ANN-Assisted Scheduling Method for Bulk Data Transfers in Optical Computing Power Networks

Xiao Lin*, Huihuang Lin*, Chenxi Zhang*, Jun Li†, Keqin Shi‡, Weiqiang Sun‡, Weisheng Hu‡
*College of Physics and Information Engineering,
Fuzhou University, Fuzhou, China, {linxiaocer,211127022,181900432}@fzu.edu.cn
†School of Electronic and Information Engineering, Soochow University, Suzhou, China, ljun@suda.edu.cn
‡State Key Laboratory of Advanced Optical Communication Systems and Networks,
Shanghai Jiao Tong University, Shanghai, China, {keqinshi,sunwq,wshu}@sjtu.edu.cn

***Abstract*—ANN-assisted scheduling method is presented to schedule bulk transfers across optical computing power networks in a single-path/SnF-multi-path manner. Studies demonstrate that the method outperforms the conventional methods in terms of blocking probability and computation time.**

***Index Terms*—Artificial neural network, bulk data transfer, computing power network, multi-path routing, optical network, store-and-forward.**

I. Introduction

Like the steam power in the steam age and the electric power in the electrical age, computing power (CP) has become a new engine of the digital economy age. Whether you are going to a gym, watching a video, or driving a car, every human activity can be empowered by the CP. The rise of intelligent services and applications (e.g., smart city, remote education and AR/VR) has fueled an unprecedented demand for the CP [1]. Computing power network (CPN) has recently emerged to efficiently leverage computing power resources geographically distributed in cloud computing centers, edge computing servers and smart devices. Electric power can be physically scheduled and moved in the power grid. However, the CP only exists in the computing infrastructures and cannot be physically scheduled to its remote users. Instead of scheduling the CP, a CPN should move data from its users to CP nodes (e.g., datacenters and supercomputer centers) where the data can be stored and processed on-the-fly. A fundamental challenge faced by the CPN is how to meet the growing demands for bulk data transfers between its users and CP nodes as well as between CP nodes [2].

Data transfers, generated from online data analysis, distributed AI training, disaster detecting and the like, are data-intensive and bandwidth-hungry [3]. Besides, they often have tight deadline constraints. Violating the deadline would be in breach of the service level agreement (SLA) contract and hence incur the penalties. As a result, CPN operators have to provision these transfers over high-bandwidth end-to-end (E2E) connections as soon as possible. However, network background traffic is typically space- and time-varying [4]. For example, the network traffic in an enterprise district peaks during working hours, whereas that in a residential district peaks after the hours. Such traffic fluctuation makes it difficult to provision such high-bandwidth and long-last connections. To accommodate the growing transfer demands, CPN operators have to constantly increase link capacity even if the average utilization is quite low [5].

Our approach to overcome the challenge faced by bulk data transfers in the CPN is based on two observations:

First, data are often transferred using single-path routing (SP). A bulk data flow may occupy a large proportion of link bandwidth for a long holding time, which may invoke the congestion on the single path and increase the blocking probability in the CPN, especially at peak hours. Alternatively, multi-path routing (MP) can distribute the data over multiple paths and hence alleviate the congestion on each single path. However, the more the paths, the more the bandwidth usage. The number of distributing paths should be carefully decided [6]. Inappropriate MP decision may escalate the network congestion and degrade the network performance.

Second, store-and-forward (SnF) approach is another solution to alleviate the network congestion [7]. SnF leverages intermediate nodes (e.g., datacenters) to temporarily store the data when the next hop is congested and forward the data at a later time. The use of SnF can split the E2E connection into multiple segments, which relaxes both bandwidth constraint and holding time constraint on each segment. However, SnF incurs extra storing delay, which may be unacceptable for some delay-sensitive users.

Intuitively, MP and SnF are attractive for the CPN, as complementary to the conventional SP transfers. However, the complex nature of MP and SnF prohibits them from becoming a practical solution. On one hand, the interplay between traffic splitting and routing in MP leads to an NP-hard problem, which has significant impacts on the performance of MP scheduling [8]. On the other hand, the interplay among storage location, route selection and temporal scheduling in SnF leads to another NP-hard problem, which also has signif-

This work was supported by the National Natural Science Foundation of China (61901118 and 62101372), the Project of Jiangsu Engineering Research Center of Novel Optical Fiber Technology and Communication Network (No. SDGC2232) and the Project of the State Key Laboratory of Advanced Optical Communication Systems and Networks (2023GZKF020), Shanghai Jiao Tong University, China.

icant impacts on the performance of SnF scheduling [9]. The combination of SnF and MP leads to a more complex SnF-MP problem, whose problem size would exponentially expand with the network size, storage locations and the number of distributing paths. As a result, the conventional optimization methods may become computationally impractical to solve the SnF-MP problem, especially under dynamic traffic and large network scenarios [10].

Machine learning (ML) has attracted a strong attention for its excellent performance in solving complex problems. However, prior studies directly learnt the optimal results, such as next hop and route selection under static network scenarios [11]. They may suffer from unsatisfied accuracy when their network scenarios change.

In this paper, we consider how to schedule bulk data transfers across the optical CPN (OCPN). On one hand, the optical layer can provide high-bandwidth and low-overhead lightpaths to carry bulk data. On the other hand, SnF-MP is complementary to the SP transfers and hence improves the flexibility of scheduling. Moreover, an artificial neural network (ANN) model is used to simplify the SnF-MP problem, and an ANN-assisted hybrid-path scheduling method (AHP) is presented. Our contributions are summarized as follows:

1) The conventional methods attempted to solve the scheduling problem on the entire network and hence suffered from high complexity. Instead, AHP decouples the SnF-MP problem into three sub-problems: i) SP/MP decision, ii) traffic splitting and routing, iii) SnF-SP scheduling. This simplifies the SnF-MP problem.
2) Instead of using an ML model to predict the scheduling results directly, AHP uses the ANN model to predict the optimal SP/MP decision. Moreover, instead of involving the entire network, the ANN model also predicts the most relevant nodes that should be involved in the SnF-MP problem. This further reduces the size of the problem and the difficulty of solving the problem.
3) Simulation results show that AHP can obtain better network performance and shorter computation time than the conventional scheduling methods.

The remainder of this paper is structured as follows: Sect. II illustrates the ANN model. Sect.III presents AHP. Then, Sect.IV gives the evaluation and discussion. Sect.V draws a conclusion.

II. ANN Model

Prior studies learnt the optimal solutions directly [11]. Instead, we use the ANN model to assist the SP/MP decision-making and the traffic splitting and routing optimization in the SnF-MP problem, which are formulated as a multicategory classification task.

A. Dataset

In practice, the admission of a transfer request mainly depends on the utilization of a few routing-related links rather than that of the entire network [6]. Let $R_{(s,d)}$ denote the set of the pre-computed link-disjoint routes from the source s to the destination d, where $R_{(s,d)} = \{R_1, ..., R_K\}$. Intuitively, data are more likely to be delivered in an SP manner if the utilization of the links along R_1 is lower. Thus, the utilization of the links along R_1 should be taken into account. To assist the SP/MP decision-making, we define two metrics to measure the usage of wavelength resources on R_1, i.e., Eq. (1), and the availability of the holding time on R_1, i.e., Eq. (2). We have

$$\alpha_1 = \frac{\sum\limits_{i,j \in R_1} w^{idle}_{(i,j)}}{\sum\limits_{i,j \in R_1} w_{(i,j)}} \tag{1}$$

$$\beta_1 = \frac{\sum\limits_{i,j \in R_1} t^{idle}_{(i,j)}}{\sum\limits_{i,j \in R_1} t^{ddl}} \tag{2}$$

where $w^{idle}_{(i,j)}$ and $w_{(i,j)}$ denote the number of idle wavelengths and the total number of wavelengths on link (i,j), $t^{idle}_{(i,j)}$ denotes the idle time of link (i,j) and t^{ddl} denotes the deadline, respectively. The idle time $t^{idle}_{(i,j)}$ is defined as the time interval between the time instant when link (i,j) is available and t^{ddl}.

Moreover, the lower the utilization of $R_{(s,d)}$, the fewer the distributing paths in MP, the fewer the nodes that should be involved in the SnF-MP optimization. To assist the traffic splitting and routing optimization, two metrics are introduced to measure the usage of wavelength resources on $R_{(s,d)}$, i.e., Eq. (3), and the availability of the holding time on $R_{(s,d)}$, i.e., Eq. (4). We have

$$\alpha = \frac{\sum\limits_{i,j \in R_{(s,d)}} w^{idle}_{(i,j)}}{\sum\limits_{i,j \in R_{(s,d)}} w_{(i,j)}} \tag{3}$$

$$\beta = \frac{\sum\limits_{i,j \in R_{(s,d)}} t^{idle}_{(i,j)}}{\sum\limits_{i,j \in R_{(s,d)}} t^{ddl}} \tag{4}$$

We run extensive simulations of an MP scheduling method [12] to generate training samples. In each simulation, once a request is admitted, we record s, d, α_1, β_1, α and β, which are used as the input of ANN model. We classify the scheduling result into three labels, as: i) if the request is scheduled in an SP manner, the label Y is 0; ii) if the request is scheduled in an MP manner and the nodes on its distributing paths belonging the l-hop direct neighbor nodes of its shortest path, Y is 1; iii) if the nodes on its distributing paths belonging the 2-hop neighbor or farther nodes of its shortest path, Y is 2.

Therefore, we can make SP/MP decision based on whether or not the predicted result $\hat{Y}$ is zero. Moreover, we can decide how many neighbor nodes should be involved in the traffic splitting and routing optimization if $\hat{Y}$ is non-zero. Note that to reduce the problem size, only 2-hop neighbor nodes would be involved in the optimization process when Y is 2. This greatly simplifies the SnF-MP problem.

TABLE I: Comparisons of the Classification Results

	Accuracy	Precision	Recall	F1
ANN	0.9231	0.8535	0.8555	0.8545
kNN	0.8888	0.7904	0.7908	0.7906
SVM	0.8569	0.7324	0.7215	0.7199

B. Architecture

Our ANN model consists of one input layer with 6 neurons, four hidden layers and one output layer with 3 neurons. The four hidden layers are fully connected and have 500, 250, 150 and 100 neurons, respectively. The multi-classification cross-entropy is used as the loss function. The ReLU function and the softmax function are used as the activation functions for the hidden layers and the output layer, respectively.

C. Training

We use the Adam optimizer to automatically adjust the learning rate in the training. We use 80% of the data for training, while the remaining 20% is used for testing. Additionally, 20% of the training data is used as the validation data to evaluate the performance of model and prevent overfitting.

D. Testing

To investigate the classification performance, we compare our ANN model with two classification methods:

1) The **k-Nearest Neighbors (kNN)** method performs the cross-validation to determine the best value of k.
2) The **Support Vector Machines (SVM)** method based on RBF kernel function and the default value of C performs its best classification performance.

The classification performance of three classification methods is shown in Table I. The results show that our ANN model has the best classification performance under all evaluation metrics. Thus, our ANN model can assist effectively the SnF-MP problem based on the network status.

III. ANN-Assisted Hybrid-Path Scheduling Method

A. Network Model and Assumptions

In Fig. 1, the OCPN employs a wavelength-division multiplexing (WDM) networking infrastructure. A typical CP node

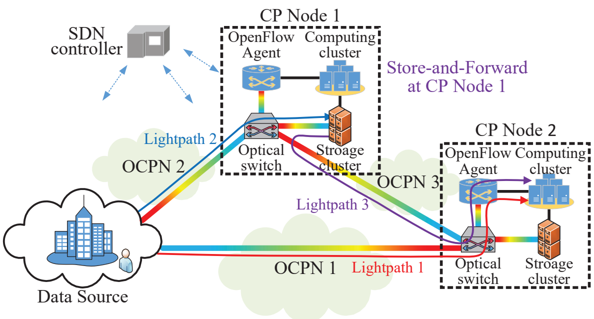


Fig. 1: An OCPN network scenario.

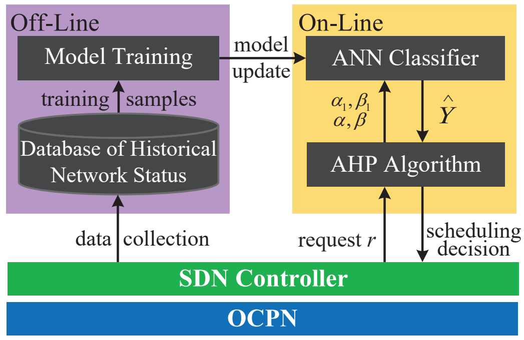


Fig. 2: Schematic of the AHP Method.

has an optical switch, a computing cluster, a storage cluster and an OpenFlow Agent (OFA). An SDN controller maintains up-to-date a global view of the entire OCPN and all admitted requests via the OFAs. Each CP node is capable of wavelength conversion and temporary storage.

Let s, d, F and t^{ddl} denote the source, the destination, the file size and the deadline of a transfer request r, respectively. Thus, r can be represented as a tuple $r = \{s, d, F, t^{ddl}\}$. Requests arrive randomly and follow a Poisson process. Let λ denote the arrival rate per hour. File sizes follow a negative exponential distribution with a mean size of F TB. The transmission of each request uses a wavelength. The holding time H for a request r, which is the duration it occupies a wavelength, is determined by dividing F by the data rate of a wavelength. To highlight the tightness of deadline, we assume that H is equal to t^{ddl}. Upon arrival, r has to start transmitting immediately if it is delivered in an SP manner. Compared with H, the processing overhead (e.g., splitting and reassembly) is assumed to be negligible [13].

Consider a request that attempts to transfer data to CP node 2 at time t. Assume that OCPN 1 cannot provision an E2E path with required holding time, while OCPN 3 cannot provision an E2E path with required wavelength at t. The request cannot be transferred in an SP manner. Alternatively, the data can be split into two smaller chunks. On one hand, a chunk can be directly sent to CP node 2 over lightpath 1 with the available holding time. On the other hand, the other can be stored on CP node 1 until OCPN 3 becomes available as long as the deadline constraint is met. SnF-MP provides extra flexibility for provisioning.

B. Overview

We present an AHP method to schedule bulk data transfers over the OCPN, whose schematic is depicted in Fig. 2. On one hand, AHP performs admission control to arriving requests based on the network status and the ANN classifier. Once admitted, AHP guarantees the request completion before the deadline. On the other hand, the historical network status information will be collected and used as new training samples. The ANN classifier will be updated periodically by learning from these new samples. As a result, the ANN classifier can quickly

Algorithm 1 ANN-Assisted Hybrid-Path Scheduling Method (AHP)

1: Input: request $r = \{s, d, F, t^{ddl}\}$, the pre-computed $R_{(s,d)}$
2: Output: $Path$ and $Chunks$
3: Calculate H based on F and the data rate of a wavelength
4: Obtain the link status information $w_{(i,j)}^{idle}$ and $t_{(i,j)}^{idle}$ from the SDN controller given t^{ddl}, where $i, j \in V$
5: Calculate α_1, β_1, α and β using Eqs. (1)-(4)
6: $\hat{Y} \leftarrow \text{ANN}(s, d, \alpha_1, \beta_1, \alpha, \beta)$
7: **if** $\hat{Y}$ = 0 **then**
8: Create an auxiliary graph G_1 of G, whose link weights are the reciprocal of $w_{(i,j)}^{idle}$ and $i, j \in V$
9: Apply the SP algorithm [14] to provision r and find $Path$ from s to d on G_1, and $Chunks \leftarrow H$
10: **else**
11: Apply Algorithm 2 to provision r and find $Path$ and $Chunks$
12: **end if**
13: **if** $Path$ is valid **then**
14: Accept r and **return** $Path$ and $Chunks$
15: **else**
16: Reject r and **return** $Path \leftarrow \varnothing$ and $Chunks \leftarrow \varnothing$
17: **end if**

adapt to the change of the network condition and the traffic pattern. The main features of AHP are twofold as follows:

First, **SnF-MP problem decoupling**. The conventional methods often formulated the MP problems as static optimization problems, where traffic splitting and routing were jointly addressed for static traffic [8]. The use of SnF complicates the conventional spatial routing process and introduces an extra temporal scheduling process. In this case, the optimization methods may become computationally impractical, especially when handling large networks and dynamic traffic. To tackle this, AHP decouples the problem into the SP/MP decision, traffic splitting and routing, and SnF-SP problems, and solves them separately. This greatly simplifies the problem.

Second, **ANN-assisted optimization**. The study showed that a majority of bulk data transfers could be delivered in an SP manner especially when the traffic load is small or medium [6]. The congestion may be escalated if an MP transfer uses too many paths. In most cases, a few paths can suffice to accommodate an MP transfer [15]. Moreover, these paths can be found within the 1-hop or 2-hop neighbor nodes of its shortest path from s to d. These key findings inspire us to further simplify the optimization process. Instead of using a complex optimization to decide the number of paths, AHP uses the ANN model to predict the optimal SP/MP decision. Instead of involving the entire network graph in the traffic splitting and routing optimization, the ANN model predicts the most relevant nodes in the network, which should be involved in the problem. This further reduces the problem size and the difficulty of solving the problem.

Algorithm 1 presents the overall procedure of AHP. Assume

Algorithm 2 SnF Multi-Path Scheduling Algorithm (SnF-MP)

1: Input: r, $t_{(i,j)}^{idle}$ and $\hat{Y}$
2: Output: $Path$ and $Chunks$
3: Initialize: $Path \leftarrow \varnothing$ and $Chunks \leftarrow \varnothing$
4: Create an auxiliary graph G_2 of G, whose link weights are the reciprocal of $t_{(i,j)}^{idle}$ and $i, j \in V$
5: Use Dijkstra's algorithm to calculate a shortest path p from s and d on G_2
6: Create a subgraph G'' of G, where G'' only contains $\hat{Y}$-hop neighbor nodes of the nodes along p
7: Formulate the traffic splitting and routing problem on G'' as an optimization model given in Sect. III-C
8: Solve the optimization model using Gurobi and obtain $Routes$ and $Chunks$, where $r_i \in Routes$ and $h_i \in Chunks$
9: **for all** $h_i \in Chunks$ **do**
10: Apply the SnF-SP algorithm [17] to provision each h_i on its route r_i and find a viable temporal schedule p_i
11: **if** h_i can be provisioned on r_i **then**
12: $Path \leftarrow Path \cup p_i$
13: **else**
14: **return** $Path \leftarrow \varnothing$ and $Chunks \leftarrow \varnothing$
15: **end if**
16: **end for**
17: **return** $Path$ and $Chunks$

that a request r arrives. First, line 3 calculates H based on F and the data rate of a wavelength. Second, line 4 obtains the link status $w_{(i,j)}^{idle}$ and $t_{(i,j)}^{idle}$ from the SDN controller. Third, line 5 uses the link status to calculate the α_1, β_1, α and β using Eqs. (1)-(4). Fourth, line 6 uses the ANN model to predict the optimal SP/MP decision. Fifth, lines 8-9 apply the algorithm [14] to provision r in an SP manner if $\hat{Y}$ is zero. Otherwise, line 11 applies Algorithm 2 to provision r in an SnF-MP manner. Finally, if $Path$ is valid, line 14 accepts r and returns $Path$ and $Chunks$; otherwise, line 16 rejects r.

Algorithm 2 aims to solve the SnF-MP problem. First, line 4 creates an auxiliary graph G_2 of G, whose link weights are the reciprocal of $t_{(i,j)}^{idle}$. Second, line 5 uses Dijkstra's algorithm to calculate a shortest path p from s and d on G_2. Third, line 6 creates a $\hat{Y}$-hop-neighborhood subgraph G'' of G. Specifically, G'' only contains $\hat{Y}$-hop neighbor nodes of the nodes along p and the links connecting them. Fourth, line 7 formulates the traffic splitting and routing problem on G'' as an optimization model, which will be elaborated upon in Sect. III-C. Fifth, line 8 uses a commercial solver [16] to solve the optimization model and obtain the set of alternate routes $Routes$ and the holding time set of chunks $Chunks$, where $r_i \in Routes$ and $h_i \in Chunks$. Finally, lines 9-16 apply the SnF-SP algorithm [17] to provision each h_i on its corresponding route r_i and find a viable temporal schedule p_i. If h_i can be provisioned on r_i, line 12 adds p_i to $Path$. Otherwise, line 14 returns invalid $Path$ and $Chunks$.

C. Problem Formulation

Herein, we formulate the traffic splitting and routing problem as an optimization problem, which aims to save the wavelength resources. To this end, we consider reduce the hop count of all the distributing paths by means of Eq. (5).

Given:

- $G''(V'', E'')$: the $\hat{Y}$-hop-neighborhood subgraph of the whole network graph G, where V'' is the set of nodes in G'', E'' is the set of links connecting V'', $\{i|i \in V''\}$, $\{j|j \in V''\}$ and $\{(i,j)|(i,j) \in E''\}$;
- $t^{idle}_{(i,j)}$: the idle time of link (i,j);
- K: the maximum number of paths allowed for splitting.

Variables:

- $x_{(i,j)}$: integer variable, denotes the holding time of a chunk using link (i,j);
- $f_{(i,j)}$: binary variable, equals one if link (i,j) is used, and zero otherwise.

Objective:

$$\min \sum_{i,j \in V''} f_{(i,j)} \tag{5}$$

$$s.t. \quad \text{Constraints}\,(6) - (11).$$

Constraints:

1) Eq. (6)-Eq. (8) are flow conservation constraints.

$$\sum_{i \in V''} x_{(s,i)} - \sum_{i \in V''} x_{(i,s)} = H \tag{6}$$

$$\sum_{i \in V''} x_{(d,i)} - \sum_{i \in V''} x_{(i,d)} = -H \tag{7}$$

$$\sum_{i \in V''} x_{(j,i)} - \sum_{i \in V''} x_{(i,j)} = 0, \ \forall j \in V'', j \neq s, j \neq d \tag{8}$$

2) Eq. (9)-Eq. (11) are traffic splitting constraints. Eq. (9) specifies that a request should not distribute the traffic over more than K paths. Eq. (10) specifies that apart from s and d, any intermediate nodes should not split or reassemble the traffic. Eq. (11) specifies that a link should not be used by more than a chunk, whose holding time should not exceed $t^{idle}_{(i,j)}$.

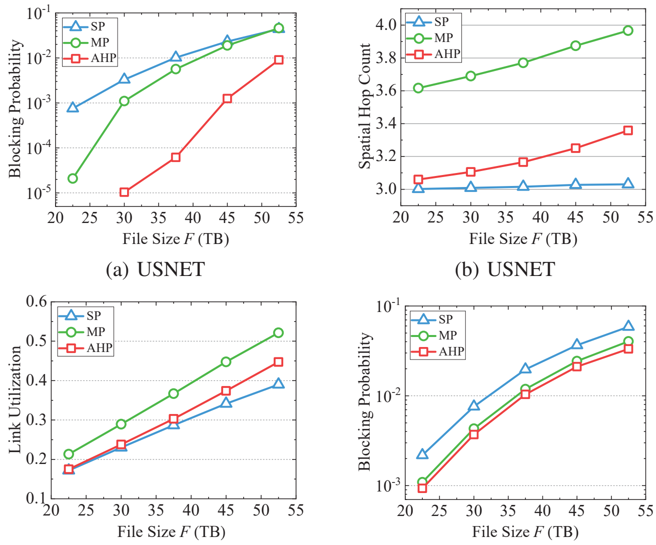
$$\sum_{i \in V''} f_{(s,i)} \leq K, \ \sum_{i \in V''} f_{(i,d)} \leq K \tag{9}$$

$$\sum_{j \in V''} f_{(i,j)} \leq 1, \ \forall i \in V'', i \neq s, i \neq d \tag{10}$$

$$f_{(i,j)} \leq x_{(i,j)} \leq t^{idle}_{(i,j)} \times f_{(i,j)}, \ \forall i,j \in V'' \tag{11}$$

D. Computational Complexity

The aforementioned optimization problem consists of $O(|E''|)$ variables and $O(|E''| + |V''|)$ constraints. The complexity of AHP is mainly decided by its complexity and hence is $O((|E''| + |V''|)^2 \cdot |E''|)$. Compared with the optimization model on the entire network graph G, AHP only needs to handle G'', where $|V''| \ll |V|$ and $|E''| \ll |E|$. It becomes more computationally advantageous to use AHP in large network graphs.



(a) USNET (b) USNET

(c) USNET (d) CENI

Fig. 3: Network performance under various F.

IV. Evaluation

In this section, AHP is compared with the existing methods in dynamic network simulations. We consider the assumptions in Sect. III-A and use the US backbone network (USNET) in simulations. CP node is deployed on each node in USNET. The storage capacity on each CP node for temporary storage is 200 TB. Preliminary results showed that decreasing the storage capacity would degrade the network performance. The results are omitted, due to the limited space. Following the assumptions in [5], 60 Gbps of link bandwidth is dedicated for requests. Each wavelength carries 10 Gbps. We average the results over 20 simulation runs, each with 500,000 requests. Two most related baseline methods are considered as follows:

- **SP** [14]: It is a single-path scheduling method and uses K-shortest-path routing without SnF capability.
- **MP** [12]: It is a multi-path scheduling method and adaptively distributes traffic among K link-disjoint paths without SnF capability.

We first investigate how the network performance changes with F. Herein, K=3, λ=6 and $F \in$ [20,55] TB. Fig. 3(a) shows that AHP has lower blocking probability than SP and MP. This is because AHP is more flexible than SP and MP by distributing traffic in an SnF-MP manner. Note that the blocking probability in AHP is zero when F=22.5 TB. Figs. 3(b) and (c) show that MP detours traffic via long-hop paths and hence has higher link utilization than SP and AHP, but MP suffers from higher blocking probability than AHP. This suggests that AHP can use the wavelength resources more efficiently than the other methods.

We investigate the blocking probability on a different topology, i.e., China Environment for Network Innovations (CENI) [18]. Fig. 3(d) shows that the result in CENI follows similar trends in Fig. 3(a). However, AHP offers less performance advantage over SP and MP in CENI than that in USNET,

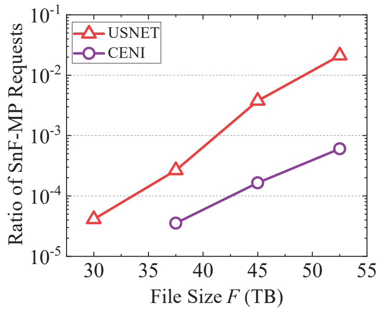


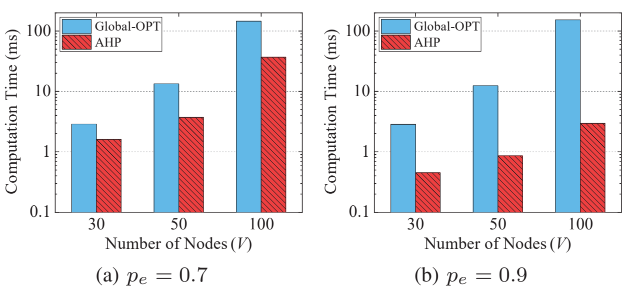
Fig. 4: Ratio of SnF-MP requests in USNET and CENI.

since USNET has higher network connectivity than CENI. Compared with USNET, AHP may find it difficult to provision multiple routing paths in CENI. To study this, we investigate the ratio of the requests that are delivered in an SnF-MP manner and all the generated requests. Fig. 4 shows that more requests can be delivered in an SP manner when F is small. In particular, all the requests use an SP manner in CENI when F=30 TB. More requests use SnF-MP scheduling to reach their destinations with F increasing. However, the increase in USNET is more significant than that in CENI due to the network connectivity.

Finally, we investigate the average computation time to solve the SnF-MP problem on a randomly generated network topology. Herein, a global optimization model (i.e., Global-OPT) is used as a baseline model, where the SnF-MP problem on the entire network is solved by Gurobi. Topologies are randomly generated with a probability p_e, where p_e denotes the probability of an edge between any two nodes. Fig. 5 shows that compared with Global-OPT, AHP can reduce the computation time. Moreover, the larger the value of p_e, the greater the decrease in the computation time in AHP. This is because the network resource becomes more sufficient when p_e increases. In this case, compared with Global-OPT, AHP is more likely to either use SP or solve the problem on small-hop-neighborhood subgraphs, which can save the computation time efficiently.

V. Conclusion

In this paper, an AHP method is presented to schedule bulk data transfers across the OCPN in a hybrid SP/SnF-MP manner. AHP employs the ANN model to make the SP/MP decision and involve the most relevant nodes in the traffic splitting and routing optimization. Studies show that AHP outperforms the conventional SP and MP methods in terms of blocking probability and computation time.



(a) $p_e = 0.7$ (b) $p_e = 0.9$

Fig. 5: Computation time in random network topologies.

References

[1] J. Liu, Y. Sun, J. Su, Z. Li, X. Zhang, B. Lei, and W. Wang, "Computing power network: A testbed and applications with edge intelligence," in *IEEE Conf. Comput. Commun. Wkshps. (INFOCOM WKSHPS)*, pp. 1–2, 2022.

[2] X. Tang, C. Cao, Y. Wang, S. Zhang, Y. Liu, M. Li, and T. He, "Computing power network: The architecture of convergence of computing and networking towards 6G requirement," *China Commun.*, vol. 18, no. 2, pp. 175–185, 2021.

[3] L. Chen, S. Liu, and B. Li, "Optimizing network transfers for data analytic jobs across geo-distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 2, pp. 403–414, 2022.

[4] B. Bao, H. Yang, Y. Wan, Q. Yao, A. Yu, J. Zhang, B. C. Chatterjee, and E. Oki, "Node-oriented traffic prediction and scheduling based on graph convolutional network in metro optical networks," in *Opt. Fiber Commun. (OFC)*, pp. 1–3, 2021.

[5] L. Luo, L. Yu, T. Ma, and H. Yu, "Flexible and efficient multicast transfers in inter-datacenter networks," in *IEEE/ACM Int. Symp. Quality of Service (IWQoS)*, pp. 1–10, 2022.

[6] Y. Liu, F. Zhou, C. Chen, Z. Zhu, T. Shang, and J.-M. Torres-Moreno, "Disaster protection in inter-datacenter networks leveraging cooperative storage," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 2598–2611, 2021.

[7] X. Lin, J. Zou, S. Yue, W. Sun, and W. Hu, "SnF scheduling of multicast transfers across inter-datacenter optical networks," in *IEEE Int. Conf. Commun. (ICC)*, pp. 1–7, 2021.

[8] E. Akin and T. Korkmaz, "Rate-based dynamic shortest path algorithm for efficiently routing multiple flows in SDN," in *IEEE Int. Conf. Commun. (ICC)*, pp. 1–7, 2019.

[9] P. Lu and Z. Zhu, "Data-oriented task scheduling in fixed- and flexible-grid multilayer inter-DC optical networks: A comparison study," *J. Lightwave Technol.*, vol. 35, pp. 5335–5346, Dec. 2017.

[10] L. Luo, K. Foerster, S. Schmid, *et al.*, "Deadline-aware multicast transfers in software-defined optical wide-area networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1584–1599, 2020.

[11] F. Tang, B. Mao, Y. Kawamoto, and N. Kato, "Survey on machine learning for intelligent end-to-end communication toward 6G: From network access, routing to traffic control and streaming adaption," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1578–1598, 2021.

[12] C. Lin, Y. Bi, H. Zhao, Z. Liu, S. Jia, and J. Zhu, "DTE-SDN: A dynamic traffic engineering engine for delay-sensitive transfer," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5240–5253, 2018.

[13] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and scheduling of time-triggered traffic in time-sensitive networks," *IEEE Trans. Industr. Inform.*, vol. 16, no. 7, pp. 4525–4534, 2020.

[14] W. Lu and Z. Zhu, "IR/AR/MR service provisioning in elastic optical networks," in *Int. Conf. Opt. Commun. Netw. (ICOCN)*, pp. 1–3, 2016.

[15] L. Hiryanto, S. Soh, K.-W. Chin, D.-S. Pham, and M. M. Lazarescu, "Multi-path routing in green multi-stage upgrade for bundled-links SDN/OSPF-ECMP networks," *IEEE Access*, vol. 9, pp. 99073–99091, 2021.

[16] "Gurobi Optimization." http://www.gurobi.com, 2023.

[17] X. Lin, S. Yue, X. Wang, W. Sun, M. Veeraraghavan, and W. Hu, "Performance study of an snf scheduling method for bulk data transfers over inter-datacenter WANs," *Opt. Switch. Netw.*, vol. 37, p. 100558, 2020.

[18] "China Environment for Network Innovations (CENI)." https://sdn.ceni.org.cn, 2023.