Future Generation Computer Systems 112 (2020) 549–560

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# Job scheduling for distributed machine learning in optical WAN

Ling Liu [a], Hongfang Yu [a,b,*], Gang Sun [a,c,**], Long Luo [a], Qixuan Jin [a], Shouxi Luo [d]

[a] Key Laboratory of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu, People's Republic of China
[b] Peng Cheng Laboratory, Shenzhen, People's Republic of China
[c] Agile and Intelligent Computing Key Laboratory of Sichuan Province, Chengdu, People's Republic of China
[d] Southwest Jiaotong University, Chengdu, People's Republic of China

ARTICLE INFO

*Article history:*
Received 20 January 2020
Received in revised form 12 April 2020
Accepted 5 June 2020
Available online 9 June 2020

*Keywords:*
Distributed machine learning (DML)
Job scheduling
Weighted job completion time (WJCT)

ABSTRACT

Large companies operate tens of data centers (DCs) across the globe to serve their customers and store data. On the other hand, many machine learning applications need a global view of such global data to pursue high model accuracy. However, for this Geo-distributed machine learning (Geo-DML), it is infeasible to move all data together over wide-area networks (WANs) due to scarce WAN bandwidth, privacy concerns and data sovereignty laws. Therefore, most Geo-DML systems leverage geo-distributed approaches to train models, where global model synchronization is required between DCs over WAN. With the rapid increase of training data and the model sizes, it is challenging to efficiently utilize scarce and heterogeneous WAN bandwidth to synchronize models. With the advancement of optical technology, network topology becomes reconfigurable in optical WAN, which brings a new opportunity for Geo-DML training over WAN.

We propose to optimize Geo-DML training with centralized joint control of the network and reconfigurable optical layers. We respectively prove the intra-job and inter-job scheduling problems are *NP-hard* and *strongly NP-hard*. For intra-job scheduling, RoWAN based on deterministic rounding algorithm, is presented to dynamically change the topology by reconfiguring the optical devices, and allocate path and rate for each flow. For inter-job scheduling, delayed SWRT is provided to schedule multiple jobs according to their priorities. The simulations in real topologies show that RoWAN reduces global model synchronization communication time of single iteration by up to 15.54%-48.2% on average in comparison with the traditional solutions. Compared to other three inter-job scheduling approaches, delayed SWRT can reduce the weighted job completion time (WJCT) by about 60%, 44.8% and 28.76%.

## 1. Introduction

Machine learning is used in a wide range of application domains such as speech recognition [1], language understanding [2,3] and computer vision [4]. These applications need to derive useful information from massive amounts of data about user activities,videos, and pictures, etc., which are often generated rapidly all over the world. Many IT leading companies, such as Google [5] and Microsoft [6], have built tens of data centers (DCs) globally to store large-scale data in each DC. Then, geo-distributed machine learning (Geo-DML) which deals with machine learning across data centers is brought up.

Unfortunately, it is impractical to gather all globally-generated data over wide-area network (WAN) to a single data center before training machine learning (ML) models, because it is extremely slow and costly due to scarce and expensive WAN bandwidth. Besides, it is also limited by data sovereignty laws and private constraints [7–9]. In fact, most Geo-DML applications adopt geo-distributed training, where each DC runs ML algorithms based on its data and executes intra-DC (local) model synchronization, and global model synchronization is needed across DCs to keep the global ML models consistent sufficiently. Most ML algorithms are iterative in nature and it may require hundreds of thousands or even millions of iterations to achieve ideal model accuracy [10]. Gaia [11] and Multi-Level Master/Slave Communication Tree [12] are two typical Geo-DML systems, shown in Fig. 1(a) and (b). With the increasing size of training data, ML models can contain up to $10^9$ to $10^{12}$ parameters [13], which means that the model sizes can be up to the order of hundreds of GB or even bigger. The large quantities of model parameters put heavy burden on the WAN with sparse and heterogeneous bandwidth across DCs.

Although some schemes [14–16] have put forward to accelerate Geo-DML in terms of reducing the globally synchronization

[*] Correspondence to: No.2006, Xiyuan Ave, West Hi-Tech Zone, 611731, Chengdu, Sichuan, P.R.China.
[**] Corresponding author.
*E-mail addresses:* yuhf@uestc.edu.cn (H. Yu), gangsun@uestc.edu.cn (G. Sun).

https://doi.org/10.1016/j.future.2020.06.007
0167-739X/© 2020 Published by Elsevier B.V.

frequency of model parameters, the size of parameters shared, or handling the heterogeneous computation and network resources over WAN, the performance improvement brought by these schemes is also limited by the scarce bandwidth of WAN. This is because the difference in WAN bandwidth between DCs can be more than 12 times [11], and the global model synchronization time between DCs is limited by the worst-bandwidth. Fortunately, the recent optical WAN technology provide a new alternative for Geo-DML. Note that recent WAN network designs such as Microsoft SWAN [6] and Google B4 [5] can centrally control the network globally by leveraging software-defined networking (SDN). Accordingly, with the help of SDN, the optical WAN allows to reconfigure the network topology with a global view by flexibly shifting wavelengths to adjacent fibers. This reconfigurable characteristic of optical WAN enables a *demand-aware* network [17] which can adjust the network topology based on the traffic demands of the upper applications.

In recent years, some efforts focusing on specific traffic demands in optical WAN have been proposed, such as bulk transfers [18,19]. And the experiment or simulation results indicate that jointly optimizing topology and scheduling can greatly reduce the transfer completion time. However, the impact of reconfigurable WAN topology on the performance is ignored in existing solutions to accelerate Geo-DML applications. Besides, the main purposes of works focusing on bulk transfers are to minimize the transfer completion time or maximize the number of successful transfers. Their algorithms are designed for single-stage transfer/job, and the completion of a data transmission represents the end of the transfers/jobs, so they are not well suited for multi-stage jobs like Geo-DML where there are dependencies between data transmissions. This is because minimizing job completion time of multi-stage job is not the same as minimizing the completion time of single-stage transfer or flow [20].

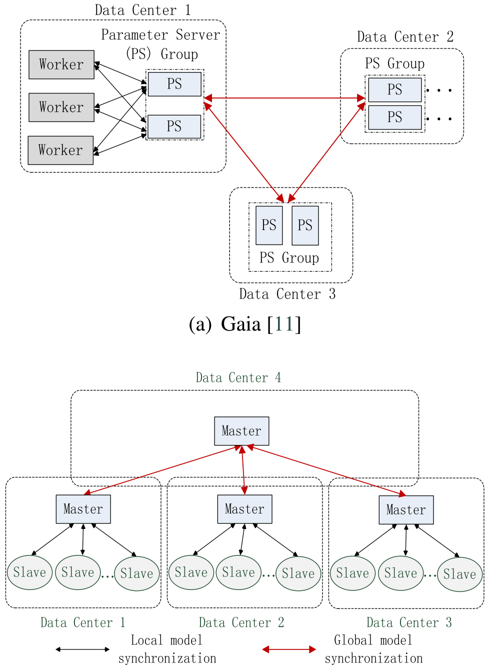
Therefore, in this paper, we make the first step towards the study of how to jointly optimize Geo-DML with scheduling in network layer and reconfigurable topology in optical layer to reduce the weighted job completion time (WJCT). Specifically, we divide our problem into intra-job scheduling and inter-job scheduling problems, and the major contributions of our work are as follows:

- Firstly, we prove the intra-job scheduling in optical WAN is *NP hard*, and propose a deterministic rounding based algorithm **RoWAN** (**R**outing and rate allocation in **o**ptical **WAN**) to solve the problem. Besides, we show the performance of RoWAN through theoretical analysis.
- Then, we prove the inter-job scheduling problem is *strongly NP hard*, and propose an online heuristic algorithm **delayed SWRT** (**delayed S**hortest **W**eighted **R**emaining **T**ime), which schedules jobs based on the weight and remaining completion time.
- Finally, we conduct simulations to evaluate the performance of RoWAN and delayed SWRT. And the results show that our algorithms can significantly improve the performance of Geo-DML.

The remainder of this paper is organized as follows. Section 2 describes the related works. Section 3 shows the background and a motivating example of our work. Section 4 describes the details of the problem formulations and our solutions. Section 5 shows the performance analysis of our solutions. Section 6 carries out extensive simulations to evaluate the performance of RoWAN and delayed SWRT. Finally, Section 7 concludes this work.

## 2. Related work

**Geo-distributed machine learning.** Large organizations often have a planetary footprint, and large amounts of data are created in the vicinity of users and systems. So geo-distributed machine learning (Geo-DML) across DCs has attracted much attention.



(a) Gaia [11]

(b) Multi-Level Master/Slave Communication Tree [12]

**Fig. 1.** Two typical Geo-DML systems. (a) Gaia is based on Parameter Server (PS) architecture. Each PS group takes responsible for the local model synchronization in its data center, and all PS groups require global model synchronization with each other across data centers. (b) In each data center, local master communicates with local slaves through Broadcast/Reduce operations (local model synchronization). Further, a global master connects with other local masters to synchronize model (global model synchronization, including model aggregation when each local master transmits local model updates to global master and model distribution when global master transmits new model to each local master).

Many works have proposed to solve the problems about Geo-DML. To efficiently utilize the sparse and heterogeneous WAN bandwidth, Hsieh K. et al. [11] propose a general Geo-DML system based on Parameter Server (PS) architecture, Gaia, which decouples the communication over a DC from the communication over WANs. It provides Approximate Synchronous Parallel (ASP), a new synchronization model between DCs, which allows the global model copy in each DC is approximately correct by reducing insignificant communication over WAN. Ignacio Cano et al. [12] provide a Multi-Level Master/Slave Communication Tree for model synchronization, where the local DC masters are used to connect with their respective slaves in each DC for local model aggregation and the global DC master is used to communicate with local DC masters for global model aggregation. Xinchen Lyu et al. [21] present a novel online solution to partition streaming data in edge of wireless networks for Geo-DML, where data are centralized and then partitioned to different network edge servers. Besides, Corentin Hardy et al. [22] propose MD-GAN, the first scheme to train a Generative Adversarial Network (GAN) over geo-distributed workers. Wang S. et al. [23] provide a control algorithm to achieve the trade-off between local update and global model aggregation in geo-distributed resource constrained edge computing systems. Most of these Geo-DML approaches focus on communication optimization of model synchronization, regardless of the underlying network topology.

**Optical WAN.** Based on the fact that modern wide-area network (WAN) communication is carried over optical devices, such as Reconfigurable Optical Add-Drop Multiplexers (ROADMs), the

WAN network topology has become reconfigurable by adjusting the wavelengths on each optical fiber. Recently, some works have focused on the performance improvement obtained by reconfigurable optical WAN. Owan [18] aims to optimize unicast bulk transfers in optical WAN. With the consideration of ROADMs and regenerators, it uses simulated annealing to find a topology, as well as the path and rate allocation for transfers. DaRTree [24] is proposed to maximize the number of multicast transfers that can finish data transmission before deadlines. The simulation results show that it can significantly improve the performance by leveraging both multicast Steiner trees and reconfigurable WAN topology. In addition, Jia S. et al. [25] present the theoretical analysis of some online transfer scheduling approximation algorithms in optical WANs under some assumptions such as release time and data size. However, none of these approaches care about multi-stage jobs like our approach.

## 3. Background and a motivating example

**Reconfigurable optical WAN.** The network-layer topology of a modern WAN is constructed over an intelligent optical layer. Typically, each WAN router is connected to a Reconfigurable Optical Add/Drop Multiplexer (ROADM) [26,27], which in turn is connected by optical fiber pairs. The fibers can carry different number of wavelengths which depends on the ROADM technologies. Through adjusting the number of wavelengths on each fiber, we are able to get different network topologies.

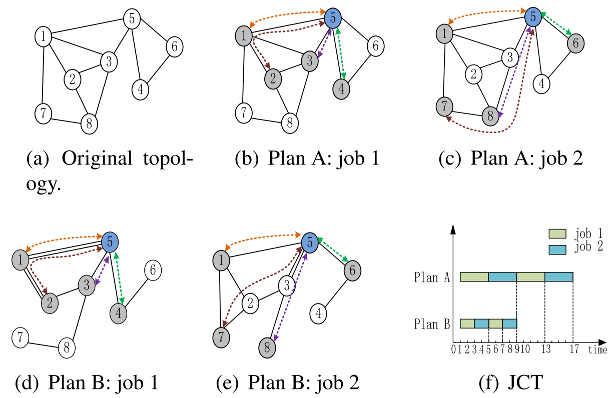
ROADMs allow to dynamically reconfigure the wavelengths as low as tens of milliseconds [18], and the reconfiguration time will be lower with the development of ROADM technologies. Note that the models considered in this work are large enough that the reconfigurable time can be ignored. Besides, the number of wavelengths deployed on each ROADM is limited by its number of optical transponders, and the receiving and transmitting parts can be bundled into bidirectional wavelengths or can be also separated. For simplicity, we use the former one in this article. In other words, the sum of the number of transmitted and received wavelengths on each ROADM is constant.

Previous works mainly focus on the potential performance improvement of single-stage multicast bulk transfers or unicast bulk transfers in reconfigurable optical WAN. While our work concentrate on the complete multi-stage DML jobs, not just single-stage flows.

**A motivating example.** Consider the eight node WAN in Fig. 2, where each black edge represents a wavelength between DCs. Assume that there are two jobs and the job information is shown in Table 1, where computation time means the local model update computation time including the time of local update computation and local model synchronization in each DC. The bandwidth of a wavelength is 1 and model synchronization of jobs uses the Master/Slave Communication Tree method. Suppose node 5 is the global master DC of both jobs, and the local master DCs of two jobs are $\{1, 2, 3, 4\}$ and $\{1, 6, 7, 8\}$, respectively. In static topology (Plan A, only scheduling), the paths of flows are shown in Fig. 2(b) and (c). We ignore the computation time of model aggregating, because it is much smaller compared to the data transmission time due to powerful computation capability of each DC. Besides, We assume model aggregation (data from local master DCs to global master DC) and distribution (data from global master DC to local master DCs) flows leverage the same paths and rates. Thus, it takes 4 ($max\{\frac{1+1}{1}, \frac{1}{1}, \frac{1}{1}\} \times 2$) units of time to transmit data for each global model synchronization. Therefore, the completion times of the two jobs are 13 and 17. On the other hand, if the paths of flows of the two jobs in reconfigurable WAN (Plan B, scheduling + dynamic topology) are shown in Fig. 2(d) and (e), respectively. It is notable that the bandwidth of $link(1, 2)$, $link(1, 5)$ (see Fig. 2(d)) and $link(3, 5)$ (see Fig. 2(e)) becomes 2 due to the wavelength reconfiguration. Thus it takes 2 ($\max\{\frac{1+1}{2}, \frac{1}{1}, \frac{1}{1}\} \times 2$) units of time for both jobs to finish a single global model synchronization. It is worth noting that only two easy jobs are involved in this example, but the inter-job scheduling problem should be considered in a production environment. The job completion times (JCTs) are 7 and 9, respectively. Fig. 2(f) shows the completion times where the weighted job completion time (WJCT) in reconfigurable optical WAN is $\frac{64-34}{64} \times 100\% \approx 47\%$ lower than that in static WAN, which demonstrates that network topology has a substantial impact on the training time. Therefore, combining scheduling in network layer and topology reconfiguration in optical layer can effectively improve the Geo-DML performance.

**Table 1**
Job Information.

| Job ID | Iteration number | Computation time | Model size | Weight |
|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 1 |
| 2 | 2 | 2 | 1 | 3 |



(a) Original topology.
(b) Plan A: job 1
(c) Plan A: job 2
(d) Plan B: job 1
(e) Plan B: job 2
(f) JCT

**Fig. 2.** Example of Geo-DML in static topology and reconfigurable topology. The initial topology is shown in Fig. 2(a), where each black edge presents a wavelength with 1 unit bandwidth. The flow scheduling solutions of two jobs in static topology (Plan A) are shown in Fig. 2(b) and (c), respectively, where the dotted lines represent the paths of flows. Fig. 2(d) and (e) show the flow scheduling solution in reconfigurable topology (Plan B). Fig. 2(f) shows the completion times of the jobs in static and reconfigurable topologies.

## 4. Problem formulation and solution

Multiple jobs scheduling is a complex and difficult problem. Therefore, we divide our problem into two sub-problems, intra-job scheduling and inter-job scheduling problems, and we will introduce the problem formulations and solutions respectively.

### 4.1. Intra-job scheduling problem

#### 4.1.1. Network model

We abstract the optical WAN as an undirected graph $G = (\mathbb{N}, \mathbb{E})$, where nodes $\mathbb{N}$ and edges $\mathbb{E}$ represent ROADMs connected to DCs and fibers connecting ROADMs, respectively. The maximum number of wavelengths node $v \in \mathbb{N}$ can carry is $Z_v$, including receiving and sending wavelengths. And each fiber $e \in \mathbb{E}$ also has a maximum number of wavelengths $R_e$. Since a fiber can have two links in opposite directions, we introduce two virtual directed links for each fiber, and the sum of wavelengths assigned to the two directed links should not be greater than the capacity of the edge.

Similar to the multiple coflows scheduling algorithms [28–30] across inter-DC WAN, we also mainly schedule a job at each moment. Note that we also pursue work-conserving property by

**Table 2**
Notation.

| Symbol | Definition |
|---|---|
| $J$ | The number of flows |
| $C$ | The bandwidth capacity of a wavelength |
| $E$ | The number of edges |
| $N$ | The number of nodes |
| $L$ | The number of directed links |
| $\mathbb{J} = \{1, 2, \dots, J\}$ | The flow set |
| $\mathbb{E} = \{1, 2, \dots, E\}$ | The edge set |
| $\mathbb{N} = \{1, 2, \dots, N\}$ | The node set |
| $\mathbb{L} = \{1, 2, \dots L\}$ | The directed link set |
| $S$ | Model size |
| $R_e$ | The available number of wavelengths that edge $e \in \mathbb{E}$ can carry |
| $Z_v$ | The available number of wavelengths that node $v \in \mathbb{N}$ can carry |
| $P_j$ | The path set of flow $j \in \mathbb{J}$ |
| $P_j^k$ | The link set of the $k$th path of flow $j \in \mathbb{J}$ |
| $Q_j^k$ | The node set of the $k$th path of flow $j \in \mathbb{J}$ |
| $h_l$ | The number of wavelength assigned to link $l \in \mathbb{L}$ |
| $g_j$ | The transmission rate of flow $j \in \mathbb{J}$ |
| $x_j^k$ | A binary variable. $x_j^k = 1$ denotes the flow $j$ is routed through $k$th path |
| $T$ | The completion time of all flows |

distributing the idle links to other jobs, which will be described later.

Although the computing capabilities and data volume of DCs can be different, leading to straggler problems, which can be solved by the solutions in [11,31]. So the start time of global model synchronization in each DC can be considered almost the same. It is notable that a Geo-DML job contains lots of iterations, and each iteration can be considered similar, so the iteration time of each iteration, including the local model update computation time (called computation time for short) and global model synchronization communication time (called communication time for short), is regarded as the same.

Without loss of generality, we do not assume the specific model synchronization method within DC and between DCs, but we assume the global model synchronization between DCs is synchronous, which means all DCs will participate in each global model synchronization. Besides, some schemes based on sparsity to reduce data size are not used, such as gradient compression [32,33] and coding [34]. So for each job, the data transmitted during global model synchronization stage are the same (i.e., the model size). Regardless of whether PS-based architecture or Master–Slave Communication Tree architecture is used for Geo-DML, once the jobs are deployed, the data flows for global model synchronization across optical WAN are determined. Then our algorithm is used to determine the underlying topology, allocate paths and rates for these flows. We model the intra-job scheduling problem with Master/Slave Communication Tree as an example. Besides, the corresponding flows in model aggregation and model distribution phases use the same paths and rates, that is, the completion times of the two phases are the same. Therefore, to minimize the communication time for intra-job scheduling problem, we only need to model the model aggregation phase, which can be formulated as follows. The notations we used are shown in Table 2.

$$\min \quad T \tag{1}$$

s.t.

$$T = max \frac{S}{g_j}, \forall j \in \mathbb{J} \tag{1a}$$

$$\sum_{l \in \mathbb{L}} I(l \in e) h_l \leq R_e, \forall e \in \mathbb{E} \tag{1b}$$

$$\sum_{j \in \mathbb{J}} \sum_{k \in P_j} g_j x_j^k I(l \in P_j^k) \leq h_l C, \forall l \in \mathbb{L} \tag{1c}$$

$$\sum_{j \in \mathbb{J}} \sum_{k \in P_j} g_j x_j^k I(v \in Q_j^k) \leq Z_v C, \forall v \in \mathbb{N} \tag{1d}$$

$$\sum_{k \in P_j} x_j^k = 1, \forall j \in \mathbb{J} \tag{1e}$$

$$x_j^k \in \{0, 1\}, \forall j \in \mathbb{J}, \forall k \in P_j \tag{1f}$$

As shown in Eq. (1), our goal is to minimize the completion time of all aggregating flows $T$. Note that for Master/Slave Communication Tree, the communication time of a single iteration equals twice $T$ when aggregating model phase and distributing model phase are viewed as two equal phases. For PS-based architecture, the global model synchronization between DCs can be all-to-all communication, then $T$ is the communication time. Constraint (1a) represents $T$ is equal to the completion time of the last completed flow. Constraint (1b) ensures the sum of assigned wavelengths on the directed links going through edge $e$ is no more than the available wavelength number of edge $e$. The indicator $I(l \in e)$ denotes whether the link $l$ passes through edge $e$. Constraint (1c) guarantees that the traffic load on each link should not exceed the link capacity, where $I(l \in P_j^k)$ denotes whether the $k$th path of flow $j$ passes through the link $l$. Constraint (1d) states the traffic load on each node should not exceed the node capacity, where $I(v \in Q_j^k)$ denotes whether the $k$th path of flow $j$ goes through the node $v$. Constraint (1e) ensures only a path is selected for each flow.

**Theorem 1.** *Problem (1) is NP-hard.*

**Proof.** We consider a special case when there are no constraints on each node, and the two directed links on each edge share the wavelengths equally. This means that each directed link has the fixed bandwidth capacity $\frac{R_e}{2} \times C$ when $R_e$ is even, or the two links have bandwidth capacity $\frac{R_e+1}{2} \times C$ and $\frac{R_e-1}{2} \times C$ respectively when $R_e$ is odd. Besides, the aggregating flows can be viewed as a *coflow*. Then the special case problem becomes the routing and rate allocation for a single *coflow* problem that is proven to be *NP-hard* [35,36]. Therefore, the original problem (1) is *NP-hard* too.

*4.1.2. Relaxed formulation*

It is mathematically intractable to solve the NP-hard problem. So we resort to design an efficient heuristic algorithm based on deterministic rounding to solve the problem (1). Note that constraint (1a) is not a linear form. To transform (1a) into a linear form, we introduce a new variable $f$ to be $\frac{1}{T}$. Then programming model (1) can be modified as follows:

$$\max \quad f \tag{2}$$

s.t.

$$f \leq \frac{g_j}{S}, \forall j \in \mathbb{J} \tag{2a}$$

(1b)–(1f)

However, there are still binary variables $x_j^k$ in model (2), so we relax the constraint (1f) to $x_j^k \in [0, 1]$. Then our problem becomes:

$$\max \quad f \tag{3}$$

s.t.

$$x_j^k \in [0, 1], \forall j \in \mathbb{J}, \forall k \in P_j \tag{3a}$$

(2a), (1b)–(1e)

It is worth noting that there is a product of two variables in constraints ($g_j$ and $x_j^k$) (1c) and (1d), so we introduce $y_j^k = f * x_j^k$, and we substitute (2a) into (1c) and (1d), then we obtain the following linear programming model:

$$\max \quad f \tag{4}$$

s.t.

$$\sum_{j\in\mathbb{J}}\sum_{k\in P_j} S y_j^k I(l \in P_j^k) \le h_l C, \forall l \in \mathbb{L} \tag{4a}$$

$$\sum_{j\in\mathbb{J}}\sum_{k\in P_j} S y_j^k I(v \in Q_j^k) \le Z_v C, \forall v \in \mathbb{N} \tag{4b}$$

$$\sum_{k\in P_j} y_j^k = f, \forall j \in \mathbb{J} \tag{4c}$$

$$y_j^k \ge 0, \forall j \in \mathbb{J}, \forall k \in P_j \tag{4d}$$

(1b)

Now, $y_j^k$, $f$ and $h_l$ are variables and problem (4) can be solved efficiently by standard linear programming (LP) solvers (e.g., GUROBI and CPLEX). Because the binary variable $x_j^k$ is relaxed, the solution of problem (2) obtained according to problem (4) may be fractional, which is not a feasible solution. To find a feasible solution, we then propose a deterministic rounding based algorithm RoWAN, shown in the next section.

#### 4.1.3. RoWAN algorithm design

In our design, once a Geo-DML job arrives at the network, we will formulate a model problem (1) and relax it into a LP problem (4), then use deterministic rounding technique to enforce the solution of the problem (4) to be a feasible one of the problem (1). The basic idea of RoWAN is: firstly the path of each flow is selected according to the solution of problem (4), and then the selected paths are used to guide the construction of topology and rates assigned to flows. The RoWAN is summarized as Algorithm 2.

In Algorithm 2, firstly we initialize all $x_j^k = 0$ (line 1–3), then we get all $y_j^k$ by solving problem (4) (line 4). Next, we route the $j$th flow to a path $k'$ such that $y_j^{k'} = max_k\ y_j^k$, so $x_j^{k'} = 1$ (line 5–8). Then, we solve problem (2) according to fixed $x_j^k$ and get rate allocated to each flow $g_j$, wavelength number assigned to each link $h_l$ and $f$. Finally, we can calculate $T$ by $T = \frac{1}{f}$ (line 10). Therefore, we can get the completion time of aggregating flows $T$ through RoWAN algorithm, and the communication time of a single iteration is $2T$ for Master/Slave Communication Tree.

### 4.2. Inter-job scheduling problem

To better understand the inter-DML job scheduling problems, firstly we present the offline formulation of multiple Geo-DML job scheduling problem and prove its *strong NP-hardness*. Then we will describe our inter-job scheduling algorithm.

#### 4.2.1. Offline formulation

The iteration time of Geo-DML contains computation time (local model update computation time) and communication time (global model synchronization communication time) over WAN. When a new Geo-DML job $i$ arrives, we can obtain its communication time ($P_i$) of each iteration based on RoWAN. Besides, we also assume the iteration number of each job is given in advance, which is reasonable because Geo-DML jobs can to be interrupted or stopped according to the preset iteration number or model accuracy. The inter-job scheduling problem can be formulated as follows, and the notations are shown in Table 3.

$$\min \quad \sum_i w_i J_i \tag{5}$$

**Algorithm 1:** RoWAN: **R**outing and rate allocation in **o**ptical **WAN**

```
   Input: $S$, $C$, $\mathbb{J}$, $\mathbb{E}$, $\mathbb{N}$, $\mathbb{L}$, $R_e$,$Z_v$, $P_j$,$P_j^k$,$Q_j^k$
   Output: $g_j$,$h_l$,$x_j^k$,$T$
1  Initialization: for flow $j \in \mathbb{J}$ do
2      $x_j^k \leftarrow 0$
3  end
4  $y_j^k$ = solve problem (4)
5  for flow $j \in \mathbb{J}$ do
       // round the solution $x_j^k$
6      $k' \leftarrow arg_k\ max\ y_j^k$
7      $x_j^{k'} \leftarrow 1$
8  end
9  $f$,$g_j$, $h_l$ = solve problem (2) according to $x_j^k$ obtained from
   Line 5-8
10 $T \leftarrow \frac{1}{f}$
11 return ($g_j$,$h_l$,$x_j^k$,$T$)
```

**Table 3**
Notation.

| Symbol | Definition |
|---|---|
| $R$ | The number of DML jobs |
| $T_i$ | Computation time of job $i$ |
| $K_i$ | The iteration number of job $i$ |
| $\mathbb{R} = \{1, 2, \dots, R\}$ | The job set |
| $\mathbb{K}_i = \{1, \dots, K_i\}$ | The iteration set of job $i$ |
| $P_i$ | The communication time of single iteration obtained by RoWAN |
| $w_i$ | The weight of job $i$ |
| $M$ | A very big number that is greater than the sum of all jobs' communication time, computation time. |
| $J_i$ | The completion time of job $i$ |
| $C_i^k$ | The communication completion time of the $k$th iteration of job $i$ |
| $S_i^k$ | The communication starting time of the $k$th iteration of job $i$ |
| $y_{ij}^{hg}$ | A binary variable. If the communication stage of $h$th iteration of job $i$ is completed before the communication stage of $g$th iteration of job $j$, $y_{ij}^{hg} = 1$, otherwise $y_{ij}^{hg} = 0$ |

s.t.

$$J_i = C_i^{K_i}, \forall i \in \mathbb{R} \tag{5a}$$

$$C_i^k = S_i^k + P_i, \forall i \in \mathbb{R}, \forall k \in \mathbb{K}_i \tag{5b}$$

$$S_i^k \ge C_i^{k-1} + T_i, \forall i \in \mathbb{R}, \forall k \in \mathbb{K}_i \tag{5c}$$

$$C_i^h + P_j \le C_j^g + M(1 - y_{ij}^{hg}), \forall i < j \in \mathbb{R}, \forall h \in \mathbb{K}_i, \forall g \in \mathbb{K}_j \tag{5d}$$

$$C_j^g + P_i \le C_i^h + M y_{ij}^{hg}, \forall i < j \in \mathbb{R}, \forall h \in \mathbb{K}_i, \forall g \in \mathbb{K}_j \tag{5e}$$

$$S_i^k \ge T_i, \forall i \in \mathbb{R}, \forall k \in \mathbb{K}_i \tag{5f}$$

$$y_{ij}^{hg} = \{0, 1\}, \forall i < j \in \mathbb{R}, \forall h \in \mathbb{K}_i, \forall g \in \mathbb{K}_j \tag{5g}$$

$$C_i^k \ge 0, \forall i \in \mathbb{R}, \forall k \in \mathbb{K}_i \tag{5h}$$

Eq. (5) is the objective of our problem. Constraint (5a) defines the job completion time (JCT) which is equal to the communication completion time of the last iteration. Constraint (5b) states that for each iteration, the communication completion time equals to the sum of communication start time and the communication time. Constraint (5c) ensures that the communication start time of $k$th iteration is greater than or equal to the communication completion time of the $(k-1)$th plus the computation time. Constraint set (5d) and (5e) are disjunctive constraints which enforce that either the $h$th iteration of job $i$ is processed before the $g$th iteration of job $j$ or the $g$th iteration of job $j$ is processed before $h$th iteration of job $i$ for any pair of jobs. Constraint (5f) indicates the communication start time is always no less than the computation time. Eqs. (5g) and (5h) are the integrality and non-negativity constraints.

**Theorem 2.** *Problem (5) is strongly NP-hard.*

**Proof.** We consider a special case when any DML job contains only one iteration, then the computation time ($T_i$) of each job can be viewed as the release date at which the job becomes available to transmit data. Besides, we regard optical WAN as a machine, and the communication stage of each DML job as the process of the job on the machine, then the communication time ($P_i$) becomes the processing time on the machine. So the special case of problem (5) becomes the problem of scheduling jobs with arbitrary release dates on a single machine for the purpose of minimizing total weighted completion time, which has proven to be *strongly NP-hard* [37]. Since the special case is *strongly NP-hard*, the original problem (5) is *strongly NP-hard* too.

#### 4.2.2. Online inter-job scheduling

The offline inter-job scheduling is *strongly NP-hard*, and it is very slow to solve this mixed-integer linear programming (MILP) problem by CPLEX or GUROBI. Besides, some new jobs may be started at any time during training processes of other jobs. Therefore, we design an online scheduling algorithm **delayed S**hortest **W**eighted **R**emaining **T**ime (delayed SWRT) for multiple Geo-DML jobs.

Considering the interlaced characteristic of the communicate and computation phases of Geo-DML, the delayed SWRT is executed when a job completes communication phase and switches to computation phase, which is the biggest difference between delayed SWRT and traditional job scheduling algorithm because traditional one is often executed once when a new job arrives or leaves. The basic idea of delayed SWRT is that when a job completes transmitting data globally, we preferentially schedule the highest priority job among all available jobs. The available job needs to meet two conditions: (1) the job has completed local model update computation phases, and (2) the product of its performed iteration number and the communication time ($P_i$) is no larger than the current time (*currentTime*), which is inspired by delayed Shortest Weighted Processing Time (delayed SWPT), an algorithm with competitive ratio of 2 for scheduling jobs on a single machine [38]. The priority of a job takes remaining time ($RT_i$) and weight into account. The $RT_i$ can be easily calculated by remaining iteration number ($RIT_i$), computation time ($T_i$) and communication time ($P_i$), because each iteration of Geo-DML can be considered similar. So

$$RT_i = RIT_i * P_i + (RIT_i - 1) * T_i.$$

And the priority of a job is calculated by

$$p_i = \frac{w_i}{RT_i}.$$

Algorithms 2 and 3 show the pseudocode of delayed SWRT.

**Starvation Avoiding and Work Conservation.** In order to avoid starvation for some jobs with small weights and long completion times, we set a time threshold $\Gamma$. If any job waits longer than $\Gamma$, the job will be scheduled firstly regardless of its priority. Besides, we also pursue work conservation so that some randomly selected jobs can take full advantage of the idle bandwidth to transmit data, which is not shown in the algorithm pseudocode.

**Algorithm 2:** delayed SWRT: delayed Shortest Weighted Remaining Time

```
Input: job set $\mathbb{R}$
Output: weighted job completion time $WJCT$
Initialization: $currentTime \leftarrow 0$;
$WJCT \leftarrow 0$;
while $\mathbb{R} \neq \emptyset$ do
    // select the highest priority job from
       available jobs
    $job, currentTime \leftarrow$ Available-Highest-Priority-Job
    ($\mathbb{R}, currentTime$);
    the remaining iteration number of $job$ minus 1;
    if the remaining iteration number of job = 0 then
        completiontime $\leftarrow currentTime +$
        job.communicationtime;
        $WJCT \leftarrow WJCT +$ job.weight*completiontime;
        remove $job$ from $\mathbb{R}$
    end
    else
        set the available time of $job \leftarrow$ Max{ $currentTime +$
        job.communicationtime + job.computationtime,
        job.communciationtime * job.performed iteration
        number}
    end
    $currentTime \leftarrow currentTime +$ job.communicationtime
end
return $WJCT$
```

**Algorithm 3:** Available-Highest-Priority-Job

```
Input: $remainjobs, currentTime$
Output: selected job $job$, start communicating time
        $startTime$
$joblist \leftarrow$ find jobs whose available time is no bigger than
$currentTime$ from $remainjobs$;
if $joblist \neq \emptyset$ then
    $job \leftarrow$ select the highest priority ( $p_i = \frac{w_i}{RT_i}$ ) job;
    $startTime \leftarrow currentTime$;
end
else
    // there are no available jobs now
    $job \leftarrow$ find the job with the minimum available time;
    $startTime \leftarrow$ job.availabletime
end
return ($job, startTime$)
```

## 5. Performance analysis

In this section, we show the performance of RoWAN through theoretical analysis and the complexity of delayed SWRT.

- **Approximation Bound Analysis.**

**Theorem 3** (*Lower Bound of RoWAN*)**.** *Assume the minimum completion time of problem (1) and the completion time based on problem (4) with rounded $x_j^k$ are is $T_o$ and $T_R$, respectively. Then, we have $T_o \leq T_R$.*

**Proof.** Note that we can get $h_l$ and $g_j$, the feasible solutions of problems (2) according to the rounded $x_j^k$ and $y_j^k$ obtained by problem (4). Now we mainly focus on proving the rounded $x_j^k$ satisfy the constraints of problem (1).

Based on (4c) and $y_j^k = f * x_j^k$, we have $\sum_{k \in P_J} y_j^k = \sum_{k \in P_J} f x_j^k = f$, then we can get $\sum_{k \in P_J} x_j^k = 1$. Besides, $x_j^k$ is limited to $\{0, 1\}$, so $x_j^k$ meets the constraint (1e). Besides, $h_l$ and $g_j$ are obtained by problem (2) with the rounded $x_j^k$, so they satisfies constraint (1(b))–(1(d)). Therefore, $x_j^k$ is a feasible solution to problem (1), and $T_R$ is a corresponding value of problem (1). In addition, it is known that an optimal solution is always better than or equal to a feasible solution. Therefore, for the minimization problem, we can obtain $T_o \leq T_R$.

**Theorem 4** (*Upper Bound of RoWAN*)**.** *$T_R \leq \beta T_o, \forall \beta \geq 1$.*

**Proof.** Let $f_{upper}$ denotes the optimal objective value of problem (4), and $C_o$ denotes the inverse of $T_o$. So $f_{upper} > C_o$ because (4) is a relaxation problem of (1). In RoWAN, we select the path $k$ with maximum $y_j^k = f_{upper} x_j^k$ for flow $j$, then we have

$$y_j^k \geq \frac{f_{upper}}{\beta} x_j^k, \forall \beta \geq 1$$

If we substitute the equation above into the constraints (4a) and (4b), then we can obtain

$$\sum_{j \in \mathbb{J}} \sum_{k \in P_j} S \frac{f_{upper}}{\beta} x_j^k I(l \in P_j^k) \leq \sum_{j \in \mathbb{J}} \sum_{k \in P_j} S y_j^k I(l \in P_j^k) \\ \leq h_l C$$

and

$$\sum_{j \in \mathbb{J}} \sum_{k \in P_j} S \frac{f_{upper}}{\beta} x_j^k I(v \in Q_j^k) \leq \sum_{j \in \mathbb{J}} \sum_{k \in P_j} S y_j^k I(v \in Q_j^k) \\ \leq Z_v C$$

Therefore, $\frac{f_{upper}}{\beta}$ is a feasible solution to problem (2) for the given $x_j^k$. While $\frac{1}{T_R}$ is the optimal solution to problem (2) based on problem (4) with given $x_j^k$, then, we can get

$$\frac{1}{T_R} \geq \frac{f_{upper}}{\beta} \geq \frac{C_o}{\beta}.$$

Finally, we can get for $\forall \beta \geq 1$,

$$T_R \leq \beta T_o.$$

- **Complexity analysis.** Note that the algorithm of finding the highest priority available job has a computational complexity of $O(R)$, where $R$ is the number of jobs. Therefore, given the communication time of each job, the computational complexity of delayed S- WRT is $O(RK)$, where $K$ is the sum of iteration number for all jobs.

## 6. Performance evaluation

In this section, we implement a flow-level discrete-event simulator to compare RoWAN, delayed SWRT with various intra-job and inter-job scheduling algorithms. The detail of simulation setup is shown in the following.

### 6.1. Simulation setup

**Network Topology:** We use two real-world inter-DC networks to be our network topologies in the simulations. One is B4 [5], the Google's private WAN, which connects Google's 12 data centers all over the world and has 19 inter-DC links. The other is Internet2, the ISP public network, which has 9 data centers and 17 inter-DC links. Firstly, we assume each link has assigned an initial uniform capacity of 50 Gbps, representing the static topology configuration. And each wavelength can carry 10 Gbps bandwidth, analogously to the evaluations in [18,24]. Besides, We define $r$ as the link capacity reconfiguration ratio and we allow a link to carry up to $r = 60\%$ more wavelengths than initially assigned at the cost of using wavelengths on adjacent links. That is, each link can have a maximum bandwidth of 80 Gbps in our experiments, except the experiment of testing the effects of $r$ on the performance.

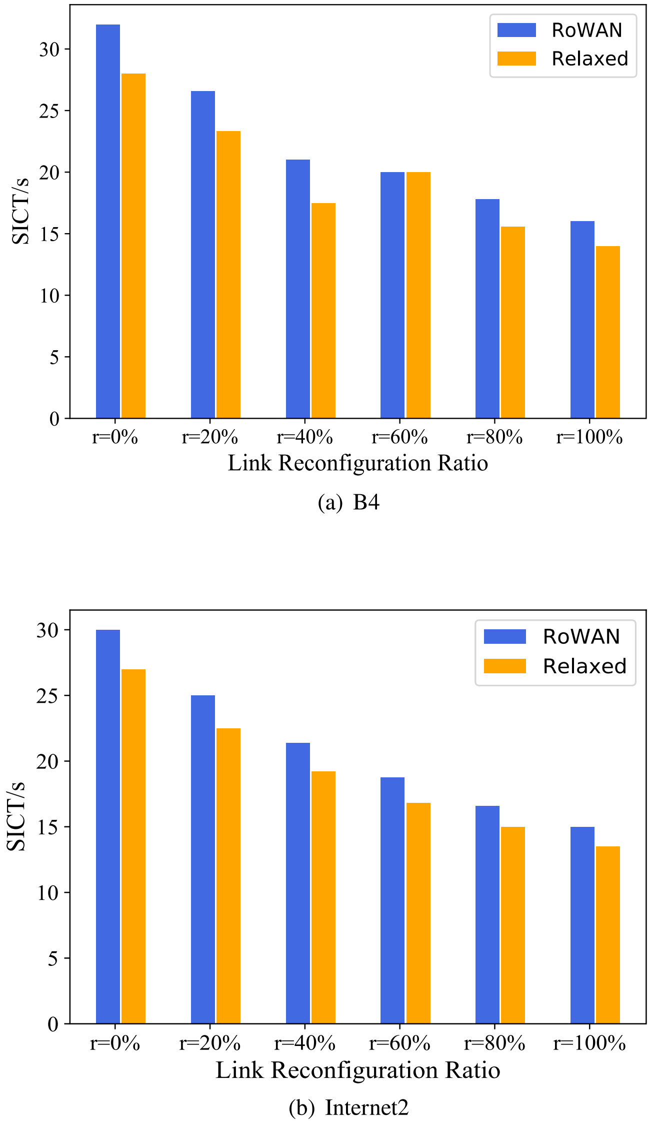
**Intra-job solutions to compare:** We compare the following solutions with RoWAN.

- **Scheduling-only**: routes all the flows by ECMP in static topology, and the order of the flows is arbitrary. The order of flows will not affect the performance because flow size of each flow is the same (i.e., model size). It is the baseline for intra-job scheduling solutions.
- **Routing-only**: route a flow on a selected path for the purpose of load balancing, and all flows fairly compete for bandwidth in static topology.
- **Routing+Scheduling**: select a path and allocate rate for each flow based on a joint optimization model in static topology, which is conceptually equivalent to the routing and rate allocation of PRO across DCs [28] and Rapiter [35]. By comparing with this algorithm, we can see the performance improvement obtained by optimizing Geo-DML in combination with dynamic topology.
- **Owan:** exploits $k$-paths to deliver bulk transfers by jointly controlling network topology, paths and rate allocation in optical WAN [18]. The key idea of Owan is to use simulated annealing to create new topology in each time slot, while we adopt rounding based algorithm.
- **Relaxed solution (Relaxed):** is the value deprived by solving problem (4), which is the upper bound of the original problem (2). Compared to this value, we can see the gap between the time obtained by RoWAN and the upper bound one.

**Inter-job solutions to compare:** We compare the following online inter-job algorithms with delayed SWRT.

- **First Come First Serve (FCFS):** schedule jobs according to their arrival times, and jobs that arrive earlier are scheduled firstly. It is the baseline for inter-job scheduling solutions.
- **Minimum Remaining Time First (MRTF):** select the job with minimum remaining completion time to schedule whenever a new job arrives or the job being scheduled finishes, which is a commonly used inter-job/inter-coflow scheduling approach [25,28,35,39].
- **Minimum Weighted Remaining Time First (MW- RTF):** calculate the ratio of the remaining completion time of each job in the network to its weight whenever a job arrives or the job being scheduled finishes, and choose the job with the smallest ratio to schedule. This is reasonable because prioritizing jobs with high weights and short completion times can help to reduce WJCT.

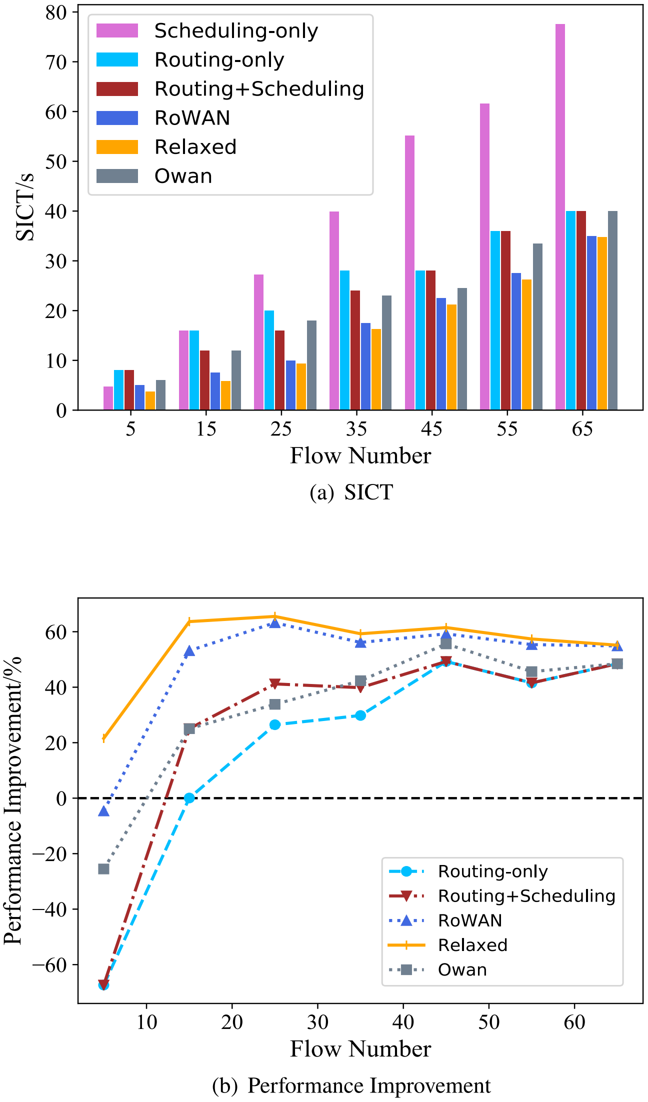
**Job workloads.** Similar to [18,24], we use synthetic models to generate Geo-DML job workloads, such as flow number, model size and computation time. Specifically, as mentioned in the *Introduction* Section, the model size can be up to several hundred GB, so we randomly choose model sizes from 0.1 GB to 100 GB for all simulations. Besides, we do not need to know the global model synchronization architecture, but assume the flows over WAN are given once jobs are deployed. Considering the limited number of DCs and links in the networks, the flow number is set to 5 to 70. Furthermore, the computation time ranges from several seconds to several minutes based on the model size and the maximum number of jobs in the network is 100.



(a) B4

(b) Internet2

**Fig. 3.** Comparison with different link reconfiguration ratios.



(a) SICT

(b) Performance Improvement

**Fig. 4.** Comparison with different flow numbers in B4.

Besides, preemption is not allowed in our algorithms. This means that no flow or job can interrupt the flow or job being scheduled.

**Performance metrics.** For intra-job scheduling, we only focus on the single iteration communication time (SICT) which denotes the global model synchronization communication time of a job's single iteration. For inter-job scheduling, we use weighted job completion time (WJCT) to show the performance of different solutions. Besides, we also define *performance improvement* of solution 2 compared to solution 1 as $\frac{t_1-t_2}{t_1}$, where $t_1$ represents the SICT/WJCT of the baseline, and $t_2$ is the result derived by other algorithms.

### 6.2. Intra-job scheduling comparison

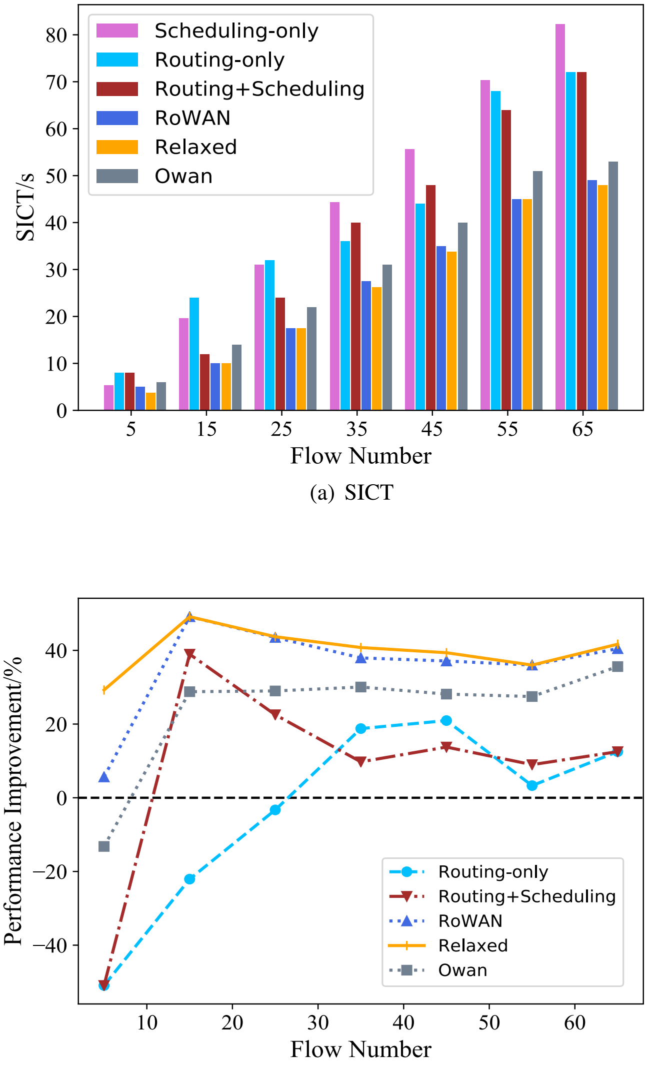
In this section, we compare the SICTs of different intra-job scheduling approaches in two topologies.

**Impact of link reconfiguration ratio** $r$. We evaluate the impact of link reconfiguration ratio $r$ on the SICT. We set $r$ to vary from 0% to 100%, which means the maximum bandwidth capacity of each link can vary from 50 Gbps to 100 Gbps. Note that, if a link gets more wavelengths, the wavelengths of its neighbor links will decrease accordingly. Fig. 3 shows the results in B4 and Internet 2 respectively.

It is clear that SICT in the static topology ($r = 0\%$) is greater than that in the reconfigurable topology ($r > 0\%$), which verifies that scheduling solution controlling network layer and optical layer could improve the performance in contrast to the solutions controlling only network layer. Besides, SICT decreases as $r$ increases in both topologies. This is because more bandwidth can be borrowed from adjacent links to serve the selected paths for flows, which is beneficial to the data transmission.

**Impact of flow number.** We also compare the impacts of different flow numbers on the performance. The results in two WAN topologies are depicted in Figs. 4 and 5, respectively. In both networks, for all solutions, as the flow number increases, so does SICT. This is because more flows will compete the bandwidth, which will prolong the completion time.

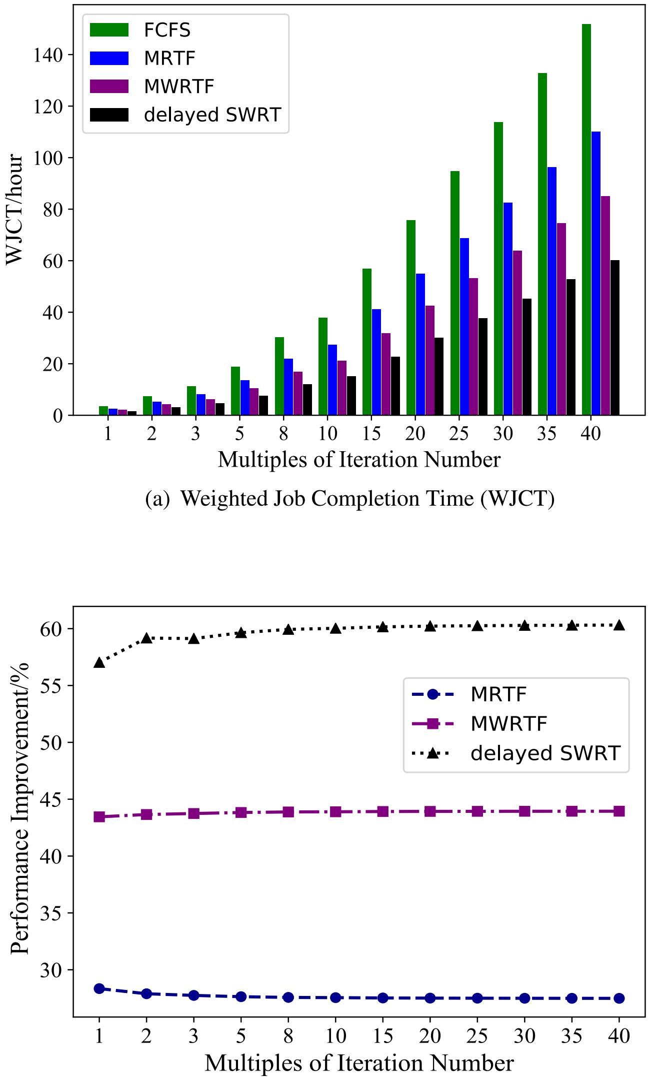
Besides, it can be seen from Figs. 4(b) and 5(b) that the performance improvement of RoWAN compared to the baseline is always greater than other solutions except the relaxed one, and it shows an upward trend when flow number is small, and then shows a decrease trend when flow number increases, despite some fluctuations. This is because with growing of flow number, more flows will compete the limited bandwidth, the



(a) SICT

(b) Performance Improvement

**Fig. 5.** Comparison with different flow numbers in Internet2.



(a) Weighted Job Completion Time (WJCT)

(b) Performance Improvement

**Fig. 6.** Comparison with different iteration numbers.

reconfigurable topology provides less adjustable wavelengths for each flow. However, with increasing of flow number, the value deprived by RoWAN in both networks is getting closer to the theoretical upper bound, which verifies the effectiveness of RoWAN. In general, in B4 network, the SICT obtained by RoWAN is reduced by 48.2%, 33.4%, 27.9%, 23% on average compared with Scheduling-only, Routing-only, Routing + Scheduling a-nd Owan. In Internet2 network, the SICT of RoWAN is 35.69%, 35.85%, 28.75%, 15.54% less than Scheduling-only, Routing-only, Routing + Scheduling and Owan.

### 6.3. Inter-job scheduling comparison

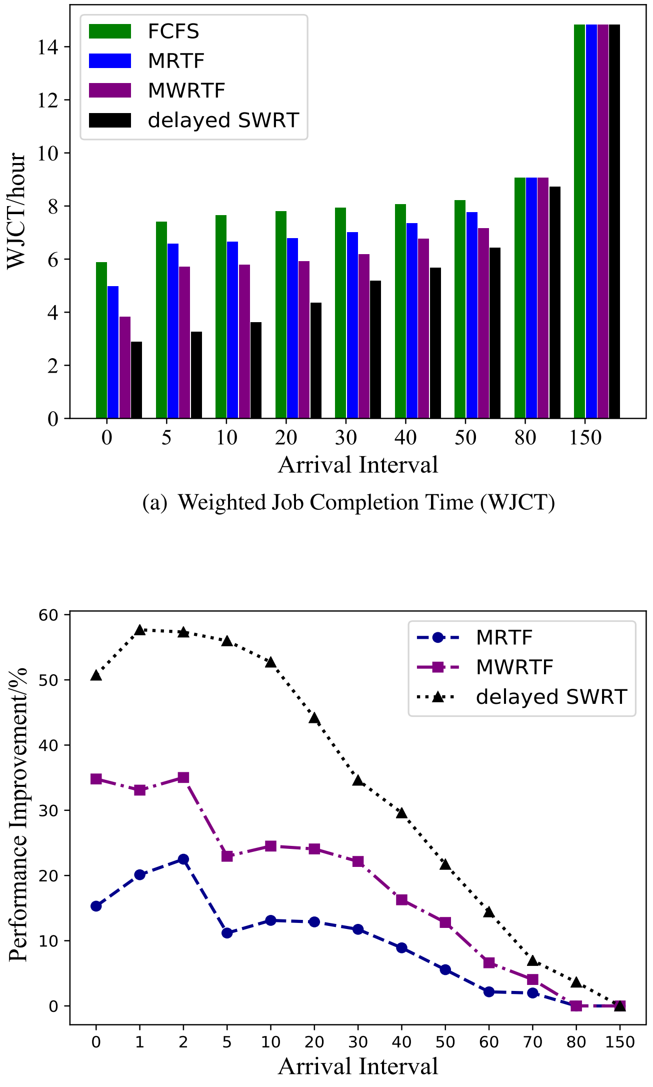
In this section, we compare the WJCTs of different inter-job scheduling approaches.

**Impact of iteration number.** In this part, we evaluate the effect of iteration number of Geo-DML job on the WJCT, and Fig. 6 shows the results. We change the iteration number of every job in multiples, and the abscissa in the figure represents the multiple of each job's original iteration number.

From Fig. 6(a), we can see that as iteration number increases, WJCT of three solutions also increases. This is reasonable because the increase of iteration number means the completion time of each job prolongs, resulting in an increase in WJCT. It can be seen that when iteration number is doubled, the WJCT of delayed SWRT is about 59.2%, 43.4%, 27.5% less than FCFS, MRTF and MWRTF. While when iteration number becomes 10 times the original one, the values become 60.03%, 44.83% and 28.76%. Besides, with the growing of iteration number, the performance improvement of delayed SWRT compared to the baseline almost shows a stable increase (see Fig. 6(b)) and is much greater than that of the other two schemes. This is because delayed SWRT effectively utilizes the interlaced nature of global model synchronization communication and local model update computation to schedule jobs.

**Impact of arrival interval.** In order to evaluate the online performance of delayed SWRT, we conduct a series of simulations with different job arrival intervals, and the results are shown in Fig. 7.

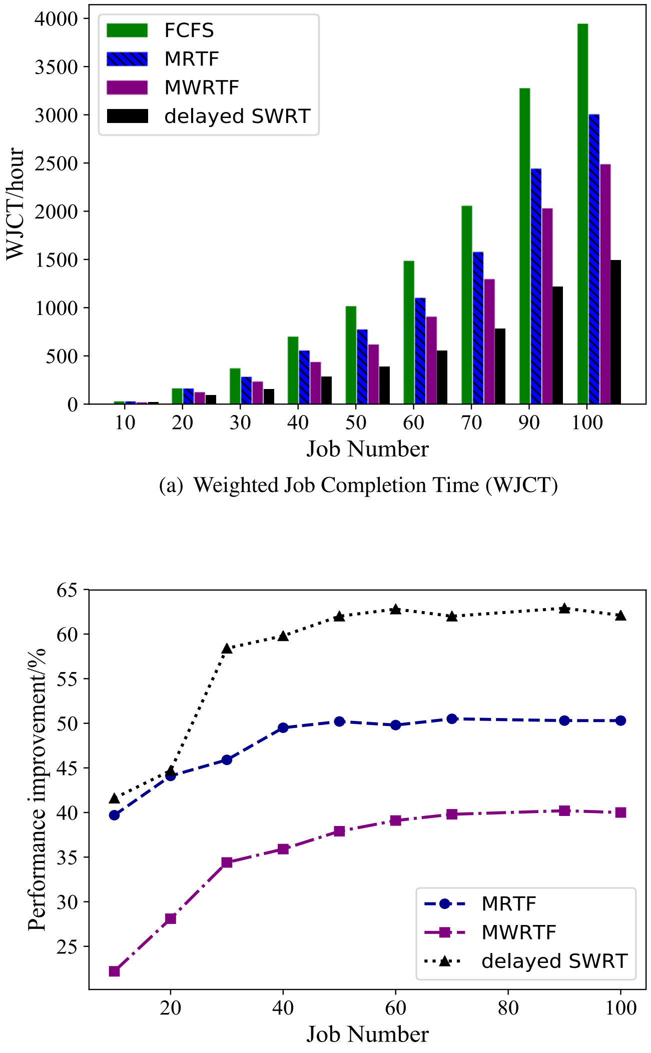
As depicted in Fig. 7(a), delayed SWRT performs better than the other three solutions almost in all scenarios. Notably, the gap between delayed SWRT and the others gradually decreases as the arrival interval increases, and when the arrival interval equals to 150 s, the results are the same. This is because when job arrival interval is longer than the completion time of a job, no



(a) Weighted Job Completion Time (WJCT)

(b) Performance Improvement

**Fig. 7.** Comparison with different job arrival intervals.



(a) Weighted Job Completion Time (WJCT)

(b) Performance Improvement

**Fig. 8.** Comparison with different job numbers.

matter which algorithm is used, the jobs will be executed in the order of arrival time. Besides, Fig. 7(b) indicates the performance improvement shows a downward trend when arrival interval changes from 5 s to 150 s, and finally becomes 0.

**Impact of job number.** To investigate how the performance of delayed SWRT is influenced by the job number, we randomly set the arrival time and iteration number of each job. Fig. 8 shows the simulation results. We can make the following observations from this figure.

As shown in Fig. 8(a), WJCTs of all solutions increase with the job number and delayed SWRT always outperforms FCFS, MRTF and MWRTF. For example, when job number is 90, the performance of delayed SWRT is up to 62.89%, 50.26%, 40.17% compared to FCFS, MRTF and MWRTF. Fig. 8(b) shows that improvement performance almost keeps increasing stably as job number grows, and the improvement performance of delayed SWRT is consistently greater than that of other two solutions. The reason why delayed SWRT performs the best is that delayed SWRT is executed once a job being transmitting data switches into the computation stage, which makes good use of the network resource during the job's computation stage.

## 7. Conclusion

Our work is motivated by the widespread use of Geo-DML and the reconfigurable topology in optical WAN. In this paper, we propose to optimize Geo-DML by combining scheduling in the network layer and topology reconfiguration in the optical layer. Firstly, as for intra-job scheduling problem, we provide a deterministic rounding based heuristic algorithm RoWAN, which decides the network topology and the routing and rate allocation for a Geo-DML job. Then, as for inter-job scheduling problem, we propose delayed SWRT that schedules DML jobs according to their priorities defined by their weights and remaining completion time. In the end, our simulations for real-world topologies indicate that the two algorithms can significantly reduce the communication time of single iteration and the WJCT of multiple Geo-DML jobs, respectively. However, the time complexity of delayed SWRT is proportional to the product of the job number and the total iteration number. Therefore, a lower time complexity inter-job scheduling solution is the future work we are interested in. In addition, real job traces, the heterogeneous computing power and training data size between DCs will also be considered in our future research.

## CRediT authorship contribution statement

**Ling Liu:** Conceptualization, Methodology, Software, Writing-Original Draft, Validation. **Hongfang Yu:** Supervision, Funding acquisition, Project administration. **Gang Sun:** Supervision, Validation, Formal analysis, Writing - review & editing. **Long Luo:** Investigation, Writing - review & editing. **Qixuan Jin:** Writing - review & editing. **Shouxi Luo:** Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This research was partially supported by the National Key Research and Development Program of China (2019YFB1802800), PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications, People's Republic of China (PCL2018KP001).

## References

[1] Z. Zhang, J. Geiger, J. Pohjalainen, A.E.-D. Mousa, W. Jin, B. Schuller, Deep learning for environmentally robust speech recognition: An overview of recent developments, ACM Trans. Intell. Syst. Technol. (TIST) 9 (5) (2018) 49.

[2] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, IEEE Comput. Intell. Mag. 13 (3) (2018) 55–75.

[3] K. Shuang, Z. Zhang, J. Loo, S. Su, Convolution–deconvolution word embedding: An end-to-end multi-prototype fusion embedding method for natural language processing, Inf. Fusion 53 (2020) 112–122.

[4] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A.L. Yuille, L. Fei-Fei, Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 82–92.

[5] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al., B4: Experience with a globally-deployed software defined wan, in: ACM SIGCOMM Computer Communication Review, Vol. 43, ACM, 2013, pp. 3–14.

[6] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, R. Wattenhofer, Achieving high utilization with software-driven wan, in: ACM SIGCOMM Computer Communication Review, Vol. 43, ACM, 2013, pp. 15–26.

[7] H. Hu, D. Wang, C. Wu, Distributed machine learning through heterogeneous edge systems, 2019, arXiv preprint arXiv:1911.06949.

[8] J. Chen, X. Ran, Deep learning with edge computing: A review, Proc. IEEE 107 (8) (2019) 1655–1674.

[9] J. Mills, J. Hu, G. Min, Communication-efficient federated learning for wireless edge intelligence in iot, IEEE Internet Things J. (2019).

[10] W. Xiao, R. Bhardwaj, R. Ramjee, M. Sivathanu, N. Kwatra, Z. Han, P. Patel, X. Peng, H. Zhao, Q. Zhang, et al., Gandiva: Introspective cluster scheduling for deep learning, in: 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18), 2018, pp. 595–610.

[11] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G.R. Ganger, P.B. Gibbons, O. Mutlu, Gaia: Geo-distributed machine learning approaching {LAN} speeds, in: 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17), 2017, pp. 629–647.

[12] I. Cano, M. Weimer, D. Mahajan, C. Curino, G.M. Fumarola, Towards geo-distributed machine learning, 2016, arXiv preprint arXiv:1603.09035.

[13] M. Li, D.G. Andersen, J.W. Park, A.J. Smola, A. Ahmed, V. Josifovski, J. Long, E.J. Shekita, B.-Y. Su, Scaling distributed machine learning with the parameter server, in: 11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14), 2014, pp. 583–598.

[14] R. Hong, A. Chandra, Decentralized distributed deep learning in heterogeneous wan environments, in: Proceedings of the ACM Symposium on Cloud Computing, ACM, 2018, p. 505.

[15] R. Hong, A. Chandra, Dlion: decentralized distributed deep learning in micro-clouds, in: 11th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 19), 2019.

[16] Q. Luo, J. He, Y. Zhuo, X. Qian, Heterogeneity-aware asynchronous decentralized training, 2019, arXiv preprint arXiv:1909.08029.

[17] L. Liu, H. Yu, G. Sun, H. Zhou, Z. Li, S. Luo, Online job scheduling for distributed machine learning in optical circuit switch networks, Knowl.-Based Syst. (2020) 1–13.

[18] X. Jin, Y. Li, D. Wei, S. Li, J. Gao, L. Xu, G. Li, W. Xu, J. Rexford, Optimizing bulk transfers with software-defined optical wan, in: Proceedings of the 2016 ACM SIGCOMM Conference, ACM, 2016, pp. 87–100.

[19] L. Luo, K.-T. Foerster, S. Schmid, H. Yu, Deadline-aware multicast transfers in software-defined optical wide-area networks, IEEE J. Sel. Areas Commun. (2020).

[20] B. Tian, C. Tian, H. Dai, B. Wang, Scheduling coflows of multi-stage jobs to minimize the total weighted job completion time, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 2018, pp. 864–872.

[21] X. Lyu, C. Ren, W. Ni, H. Tian, R.P. Liu, E. Dutkiewicz, Optimal online data partitioning for geo-distributed machine learning in edge of wireless networks, IEEE J. Sel. Areas Commun. 37 (10) (2019) 2393–2406.

[22] C. Hardy, E. Le Merrer, B. Sericola, Md-gan: Multi-discriminator generative adversarial networks for distributed datasets, in: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2019, pp. 866–877.

[23] S. Wang, T. Tuor, T. Salonidis, K.K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1205–1221.

[24] L. Luo, K.-T. Foerster, S. Schmid, H. Yu, Dartree: deadline-aware multicast transfers in reconfigurable wide-area networks, in: Proceedings of the International Symposium on Quality of Service, ACM, 2019, p. 28.

[25] S. Jia, X. Jin, G. Ghasemiesfeh, J. Ding, J. Gao, Competitive analysis for online scheduling in software-defined optical wan, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.

[26] C. Zhang, J. Li, H. Wang, A. Guo, C. Janz, Evaluation of dynamic optical service restoration on a large-scale roadm mesh network, IEEE Commun. Mag. 57 (4) (2019) 138–143.

[27] J. Kundrát, O. Havliš, J. Jedlinský, J. Vojtěch, Opening up roadms: Let us build a disaggregated open optical line system, J. Lightwave Technol. 37 (16) (2019) 4041–4051.

[28] Y. Guo, Z. Wang, H. Zhang, X. Yin, X. Shi, J. Wu, Joint optimization of tasks placement and routing to minimize coflow completion time, J. Netw. Comput. Appl. 135 (2019) 47–61.

[29] S. Liu, L. Chen, B. Li, Siphon: expediting inter-datacenter coflows in wide-area data analytics, in: 2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18), 2018, pp. 507–518.

[30] X.S. Huang, X.S. Sun, T.S. Ng, Sunflow: Efficient optical circuit scheduling for coflows, in: The 12th International on Conference on Emerging Networking EXperiments and Technologies, Irvine, CA, USA, 2016, pp. 297–311.

[31] J. Cipar, Q. Ho, J.K. Kim, S. Lee, G.R. Ganger, G. Gibson, K. Keeton, E. Xing, Solving the straggler problem with bounded staleness, in: Presented As Part of the 14th Workshop on Hot Topics in Operating Systems, 2013.

[32] H. Lim, D.G. Andersen, M. Kaminsky, 3lc: Lightweight and effective traffic compression for distributed machine learning, 2018, arXiv preprint arXiv:1802.07389.

[33] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, R. Pedarsani, Robust and communication-efficient collaborative learning, in: Advances in Neural Information Processing Systems, 2019, pp. 8386–8397.

[34] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, K. Ramchandran, Speeding up distributed machine learning using codes, IEEE Trans. Inform. Theory 64 (3) (2017) 1514–1529.

[35] Y. Zhao, K. Chen, W. Bai, M. Yu, C. Tian, Y. Geng, Y. Zhang, D. Li, S. Wang, Rapier: Integrating routing and scheduling for coflow-aware data center networks, in: 2015 IEEE Conference on Computer Communications (INFOCOM), IEEE, 2015, pp. 424–432.

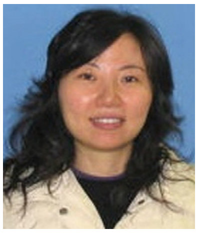
[36] Y. Chen, J. Wu, Multi-hop coflow routing and scheduling in data centers, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.

[37] J.K. Lenstra, A.R. Kan, P. Brucker, Complexity of machine scheduling problems, in: Annals of Discrete Mathematics, Vol. 1, Elsevier, 1977, pp. 343–362.

[38] E.J. Anderson, C.N. Potts, On-line scheduling of a single machine to minimize total weighted completion time, in: Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2002, pp. 548–557.

[39] T. Zhang, R. Shu, Z. Shan, F. Ren, Distributed bottleneck-aware coflow scheduling in data centers, IEEE Trans. Parallel Distrib. Syst. (2018).

**Ling Liu** is pursuing her Doctor degree in Communication and Information System at University of Electronic Science and Technology of China. Her research interests include distributed machine learning and network scheduling.



**Hongfang Yu** received her B.S. degree in Electrical Engineering in 1996 from Xidian University, her M.S. degree and Ph.D. degree in Communication and Information Engineering in 1999 and 2006 from University of Electronic Science and Technology of China, respectively. From 2009 to 2010, she was a Visiting Scholar at the Department of Computer Science and Engineering, University at Buffalo (SUNY). Her research interests include network survivability, network security and next generation Internet.

**Gang Sun** is an associate professor of Computer Science at University of Electronic Science and Technology of China (UESTC). His research interests include network virtualization, cloud computing, high performance computing, parallel and distributed systems, ubiquitous/pervasive computing and intelligence and cyber security.

**Long Luo** is currently working toward the Ph.D degree from the University of Electronic Science and Technology of China (UESTC). She received the BS degree in communication engineering from Xi'an University of Technology in July 2012 and MS degree in communication engineering from the UESTC in July 2015. Her research interests include datacenters, cloud computing, software-defined networks, and datadriven networking.

**Qixuan Jin** is pursuing his Master degree in Communication and Information System at University of Electronic Science and Technology of China. His research interests include distributed machine learning and algorithms.

**Shouxi Luo** received his B.S. degree in Communication Engineering and Ph.D. degree in Communication and Information System from University of Electronic Science and Technology of China in 2011 and 2016, respectively. From Oct. 2015 to Sep. 2016, he was an Academic Guest at the Department of Information Technology and Electrical Engineering, ETH Zurich. His research interests include data center networks and software-defined networks.