Reliable and efficient RAR-based distributed model training in computing power network

Reliable and Efficient RAR-based Distributed Model Training in Computing Power Network

LING CHEN[1], YAJIE LI[1,*], CARLOS NATALINO[2], YONGCHENG LI[3], BOXIN ZHANG[1], YINGBO FAN[1], WEI WANG[1], YONGLI ZHAO[1], AND JIE ZHANG[1]

[1] *Beijing University of Posts and Telecommunications, State Key Laboratory of Information Photonics and Optical Communications, Beijing, 100876, China*
[2] *Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden*
[3] *Soochow University, the School of Electronic and Information Engineering, Soochow, 215021, China*
[*] *Corresponding author: yajieli@bupt.edu.cn*

Compiled July 15, 2024

Computing power network (CPN) is a novel network technology that integrates computing power from cloud, edge, and terminals using IP/optical cross-layer networks for distributed computing. CPNs can provide an effective solution for distributed model training (DMT). As a bandwidth optimization architecture based on data parallelism, ring all-reduce (RAR) is widely used in DMT. However, any node or link failure on the ring can interrupt or block the requests deployed on the ring. Meanwhile, due to the resource competition of batch RAR-based DMT requests, inappropriate scheduling strategies will also lead to low training efficiency or congestion. As far as we know, there is currently no research that considers the survivability of rings in scheduling strategies for RAR-based DMT. To fill this gap, we propose a new scheduling scheme for RAR-based DMT requests in CPNs to optimize computing, wavelength resource with time dimension while ensuring reliability. In practical scenarios, service providers may focus on different performance metrics. We formulate an integer linear programming (ILP) model and a RAR-based DMT deployment algorithm (RDDA) to solve this problem considering four optimization objectives under the premise of minimum blocking rate: minimum computing resource consumption, minimum wavelength resource consumption, minimum training time, and maximum reliability. Simulation results demonstrate that our model satisfies the reliability requirements while achieving corresponding optimal performance for DMT requests under four optimization objectives.

Optica: https://doi.org/10.1364/JOCN.511165
Authors' version: https://research.chalmers.se/en/publication/541072
IEEEXplore: https://ieeexplore.ieee.org/abstract/document/10507175

1. INTRODUCTION

In the era of 5G and artificial intelligence (AI), computing power has been a critical resource, since more and more applications and services require vast amounts of resources to support complex data processing and analysis tasks. As a new type of network technology, computing power networks (CPNs) can effectively manage and allocate computing, storage, network, and other resources among service nodes through a network control plane [1]. CPNs are distributed computing networks that can integrate computing power of cloud, edge, and terminals through an IP/optical cross-layer network [2]. Therefore, CPNs can achieve more efficient resource utilization and faster application deployment. Pre-trained models are one of the most promising AI technologies in recent years. For instance, ChatGPT (an AI chatbot developed by OpenAI) is an AI application of this technology. AI approaches to address problems can be abstracted in terms of two-stage processes, i.e., model building and inference in [3]. Model training is a critical process, as the performance of the model during training can determine how well it works when put into application for end-users. In addition, model training builds predictive models by processing large volumes of raw data. The complexity and size of the data sets used for training can result in time-consuming and resource-intensive processes. CPNs provide an effective solution for model training. As one of the largest and most complex pre-trained language models to date, GPT-3 model consumes thousands of GPUs in the training lasting for several weeks to a few months. The training optimizes 175 billion parameters using over 45 TB of text data for training [4]. To reduce training time, GPT-3 employed distributed training to accelerate training speed.

When data are inherently distributed or too large to store and/or process on a single machine, distributed model training (DMT) is one of the solutions to make it possible to train a model, which mainly includes model parallelism and data parallelism. The principle of model parallelism is to partition the model across several machines, such that the computing responsibility is assigned to different machines [5]. This way needs to transfer feature map between different machines and has limited parallelism. Thus, model parallelism is usually used for large models that cannot be stored on a single machine. Instead, data parallelism partitions the training data and copies the entire model to multiple machines to execute in parallel. Data parallelism is easier to implement than model parallelism, so it has been a popular solution in AI frameworks. The process of DMT based on data parallelism can be briefly described as follows: firstly, each node gets the gradient values by computing forward- and backward-propagation on its assigned training data. After that, each node collects the gradients generated by other nodes and merges these gradients. Finally, the nodes update their model weights with the merged gradients. The paper in [6] compared two typical architectures based on data parallelism, i.e., parameter server (PS) and ring all-reduce (RAR). The PS architecture consists of a set of parameter servers and workers, in which parameter servers are responsible for storing model parameters and aggregating global gradients, while the workers are for calculating the local gradient of parameters. However, as the number of workers increases, the PS architecture may suffer from communication bottlenecks and single-point-of-failure. Compared with PS architecture, RAR architecture averages gradients and sends them to all nodes by forming a logical ring among worker nodes, where each worker sends data only to its successor and receives data only from its predecessor. As shown in Fig. 1, RAR algorithm proceeds in two phases: scatter-reduce and allgather. Suppose there are N workers in a ring and the amount of updated gradients of every worker is d. Each worker splits its local gradients into N subsets, each of which is d/N. During the scatter-reduce phase, these workers add up received subsets to their own subsets and average them, until each worker owns a sub-final block of global gradient data. Next, in the allgather phase, workers exchange these blocks such that every worker gets an overall result. The training is completed through multiple epochs until the required model accuracy is reached. In one epoch, scatter-reduce is usually carried out $(N-1)$ times, and then allgather $(N-1)$ times. To sum up, every worker sends d/N amount of data for $2(N-1)$ times, and the total amount of data transmitted is $2d(N-1)/N$. The total communication amount of the RAR architecture does not increase linearly with the number of workers, which can efficiently reduce communication overhead and has better scalability. Hence, the RAR architecture is suitable for distributed scenarios with large-scale datasets and a large number of computing nodes, such as large-scale model training.

Nevertheless, RAR-based DMT faces an important challenge: if a node or a link within the ring suffers from failure, the RAR-based DMT will be interrupted because of the ring characteristic. Therefore, the ring reliability becomes a critical factor for RAR-based DMT. To the best of our knowledge, there is no research on reliable RAR-based DMT in CPNs at present. Therefore, we focus on how to provide reliable and efficient RAR-based DMT services in CPNs.

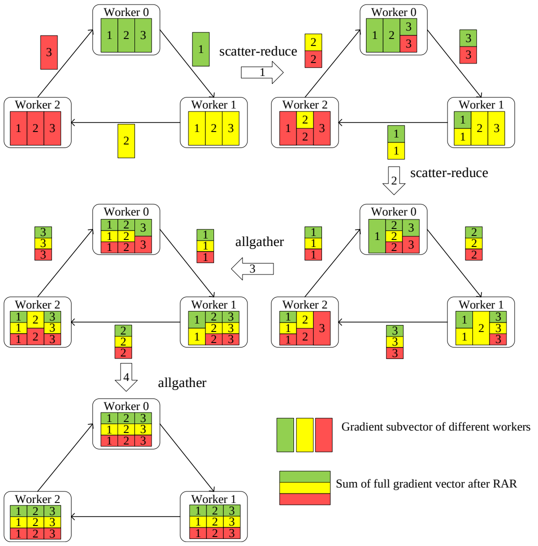
Worker 0
Worker 1
Worker 2
scatter-reduce
1
2
allgather
3
4
Gradient subvector of different workers
Sum of full gradient vector after RAR

Fig. 1. Illustration of RAR process

In a practical scenario, computing power and wavelength resources are known, while DMT requests can be collected and scheduled in advance, hence we can treat the work as a static planning problem. Moreover, we introduce time domain to extend the dimension of resource allocation and enable scheduling. Integer linear programming (ILP) is a powerful optimization method primarily employed for linear programming problems. ILP is commonly employed to address practical applications, such as allocation problems and resource scheduling problems. ILP aims to find values for a set of integer variables that satisfy specified linear constraints, while maximizing or minimizing a linear objective function. This method iteratively performs integer solution searches to gradually approach the optimal solution that satisfies the constraints. We use a mathematical programming optimizer Gurobi [7] to establish an ILP model. Gurobi employs optimization algorithms such as cutting plane algorithms to efficiently search the solution space and find the optimal solution.

In this paper, we investigate the scheduling of DMT requests in CPNs, jointly considering the allocation of computational resources and wavelength resources in the temporal dimension to maximize the provision of reliable DMT services. To address this problem, we propose an ILP model based on a small-scale topology. Meanwhile, we design a RAR-based DMT deployment heuristic algorithm to deal with a large-scale topology. We conduct simulations in the scenario of metropolitan micro data center networks to examine the performances of our ILP model and heuristic algorithm. The simulation results show that our model and algorithm can meet the reliability requirements and obtain the corresponding optimal performance for the DMT requests deployed with different optimization objectives. Note that, although our simulation focuses on the scenario of metropolitan micro data center networks, the applicability of our model is not limited to this context. By adjusting the topology parameters, our model can be flexibly applied to various other CPN scenarios.

The rest of this paper is organized as follows. In Section 2, we review the work related to DMT and job scheduling. Section 3 introduces the network model and the problem formulation to describe the differences in DMT requests deployment under different optimization schemes. Section 4 proposes the ILP model and Section 5 designs a heuristic algorithm to solve the problem in a large-scale topology. In Section 6, simulations are carried out to evaluate the performance of our ILP model and heuristic algorithm. In the end, Section 7 concludes this paper.

2. RELATED WORK

To meet the demands of emerging computational requirements and different network scenarios, some researchers have attempted to design CPN frameworks. The authors of [8] proposed a computing and networking interconnection architecture based on IP routing extension, which realized consistency in user experience and flexible and dynamic deployment of services. In [9], the work proposed key techniques for realizing a mobile CPN, which included quantifying computing resources, jointly optimizing communication and computing resources, enabling interactions between different computing capability providers, and so on. Besides, the work [10] proposed a computing power resource modeling approach and used completion time to represent the computing power for users. The authors of [11] presented a CPN composed of a computing layer, an IP layer, and an optical layer. The computing layer provided computing resources, the IP layer aggregated traffic through E-Switch, and the optical layer provided wavelength bypass through ROADM.

At present, DMT has been the topic of many works. The authors in [12] proposed a scheme called Liquid, an efficient GPU resource management platform for distributed deep learning (DDL) jobs. Liquid works by pre-scheduling data transmission, fine-grained GPU sharing, and event-driven communication to avoid over-allocating resources. In addition to optimizing network resources, some works studied scheduling strategies from the perspective of time. The work in [13] designed an online algorithm to maximize the overall utility of all jobs, which depended on their completion time, by adjusting the number of concurrent workers and parameter servers for each job over its course.

Owing to the communication bottleneck of the PS architecture as described in Section 1, RAR architecture has become popular and has been supported by mainstream DDL frameworks, such as PyTorch in [14]. The authors of [15] studied various factors that affect cluster utilization. They noted the importance of locality for DMT jobs and the interference from another job with GPU utilization. The paper in [16] proposed a dynamic DNN training clusters expansion scheme that can add nodes to up-and-running training cluster with minimal performance impact. This showed the favorable extensibility of RAR architectures. Besides, there are some studies about the scheduling strategy and application scenarios of RAR architecture. In order not to destroy the cluster structure, the work in [17] realized multi-task elastic scheduling by setting up a scheduler outside the cluster to control the cluster from the outside. The authors designed a contention-aware resource scheduling algorithm for RAR-based DDL training jobs to minimize the makespan of all RAR-based training jobs in [18]. PACE [19] aimed to maximize the overlap between communication and computation by utilizing a directed acyclic graph, the core of which was a theoretically optimal algorithm of preemptive communication scheduling in modern ML frameworks. In [20], the paper formulated a general online performance optimization framework for RAR-based DMT by decomposing RAR-based training scheduling problem over the the temporal domain and utilizing a generalized virtual graph embedding technique to improve scheduling efficiency. Besides, the paper used field-programmable gate arrays (FPGAs) to accelerate all-reduce operations and data compression to optimize bandwidth utilization for AI training [21]. Additionally, the authors explored the security problem of RAR architecture and found that the malicious workers indeed affect the models performance [22].

Optical networks can provide an ideal infrastructure for distributed computing, which can effectively support large-scale data transmission and high-speed calculations. There have been many studies on deploying distributed computing over optical networks. Traditional DMT is based on PS architecture, which has been used in the form of cloud-edge coordination [23, 24]. In this case, servers are generally deployed on cloud nodes, while workers are deployed on edge nodes. The authors of [25] studied the influence of different partition schemes of Deep Neural Network (DNN) models between cloud and edge on the usage of network resources in dynamic network scenarios. Based on this, the paper proposed an ILP model to provide efficient DMT services in elastic optical networks (EON) by optimizing the partition and distribution of training data, jointly considering computing resources and bandwidth resources [26]. The authors investigated how to deliver distributed ML services in WSS-based all-optical datacenter network with torus topology in [27, 28]. In [27], the paper proposed a two-dimensional matrix-based top-of-rack (ToR), TS and wavelength assignment algorithm which firstly assigned ToRS and TSs for all Ring services one by one and then tuned the wavelength of ToR for different services. Besides, the paper in [28] formulated an ILP model and an efficient heuristic algorithm to minimize the total service execution time and the average lightpath signal loss of distributed ML. Moreover, the paper [29] introduced a method called Super-Cloudlet, which enables dynamic resource management in a federated edge computing system. This approach utilizes commercial optical transport network (OTN) equipment to interconnect neighboring cloudlets through optical circuits, forming a collaborative unit that facilitates shared computing resources during load peaks. This method presents a robust solution for distributed training scheduling supported by OTN.

In addition, there are some scheduling studies on other jobs. From the point of view of customers, this paper [30] focused on the optimal allocation of network-aware resources among data centers in the cloud to solve the budget-optimal joint resource allocation problem, in order to minimize the rental cost of each customer. In [31], the paper considered both static and dynamic scenarios of task scheduling in EON, established an ILP model in static scenarios to minimize the completion time of all jobs, and designed a heuristic algorithm in dynamic scenarios to minimize job blocking when jobs arrive dynamically. Moreover, jobs also can be divided into two types, latency-sensitive and delay-tolerant. The authors of [32] designed and implemented an optimized job scheduling algorithm to minimize the delays for latency-critical applications. The paper [33] proposed a joint frequency and time domain optimization of static scheduling for reservation requests in EON by formulating an ILP model. These authors of [34–36] studied a joint optimization algorithm of multi-job scheduling and light path provisioning for minimizing average completion time and bandwidth occupied in fog or edge computing micro datacenter networks. Differently, [34, 35] optimized the completion time of a single job, while [36] scheduled jobs according to their urgency degree.

We can see that the existing work of scheduling DMT requests seldom consider both computing and network resources with time dimension in CPNs. Furthermore, it is necessary to consider reliability in the scheduling work of RAR-based DMT. Yet, to the best of our knowledge, there is no work on the reliable RAR-based DMT in CPNs.

3. PROBLEM DESCRIPTION

CPNs can be denoted as a directed graph $G(V, E)$, where V and E are the the sets of CPN nodes and fiber links, respectively. CPN nodes consist of computing, electrical, and optical layers, enabling functionalities such as data aggregation and wavelength multiplexing. Besides, time domain can be decomposed into discrete time slots (TSs). The computing resources used in model training are usually GPUs, which are a kind of single-allocation resource, i.e., it cannot be shared at fine granularity among users from the perspective of cluster utilization [15]. Hence, we can regard the computing resources on nodes as computing units (CUs), which can be available to only one DMT request at each TS. Each fiber can carry multiple wavelengths. Given the relatively small size of model gradient data exchanged during the training process, each wavelength can accommodate multiple requests at each TS.

The properties of DMT requests can be collected in advance. A RAR-based DMT request can be denoted as the tuple $R_j(D_j, s_j, t_j^a, t_j^d, \theta_j, r_j)$, where j is the request index. D_j is the amount of training data, s_j is the source node where R_j arrives, t_j^a is the arrival time when R_j reaches the node s_j, and t_j^d is the deadline of completing the training, so R_j can only be deployed between $[t_j^a, t_j^d]$. Besides, the gradient change rate of the loss function can be used to measure the convergence of the model. θ_j is the threshold for the gradient change rate, which means the model training will stop when the gradient change rate of the loss function reaches or falls below this value. A smaller θ_j means that the model needs to achieve a higher level of convergence (potentially representing a higher accuracy) before stopping the training, which may require more iterations. r_j is the reliability requirement.

Fig. 2 illustrates the problem of the RAR-based DMT deployment in a CPN with four computing nodes and five fiber links. The target of our work is to efficiently deploy DMT requests while satisfying reliability requirements. In the illustration, we assume that there are 4 CUs per node, 2 wavelengths (labeled with W) per link, and the same length for all links. A DMT request R_j arrives at its source node B at TS=1 with a certain size of training data D_j. R_j needs to be completed by TS=4. Besides, the training process encompasses computation, transmission, optical-electrical-optical (O-E-O) conversion, and electrical layer multiplexing at various stages. Since the O-E-O conversion time and electrical layer multiplexing time are relatively small compared to the computation time and transmission time, they can be neglected, and this part of the time is not calculated and discussed in detail.

$$T_{com}^j = \frac{D_j \times \lambda}{\nu \times N_j}, \forall j \in R. \tag{1}$$

$$T_{tran}^j = \frac{2(N_j - 1) \times \Delta}{N_j \times \omega}, \forall j \in R. \tag{2}$$

We can quantify how many CUs are required to train R_j within one iteration by Eq. (1), where λ is the amount of computation required per unit of training data, ν is the computing power contained in each CU, and N_j is the number of computing nodes involved in the request. Thus, T_{com}^j represents the time that the subset of training data to be computed using a single CU. Eq. (2) is the transmission time in one iteration, where Δ is the updated gradient parameter size and ω is the data transmission rate.

For the sake of clarity, we consider two different deployment schemes for R_j as examples. Resource-saving deployment schemes would consume fewer resources at the expense of more time, while time-saving schemes consume less time at the expense of resources. The grids in the Fig. 2 represent the allocation of resources. The horizontal axis of the grid represents the number of TSs, while the vertical axis represents the quantity of CUs or wavelengths. If the resource utilization is oriented horizontally, it indicates that the deployment scheme consumes more time. Conversely, if the resource utilization is oriented vertically, it means that the scheme consumes more resources. Specifically, the resource-saving deployment scheme is highlighted in red on the left, which allocates R to a ring with three nodes A, B and D. The training data are equally divided into the three parts, and the partitioned data and a complete copy of the model is assigned to the three nodes involved in the computation, which must include its source node. The other is time-saving scheme, marked in green on the right, which deploys R_j on a ring containing four nodes A, B, C and D and the training data are evenly split among these four nodes. During the model training, participating nodes train the model copy with the partitioned data to get the local updated gradient, and then through scatter-reduce and allgather to obtain global gradient. The model will be trained through a series of iterations until it meets the required θ_j. In the whole training process, the allocation of CU and wavelength on the ring is synchronous and continuous in time. Besides, the allocation of CU needs to satisfy the constraint of non-overlap. During multi-hop transmission, wavelength allocation needs to meet the wavelength consistency constraint.

In detail, both schemes occupy a total of 12 CU-TS blocks, but the resource-saving scheme occupies 12 W-TS blocks, while the time-saving scheme occupies only 4 W-TS blocks. The resource-saving scheme activates 3 CUs and one wavelength of ring ABD between TS=1 and TS=4. The time-saving scheme activates 12 CUs and one wavelength of ring $ABCD$ at TS=1. To conclude, the resource-saving scheme activates fewer computing resources but takes more time to complete the training, while the time-saving scheme spends more computing resources and less time on the training. The resource-saving scheme may require more activation of wavelengths when dealing with a larger number of requests. In addition to the differences in computing resource activation and training time, the two schemes can also be distinguished in terms of reliability. We can calculate the reliability of both solutions through Eq. (3), where N and L are the set of participating computing nodes and links on the selected ring, e_n is the the failure probability of node n, e_l is the failure probability of the link per kilometer (km), ξ_l is the length of link l. Note that in our work, we only consider independent faults, where a fault in one node or link does not result in the failure of other nodes and links. Since the ring of the time-saving scheme passes through 4 nodes and 4 links, while the resource-saving scheme passes through only 3

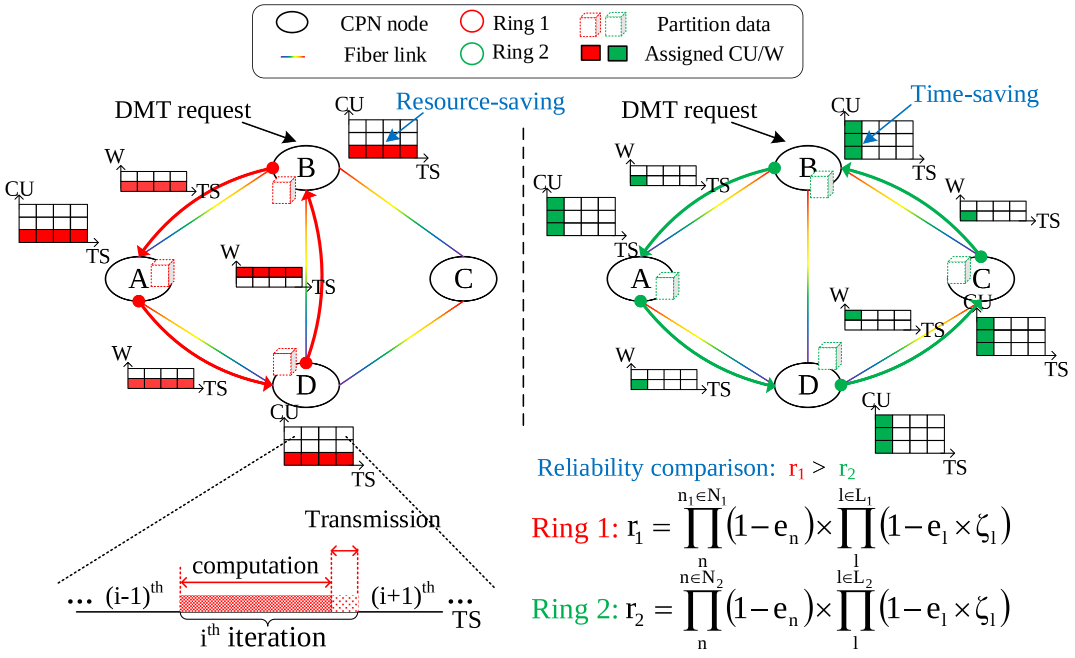
CPN node
Fiber link
Ring 1
Ring 2
Partition data
Assigned CU/W
DMT request
Resource-saving
CU
W
TS
A
B
C
D
Transmission
computation
... (i-1)th
(i+1)th ...
ith iteration
DMT request
Time-saving
Reliability comparison: r1 > r2
Ring 1: $r_1 = \prod_{n}^{n_1 \in N_1}(1-e_n) \times \prod_{l}^{l \in L_1}(1-e_l \times \zeta_l)$
Ring 2: $r_2 = \prod_{n}^{n \in N_2}(1-e_n) \times \prod_{l}^{l \in L_2}(1-e_l \times \zeta_l)$

Fig. 2. Illustration of RAR-based DMT request deployment in a CPN. The resource-saving deployment scheme is highlighted in red (left). The time-saving scheme is highlighted in green (right).

nodes and 3 links, the latter ring size is smaller than the former, so its reliability is better than the former.

$$\gamma = \prod_{n \in N}(1 - e_n) \times \prod_{l \in L}(1 - e_l \times \xi_l) \quad \textbf{(3)}$$

The above is the process of providing service for one DMT request. In fact, CPNs need to provide services for a batch of requests with limited resources. Different scheduling schemes will affect the efficiency of DMT deployment, resource utilization, training time, and reliability. Hence, an appropriate scheduling plan is imperative for service providers to provide reliable and efficient RAR-based DMT services.

4. ILP MODEL

In this section, the solution to the RAR-based DMT is formulated as a static scheduling problem since all requests and the network state are known in advance. Moreover, we use an ILP model to find the optimal solution for the deployment of the RAR-based DMT requests.

A. Input

The model takes as input a topology with limited resources, a predefined time horizon, and the properties of the DMT request batch. Formally:

- **N**: set of computing nodes.
- **L**: set of fiber links.
- $\mathbf{C_n}$: set of CUs on computing node n.
- $\mathbf{W_l}$: set of wavelengths on fiber link l.
- **T**: set of TSs.
- **R**: set of DMT requests.
- **P**: set of candidate rings.
- ν: computing power contained in each CU.
- λ: required computing power per unit of training data.
- ξ_l: the length of link l.
- ω: data rate per wavelength.
- ρ: maximum number of requests each wavelength can accommodate.
- Δ: updated gradient parameter size of each iteration.
- τ: the duration of a TS, in minutes.
- N_j: the number of nodes occupied by the request j.
- $I(\theta_j)$: the number of training iterations of the request j.
- μ_k: the number of nodes on the ring k.
- γ_k: the reliability of the ring k.
- T^j_{com}: the computing time of the request j in one iteration.
- T^j_{tran}: the transmission time of the request j in one iteration.
- χ: a large number.

B. Variables

The model uses the following variables.

- $\eta^j \in \{0,1\}$: Boolean variable that assumes value 1 if DMT request j is successfully deployed, 0 otherwise.
- $C^j_{t,n,c} \in \{0,1\}$: Boolean variable that assumes value 1 if CU c on node n is used by request j at TS t, 0 otherwise.
- $f^j_{t,l,b} \in \{0,1\}$: Boolean variable that assumes value 1 if wavelength b on link l is used by request j at TS t, 0 otherwise.
- $S^j_k \in \{0,1\}$: Boolean variable that assumes value 1 if ring k is used by request j, 0 otherwise.
- $O^j_n \in \{0,1\}$: Boolean variable that assumes value 1 if request j is deployed on node n, 0 otherwise.
- $O_{n,c} \in \{0,1\}$: Boolean variable that assumes value 1 if CU c of node n is deployed, 0 otherwise.
- $O_{l,b} \in \{0,1\}$: Boolean variable that assumes value 1 if wavelength b of link l is deployed, 0 otherwise.
- $O^j_{n,c} \in \{0,1\}$: Boolean variable that assumes value 1 if CU c of node n is occupied by request j, 0 otherwise.
- $O^j_{l,b} \in \{0,1\}$: Boolean variable that assumes value 1 if wavelength b of link l is occupied by request j, 0 otherwise.
- $O^j_{n,t} \in \{0,1\}$: Boolean variable that assumes value 1 if node n is occupied by request j at TS t, 0 otherwise.
- $O^j_{l,t} \in \{0,1\}$: Boolean variable that assumes value 1 if link l is occupied by request j at TS t, 0 otherwise.
- $y^j_{k,n} \in \{0,1\}$: Boolean variable that assumes value 1 if node n on the ring k is occupied by request j, 0 otherwise.
- $\Gamma^j \in \mathbb{R}^+$: Continuous variable that indicates the reliability of request j deployed in a certain ring.
- $ON^j_n \in \mathbb{Z}^+$: Integer variable that indicates the number of CUs on node n is used by request j.
- $Ts^j_n \in \mathbb{Z}^+$: Integer variable that indicates the start deployment time of request j at node n.
- $Te^j_n \in \mathbb{Z}^+$: Integer variable that indicates the deployment departure time of request j at node n.
- $Ts^j_l \in \mathbb{Z}^+$: Integer variable that indicates the start deployment time of request j on link l.
- $Te^j_l \in \mathbb{Z}^+$: Integer variable that indicates the deployment departure time of request j on link l.

C. Objective

The goal of the ILP model is to accommodate as many DMT requests as possible while respecting resource and reliability constraints. Hence, the primary objective is to minimize the blocking rate of batch requests, i.e., $1-\frac{1}{|R|}\sum \eta^j$. In practical network scenarios, service providers may focus on different performance metrics, such as hardware cost, and computation time. Thus, we formulate a comprehensive objective function. Let ζ represent other performance metrics. The first addend is in the canonical range of $[0,1]$. α is used to adjust the value of the second addend, aiming to contain the value also in the range $[0,1]$. The general objective function is as follows:

$$Minimize \quad (1-\frac{1}{|R|}\sum_{\forall j}\eta^j)+\alpha\times\zeta \tag{4}$$

To compare and verify the effectiveness of our model, four optimization schemes are devised by varying the performance metric that ζ represents: *(1)* The *MinCU* scheme could activate the least computing resources, where $\zeta_1 = \sum O_{n,c}$ and $\alpha_1 = 1/\sum|C_n|$; *(2)* The *MinW* scheme could use the least number of wavelength resource, where $\zeta_2 = \sum O_{l,b}$ and $\alpha_2 = 1/\sum|B_l|$; *(3)* The *MinT* scheme could minimize the training time across all requests, where $\zeta_3 = \sum Te^j_{s_j}/t^d_j$ and $\alpha_3 = 1/|R|$; and *(4)* The *MaxR* scheme could maximize the reliability across all requests, where $\zeta_4 = \sum(1-\Gamma^j)$ and $\alpha_4 = 100/|R|$.

D. Constraints

D.1. Task Compliment Constraints

$$\sum_{\forall t,\forall c} C^j_{t,n,c} \geq I(\theta_j)\times\frac{T^j_{com}\times O^j_n + T^j_{tran}\times ON^j_n}{\tau},$$
$$\forall j\in R, n\in N. \tag{5}$$

$$\sum_{\forall l,\forall b} O^j_{l,b} \geq S^j_k\times\mu_k, \forall j\in R, k\in P. \tag{6}$$

$$\sum_{\forall k} S^j_k = \eta^j, \forall j\in R. \tag{7}$$

Eq. (5) and Eq. (6) allocate to each request the computing resources and wavelength resources it needs. Eq. (7) ensures that only one ring will be selected for each request.

D.2. Resource Constraints

$$\sum_{\forall j} C^j_{t,n,c} \leq 1, \forall n\in N, t\in T, c\in C_n. \tag{8}$$

$$\sum_{\forall j} f^j_{t,l,b} \leq \rho, \forall l\in L, t\in T, b\in B_l. \tag{9}$$

$$\sum_{\forall b} f^j_{t,l,b} \leq 1, \forall j\in R, l\in L, t\in T. \tag{10}$$

Eq. (8) limits that each CU can only be assigned to one request within a TS. Besides, each wavelength can accommodate ρ requests within a TS limited by Eq. (9) and Eq. (10).

$$\sum_{\forall k} S^j_k\times\mu_k \geq N_j\times\eta^j, \forall j\in R. \tag{11}$$

$$\Gamma^j = \sum_{\forall k} S^j_k\times\gamma_k. \tag{12}$$

$$\Gamma^j \geq \eta^j\times r_j, \forall j\in R. \tag{13}$$

Equations (11–13) ensure that the amount of computing nodes and the reliability of the selected ring meet the request requirement, where γ_k can be calculated by Eq. (3).

$$O^j_{n,c} \leq \sum_{\forall t} C^j_{t,n,c} \leq O^j_{n,c} \times \chi, \forall j \in R, n \in N, c \in C_n. \tag{14}$$

$$ON^j_n = \sum_{\forall c} O^j_{n,c}, \forall j \in R, n \in N. \tag{15}$$

$$O^j_n \leq ON^j_n \leq O^j_n \times \chi, \forall j \in R, n \in N. \tag{16}$$

$$O^j_{s_j} = \eta^j, \forall j \in R. \tag{17}$$

Eq. (14) is the definition of $O^j_{n,c}$. Eq. (15) calculates the number of CUs occupied by request R_j on each node. Eq. (16) defines the variable O^j_n, which indicates whether node n provides computing resources for R_j. Besides, Eq. (17) means that the source node s_j must participate in the training of R_j.

$$O^j_{l,b} \leq \sum_{\forall t} f^j_{t,l,b} \leq O^j_{l,b} \times \chi, \forall j \in R, l \in L, b \in B_l. \tag{18}$$

$$\sum_{\forall b} O^j_{l,b} = 0,\ if\ S^j_k = 1,\ \forall j \in R, k \in P, l \notin k. \tag{19}$$

$$S^j_k \leq \sum_{\forall b} O^j_{l,b}, \forall j \in R, k \in P, l \in k. \tag{20}$$

Equations (18-20) represent the relationship between rings and links, ensuring that the wavelength resources for the request are provided by the links on the selected ring.

$$y^j_{k,n} = 0, \forall j \in R, k \in P, n \notin k. \tag{21}$$

$$y^j_{k,n} \leq S^j_k, \forall j \in R, k \in P, n \in N. \tag{22}$$

$$y^j_{k,n} \leq O^j_n, \forall j \in R, k \in P, n \in N. \tag{23}$$

$$\sum_{\forall n} y^j_{k,n} = S^j_k \times N_j, \forall j \in R, k \in P. \tag{24}$$

$$\begin{aligned} y^j_{k,n} \geq |O^j_{l_1,b_1} - O^j_{l_2,b_2}|,\ if\ S^j_k = 1, \forall j \in R, \\ k \in P, n \in k, l_1, l_2 \in k, b_1 \in B_{l_1}, b_2 \in B_{l_2}. \end{aligned} \tag{25}$$

Equations (21-23) represent the relationship between rings and nodes. Eq. (24) ensures that the computing resources are provided by N_j nodes on the selected ring. In addition, Eq. (25) ensures that the wavelength meets the wavelength consistency constraints for multi-hop transmission, where l_1 and l_2 are the links associated with node n on the ring k.

$$O_{n,c} \leq \sum_{\forall j} O^j_{n,c} \leq O_{n,c} \times \chi, \forall n \in N, c \in C_n. \tag{26}$$

$$O_{l,b} \leq \sum_{\forall j} O^j_{l,b} \leq O_{l,b} \times \chi, \forall l \in L, b \in B_l. \tag{27}$$

Eq. (26) and Eq. (27) are used to calculate whether the CU c of a node or the wavelength b of a link is occupied.

D.3. Time Constraints

$$O^j_{n,t} \leq \sum_{\forall c} C^j_{t,n,c} \leq O^j_{n,t} \times \chi, \forall j \in R, n \in N, c \in C_n. \tag{28}$$

$$O^j_{l,t} \leq \sum_{\forall b} f^j_{t,l,b} \leq O^j_{l,t} \times \chi, \forall j \in R, l \in L, b \in B_l. \tag{29}$$

$$Ts^j_n \leq t \leq Te^j_n,\ if\ O^j_{n,t} = 1,\ \forall j \in R, n \in N, t \in T. \tag{30}$$

$$Ts^j_l \leq t \leq Te^j_l,\ if\ O^j_{l,t} = 1,\ \forall j \in R, l \in L, t \in T. \tag{31}$$

Equations (28-31) define the start time and end time for allocating computing and wavelength resources to training requests.

$$\begin{aligned} \sum_{t+2 \leq t_1 \leq |T|} C^j_{t_1,n,c} \leq 1(1 - C^j_{t,n,c} + C^j_{t+1,n,c}) \times \chi, \\ \forall j \in R, n \in N, c \in C_n, t \leq |T| - 2. \end{aligned} \tag{32}$$

$$\begin{aligned} \sum_{t+2 \leq t_1 \leq |T|} f^j_{t_1,l,b} \leq 1(1 - f^j_{t,l,b} + f^j_{t+1,l,b}) \times \chi, \\ \forall j \in R, l \in L, b \in B_l, t \leq |T| - 2. \end{aligned} \tag{33}$$

$$\sum_{\forall t} 0^j_{n,t} = Te^j_n - Ts^j_n + 1,\ if\ O^j_n = 1,\ \forall j \in R, n \in N. \tag{34}$$

$$\sum_{\forall t} 0^j_{l,t} = Te^j_l - Ts^j_l + 1,\ if\ S^j_k = 1,\ \forall j \in R, k \in P, l \in k. \tag{35}$$

Equations (32-35) ensure the request can continuously occupy computing and wavelength resources within the time window $[Ts^j_{s_j}, Te^j_{s_j}]$.

$$O^j_{n,t} = 0, \forall j \in R, n \in N, t \notin [t^a_j, t^d_j]. \tag{36}$$

$$O^j_{l,t} = 0, \forall j \in R, l \in L, t \notin [t^a_j, t^d_j] \tag{37}$$

$$t^j_a \leq Ts^j_{s_j} \leq Te^j_{s_j} \leq t^j_d,\ if\ \eta^j = 1,\ \forall j \in R. \tag{38}$$

$$Ts^j_n = Ts^j_{s_j},\ Te^j_n = Ts^j_{s_j},\ if\ O^j_n = 1,\ \forall j \in R, n \in N. \tag{39}$$

$$Ts^j_l = Ts^j_{s_j},\ Te^j_{s_j} = Te^j_l,\ if\ S^j_k = 1,\ \forall j \in R, k \in P, l \in k. \tag{40}$$

$$S^j_k \leq O^j_{l,t},\ if\ O^j_{s_j,t} = 1,\ \forall j \in R, k \in P, l \in k, t \in T. \tag{41}$$

Equations (36-41) guarantee that nodes and links can provide computing and wavelength resources at the same time within the request time window $[t^j_a, t^j_d]$.

5. HEURISTIC ALGORITHM

The complexity of the ILP model is influenced by the number of requests and the size of the topology. However, due to unacceptable running times in scenarios involving large topologies, we design a RAR-based DMT deployment algorithm referred to as RDDA, to provide reliable RAR-based DMT services within a large network topology. The difference in RDDA among the four schemes is mainly in the candidate rings sorting method: *(1)* The *MinCU* scheme first sorts all combinations of nodes in candidate rings in ascending order based on the number of CUs to be activated within the given time window $[t^a_j, t^d_j]$. When the number is the same, it further sorts them in ascending order based on the number of contained nodes. *(2)* In the *MinW* scheme, candidate rings are firstly sorted in ascending order based on the number of wavelengths required to be activated within the time window $[t_j^a, t_j^d]$. In case of a tie in the activated wavelength number, a secondary sorting is performed in ascending order based on the number of links included in each ring. *(3)* The *MinT* scheme prioritizes candidate rings based on the time it takes to provide the required number of CUs. The faster, the higher the priority. *(4)* The *MaxR* scheme sorts the reliability of candidate rings in descending order, prioritizing those with higher reliability for iterating.

Algorithm 1. A RAR-based DMT deployment algorithm

```
Input: batch DMT requests and network state
Output: detail of all request deployments
for each DMT request $R_j(D_j, s_j, t_j^a, t_j^d, \theta_j, r_j)$ do
    get the set of rings $SR$ containing source node $s_j$
    for each ring $\in CC$ do
        calculate the ring reliability $\gamma$ according to Eq. (3)
        if $r_j \leq \gamma$ then
            add the ring into the set of candidate rings $CR$
        end if
    end for
    for each ring $\in CR$ do
        add all node combinations on this ring containing $s_j$ to the candidate node combination set $NC$
    end for
    sort all combinations in $NC$ according to the respective scheme
    for each combination $\in NC$ do
        calculate the number of CUs needed for this combination according to Eq. (1)
        if this combination can provide the required CUs and wavelength within $[t_j^a, t_j^d]$ then
            allocate CUs and wavelength for the request $R_j$
            Break
        else
            Continue
        end if
    end for
    if the request is not deployed then
        record the request into the set of blocking requests
    end if
    record the details of $R_j$ deployment
end for
recorded network state
return network state and deployment details for all requests
```

The RDDA jointly considers computing and wavelength resources in the temporal dimension. The RDDA is also designed for static scenarios where all requests and network status can be obtained in advance. DMT requests are scheduled based on their arrival and deadline times. Requests that arrive earlier are processed first. In cases of simultaneous arrivals, those with closer deadlines are prioritized for processing. When processing a request R_j, RDDA begins by searching for rings within the network that include the source node s_j. Calculate the reliability of these rings, and if they are not less than the required r_j, they are added to the candidate ring set. A candidate ring may have multiple node combinations, and RDDA scores and ranks each combination. Different schemes employ distinct scoring and ranking methods. Iterate through these combinations in sequence. Calculate the computing resource required for each node, and verify whether all nodes in the combination can consistently provide these computing resources over a continuous TSs within the time window $[t_a^j, t_d^j]$. If this condition is satisfied, proceed to verify whether the links on the ring can offer the required wavelength resources during the corresponding TSs. If wavelength resources are also sufficient, deploy the request, halt the iteration and move on to the next request. Otherwise, the search continues with other combinations. If all candidate combinations are iterated without successfully deploying R_j, R_j is blocked, and the process proceeds to the next task. Resource allocation details for all requests are recorded, and relevant metrics are computed. The complexity of the resource allocation algorithm depends on factors such as network size, the quantity of computing resources in each node, and the number of wavelengths per link.

6. SIMULATION RESULTS

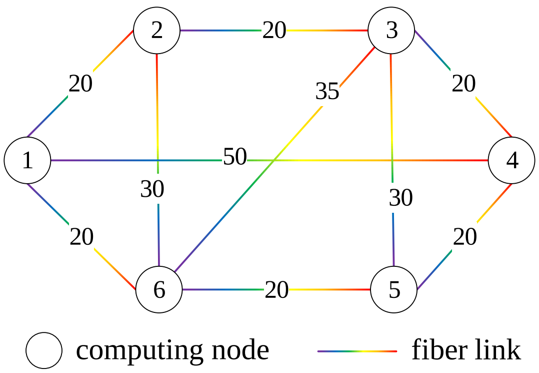

Fig. 3. 6-node topology of ILP model considered in this paper

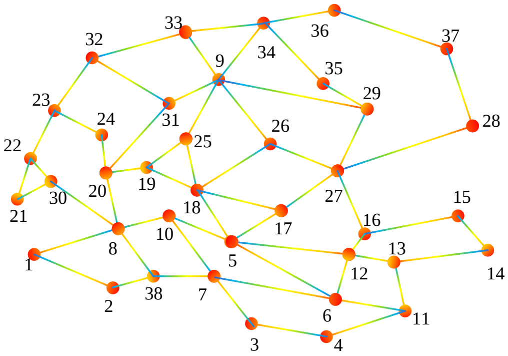

Fig. 4. 38-node topology of heuristic algorithm considered in this paper

In this section, we use a small topology in Fig. 3 and a large topology in Fig. 4 to validate our model. In Fig. 3, the topology comprises 6 nodes and 10 links, with link lengths ranging from 20 km to 50 km. Each node has 8 to 12 CUs with each CU having a computational power of 20 × 10 $TFLOPS$. Each link is provisioned with 10 wavelengths with the capacity to concurrently accommodate three requests. Besides, the large topology described in Fig. 4 consists of 38 nodes and 59 links, with each link having a length of 20 km. Each node has 160 to 320 CUs, with each CU capable of 10 TFLOPS of computing power [37]. There are 20 wavelengths per link with a data rate of 10 Gbit/s per wavelength. The time dimension for the small topology consists of 12 TSs, while the large topology comprises 48 TSs, with each TS being 30 minutes. Note that, we can change the time granularity of τ depending on different model sizes and computing power, such as hours or days. Moreover, the number of updated gradients per iteration is influenced by the model structure and the training data size. Based on previous studies [26], we can simplify the updated gradient size to about 1 GB according to the training data size in our simulation. In addition, the upper bound on iterations can be simplified as $I(\theta_j) = \beta \times \log(1/\theta_j)$, where β depends on the data size and condition number of the local problem in [38, 39]. This upper bound indicates that beyond it, additional iterations might not further improve the performance of the model, and may even cause a decrease in performance. In our work, we adopt the upper bound as the number of training iterations, and β is set to D_j. Parameters of DMT requests are randomly generated from a uniform distribution within the range listed in Table 1. Additionally, our ILP model is implemented in Python and Gurobi 9.5.1 while RDDA is implemented in Python with 3.2 GHz CPU and 16 GB of RAM.

In order to reduce the complexity of the ILP model, we simplify the formulation by adopting the following constraints: *(1)* Fix the number of computing nodes required at 3 or 4, that is $N_j \in \{3,4\}$; *(2)* Preset several rings containing 3 to 6 nodes, i.e., $\mu_k \in [3,6]$, from which the requests must select the appropriate ring; *(3)* The arrival time and deadline of requests are concentrated within the first 3 TSs and the last 3 TSs. Here, the value 3 is merely illustrative and can be replaced with other suitable numbers in practice. The approach is designed to prevent situations where requests cannot be scheduled due to the imposition of an impractical training time window; *(4)* The current model only allows ring topologies due to Equations (21-24). If linear topologies need to be considered, the mentioned constraints need to be modified.

Table 1. Simulation Parameters

Parameter	Value	Parameter	Value
C_n	$[8,12],[160,320]$	$\nu(TFLOPS)$	$20 \times 10, 10$
λ	10^{15}	$\xi_l(km)$	$[20,50],\{20\}$
W_l	$\{10\},\{20\}$	$\omega(Gbit/s)$	10
$\Delta(GB)$	1	χ	10000
$\|T\|(TS)$	12,48	$\tau(mins)$	30
e_n	10^{-6}	$e_l/[km]$	10^{-5}
$D_j(GB)$	$[100,200]$	r_j	$[0.99,0.999]$
s_j	$[1,6],[1,38]$	θ_j	$[0.1,0.4]$
t_j^a	$[1,3],[1,10]$	t_j^d	$[10,12],[39,48]$

The metrics we are concerned with include blocking rate, activated CU ratio, activated wavelength ratio, average ahead-of-time (AOT) ratio, and average reliability. Note that the activated CU ratio refers to the number of activated CUs divided by the total number of CUs. Similarly, the activated wavelength ratio is the ratio of the number of activated wavelengths to the total number of wavelengths. AOT ratio represents the ratio of the time a DMT request is completed ahead of its deadline set by the request, which is defined in Eq. (42).

$$AOT\ ratio = \frac{1}{|R|} \sum_{\forall j} \frac{(t_d^j - Te_{s_j}^j) \times \eta^j}{t_d^j} \tag{42}$$

For the small topology, we simulate our ILP model within 12 TSs, with the total number of requests ranging from 2 to 12 (in step of 2). For the large topology, we utilize the RDDA within 48 TSs, with the total number of requests ranging from 50 to 500 (in steps of 50). To reduce the impact of data randomness, we perform simulations by randomly generating multiple sets of random DMT request attributes for each batch size. Each set is simulated under four schemes. Subsequently, the results are averaged for comparison and analysis.

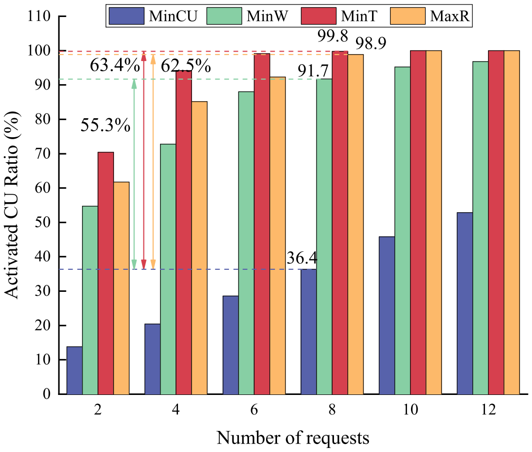

Fig. 5. Activated CU ratio of ILP model in small topology.

In the simulation for the small topology, there is no instance of request blocking. Fig. 5 shows the results of the activated CU ratio of small topology. As the number of requests grows, more CUs are needed. When the number of DMT requests is 10, *MinT* and *MaxR* schemes activate nearly all CUs. We can also observe that the *MinCU* scheme needs the least computing resources. For example, when 8 requests need to be deployed, the activated CU ratio of *MinCU* scheme is 36.4%, which is 55.3%, 62.5%, and 63.4% lower than that of *MinW*, *MaxR*, and *MinT* scheme, respectively. The *MinW* scheme achieves the second lowest. Since node and link resources are allocated simultaneously, *MinW* tends to reuse the wavelength resources, which is beneficial to the reuse of node computing resources. On the other hand, *MinT* scheme prefers to activate new CU to complete training as early as possible, which directly leads to the consumption of more CUs, so *MinT* consumes the most CUs.

The activated wavelength ratio of the four schemes are shown in Fig. 6. The wavelength activation ratio increases with the number of requests. *MinW* scheme activates the fewest wavelengths, followed by *MaxR*, *MinT*, and *MinCU* schemes in that order. When the number of requests is 8, the activated wavelength ratio of *MinW* scheme is about 7%, *MaxR* is 22.6%, *MinT* is 23.3% and *MinCU* is 24.3%. The deployment requests of *MinW* scheme have two principles: one is to pass through as few links as possible, and the other is to reuse activated wavelength as much as possible. *MaxR* scheme prefers rings with fewer links or shorter links. Hence, MaxR scheme is beneficial to saving wavelength resources in terms of multiplexing and reducing the number of links occupied.

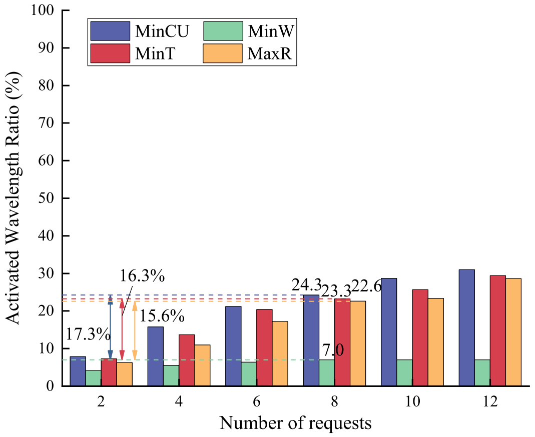

Fig. 6. Activated wavelength ratio of ILP model in small topology.

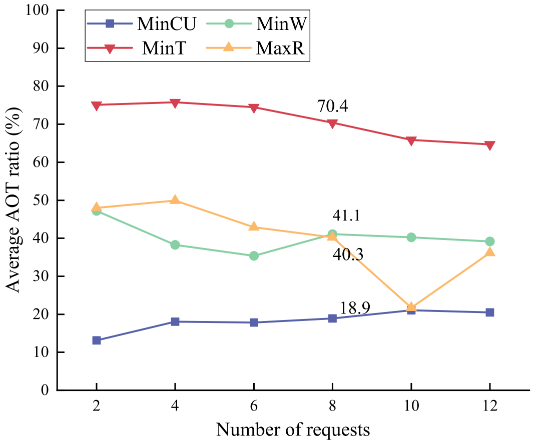

Fig. 7. Average AOT ratio of ILP model in small topology.

The results of the average AOT ratio of the four schemes are shown in Fig. 7. The overall trend in AOT ratio for *MinT* scheme is downward. Because when the number of requests is few, *MinT* scheme deploys requests as soon as possible by activating new CUs. However, computing resources are limited, and with the increase in the number of requests, not all requests can be deployed immediately. This makes some requests have to wait for other requests to complete before there are available resources to use. When the number of requests is 8, the average AOT ratio of *MinT* scheme is about 51.5%, 30.1%, and 29.3% higher than the other three schemes. Compared to other schemes, *MinT* spends more resources to reduce training time. The other three schemes prioritize the reuse of various resources, resulting in longer training times and smaller AOT ratio. These three schemes fluctuate within a certain range. To simplify the complexity of ILP model, the DMT requests fix the number of nodes participating in the computation, so for *MaxR* scheme, it only considers the number and length of links passed through in the ring. Neither *MinW* scheme nor *MaxR* scheme takes into account the efficiency of computing resource utilization. Moreover, since the computation time dominates during the training process, while transmission is relatively small, the time cost of training it takes to train a request is primarily influenced by the allocation of how many CUs are used for computation. Therefore, the AOT ratio for *MinCU* scheme is the smallest.

Fig. 8 shows the blocking performance of RDDA in the large-scale topology. We can see that requests start to be blocked for both *MinW* scheme and *MaxR* schemes when the request number is set to 200. In comparison, the threshold number of requests blocking is 300 and 450 for *MinCU* and *MinT* scheme, respectively. The main reason for the blocking is a shortage of computing resources. Besides, with the constraint that the source node must participate in computing if the source node cannot provide sufficient computing resources within its time window, the request will be blocked directly. In addition, blocking may also result from the high-reliability requirements of the requests, which can only choose the small rings, and the training data can only be split to a small number of computing nodes, which can not provide sufficient resources. For *MinCU* and *MinT* schemes, blocked requests are usually at low rank. The primary reason for their blocking is the inherently limited number of CUs at their source nodes, which are already occupied by previous requests, thus preventing them from completing their training before their respective deadlines. For a DMT request, the training data is evenly distributed to the computing nodes on the ring. To ensure that the computing nodes can compute synchronously, the number of CUs provided to the request by nodes in each TS is the same. Hence, nodes with fewer available CUs in a ring will be the primary limiting factor for request allocation. Compared to the *MinT* scheme, in the *MinCU* scheme, the duration of each request is longer. *MinCU* scheme increases the likelihood of depleting the resources of nodes inherently possessing few resources, exacerbating this bottleneck effect. Hence, the blocking rate of the *MinCU* scheme is higher than that of the *MinT* scheme. For *MinW* and *MaxR* schemes, blocking is not concentrated on low-ranked requests, and requests at intermediate ranks may also be blocked. This is primarily because these requests prioritize selecting smaller-sized rings for deployment, resulting in a concentrated deployment of requests. This concentration of deployment can exhaust resources in certain nodes. Consequently, requests originating from these nodes are directly blocked.

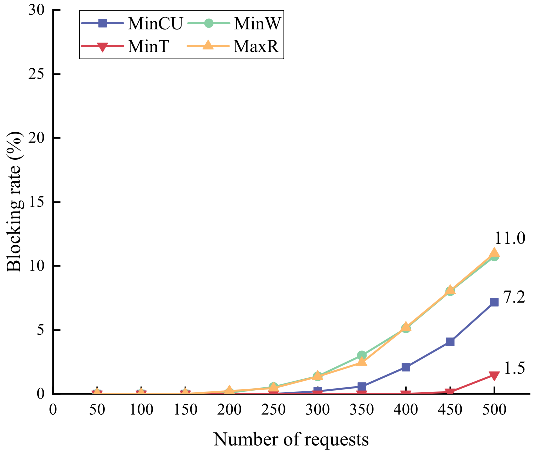

Fig. 8. Blocking rate of RDDA in large topology.

Fig. 9 shows the results of the activated CU ratio of large topology. The lowest activation cost of CU is achieved by the *MinCU* scheme, followed by the *MinW* scheme or the *MaxR* scheme, with the highest cost resulting from the *MinT* scheme. The trend of Fig. 9 is basically consistent with that of Fig. 5. The activated CU ratio of *MaxR* scheme in RDDA is slightly lower than that of *MinW* scheme when the number of requests is more than 200, while the activated CU ratio of *MaxR* scheme of the ILP model is larger than that of *MinW* scheme. The reason for the difference is that in the ILP model, to reduce model complexity, the number of computing nodes required by the request is fixed, while it is not fixed in RDDA. The *MaxR* scheme tends to deploy requests with fewer nodes for higher reliability. Compared to *MinW* and *MaxR* schemes, the *MinT* scheme tends to utilize more computing nodes to complete the training. This is one of the primary reasons for their lower CU activation rates. Additionally, the higher blocking rates of the *MinW* and *MaxR* schemes contribute to their lower CU activation rates. We can also observe that when the number of requests is 100, the CU activation ratio for the *MinCU* scheme are lower than other three schemes about 57%, 59%, and 70%. When the number of requests is 500, the *MinCU* scheme is approximately 8% lower than *MaxR*, 10% lower than *MinW*, and 11% lower than *MinT* scheme. This indicates that as the batch size of DMT requests increases, the utilization of CUs in the network approaches saturation.

The activated wavelength ratio of the four schemes are shown in Fig. 10. The activated wavelength ratio increases with the growth in the number of requests. The increasing order of wavelength consumption is as follows: *MinW*, *MaxR*, *MinT*, and *MinCU* scheme. When the number of requests is 100, we can see that the wavelength activation ratio of the *MinW* scheme is around 5%, the *MaxR* scheme is 6%, the *MinT* scheme is 6.6%, and the *MinCU* scheme is 9.6%. When the number is 500, the wavelength activation rate of the *MinCU* scheme is the highest, only reaching 38%. This indicates that the wavelength resources in the network are sufficient. The primary limiting factor for deploying requests in the network is computational resources, not wavelength resources. In addition, the *MaxR* scheme is close to the *MinW* scheme, since smaller ring sizes are given higher priority in the *MaxR* scheme, effectively reducing wavelength consumption. From Fig. 10, it is evident that when there are a large number of requests, the activated wavelength

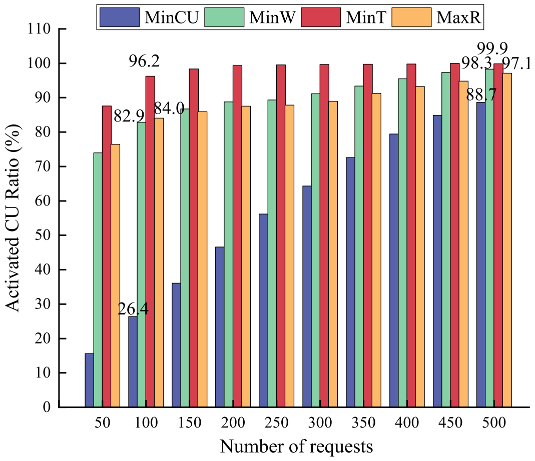

Fig. 9. Activated CU ratio of RDDA in large topology.

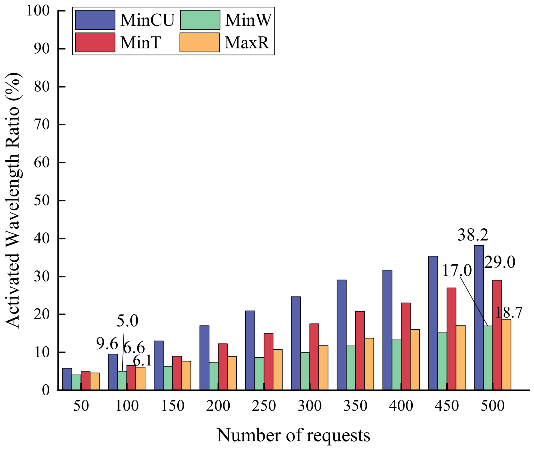

Fig. 10. Activated wavelength ratio of RDDA in large topology.

ratio of the *MinCU* scheme is significantly higher than the other schemes. The reasons are as follows: RAR-based DMT characteristics dictate a small amount of updated data to be transmitted, requiring only one wavelength per request. In comparison, the *MinCU* scheme, with its longer request duration, requires more wavelength resources than other schemes.

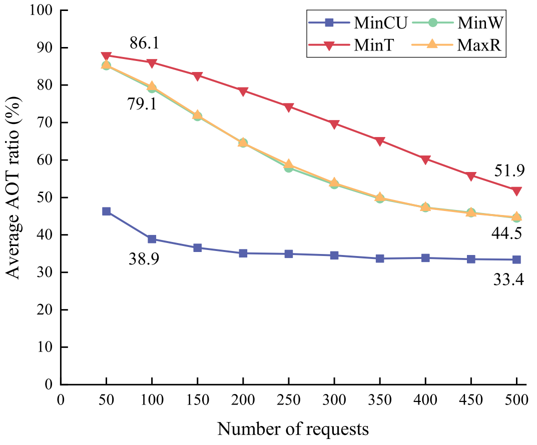

Fig. 11. Average AOT ratio of RDDA in large topology.

The results of average AOT ratio in the large topology of the four schemes are shown in Fig. 11. The AOT ratio for four schemes generally decrease as the number of requests increases. This is attributed to the fact that when the batch size of DMT requests is large, the utilization of computing resources becomes saturated, leaving fewer available CUs to expedite training. Besides, the AOT ratio curves for *MaxR* and *MinW* schemes are close to each other, showing that these two schemes share a similar approach to selecting rings. The AOT ratio of the *MinCU* scheme is the lowest, indicating the higher time cost of training compared to the other schemes. Consequently, as the number of requests increases, the reduction in the AOT ratio for the *MinCU* scheme is not very significant. Therefore, its curve exhibits the flattest slope among all schemes.

In Fig. 12 and Fig. 13, the bar charts represent the reliability gain results, and the dotted lines represent the reliability results. Reliability gain can be obtained by calculating the difference between the required and the achieved reliability, i.e., $(R_{ring} - R_j)/R_j$. We can observe that the reliability gain of all schemes is positive, indicating that all four schemes meet the reliability requirements. The performance differs between the ILP model and RDDA. In the small topology, the reliability curves of *MinCU* and *MinT* schemes are interleaved, while in the large topology, the reliability of the *MinCU* scheme is higher than that of the *MinT* scheme. The reasons are as follows. The calculation of reliability comprehensively takes into account both the nodes engaged in computations and the links traversed. However, to simplify complexity, the ILP model fixes the number of nodes participating in computation, thereby making the reliability of four schemes distinguishable solely through the links traversed. *MinCU* and *MinT* schemes exhibit no particular preference concerning links when selecting rings, resulting in minor differences in reliability between the two. In the large topology, the *MaxR* scheme has the highest reliability, followed by the *MinW* scheme, while the *MinT* scheme has the lowest reliability. Since in RDDA, the number of nodes involved in request computations remains flexible, *MinT* scheme tends to utilize more computing nodes to expedite training completion, while *MinCU* leans towards using fewer nodes. Consequently, these four schemes exhibit variations in reliability.

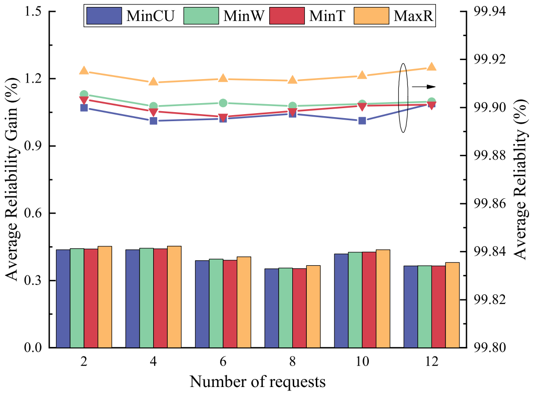

Fig. 12. Average reliability and gain of ILP model in small topology.

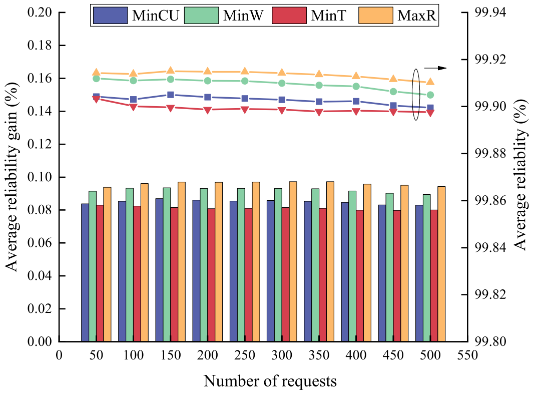

Fig. 13. Average reliability and gain of RDDA in large topology.

We compared the ILP model and the RDDA at the 6-node network for a number of requests ranging from 2 to 12. Results for the ILP are shown as solid lines and results for the RDDA are shown as dashed lines. The blocking rate is zero for both algorithms. The four schemes for the ILP model and RDDA almost show the same trend. We can see from Fig. 14 to Fig. 16 that the ILP model can obtain optimal performance in the corresponding metrics. For the CU activation ratio, the *MinCU* scheme of the ILP is 1.85% to 16.07% lower than that of the RDDA. For the wavelength activation ratio, the *MinW* scheme of the ILP is 0.85% to 2.6% lower than that of RDDA. For reliability, the *MaxR* scheme of the RDDA performs as well as the ILP while the other performances are not guaranteed. For example, the *MinW* scheme of the ILP does require fewer wavelength resources than the RDDA but may require more computing resources than the RDDA. Besides, we can find there are some differences between the ILP model and RDDA in the result of activated wavelength ratio in Fig. 15. Because each wavelength can be used by several requests, except for *MinW* scheme, the other three schemes of RDDA occupy wavelengths on the principle of first-fit, which will fill up one wavelength before using a new wavelength, while the other three of ILP occupy the wavelengths randomly subject to satisfying the constraints. Therefore, except for the *MinW* scheme, the activated wavelength ratio of RDDA performs better than that of ILP. This could be mitigated in the ILP by including an additional term to the objective, minimizing active wavelengths.

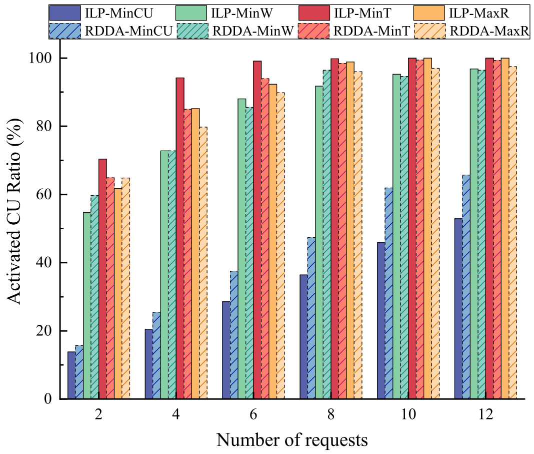

Fig. 14. Activated CU ratio of ILP model and RDDA in small topology.

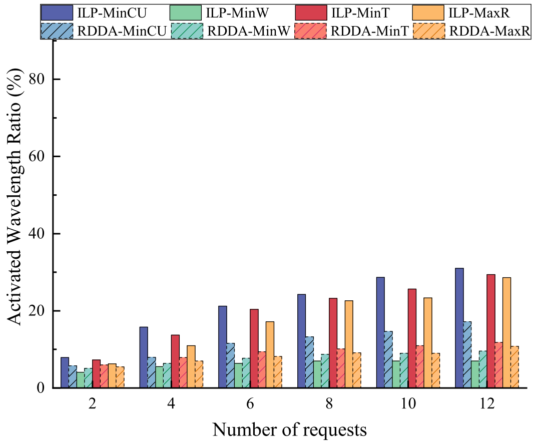

Fig. 15. Activated wavelength ratio of ILP model and RDDA in small topology.

As indicated in Table 2, the ILP model's running time experiences exponential growth with the expansion of request quantities. In comparison, RDDA demonstrates superior performance when dealing with large-scale topologies. Consequently, for scenarios where the number of requests surpasses a certain threshold, the ILP model may face challenges in providing solutions within a reasonable time, primarily owing to its high time complexity. In such contexts, RDDA becomes a more feasible choice.

7. CONCLUSION

In this paper, we investigate how to provide reliable and efficient DMT services in CPNs. We introduce the concept of ring reliability to take into account both node and link reliability in the RAR architecture, which is one of the criteria for selecting a candidate ring in the DMT requests scheduling process. To provide an efficient and reliable DMT service, we formulate an ILP model and a heuristic algorithm to schedule DMT requests in CPNs. In practical network scenarios, different service providers may consider different performance metrics. Hence, we design four optimization schemes, namely *MinCU*, *MinW*, *MinT*, and *MaxR*, to seek the optimal solutions for achieving minimum CU activation, minimum wavelength activation, minimum training time, and maximum reliability, respectively.

The results of both *MinW* and *MaxR* schemes are similar in terms of AOT ratio, reliability, and blocking performance since they share a common component in ring selection decisions, that is, the ring with a smaller number of links is prior. No matter whether it is small topology or large topology, with the increase in the number of requests, the gap of CU activation ratio of the four schemes is getting smaller, while the gap of wavelength activation ratio is getting larger. It shows that in our network scenarios, the main limiting resources for request deployment are computing resources, and CU utilization is close to saturation when the number of requests is 500, which can also be verified by the results of AOT ratio of the large topology. *MinCU*, *MinW*, and *MaxR* schemes all favor centralized deployment to reduce resource activation ratio or achieve higher reliability. Conversely, the *MinT* scheme prefers decentralized deployment to utilize more resources for accelerated training completion. Based on the simulation results of a large topology, when the number of requests is 500, the blocking rate of the *MinT* scheme is about 6% and 10% lower than the other three schemes. Hence, centralized deployment easily leads to intense resource competition and diminishes request deployment success rates, whereas decentralized deployment enables more requests to be deployed. In conclusion, all four schemes meet the reliability requirements and have achieved optimal perfor-

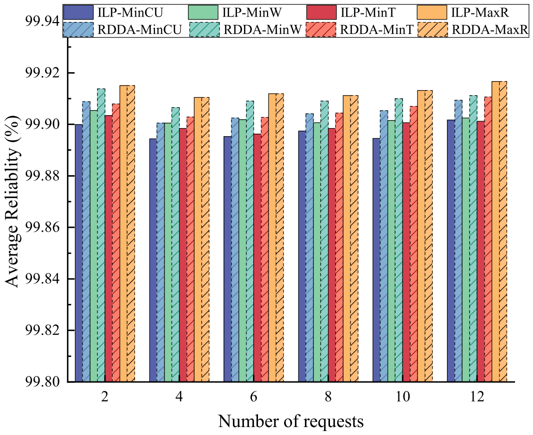

Fig. 16. Average reliability of ILP model and RDDA in small topology.

Table 2. Average Running Time (Second)

	ILP			RDDA		
requests	2	6	10	100	300	500
MinCU	21	14937	81382	21	61	102
MinW	8	105	1771	16	51	84
MaxR	3	20	62	24	74	140
MinT	4	126	77675	16	49	91

mance under their respective optimization objectives. Therefore, our work can provide reliable and efficient DMT service.

FUNDING

This work is supported in part by the Project of Jiangsu Engineering Research Center of Novel Optical Fiber Technology and Communication Network, Soochow University(SDGC2117), Beijing Natural Science Foundation(4232011), and NSFC (61831003, 62021005, 62101063).

REFERENCES

1. ITU-T, “Recommendation ITU-T Y.2501: Computing power network-framework and architecture,” Tech. rep., International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) (2021).
2. B. Lei and G. Zhou, “Exploration and practice of computing power network (cpn) to realize convergence of computing and network,” in *Optical Fiber Communications Conference and Exhibition (OFC),* (2022), p. M4A.2.
3. S. B. Calo, M. Touna, D. C. Verma, and A. Cullen, “Edge computing architecture for applying AI to IoT,” in *IEEE International Conference on Big Data (Big Data),* (IEEE, 2017), pp. 3012–3016.
4. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” Adv. neural information processing systems **33**, 1877–1901 (2020).
5. J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang *et al.*, “Large scale distributed deep networks,” Adv. neural information processing systems **25** (2012).
6. S. Alqahtani and M. Demirbas, “Performance analysis and comparison of distributed machine learning systems,” arXiv preprint arXiv:1909.02061 (2019).
7. L. Gurobi Optimization, “Gurobi optimizer reference manual,” (2022).
8. B. Lei, Q. Zhao, and J. Mei, “Computing power network: An interworking architecture of computing and network based on IP extension,” in *IEEE International Conference on High Performance Switching and Routing (HPSR),* (2021), pp. 1–6.
9. X. Shi, Q. Li, D. Wang, and L. Lu, “Mobile computing force network (MCFN): Computing and network convergence supporting integrated communication service,” in *International Conference on Service Science (ICSS),* (2022), pp. 131–136.
10. J. Li, H. Lv, B. Lei, and Y. Xie, “A computing power resource modeling approach for computing power network,” in *International Conference on Computer Communications and Networks (ICCCN),* (2022), pp. 1–2.
11. H. Ma, J. Zhang, Z. Gu, H. Yu, T. Taleb, and Y. Ji, “DeepDefrag: Spatio-temporal defragmentation of time-varying virtual networks in computing power network based on model-assisted reinforcement learning,” in *European Conference on Optical Communication (ECOC),* (2022), p. Tu5.59.
12. R. Gu, Y. Chen, S. Liu, H. Dai, G. Chen, K. Zhang, Y. Che, and Y. Huang, “Liquid: Intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed gpu clusters,” IEEE Transactions on Parallel Distributed Syst. **33**, 2808–2820 (2022).
13. Y. Bao, Y. Peng, C. Wu, and Z. Li, “Online job scheduling in distributed machine learning clusters,” in *IEEE Conference on Computer Communications (INFOCOM),* (2018), pp. 495–503.
14. S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala, “Pytorch distributed: Experiences on accelerating data parallel training,” Proc. VLDB Endow. **13**, 30053018 (2020).
15. M. Jeon, S. Venkataraman, A. Phanishayee, J. Qian, W. Xiao, and F. Yang, “Multi-tenant gpu clusters for deep learning workloads: Analysis and implications (tr),” Tech. Rep. MSR-TR-2018-13, Microsoft (2018).
16. S. Oh, K. Kim, and E. Seo, “A dynamic scaling scheme of cloud-based dnn training clusters,” in *IEEE International Conference on Smart Cloud (SmartCloud),* (IEEE, 2020), pp. 165–168.
17. Q. Zhou, Y. Zhang, and X. Li, “Multitasking elastic scheduling cluster in tensorflow,” in *International Conference on Control, Robotics and Cybernetics (CRC),* (2020), pp. 156–160.
18. M. Yu, B. Ji, H. Rajan, and J. Liu, “On scheduling ring-all-reduce learning jobs in multi-tenant gpu clusters with communication contention,” in *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing,* (2022), pp. 21–30.
19. Y. Bao, Y. Peng, Y. Chen, and C. Wu, “Preemptive all-reduce scheduling for expediting distributed dnn training,” in *IEEE Conference on Computer Communications (INFOCOM),* (2020), pp. 626–635.
20. M. Yu, Y. Tian, B. Ji, C. Wu, H. Rajan, and J. Liu, “Gadget: Online resource optimization for scheduling ring-all-reduce learning jobs,” in *IEEE Conference on Computer Communications (INFOCOM),* (2022), pp. 1569–1578.
21. R. Ma, E. Georganas, A. Heinecke, S. Gribok, A. Boutros, and E. Nurvitadhi, “FPGA-based AI smart NICs for scalable distributed AI training systems,” IEEE Comput. Archit. Lett. **21**, 49–52 (2022).
22. J. Wang, P. Liu, Z. Guo, S. Liu, and C. Yao, “Exploring the impact of attacks on ring allreduce,” in *5th Asia-Pacific Workshop on Networking (APNet 2021),* (2021), pp. 12–13.
23. H. Li, K. Ota, and M. Dong, “Learning IoT in edge: Deep learning for the internet of things with edge computing,” IEEE Netw. **32**, 96–101 (2018).
24. Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” ACM Trans. Intell. Syst. Technol. **10** (2019).
25. M. Liu, Y. Li, Y. Zhao, H. Yang, and J. Zhang, “Adaptive DNN model partition and deployment in edge computing-enabled metro optical interconnection network,” in *Optical Fiber Communications Conference and Exhibition (OFC),* (IEEE, 2020), p. Th2A.28.
26. Y. Li, Z. Zeng, J. Li, B. Yan, Y. Zhao, and J. Zhang, “Distributed model training based on data parallelism in edge computing-enabled elastic optical networks,” IEEE Commun. Lett. **25**, 1241–1244 (2021).
27. Z. Zhai, J. Lin, Y. Li, D. Zheng, Z. Chang, L. Zong, N. Deng, T. Chang, and G. Shen, “Delivering ring allreduce services in WSS-based all-optical rearrangeable clos network,” in *Asia Communications and Photonics Conference,* (Optica Publishing Group, 2021), pp. T4A–139.
28. J. Lin, G. Shen, Z. Zhai, D. Zheng, Y. Li, Z. Chang, L. Zong, N. Deng, and T. Chang, “Delivering distributed machine learning services in all-optical datacenter networks with torus topology,” in *Asia Communications and Photonics Conference (ACP),* (2021), pp. 1–3.
29. B. Mirkhanzadeh, T. Zhang, M. Razo-Razo, M. Tacca, and A. Fumagalli, “Super-Cloudlet: Rethinking edge computing in the era of open optical networks,” in *International Conference on Computer Communications and Networks (ICCCN),* (2021), pp. 1–11.
30. P. Yi, H. Ding, and B. Ramamurthy, “Budget-optimized network-aware joint resource allocation in grids/clouds over optical networks,” J. Light. Technol. **34**, 3890–3900 (2016).
31. J. Wu, J. Zhao, and S. Subramaniam, “Co-scheduling computational and networking resources in elastic optical networks,” in *IEEE International Conference on Communications (ICC),* (2014), pp. 3307–3312.
32. B. Jamil, M. Shojafar, I. Ahmed, A. Ullah, K. Munir, and H. Ijaz, “A job scheduling algorithm for delay and performance optimization in fog computing,” Concurr. Comput. Pract. Exp. **32**, e5581 (2020).
33. H. Chen, Y. Zhao, and J. Zhang, “Static provisioning for advance reservation in elastic optical networks,” in *International Conference on Optical Communications and Networks (ICOCN),* (2017), pp. 1–3.
34. Z. Liu, J. Zhang, Y. Li, L. Bai, and Y. Ji, “Joint jobs scheduling and lightpath provisioning in fog computing micro datacenter networks,” J. Opt. Commun. Netw. **10**, 152–163 (2018).
35. Z. Liu, J. Zhang, L. Bai, and Y. Ji, “Joint jobs scheduling and routing for metro-scaled micro datacenters over elastic optical networks,” in *Optical Fiber Communications Conference and Exposition (OFC),* (2018), p. M2E.3.

36. Y. Li, J. Zhang, Z. Liu, and Y. Ji, "Joint optimization for combined jobs scheduling and routing in the edge computing based EON," in *Asia Communications and Photonics Conference (ACP),* (2018), pp. 1–3.
37. C. A. of Information and C. Research, *China Arithmetic Development Index White Paper* (China Academy of Information and Communication Research, 2021).
38. N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE Conference on Computer Communications (INFOCOM),* (2019), pp. 1387–1395.
39. J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," optimization Methods Softw. **32**, 993–1005 (2017).