biggest challenges in federated learning is that the framework works poorly when the devices have different sizes and shapes of data(non-iid). Many studies have proved that machine learning algorithms can fail to converge in the server's objective functions when the central server aggregates the models' updates. The inability of federated learning is critical in the practical value of federated learning and has drawn much attention over the years of study in federated learning[12]. However, the optimal solution or preferred solution has remained undecided. In the following section, we will discuss the efforts that researchers use bipartite matching to solve the heterogenous data problems in federated learning.

III. Bipartite Matching on Federated Learning

Bipartite graphs generally have two major types of applications in federated learning. One way a bipartite graph helps with federated learning is to use bipartite matching to match the corresponding datasets, and another one is using matching to pair up the clients and sub-servers(Fig. 3). There are also some applications in bipartite matching that helped a part of one of the steps in federated learning. We will introduce the applications and the formulation in the following sections.

A. Entity Resolution and Federated Learning

a) Overview of the Framework: As part of their paper, they[13] modify a token-based entity resolution algorithm to avoid matching rows of different classes and perform experiments in which entity resolution relies on noisy data. This scenario is highly relevant to real-world applications. Several features can be used to categorize work in this area, including (a) vertical or horizontal dividing of the data and (b) models that are being learned. Most of the previous studies on secure distributed learning have considered a horizontal data dividing to different entities record with the same features.

Different features can be recorded for the same entity in a vertical data partition. It is more challenging to partition data vertically than horizontally [14]. Observe that any conventional learning algorithm can learn from the data gathered in their latter case. Although gathering the data in one place would solve the problem in this vertical partition case, they still would need to figure out which entities of the various datasets correspond to which other entities. Data partitioning on a vertical basis is more appropriate when different companies are located in the same market, so they can aggregate different features for the same clients[13].

b) Role of the Bipartite Matching in the Algorithm: A feasible solution exists in which all vertex edges connect to vertices to form a balanced complete bipartite graph with non-negative weights. They want to maximize the sum of weights. For entity resolution, they consider cosine similarity as the criterion to be maximized and note that maximizing this criterion is equivalent to maximizing the same criterion for "(1+cosine similarity)", which is not negative. The greedy approach provides an approximation to entity resolution in $O(|V|^2 log|V|)$ [13]for a non-optimized implementation; they denote C^* as the optimal value of the total cosine similarity. To even reach $O(|V|^3)$ time complexity, one must use a significantly more sophisticated implementation of the Hungarian algorithm (KM), which theoretically achieves the optimum [15].

The greedy algorithm is used not just for computational efficiency and its simplicity of implementation but because they do not really seek the optimal solution to entity-resolution but rather one that will be useful for learning. It[13] can already be remarked that greedy Algorithm provides a very good constant approximation to C^* from both perspectives - having a good approximation to the entity resolution criterion while having the best possible solution for modeling.

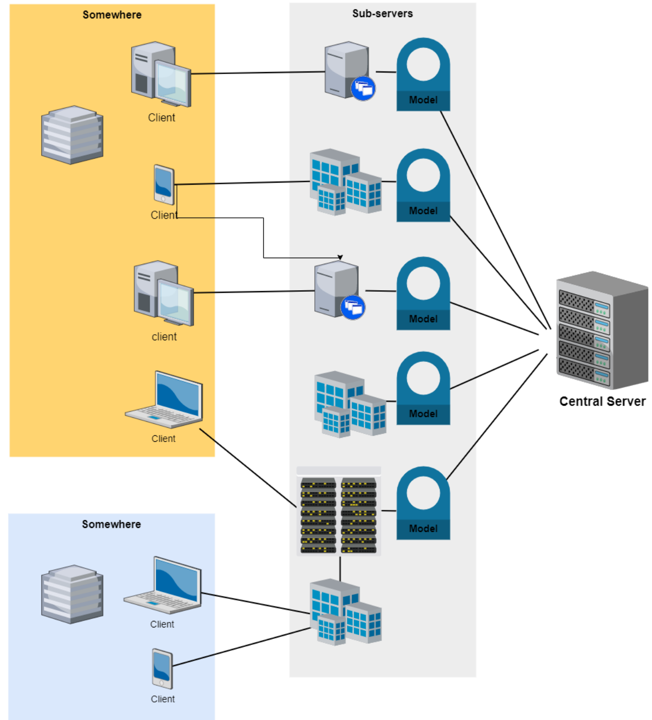

Fig. 3. General Idea in Server mapping

B. Federated Learning Over Wireless Channels With Differential Privacy

a) Overview of their Work: In this paper, the authors[16] investigate how to minimize delays for FL over wireless communication networks while taking into account the possibility that different privacy protections may apply and that data may be imbalanced. A MAMAB(multi-agent multi-armed bandit) framework is used to solve this question, with constraints on overall training performance and client DP requirements. By using the upper confidence bound (UCB) method, the authors estimate rewards in MAMAB based on the long-term constraint imposed by the Lyapunov technique on training performance and DP(Differential Privacy) requirements. Then, they[16] converted the MAMAB framework to a matching problem in max-min bipartite matching at each round of communication, by estimating rewards by UCB methods.

A max-min bipartite matching problem can be solved with two proposed solutions using this estimate to schedule clients at each communication round. The first solution(modified Hungarian algorithm) is capable of achieving optimal matching results but is highly complex, while the second solution(greedy matching with a better alternative (GMBA)) is less complex but performs well[16].

Using the MAMAB-based FL framework, they analyzed its feasibility. According to their study, the optimality gap for the agent-based collaborative MAMAB framework is $O(V^2 NlogT)$, [16] where V represents the client participation ratio constraint. In contrast, N and $logT$ represent the cost of communication dynamics and computation dynamics during the learning process, respectively. Particularly, N and T refer to the number of channels and rounds of communication available.

A comprehensive set of experimental results is presented to demonstrate the feasibility and effectiveness of their proposed algorithm. As a result, they claimed that their algorithms are capable of outperforming baseline algorithms by fully utilizing the interaction between communication and computation.

b) Role of the Bipartite Matching in the Algorithm: Different from the conventional matching methodology, they[16] suggest to maximize the the minimum of $R_i(t)$, where $i \in U$ and R represent the rounds in communications. The formulation of this question is given as follows:

$$\max_{a(t)} \min_{i \in U} \sum_{j \in N} a_{i,j}(t) e_{i,j}(t)$$

under some constraint $C1$, $C2$, $C3$.

The Objective function above maximize the minimum of the estimated reward for all clients, i.e., $j \in N a_{i,j}(t) e_{i,j}(t)$, on all avaliable matchings. The assumption in this problem noted, there is a complete weighted bipartite graph $G = (U, N, E)$, where U and N represent the sets of the clients and channels(servers), respectively. E is the set of edges which has each edge's value noted as $e_{i,j}(t)$.

In their initialization[16], the server will setup a client set as $A = 1, ..., N$, which will include all clients but the initialization is zero matrix $\hat{a}(t)$. That server will randomly select a greedy order from the set of all-order, i.e., $o \in O$, and then assign available channels or so-called sub-server to all clients. They define o as a sequence of clients $(o_1, ..., o_U)$, s.t. $o_i \in U$ and $o_i = o_j$ for all $i = j$, and O denotes the set of all orders. Then, via the estimated reward, the $i-th$ client in o is assigned to an optimized channel from all available channels[16] as: $\hat{a}_{i',j'}(t) = 1$ and $j' = arg \max_{j \in A} e_{i,j}(t)$. After updating, selected available sub-servers j will be removed from the candidate pool A Until finishing updating the clients. Finally, update $a(t)$ as the better one in $\{a(t-1), \hat{a}(t)\}$.

The authors claim this approach is more efficient and performs better than the state-of-the-art approach.

C. Multi-agent and Swapping Reinforcement Learning

a) Overview of their Work: Gao et al. propose FedSwap[17], an efficient and fair federated learning-based 5G DSA(Dynamic Spectrum Access) system. they deploy an improved multi-agent reinforcement learning model to each UE(User Equipments), which decreases channel collisions and increases resource utilization efficiency. They design a novel swapping mechanism at the aggregation phase to effectively address the fairness issue. In addition, swapping is conducted on the second FC(Fully Connected) layer. It is important to note that after the DQN(Deep Q Network) models converge, the parameters of the second FC layer still vary between UEs. Different scheduling plans for UEs are a result of such divergence. Changing the values of these divergent parameters may affect potential scheduling plans. Hence, the values of these layers cannot be changed directly but must be swapped between the UEs, and there comes the role of bipartite matching.

A fair channel[17] access is achieved by exchanging the corresponding scheduling plan. The DQN models must be swapped using a fair swapping algorithm[18] between multiple UEs. However, it is impossible for the swapping processor to directly distribute a certain channel to a certain UE, since it doesn't know the potential schedule of the model. In other words, their swapping algorithm uses the model exchange history, which records the interval since a model was assigned to a UE last. We will adopt their chart(Fig. 4) to better illustrate this mechanism.

b) Role of the Bipartite Matching in the Algorithm: Above approach means that if there is a longer interval between the swapping processor and the UE, the same model is not distributed to the UE in a short period of time. As a result, UEs will be able to request varying channels by using different DQN models. UEs and models are considered as disjoint vertex sets of the bipartite graph to achieve a larger interval, using the widely used bipartite matching algorithm, KM algorithm(Hungarian matching algorithm) [15].

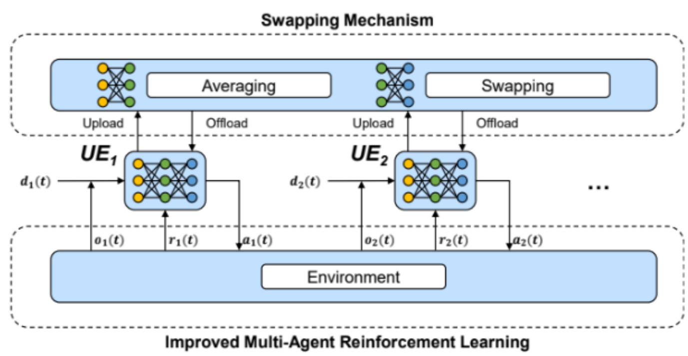

Fig. 4. Their demonstration on this mechanism

D. Clustered Vehicular Federated Learning

In [20], they have researched the problem of clustered Federated Learning in vehicular networks. A mobile-aware learning process for clustered FL was designed to bridge the gap between vehicular networks and FL clustering. According to the authors, v2v(Vehicular to Vehicular) communication in vehicular networks is an asset to overcome the FL communication bottleneck. A subset of vehicles is selected to act as cluster heads in each communication round, and the remainder are matched with them. gNodeBs(a methodology in 5G base station study)[20] with good wireless communication channels and diverse datasets are preferentially selected.

Moreover, clustering based on similarities between updates is introduced to overcome the slow convergence of single-joint FL models[22], mainly when concept shifts are present. A new model[20] is created in this step, which is then sent to non-participants and newly joined vehicles for evaluation, where they score their preferences. Each vehicle is matched to its preferred model (cluster head) based on the preference values.

Formulating cluster matching and cluster head selection as optimization problems with learning objectives is possible.

In order to select and allocate cluster heads, they used a greedy algorithm, and to form clusters, they set up a maximum-weighted bipartite matching model. Experiments demonstrate the efficacy of this approach can not only accelerate learning but also increase the importance of clustering according to the updates to limit concept shift[20].

E. FL with Cross-Platform Dispatching and Matching

In the paper[19], Wang et al. propose federated order dispatching for cross-platform ride hailing, in which multiple platforms can make dispatching decisions without sharing any local data. The authors propose a new framework named Fed-LTD, which allows both dispatching models and dispatch decisions to be shared for addressing effectiveness(Fig.5), efficiency, and privacy challenges. To make their solution more practical[19], they have also devised techniques for preserving privacy and optimizing efficiency. Using real data, the experimental results show that the proposed solution offers obvious advantages in terms of total accuracy and running time.

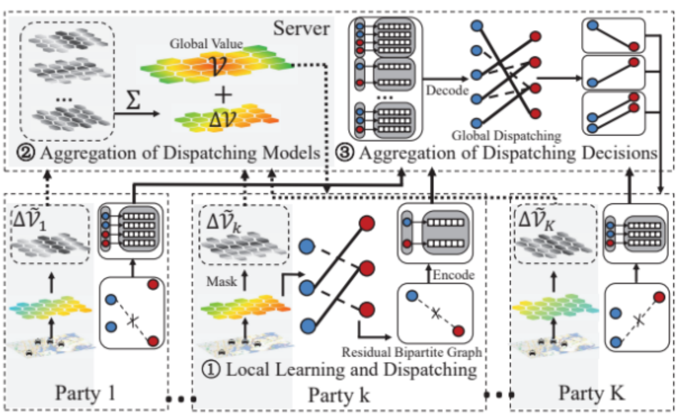

Fig. 5. Overview of the Fed-LTD

a) Role of the Bipartite Matching in the Algorithm: In the process of Order Dispatching, they[19] define their ridesharing platforms federation consisting of K platforms (or sub-servers) $P_1, P_2, ..., P_K$ and a central server S. The server can run the model as a central learner without any data, pre-knowledge, or one of the parties or sub-servers. They assume that each party is not fully-trusted, so-called semi-honest in their paper, and there is a possibility that the server is malicious to poison the data, which will result in data poisoning[21]. Each Party or sub-server P_k has its bipartite graph $G_k = (U_k \cup V_k, E_k)$ such that the nodes are its own assigned drivers and orders. They denote G as a global bipartite graph under the non-federated setting. Then, they formulated the concept of the global optimum. This approach essentially is aggregating the bipartite matching in the aggregation process.

$$\mu(G) = \max_{<M^{(t)}>} \sum_{t=1}^{T} SUM(M^{(t)}(G^{(t)}))$$

Global optimal can be considered and solved as the optimal of bipartite matching results in this non-federated setting.

IV. Challenges and Future Work

A. Limitations on Bipartite Matching

Given that most of the above-mentioned algorithms used the KM algorithm to solve the formulated question, we realize that the complexity of the conventional is $O(|V|^3)$. Inevitably, we will build up a sizeable bipartite graph for later use since there are many clients. Dealing with a system that is not time efficient before solving the question itself is not ideal and practical. We may need to consider the trade-off between the problems it solves and the problems it brings. Also, it is inevitable to have information entropy during the layers transmissionm, and that can cause a information loss and a potential performance degrade.

B. Limitations on the Usages

Solving the bipartite matching is a challenging task in the first place, and the above applications, which indicate the current applications, will have a massive size of clients to match with the sub-servers. Considering the questions we mentioned, we, ideally, want this expense to be worthy. However, the proposed algorithms are either too personalized, or with limited improvement in performance, or it is hard to prove the improvement in performance is sustainable and robust. Also, the security for the sub-servers will need to be implemented, and that is a new crucial problem we have to deal with.

C. Future Directions

For further work, we think it will be interesting to migrate this framework to other data or problems to test the approach's robustness. Another possible direction is what is the best way to formulate the question with bipartite graph. Can we do better? Also, it will be interesting to further investigate the trade-off and the information loss during the process.

V. Conclusion

In this study, we investigated the uses of the bipartite graph and matching in Federated learning. There are mainly two ways of thinking about the application. One is using the bipartite graph to do data matching, which was proposed in the early phase of federated learning development; the other one is the client matching techniques. We focus more on the second way because it is ensembling the bipartite approaches and the federated learning clients matching and aggregation problems, and it is more worthy of investigating, given that the relationship between clients and server can be a huge problem sometimes. Some of the proposed frameworks involve the use of sub-servers to group the clients and set up a reinforcement on the sub-servers to maintain feasibility. They claim that the novel approach can bring improvement in performance, and benefit te organization.

However, it still lacks of proof that these frameworks will provide better access to the system. The claim could be harmfully influenced by the efficiency and interruptions to the sub-servers. The practical value is questionable, and the claim of performance improvement might need more proof of robustness and sustainability.

Acknowledgment

The previous proposal is about applying the greedy algorithm in bipartite matching. However, there are sufficient analyses and quantitive analyses about bipartite matching. So the author decided to move to bipartite matching applications in federated learning. Thanks to the professor's previous statement about the freedom of changing the topic after the proposal.

References

[1] Wei, K., Li, J., Ma, C., Ding, M., Chen, C., Jin, S., ... & Poor, H. V. (2021). Low-latency federated learning over wireless channels with anifferential privacy. IEEE Journal on Selected Areas in Communications, 40(1), 290-307.

[2] Gao, Z., Li, A., Gao, Y., Li, B., Wang, Y., & Chen, Y. (2021, November). FedSwap: A Federated Learning based 5G Decentralized Dynamic Spectrum Access System. In 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD) (pp. 1-6). IEEE.

[3] Liu, L., Zhang, J., Song, S. H., & Letaief, K. B. (2020, June). Client-edge-cloud hierarchical federated learning. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

[4] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2), 1-210.

[5] Kahn, J. (2001). An entropy approach to the hard-core model on bipartite graphs. Combinatorics, Probability and Computing, 10(3), 219-237.

[6] Gao, W., Friedrich, T., Neumann, F., & Hercher, C. (2018, July). Randomized greedy algorithms for covering problems. In Proceedings of the Genetic and Evolutionary Computation Conference (pp. 309-315).

[7] Aharoni, R. (1984). König's duality theorem for infinite bipartite graphs. Journal of the London Mathematical Society, 2(1), 1-12.

[8] Asratian, A. S., Denley, T. M., & Häggkvist, R. (1998). Bipartite graphs and their applications (Vol. 131). Cambridge university press.

[9] DeVore, R. A., & Temlyakov, V. N. (1996). Some remarks on greedy algorithms. Advances in computational Mathematics, 5(1), 173-187.

[10] Arnosti, N. (2022). Greedy matching in bipartite random graphs. Stochastic Systems, 12(2), 133-150.

[11] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[12] Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Computing Surveys (CSUR), 54(6), 1-36.

[13] Nock, R., Hardy, S., Henecka, W., Ivey-Law, H., Patrini, G., Smith, G., & Thorne, B. (2018). Entity resolution and federated learning get a federated resolution. arXiv preprint arXiv:1803.04035.

[14] Nock, R., Hardy, S., Henecka, W., Ivey-Law, H., Patrini, G., Smith, G., & Thorne, B. (2018). Entity resolution and federated learning get a federated resolution. arXiv preprint arXiv:1803.04035.

[15] Wright, M. B. (1990). Speeding up the Hungarian algorithm. Computers & Operations Research, 17(1), 95-96.

[16] Wei, K., Li, J., Ma, C., Ding, M., Chen, C., Jin, S., ... & Poor, H. V. (2021). Low-latency federated learning over wireless channels with differential privacy. IEEE Journal on Selected Areas in Communications, 40(1), 290-307.

[17] Gao, Z., Li, A., Gao, Y., Li, B., Wang, Y., & Chen, Y. (2021, November). FedSwap: A Federated Learning based 5G Decentralized Dynamic Spectrum Access System. In 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD) (pp. 1-6). IEEE.

[18] Tiwari, S. S. A., Suman, S., & Viswanathan, P. (2014). LONG TERM EVOLUTION (LTE) PROTOCOL Verification of MAC Scheduling Algorithms in NetSim.

[19] Wang, Y., Tong, Y., Zhou, Z., Ren, Z., Xu, Y., Wu, G., & Lv, W. (2022, August). Fed-LTD: Towards Cross-Platform Ride Hailing via Federated Learning to Dispatch. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 4079-4089).

[20] Taïk, A., Mlika, Z., & Cherkaoui, S. (2022). Clustered Vehicular Federated Learning: Process and Optimization. IEEE Transactions on Intelligent Transportation Systems.

[21] Tolpegin, V., Truex, S., Gursoy, M. E., & Liu, L. (2020, September). Data poisoning attacks against federated learning systems. In European Symposium on Research in Computer Security (pp. 480-501). Springer, Cham.

[22] Briggs, C., Fan, Z., & Andras, P. (2020, July). Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-9). IEEE.