Energy-Efficient Link Selection for Decentralized Learning via Smart Devices with Edge Computing

Cheng-Wei Ching[†], Chung-Kai Yang[§‖], Yu-Chun Liu[‡‖], Chia-Wei Hsu[†‖], Jian-Jhih Kuo[†*],
Hung-Sheng Huang[†], and Jen-Feng Lee[†]

***Abstract*—Data privacy preservation has drawn much attention in emerging machine learning applications. Decentralized learning is thus developed to guarantee data security and get rid of the involvement of parameter server to avoid transmission bottleneck. However, the previous research focuses on data compression and exchange rules of model parameters among smart devices but neglects the interplay between link cardinality and transmission power consumption. To jointly optimize these issues, in this paper, we first formulate a new optimization problem, named GreenDL, prove its hardness, and then propose an approximation algorithm termed CoTRAIN. Experiment and simulation results manifest that CoTRAIN reduces more than 20% power compared with traditional methods without sacrificing the convergence rate.**

I. Introduction

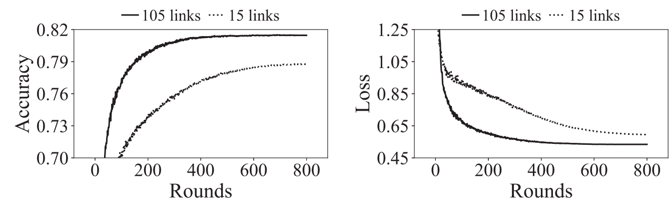
Nowadays, artificial intelligence (AI) has drawn much attention and innovates numerous advanced applications [1]. However, AI models usually require considerable amount of data for training, thereby giving rise to two following issues. One is data security and privacy. The data required for training are usually stored in smart devices of users [2]. The leakage of data associated with personal information is thus the last thing that users would like to encounter when enjoying AI-related services [2]. Another is the need of powerful platforms for training. To collect and process enormous amount of data from smart devices *in a centralized fashion*, powerful servers meeting requirements of storage, computing, and bandwidth for emerging service providers are getting more and more impractical when the number of smart devices grows drastically.[1]

To ease the above issues, collaborative learning has been proposed to train a target model by multiple smart devices of users with their local data [1], [4], [5]. Usually, a centralized server is required to aggregate different local updates of model parameters from the smart devices for the next round of training, whereas it may become a crucial network bottleneck and limit the scalability [6]. To this end, decentralized learning (DL) further removes the central server. Thus, each smart device *only* shares local updates of model parameters with its neighboring smart devices in the mobile edge network via device-to-device (D2D) transmission [7] to locally derive *new average model parameters* for the next round of training in DL. Remark that messages here are forwarded to neighboring devices by *one-hop broadcasting* to improve the transmission efficiency and reduce the number of transmissions.[2] It can be envisaged that smart devices act as both a central parameter server and a training unit at the same time. Eventually, DL

[†]Dept. of Computer Science, National Chung Cheng University, Taiwan
[§]Dept. of Computer Science, National Tsing Hua Univerisity, Taiwan
[‡]Dept. of Electrical Engineering, National Chung Cheng University, Taiwan
* indicates the corresponding author; ‖ denotes the equal contributions.
Corresponding author's email: lajacky@cs.ccu.edu.tw

[1]Cisco predicts smart devices will grow to 13.1 billions by 2023 [3].

[2]The local update messages can be further compressed for better transmission efficiency [8], [9], whereas it is beyond the scope of this paper.



(a) Effect on accuracy (b) Effect on loss

Fig. 1. Effect of different number of links on DL in a network with 15 devices.

will converge and achieve *consensus* among the smart devices. DL has two main advantages as follows: 1) It guarantees data privacy since data are only accessed by their owners. 2) The central server for parameter aggregation is no longer required.

Intuitively, a network with more links has more data exchanges and is more likely to yield better training performance (e.g., higher accuracy and fewer training rounds for convergence) for DL. To further verify the interplay between *link cardinality* (i.e., number of links) and training performance, we implement a DL framework [4] on a small network with 15 smart devices.[3] The effects of different link cardinality on the accuracy and loss are shown in Fig. 1, where the two tested networks have 105 and 15 links, respectively. The accuracy rates of 105-link and 15-link networks respectively achieve 70% at the 37^{th} and 104^{th} rounds while 75% at the 65^{th} and 282^{th} rounds, and eventually converge to 81.1% and 79.1%. The above results show that more links in a network benefit the training performance of DL. To increase the link cardinality in a network, devices are inevitable to use higher transmission power to make more one-hop neighbors and then incur higher energy consumption during a round, whereas energy-efficient communications are crucial for smart devices [10]. However, the trade-off between *training performance* and *transmission power consumption* has not been carefully explored for DL to select adequate and energy-efficient links in networks.

Optimizing *transmission power consumption* while ensuring *link cardinality* in the network for DL to guarantee *training performance* leads to new challenges as follows: 1) *Discrete power levels*. Devices have to set a higher transmission power to cover farther devices. Nonetheless, smart devices are usually unable to support configurations over a continuous domain but have a *limited*, *discrete* set of possible transmission power levels [11]. A smart device may seriously waste transmission power if covering *inappropriate* devices whose locations are slightly beyond the coverage of its previous transmission power level. 2) *Symmetric wireless links*. Raising the transmission power level of only one smart device may not connect the device to the other devices. Two devices have to set a sufficient

[3]The detailed implementation and setting are described in Section IV.

power level to cover each other in the transmission range. In addition, the network with the selected links should be *connected*. 3) *Density-aware power selection*. Connecting devices *close* to each other tends to reduce the overall transmission power. Moreover, the devices in denser areas using high transmission power levels are inclined to have more links than those in sparser areas. Overall, the problem is quite challenging since it has to jointly decide which links should be selected to achieve the required *link cardinality* to ensure the *training performance* and which transmission power level should be used for each device to minimize the *transmission power consumption*.

To address the challenges, we present **Green** Transmission Power Level Allocation Problem for **D**ecentralized **L**earning (GreenDL). With the given parameters: 1) the smart devices in the network, 2) the required transmission power level to connect each device pair, and 3) a *link cardinality ratio* (i.e., a user-defined ratio of requested link cardinality to total number of possible device pairs), GreenDL asks for a set of links that satisfies the *link cardinality ratio* and yields the minimum overall *transmission power consumption* of devices. Then, we prove that GreenDL is NP-hard and design an efficient **Co**llaborative Densi**t**y-Awa**r**e Power Level **A**llocation and L**i**nk Selectio**n** Approximation Algorithm (CoTRAIN). To jointly address the above three challenges, CoTRAIN introduces a novel notion, *niche link set*, that is, a set of links which connect the devices that are close to each other in a *dense* area such that the links have the *lowest power consumption per link cardinality* in the network. Then, CoTRAIN introduces *niche indicator* (detailed later) to evaluate sets of links and recognize the niche link set in the network. Therefore, CoTRAIN iteratively augments the set of selected links with the niche link set until the link cardinality ratio is satisfied. Experiment and simulation results manifest that CoTRAIN outperforms naive approaches by 20%.

II. The GreenDL Problem

This paper considers a mobile edge network that consists of 1) a set V of smart devices that are able to use device-to-device (D2D) communication, 2) a *limited*, *discrete* set of possible transmission power levels P, and 3) a set E of all possible links between devices in V, where each possible link $e \in E$ has a minimum transmission power level $c(e)$.[4] GreenDL is formulated as an *integer linear programming* (ILP) as follows.

Recall that the training performance can be guaranteed by selecting an adequate number of links. Let $\varphi \in [0, 1]$ denote a user-defined lower bound for selected links to bound the link cardinality (i.e., *link cardinality ratio*), and binary variable y_e denote whether link $e \in E$ is selected. To ensure the training performance, we require an adequate number of links, i.e.,

$$\sum_{e \in E} y_e \geq \varphi |E|. \tag{1}$$

Let binary variable x_{vp} denote whether device $v \in V$ sets the transmission power level power as $p \in P$. Each device can only choose a configuration for transmission power, and then

$$\sum_{p \in P} x_{vp} = 1, \qquad \forall v \in V. \tag{2}$$

[4]To explore the intrinsic property of GreenDL, both two end devices u and v are assumed to set the same transmission power level, which is no less than the minimum transmission power level $c(e)$, for communications in this paper.

Let $V(e)$ denote the two corresponding devices incident to link e. Both devices of a possible link e have to choose a transmission power level no less than the minimum transmission power $c(e)$ to cover each other in the transmission range. Thus,

$$y_e \leq \sum_{p \in P : p \geq c(e)} x_{vp}, \qquad \forall e \in E, \forall v \in V(e). \tag{3}$$

Then, the induced graph by the selected links should be connected, leading to the following *flow-based* constraints. Note that r is an arbitrary device selected from V and every device $d \in V \setminus \{r\}$ should have a path from r in the induced graph to ensure the *network connectivity*. Let binary variable z_{uv}^d denote whether the flow is steered along the link from u to v to build a path from r to d, where $u, v \in V$. Therefore,

$$\sum_{u \in V} z_{ru}^d - \sum_{u \in V} z_{ur}^d = 1, \qquad \forall d \in V \setminus \{r\}. \tag{4a}$$

$$\sum_{u \in V} z_{du}^d - \sum_{u \in V} z_{ud}^d = -1, \quad \forall d \in V \setminus \{r\}. \tag{4b}$$

$$\sum_{u \in V} z_{vu}^d - \sum_{u \in V} z_{uv}^d = 0, \qquad \forall v, d \in V \setminus \{r\}, v \neq d. \tag{4c}$$

Lastly, combined with the above *flow-based* constraints, the last constraint in the following ensures that the network induced by the selected links is connected.

$$y_e \geq z_{uv}^d, \quad \forall e \in E, \forall u, v \in V(e), u \neq v, \forall d \in V \setminus \{r\}. \tag{5}$$

Definition 1. Given a *complete* network $G = (V, E)$, the minimum transmission power level $c(e)$ of each possible pair $e \in E$, and a link cardinality ratio φ, the **Green** Transmission Power Level Allocation Problem for **D**ecentralized **L**earning (GreenDL) asks for a set of links $\subseteq E$ and the transmission power level for each device $v \in V$ to meet the constraints (1)−(5) while minimizing the total transmission power consumption, i.e.,

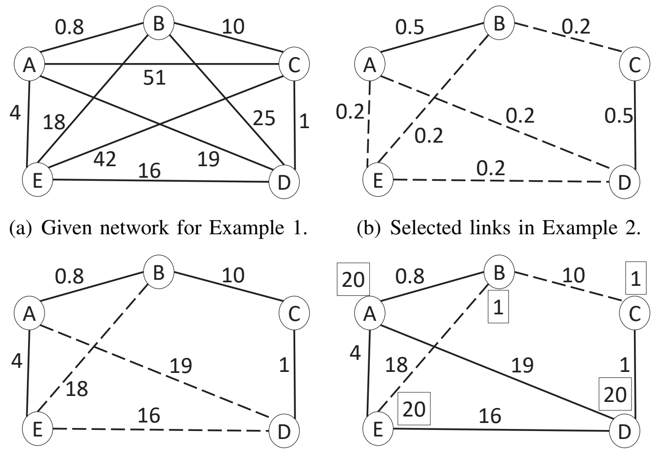
$$\text{min.} \sum_{v \in V} \sum_{p \in P} x_{vp} \cdot p \tag{6}$$

Note that GreenDL has *decision variables* x_{vp}, y_e, and $z_{u,v}^d$, and it is NP-hard since a NP-hard problem, Min-Power Symmetric Connectivity problem [12], is its special case.

Example 1. The toy example shows the effect of different link selection strategies for GreenDL. The given network $G = (V, E)$ associated with link information is depicted in Fig. 2(a) and the transmission power levels of devices are summarized in Table I. Two naive extensions of Kruskal's algorithm termed KR1 and KR2 can be applied to GreenDL. Assume that the *link cardinality ratio* is 0.5, and thus five links should be selected. KR1 iteratively selects the links with the lowest $c(e)$ until the number of selected links is satisfied. The selected links are $\overline{AB}, \overline{CD}, \overline{AE}, \overline{BC}, \overline{DE}$, and the total power consumption is $20 + 20 + 20 + 20 + 20 = 100$. KR2 iteratively selects the link that will increases the lowest extra power among all links. It first selects $\overline{AB}, \overline{CD}$ since their extra power is $1 + 1 = 2$. Then, it selects $\overline{BC}$ since the extra power is $(20-1) \times 2 = 38$. Similarly, $\overline{AD}, \overline{AE}$ are selected and the power consumption is also 100. By contrast, CoTRAIN (detailed in Section III) selects links $\overline{AB}, \overline{AD}, \overline{AE}, \overline{CD}, \overline{DE}$ and achieves the *optimal*

TABLE I
POWER LEVELS FOR EXAMPLE OF GREENDL

Power Level	1	2	3	4	5
Transmission Power	1	20	40	60	80



(a) Given network for Example 1. (b) Selected links in Example 2. (c) Constructed MST in Example 3. (d) Removed links in Example 4.

Fig. 2. Example of GreenDL to demonstrate CoTRAIN. (a) The number next to each link indicates the minimum transmission power required to connect the two devices. The power required for $\overline{BE}$ is 18, whereas the power level 2 is 20 (see Table I). Thereby, B, E must set power mode to level 2 if $\overline{BE}$ is selected. (b) The link sets $\mathcal{L}_1$ and $\mathcal{L}_2$ chosen by NL Selection are in the solid and dashed lines based on the optimal fractional solutions at the 1^{st} and 2^{nd} iterations, respectively. (c) The links of $\mathcal{T}$ selected by NC Provision are in the solid lines. (d) The number in a box next to each device is the transmission power level of that device in the solution output by RL Deletion.

power consumption $20+1+1+20+20=62$. Therefore, CoTRAIN can reduce the cost by 38% in this example. ■

III. ALGORITHM DESIGN

Two naive extensions from the well-known approaches for the minimum spanning tree (MST) problem can be applied to solve GreenDL. They respectively follow the ideas behind Kruskal's algorithms, termed KR1 and KR2. KR1 iteratively selects the unselected link e with the lowest $c(e)$ among all links until the link cardinality ratio is satisfied. Different from KR1, KR2 iteratively selects the unselected link that will increase the lowest extra power consumption among all links instead. Note that both avoid selecting links that will generate cycles in the solution until a MST forms to ensure the network connectivity first. However, both neglect the interplay among the three challenges described in Section I such that most links selected in their solutions are rarely in dense areas, and thus may waste power to cover a few links.

To efficiently solve GreenDL, we design an approximation algorithm named CoTRAIN to carefully address the above challenges. Instead of selecting a *single* link with the lowest minimum transmission power level for each time, CoTRAIN takes a *forward view* to select *multiple links* that have the *lowest power consumption per link cardinality* (i.e., *niche link set*). In this way, CoTRAIN can avoid excessively increasing the transmission power of devices in sparse areas and significantly reduce the transmission power. It introduces *niche indicator* to evaluate link sets and iteratively find the niche link set (or a close one) to augment the links until the *link cardinality ratio* is met, and then imposes a MST for the *network connectivity*. Finally, it removes the redundant links to save more energy.

Specifically, let E_t be the unselected links in E after link selection of the t^{th} iteration and $E_0 = E$, initially. Finding the niche link set in the network $\{V, E_{t-1}\}$ at the t^{th} iteration is equal to solving the following integer programming (IP), where variables (x^t, y^t) are akin to (x, y) in Definition 1.

$$\text{min.} \quad \frac{\sum_{v\in V}\sum_{p\in P} x^t_{vp}\cdot p}{\sum_{e\in E_{t-1}} y^t_e} \tag{7a}$$

$$\text{s.t.} \quad y^t_e \le \sum_{p\in P: p\ge c(e)} x^t_{vp}, \quad \forall e\in E_{t-1}, \forall v\in V(e) \tag{7b}$$

$$\sum_{e\in E_{t-1}} y^t_e \ge 1 \tag{7c}$$

$$x^t_{vp}, y^t_e \in \{0,1\}, \quad \forall v\in V, \forall p\in P, \forall e\in E_{t-1} \tag{7d}$$

However, the above IP is still *non-trivial*. To this end, CoTRAIN subtly 1) *relaxes* the integral restriction $x^t_{vp}, y^t_e \in \{0,1\}$ to $\overline{x}^t_{vp}, \overline{y}^t_e \ge 0$ and 2) *scales* $\sum_{e\in E_{t-1}} y^t_e \ge 1$ to $\sum_{e\in E_{t-1}} \overline{y}^t_e = 1$ to get a *linear programming* (LP) as follows.

$$\text{min.} \quad \sum_{v\in V}\sum_{p\in P} \overline{x}^t_{vp}\cdot p \tag{8a}$$

$$\text{s.t.} \quad \overline{y}^t_e \le \sum_{p\in P: p\ge c(e)} \overline{x}^t_{vp}, \quad \forall e\in E_{t-1}, \forall v\in V(e) \tag{8b}$$

$$\sum_{e\in E_{t-1}} \overline{y}^t_e = 1 \tag{8c}$$

$$\overline{x}^t_{vp}, \overline{y}^t_e \ge 0, \quad \forall v\in V, \forall p\in P, \forall e\in E_{t-1} \tag{8d}$$

In this way, the *fractional optimal solution* $(\overline{x}^t, \overline{y}^t)$ can be acquired by a polynomial-time LP solver (e.g., Gurobi), and CoTRAIN can find a near-optimal niche link set (i.e., close to the niche link set) based on the *clue* given by $(\overline{x}^t, \overline{y}^t)$.

To acquire the near-optimal niche link set, CoTRAIN constructs *multiple* candidate niche link sets in each iteration and chooses the one with the *largest* niche indicator. Specifically, at the t^{th} iteration, CoTRAIN constructs $\mathcal{N}_t$ candidate niche link sets, $\mathcal{C}^t_0$, $\mathcal{C}^t_1$, ..., $\mathcal{C}^t_{\mathcal{N}_t-1}$, and the i^{th} candidate is

$$\mathcal{C}^t_i = \{e\in E_{t-1} \mid 0.5^{(i+1)} < \overline{y}^t_e \le 0.5^i\}, \tag{9}$$

where $\mathcal{N}_t = 2\lceil\log|E_{t-1}|\rceil - 1$ and $0 \le i \le \mathcal{N}_t - 1$. Then, the niche indicator of each niche link set $\mathcal{C}^t_i$ is defined as

$$\mathcal{I}(\mathcal{C}^t_i) = |\mathcal{C}^t_i| - \frac{2^i}{\mathcal{N}_t+1}. \tag{10}$$

Later we show that the *niche link set* and *niche indicator* are the cornerstones of CoTRAIN to ensure the approximation ratio.

A. Algorithm Description

CoTRAIN includes the following three phases: 1) Niche Link Selection (NL Selection), 2) Network Connectivity Provision (NC Provision), and 3) Redundant Link Deletion (RL Deletion). Particularly, NL Selection first iteratively constructs multiple candidate niche link sets, chooses the best set among them, and adds the links of the chosen set into the selected links until the link cardinality ratio is satisfied. NC Provision then imposes a MST to connect all devices to guarantee network connectivity. Finally, RL Deletion eliminates redundant links to reduce total power consumption. To achieve the approximation ratio, it is important for NL Selection to evaluate the constructed candidate niche link sets by niche indicator to find the near-optimal niche link set.

1) Niche Link Selection (NL Selection): NL Selection *iteratively* finds the near-optimal niche link set and adds the links of the set to the selected links until the link cardinality ratio is satisfied. Recall that E_t is the unselected link set in E after link selection of the t^{th} iteration and $E_0 = E$, initially. For the t^{th} iteration, NL Selection obtains the optimal fractional solution $(\overline{x}^t, \overline{y}^t)$ of LP (8) with the network[5] $\{V, E_{t-1}\}$ by any existing LP solver (e.g., Gurobi). Then, it constructs $\mathcal{N}_t$ *candidate niche link sets* according to eq. (9) and chooses the set with the largest niche indicator (see eq. (10)) among the $\mathcal{N}_t$ candidate sets, where $\mathcal{N}_t = 2\lceil \log |E_{t-1}| \rceil - 1$. The candidate niche link set chosen in the t^{th} iteration can be written as

$$\mathcal{L}_t = \underset{\mathcal{C}_i^t : 0 \le i \le \mathcal{N}_t - 1}{\arg\max} \ \mathcal{I}(\mathcal{C}_i^t). \quad (11)$$

Afterward, NL Selection adds the links in $\mathcal{L}_t$ into the selected links, and then employs the sub-network with $\{V, E_t\}$, where $E_t \leftarrow E_{t-1} \setminus \mathcal{L}_t$, for the next iteration (i.e., the $(t+1)^{th}$ iteration) to compute the optimal fractional solution of LP (8) and $\mathcal{L}_{t+1}$. NL Selection repeats link selection until the number of selected links is at least $(1-\epsilon)\varphi|E|$, where ϵ is a positive tunable parameter to limit the loss percentage of requested link cardinality. Now, the link set selected by NL Selection is $\mathcal{L} = \bigcup_{t=1}^{k} \mathcal{L}_t$, where k is the number of executed iterations.

Example 2. Following Example 1, NL Selection is shown in Fig. 2(b), where $\epsilon = 0.1$ and $(1-\epsilon)\varphi|E| = 4.5$. The non-zero variables in the solution of LP (8) are $\overline{y}^1_{\overline{AB}} = \overline{y}^1_{\overline{CD}} = 0.5$ in the 1^{st} iteration. Thus, $\mathcal{C}_1^1 = \{\overline{AB}, \overline{CD}\}$ is chosen as the $\mathcal{L}_1$ since the two links satisfy $0.5^{(1+1)} < \overline{y}_e^t \le 0.5^1$ based on eqs. (9) and (11), and $E_1 = E \setminus \{\overline{AB}, \overline{CD}\}$. Remark that $\mathcal{C}_0^1 = \mathcal{C}_2^1 = \mathcal{C}_3^1 = ... = \mathcal{C}^1_{\mathcal{N}_1 - 1} = \emptyset$. Then, the non-zero variables in the solution of LP (8) are $\overline{y}^2_{\overline{AD}} = \overline{y}^2_{\overline{AE}} = \overline{y}^2_{\overline{BC}} = \overline{y}^2_{\overline{BE}} = \overline{y}^2_{\overline{DE}} = 0.2$ in the 2^{nd} iteration. Thus, $\overline{AD}, \overline{AE}, \overline{BC}, \overline{BE}, \overline{DE}$ are included in $\mathcal{C}_2^2$ and chosen as $\mathcal{L}_2$. Lastly, since $|\mathcal{L}| = |\mathcal{L}_1 \cup \mathcal{L}_2| = 7 \ge (1-\epsilon)\varphi|E| = 4.5$, NL Selection stops. ■

2) Network Connectivity Provision (NC Provision): NC Provision employs Kruskal's algorithm to get a MST $\mathcal{T}$ of the network G to ensure the *network connectivity*. To precisely calculate the power consumption, each link e is associated with a new cost $p(e) = \min_{p \in P: p \ge c(e)} p$ for computing $\mathcal{T}$. After NC Provision, the selected link set becomes $(E \cap \mathcal{T}) \cup \mathcal{L}$.

Example 3. Following Example 2, Fig. 2(c) shows the MST $\mathcal{T}$ built by NC Provision. It has $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, and $\overline{AE}$. Thus, the selected links are $\overline{AB}, \overline{AD}, \overline{AE}, \overline{BC}, \overline{BE}, \overline{CD}, \overline{DE}$. ■

3) Redundant Link Deletion (RL Deletion): The final phase eliminates redundant links so as to further decrease the power consumption. Let $\mathcal{E}$ and $\mathcal{P}(\mathcal{E})$ denote the current selected links (i.e., $\mathcal{E} \leftarrow (E \cap \mathcal{T}) \cup \mathcal{L}$) and its transmission power consumption. Since each device in the network should have an adequate transmission power level,

$$\mathcal{P}(\mathcal{E}) = \sum_{v \in V} \max_{e \in E(v) \cap \mathcal{E}} p(e), \text{ where } p(e) = \min_{p \in P: p \ge c(e)} p, \quad (12)$$

and $E(v)$ denotes the selected links incident to device $v \in V$. RL Deletion iteratively removes the link e, the removal of which saves the most power until $|\mathcal{E}| \le \varphi|E|$, i.e.,

$$e = \arg\max_{e \in \mathcal{E}} \mathcal{P}(\mathcal{E}) - \mathcal{P}(\mathcal{E} \setminus \{e\}). \quad (13)$$

However, to keep the *network connectivity*, RL Deletion doesn't remove link e if removing e splits the induced network.

Example 4. Following Example 3, $\overline{BC}, \overline{BE}$ in Fig. 2(d) will be removed by RL Deletion since removing them can achieve the greatest power saving while keeping the *network connectivity*. The power consumption was $20+20+20+20+20 = 100$ and can be reduced to $20+1+1+20+20 = 62$. ■

B. Theoretical Analysis

Time Complexity. The complexity is dominated by NL Selection. Let T_{LP} denote the complexity of solving LP (8), which is clearly in polynomial time. Since NL Selection selects at most $|E|$ links, the overall time complexity is $O(T_{LP}|E|)$. The details are omitted due to the page limit. ■

Theorem 1. CoTRAIN is a $(O(\log|E|(\ln\frac{1}{\epsilon} + \frac{1-\varphi}{\epsilon\varphi})), 1-\epsilon)$-approximation algorithm, where ϵ is a positive tunable parameter to limit the loss percentage of requested link cardinality.

Proof. We sequentially prove that 1) at the t^{th} iteration of NL Selection, the chosen near-optimal niche link set $\mathcal{L}_t$ has the power consumption per link cardinality $\mathcal{P}(\mathcal{L}_t)/|\mathcal{L}_t|$ less than $\lceil \log |E_{t-1}| \rceil \cdot \mathcal{P}(\mathcal{L}_t^*)/|\mathcal{L}_t^*|$, where $\mathcal{L}_t^*$ denotes the optimal niche link set at the t^{th} iteration, i.e., the optimal solution of IP (7), 2) NL Selection outputs a set of links $\mathcal{L}$ with the power consumption $\mathcal{P}(\mathcal{L}) < \lceil \log |E| \rceil (1 + \ln\frac{1}{\epsilon} + \frac{1-\varphi}{\epsilon\varphi}) \mathcal{P}(OPT)$ while ensuring $|\mathcal{L}| \ge (1-\epsilon)\varphi|E|$, where OPT is the optimal solution of GreenDL, and 3) the power consumption of MST by NC Provision is $\mathcal{P}(\mathcal{T}) \le 2 \cdot \mathcal{P}(OPT)$. Finally, we will prove the theorem with the three above statements.

For the 1^{st} statement, we respectively prove the two cases when $|E_{t-1}| \ge 9$ and $|E_{t-1}| < 9$. For $|E_{t-1}| \ge 9$, we first prove that the niche indicator of chosen candidate niche link set $\mathcal{L}_t$ is at least zero and then the 1^{st} statement. Since in the optimal fractional solution of LP (8), the variable $\overline{y}_e^t$ of each link e that is not in any constructed candidate niche link set must be at most $0.5^{\mathcal{N}_t}$, where $\mathcal{N}_t = 2\lceil \log |E_{t-1}| \rceil - 1$. Then,

$$\sum_{e \in E_{t-1} \setminus \bigcup_{i \in [0, \mathcal{N}_t - 1]} \mathcal{C}_i^t} \overline{y}_e^t \le |E_{t-1}| \cdot 0.5^{\mathcal{N}_t} < \frac{1}{\mathcal{N}_t + 1}. \quad (14)$$

The last inequality explicitly holds when $|E_{t-1}| \ge 9$. Thereby, there exists some index $\hat{j} \in [0, \mathcal{N}_t - 1]$ such that the candidate niche link set $\mathcal{C}_{\hat{j}}^t$ has $\sum_{e \in \mathcal{C}_{\hat{j}}^t} \overline{y}_e^t \ge \frac{1}{\mathcal{N}_t + 1}$ because the number of candidate niche link sets is $\mathcal{N}_t$ and $\sum_{e \in E_{t-1}} \overline{y}_e^t = 1$ due to constraint (8c). Also, $\overline{y}_e^t \le 0.5^{\hat{j}}$ for each $e \in \mathcal{C}_{\hat{j}}^t$ implies that

$$\mathcal{I}(\mathcal{C}_{\hat{j}}^t) = |\mathcal{C}_{\hat{j}}^t| - \frac{2^{\hat{j}}}{\mathcal{N}_t + 1} \ge 2^{\hat{j}} \sum_{e \in \mathcal{C}_{\hat{j}}^t} \overline{y}_e^t - \frac{2^{\hat{j}}}{\mathcal{N}_t + 1} \ge 0. \quad (15)$$

[5] Note that when the number of unselected links is smaller than nine, i.e., $|E_{t-1}| < 9$, NL Selection can compute the optimal niche link set since it only has to examine at most $2^8 = 256$ possibilities, which takes a small constant time. Therefore, it suffices to deal with the cases where $|E_{t-1}| \ge 9$.

Thus, $\mathcal{I}(\mathcal{L}_t) \geq \mathcal{I}(\mathcal{C}^t_{\hat{j}}) \geq 0$ due to eq. (11). Let $\hat{i}$ be the index of $\mathcal{L}_t$, i.e., $\mathcal{L}_t = \mathcal{C}^t_{\hat{i}}$, and recall that $p(e) = \min_{p \in P: p \geq c(e)} p$. With $\overline{y}^t_e > 0.5^{\hat{i}+1}$ for each $e \in \mathcal{C}^t_{\hat{i}}$ and constraint (8b),

$$\mathcal{P}(\mathcal{L}_t) = \mathcal{P}(\mathcal{C}^t_{\hat{i}}) = \sum_{v \in V} \max_{e \in E(v) \cap \mathcal{C}^t_{\hat{i}}} p(e)$$
$$< 2^{\hat{i}+1} \sum_{v \in V} \max_{e \in E(v) \cap \mathcal{C}^t_{\hat{i}}} \overline{y}^t_e \cdot p(e) \leq 2^{\hat{i}+1} \sum_{v \in V} \sum_{p \in P} \overline{x}^t_{vp} \cdot p. \quad (16)$$

Moreover, $\mathcal{I}(\mathcal{C}^t_{\hat{i}}) \geq 0$ and $\mathcal{I}(\mathcal{C}^t_{\hat{i}}) \geq \mathcal{I}(\mathcal{C}^t_{\hat{j}})$. Thus,

$$|\mathcal{L}_t| = |\mathcal{C}^t_{\hat{i}}| \geq \frac{2^{\hat{i}}}{\mathcal{N}_t + 1} = \frac{2^{\hat{i}}}{\mathcal{N}_t + 1} \sum_{e \in E_{t-1}} \overline{y}^t_e. \quad (17)$$

Note that the solution value of LP (8) is no greater than that of IP (7) due to LP relaxation. Hence, by eqs. (16) and (17),

$$\frac{\mathcal{P}(\mathcal{L}_t)}{|\mathcal{L}_t|} = \frac{\mathcal{P}(\mathcal{C}^t_{\hat{i}})}{|\mathcal{C}^t_{\hat{i}}|} < \frac{2^{\hat{i}+1}\mathcal{P}(\mathcal{L}^*_t)}{\frac{2^{\hat{i}}}{\mathcal{N}_t+1}|\mathcal{L}^*_t|} = \lceil \log |E_{t-1}| \rceil \frac{\mathcal{P}(\mathcal{L}^*_t)}{|\mathcal{L}^*_t|} \quad (18)$$

We have proved the case where $|E_{t-1}| \geq 9$. For $|E_{t-1}| < 9$, the instance is quite small such that CoTRAIN can enumerate all possible solutions to obtain the optimal niche link set and the time complexity can be regarded as a small constant time. Thus, the following inequality always holds.

$$\mathcal{P}(\mathcal{L}_t)/|\mathcal{L}_t| < \lceil \log |E_{t-1}| \rceil \cdot \mathcal{P}(\mathcal{L}^*_t)/|\mathcal{L}^*_t|. \quad (19)$$

We then prove the 2^{nd} statement. Suppose that NL Selection stops at the k^{th} iteration, and let n_t denote the number of links that remains to be covered after t^{th} iteration. Since NL Selection stops in the k^{th} iteration, $\sum_{t=1}^{k-1} |\mathcal{L}_t| < (1-\epsilon)\varphi|E|$. Then, we have

$$n_0 = \varphi|E|, \quad \text{and} \quad n_{k-1} = \varphi|E| - \sum_{t=1}^{k-1} |\mathcal{L}_t| > \epsilon\varphi|E|. \quad (20)$$

Let OPT_t denote the optimal solution at the t^{th} iteration of NL Selection to meet link cardinality ratio. Since $\mathcal{L}^*_t$ is the optimal niche link set, $\mathcal{P}(\mathcal{L}^*_t)/|\mathcal{L}^*_t| \leq \mathcal{P}(OPT_t)/n_{t-1}$. Moreover, by combining with eq. (19), we have

$$\frac{\mathcal{P}(\mathcal{L}_t)}{n_{t-1} - n_t} = \frac{\mathcal{P}(\mathcal{L}_t)}{|\mathcal{L}_t|} < \lceil \log |E_{t-1}| \rceil \frac{\mathcal{P}(OPT_t)}{n_{t-1}}. \quad (21)$$

Remark that $\mathcal{P}(OPT_t) \leq \mathcal{P}(OPT)$ and $|E_{t-1}| \leq |E|$ for each $t \in [1, k-1]$. By combining eq. (21) with eq. (20),

$$\sum_{t=1}^{k-1} \mathcal{P}(\mathcal{L}_t) \leq \sum_{t=1}^{k-1} [(\lceil \log |E_{t-1}| \rceil \cdot \frac{n_{t-1} - n_t}{n_{t-1}}) \cdot \mathcal{P}(OPT_t)]$$
$$\leq \lceil \log |E| \rceil \cdot \mathcal{P}(OPT) \sum_{t=1}^{k-1} (\frac{n_{t-1} - n_t}{n_{t-1}})$$
$$\leq \lceil \log |E| \rceil \cdot \mathcal{P}(OPT) \cdot \ln(\frac{n_0}{n_{k-1}})$$
$$< \lceil \log |E| \rceil \cdot \mathcal{P}(OPT) \cdot \ln(\frac{1}{\epsilon}). \quad (22)$$

We then turn to analyze the k^{th} iteration. In the worst case, $\mathcal{L}_k$ might cover all the links left, i.e., $(1-\varphi)|E| + n_{k-1}$. Thus,

$$\frac{\mathcal{P}(\mathcal{L}_k)}{(1-\varphi)|E| + n_{k-1}} \leq \frac{\mathcal{P}(\mathcal{L}_k)}{|\mathcal{L}_k|} \leq \lceil \log |E| \rceil \cdot \frac{\mathcal{P}(OPT)}{n_{k-1}}. \quad (23)$$

Combining eq. (23) with eq. (20),

$$\mathcal{P}(\mathcal{L}_k) < \lceil \log |E| \rceil (1 + \frac{1-\varphi}{\epsilon\varphi}) \cdot \mathcal{P}(OPT). \quad (24)$$

Note that NL Selection stops when $|\mathcal{E}| \geq (1-\epsilon)\varphi|E|$. In addition, $\mathcal{P}(\mathcal{E}) = \sum_{t=1}^{k} \mathcal{P}(\mathcal{L}_t)$. By eq. (22) and (24), we have

$$\mathcal{P}(\mathcal{L}) < \lceil \log |E| \rceil (1 + \ln \frac{1}{\epsilon} + \frac{1-\varphi}{\epsilon\varphi}) \cdot \mathcal{P}(OPT). \quad (25)$$

Last, we prove the 3^{rd} statement. The power level of each device $v \in \mathcal{T}$ is $\max_{e \in E(v) \cap \mathcal{T}} p(e)$. Also, OPT must contain a spanning tree $\mathcal{F}$ (may not be the minimum one). Thus,

$$\mathcal{P}(\mathcal{T}) = \sum_{v \in \mathcal{T}} \max_{e \in E(v) \cap \mathcal{T}} p(e) \leq 2 \sum_{e \in \mathcal{T}} p(e) \leq 2 \sum_{e \in \mathcal{F}} p(e)$$
$$\leq 2 \sum_{v \in \mathcal{F}} \max_{e \in E(v) \cap \mathcal{F}} p(e) \leq 2\mathcal{P}(\mathcal{F}) \leq 2\mathcal{P}(OPT). \quad (26)$$

The inequalities hold since each link only connects two devices and the total transmission power consumption of any spanning tree is at least its total link cost. Combining eqs. (25) and (26), CoTRAIN outputs the link set $\mathcal{E}$ after NC Provision.

$$\mathcal{P}(\mathcal{E}) < (2 + \lceil \log |E| \rceil (1 + \ln \frac{1}{\epsilon} + \frac{1-\varphi}{\epsilon\varphi})) \cdot \mathcal{P}(OPT). \quad (27)$$

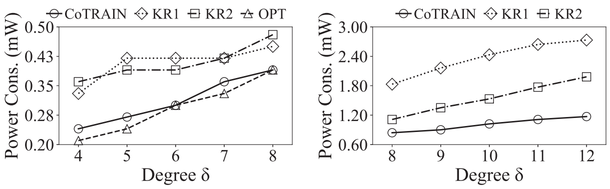
Finally, the theorem follows since link deletion by RL Deletion does not increase the power consumption and it stops once $|\mathcal{E}| \leq \varphi|E|$, i.e., $|\mathcal{E}|$ is still at least $(1-\epsilon)\varphi|E|$. ☐

IV. Performance Evaluation

A. Experiment and Simulation Settings

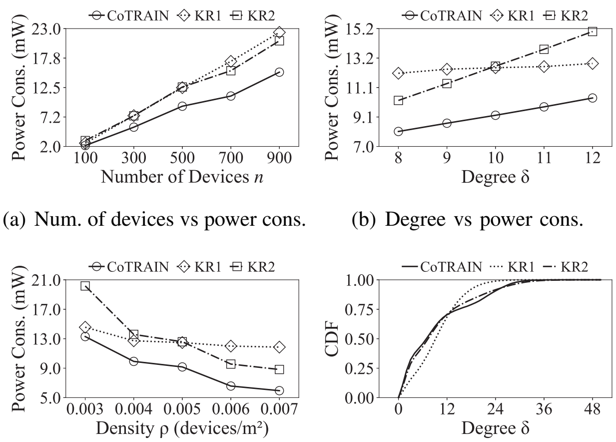
Extensive simulations are conducted to compare CoTRAIN with KR1, KR2, and the optimal solution OPT in a small (15 smart devices) and a median (105 smart devices) networks extracted from the real network dataset of Santander City [13], as well as large synthetic networks with n smart devices, where $n \in [100, 900]$. Assume that only 5 transmission power configurations are available for *D2D communication*, where the minimum and maximum ones are 0.02 mW and 2.4 mW, and the maximum link rage of D2D is 100 m. The other parameters are referred to the settings in [14]. We alter the *demand average degree* δ (i.e., average number of links per device) to observe the metric of power consumption. For synthetic networks, we also change n and the *device density* ρ (i.e., number of devices per unit of area). The default values of n, δ, and ρ are 500 devices, 10 links/device, and 0.005 devices/m^2, respectively.

To implement DL, we distribute 50000 images extracted from CIFAR-10 dataset [15] evenly to 15 devices in a small real network for training, and leave 10000 images for testing. The input images for training are preprocessed according to [16]. Tensorflow and Keras are used to implement a convolutional neural network (CNN), which has 2 convolutional layers (CL) and 3 fully connected layers (FL). Both 2 CLs have 64 channels and each layer is followed by a 3×3 max pooling with a stride of 2 and normalization. The first 2 FLs have 384 and 192 units (each of them with ReLu activation followed by 1 dropout), and the last FL is the final softmax output layer with 10 units. The learning rate, learning rate decay, number of local epochs, and local minibatch size are set to 0.2, 0.99, 5, and 64, respectively.



(a) Small real network (b) Median real network

Fig. 3. Effect of different degree on power consumption.



(a) Num. of devices vs power cons. (b) Degree vs power cons.

(c) Density vs power cons. (d) CDF of device degree

Fig. 4. Effect of different parameters in large synthesis networks.

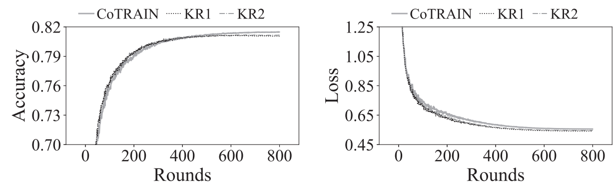
B. Discrete Power Levels and Symmetric Wireless Links

Overall, power consumption increases when δ and n go up as shown in Figs. 3 and 4. KR1 selects the links with lowest required power yet ignores discrete power levels and symmetric wireless links. The selected links tends to be sparse and waste much energy of devices to meet the power level for connecting devices. KR2 selects the links with the lowest increase on overall power consumption. Some devices may be forced to use the maximum power level to cover links. In contrast, the links of each candidate niche link set in CoTRAIN tend to be energy-saving. Then, by niche indicator, CoTRAIN can choose the set with more links to approximate the optimal solution.[6]

C. Effect of Density-aware Power Selection on Device Degree

Fig. 4(d) shows the distribution of device degree. KR1 tends to select links uniformly in the network since it overlooks the relation among links. More than 70% of devices have $5 \sim 20$ neighbors whereas those devices only account for 40% in KR2 and CoTRAIN. KR2 selects the links with lowest increase of power consumption such that it has the most devices with more than 30 selected links. When n is high, δ is low, or ρ is high, KR2 outperforms KR1 since more links for selection highlight the importance of relation among links as show in Figs. 4(a), 4(b), and 4(c). However, none of them can always generate desired solutions. By contrast, CoTRAIN can always balance the two factors to reduce the power consumption (see Table II) without sacrificing the convergence rate as shown in Fig. 5.

[6]The optimal solution of GreenDL is obtained only for the small network since it is NP-hard and the running time is exponential to the network size.



(a) Effect on accuracy (b) Effect on loss

Fig. 5. Convergence of different algorithms in small real network.

TABLE II
TRANSMISSION POWER CONSUMPTION IN SMALL REAL NETWORK (MW)

Accuracy	72%	75%	78%	81%
CoTRAIN	17.71 (1x)	25.52 (1x)	44.73 (1x)	127.28 (1x)
KR1	21.85 (1.23x)	32.77 (1.28x)	59.25 (1.32x)	189.92(1.49x)
KR2	19.90 (1.12x)	31.61 (1.24x)	54.63 (1.22x)	162.71 (1.28x)

V. CONCLUSIONS

This paper studies a new optimization problem, GreenDL, to explore the trade-off between transmission power consumption and training performance in depth. GreenDL is very intractable due to the new challenges, i.e., discrete power levels, symmetric wireless links, and density-aware power selection. We propose a novel algorithm CoTRAIN to subtly take a *forward view* to selects link sets by exploiting insightful *niche indicator* to assess link sets. Experiment and simulation results manifest that CoTRAIN outperforms traditional approaches more than 20% without sacrificing the convergence rate. For future research, it would be interesting to model the relation between the convergence rate and number of links to save more energy.

REFERENCES

[1] C. Zhang *et al.*, “Deep learning in mobile and wireless networking: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, pp. 2224–2287, 2019.

[2] T. Yang *et al.*, “Applied federated learning: Improving google keyboard query suggestions,” *arXiv preprint arXiv:1812.02903*, 2018.

[3] (2020) Cisco annual internet report (2018–2023). White paper.

[4] A. Koloskova *et al.*, “Decentralized stochastic optimization and gossip algorithms with compressed communication,” *ICML*, 2019.

[5] Y. Li *et al.*, “Pipe-SGD: A decentralized pipelined SGD framework for distributed deep net training,” in *NIPS*, 2018.

[6] S. Shi, X. Chu, and B. Li, “MG-WFBP: Efficient data communication for distributed synchronous SGD algorithms,” in *IEEE INFOCOM*, 2019.

[7] S.-R. Yang *et al.*, “Multi-access edge computing-assisted D2D streaming for proximity-based social networking,” in *IEEE GLOBECOM*, 2019.

[8] J. Wangni, J. Wang, J. Liu, and T. Zhang, “Gradient sparsification for communication-efficient distributed optimization,” in *NIPS*, 2018.

[9] D. Alistarh *et al.*, “QSGD: Communication-efficient SGD via gradient quantization and encoding,” in *NIPS*, 2017.

[10] Q. Ju *et al.*, “Predictive power management for internet of battery-less things,” *IEEE Trans. Power Electron.*, vol. 33, pp. 299–312, 2017.

[11] M. Condoluci *et al.*, “Toward 5G densenets: architectural advances for effective machine-type communications over femtocells,” *IEEE Commun. Mag.*, vol. 53, pp. 134–141, 2015.

[12] E. Althaus *et al.*, “Power efficient range assignment for symmetric connectivity in static ad hoc wireless networks,” *Wireless Networks*, vol. 12, pp. 287–299, 2006.

[13] C. Marche, L. Atzori, V. Pilloni, and M. Nitti, “How to exploit the social internet of things: Query generation model and device profiles’ dataset,” *Comput. Netw.*, vol. 174, p. 107248, 2020.

[14] A. Abdallah, M. M. Mansour, and A. Chehab, “Power control and channel allocation for D2D underlaid cellular networks,” *IEEE Trans. Commun.*, vol. 66, pp. 3217–3234, 2018.

[15] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.

[16] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2017.