

Crosstalk-Avoid Virtual Optical Network Embedding Over Elastic Optical Networks With Heterogeneous Multi-Core Fibers

Qihan Zhang¹, Member, IEEE, Xu Zhang¹, Member, IEEE, Xiaoxue Gong¹, Member, IEEE,
and Lei Guo¹, Member, IEEE

Abstract—With the advent of the era of the 5th generation fixed networks (F5G), some bandwidth-hungry services such as 4K/8K video, telemedicine, and virtual reality (VR) gaming are emerging, which will bring more challenges to the optical network. Typically, both flexible bandwidth allocation and dynamic services deployment will encounter a bottleneck when huge volumes of access data are flooded. To break it, virtual optical network embedding (VONE) over the space division multiplexing (SDM) based elastic optical network (EON) using multi-core fiber (MCF) is adopted. However, the problem of inter-core crosstalk (IC-XT) between two adjacent cores in an MCF link has not been solved. In this paper, we first design a heterogeneous MCF (HMCF) structure. The IC-XT can be significantly reduced when virtual networks are embedded over EONs with HMCF. Then some integer linear programming (ILP) models are established to formulate the process of VONE over EONs with HMCF. Next, to avoid the high time complexity of ILP, we propose a novel VONE algorithm supporting joint IC-XT-avoid cores and frequency slots assignment in EONs with HMCF. Finally, simulation results demonstrate that the proposed algorithm can effectively improve 72% of the available frequency ratio (AFR) and reduce 35% of the fragmentation ratio (FR) compared with conventional VONE over MCF-based networks without any IC-XT reducing strategy, which squeezes more bandwidth resources.

Index Terms—Heterogeneous multi-core fiber, inter-core crosstalk, routing, spectrum and core assignment, virtual optical network embedding.

Manuscript received 21 June 2022; revised 27 August 2022; accepted 30 August 2022. Date of publication 2 September 2022; date of current version 20 December 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0206800, the National Natural Science Foundation of China under Grants 62201105, 62075024, and 62025105, in part by the Nature Science Foundation of Chongqing under Grants cstc2019jcyj-msxmX0615 and cstc2021jcyj-msxmX0404, in part by the Chongqing Municipal Education Commission under Grants CXQT21019 and KJQN202100643, in part by the China Postdoctoral Science Foundation under Grant 2021M700563, in part by the Chongqing Postdoctoral Funding Project under Grant 2021XMX3052, and in part by the Open Fund of IPOC (BUPT) under Grant IPOC2019A007. (*Corresponding authors:* Xu Zhang; Xiaoxue Gong.)

Qihan Zhang is with the School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China (e-mail: qianzhang@stumail.neu.edu.cn).

Xu Zhang, Xiaoxue Gong, and Lei Guo are with the Institute of Intelligent Communication and Network Security, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: zhangxu@cqupt.edu.cn; gongxx@cqupt.edu.cn; guolei@cqupt.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JLT.2022.3203861>.

Digital Object Identifier 10.1109/JLT.2022.3203861

I. INTRODUCTION

F5G makes it more possible for the everlasting bandwidth-hungry services like ultra/super-high definition video streaming (4K/8K video), telemedicine, and VR gaming, leading to dynamic bandwidth requirement for the full-fiber connection network [1]. In F5G, multiple services for multiple customer types can be deployed based on a shared infrastructure to enable flexible service deployment and fulfill cost-effectively [2]. Therefore, network slicing or VONE [3] technology is introduced for network flexibility as a tool to migrate current ossified fixed networks towards new F5G networks. However, the currently deployed infrastructure is wavelength division multiplexing (WDM) optical communication systems that rely on single-mode fibers (SMF), in which optical signals are transmitted in parallel through non-overlapping, fixed-spaced channels in the frequency domain [4]. Inherently, the inefficient utilization of the spectrum is too coarse to adapt to the rapid growth of dynamic traffic, which stimulates the concept of the flexible-grid EON. In EONs, the optical signals are customized by employing the optical orthogonal frequency division multiplexing (OOFDM) in several frequency slots, leading to the fine-granularity spectrum allocation and flexible services deployment. Although EONs can significantly improve the ability of dynamic spectrum allocation, the physical capacity limits of the SMF will still be exhausted [4]. To solve the physical capacity limits, a commercially available SMF ribbon cable appears, which evolves a promising technology called SDM by using weakly coupled single-mode MCF [5]. Through SDM, except for the flexible spectral domain, different optical signals can be co-propagated in multiple spatial channels called cores in the same fiber structure. It utilizes the spatial dimension to provide multiple times as much physical capacity as the SMF [6], [7], [8], [9]. The optical networks in project blueSPACE [10] is an embryonic form of F5G, which proves the trends of VONE over space division multiplexed elastic optical networks (SDM-EONs) in the torrent of F5G evolution.

A. Related Work

Intuitively, the VONE over SDM-EONs becomes very complex because of the new increased spatial dimension. Indeed it is more challenging due to the inter-core interference named IC-XT illustrated in Fig. 1 when the same center frequency of

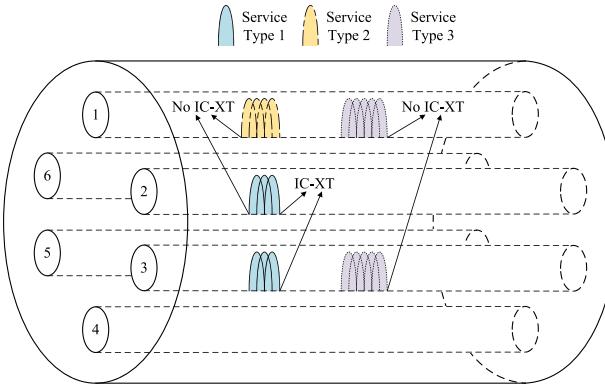


Fig. 1. IC-XT in MCF.

two services in two adjacent cores is utilized, which is one of the primary physical impairment characteristics in SDM-EONs [5]. Since the birth of SDM, academia has been fighting against the IC-XT almost in two types of suppression approaches. One is hardware techniques, in which the fiber micro-structure has been ingeniously designed. These techniques include multi-core photonic crystal fiber [11], hole-assisted MCF [12], trench-assisted MCF [13], unequal core pitch MCF [14], large cladding diameter [15], and heterogeneous MCF (HMCF) [16], leading to the measured IC-XT being less than -30dB. The other one is software methods, in which the input signals have been reasonably allocated. These methods consist of counter-propagating interleaving [17], MIMO electronic equalization [18], conventional IC-XT-avoid, IC-XT-worstcase, and IC-XT-aware routing, spectrum, and core assignment (RSCA) [19], and machine learning introduced RSCA [20], making the IC-XT acceptable. There is no doubt that the combination of the two suppression approaches will further reduce IC-XT dramatically.

Many related works focused on the RSCA problem, which began with Muhammad et al. [21]. They proposed a static RSCA problem for MCF-based networks and took the IC-XT issue into account at first. However, from the perspective of the problem model, the VONE problem increases the constraints of node mapping compared with the RSCA problem. These constraints make the VONE problem more complex, bringing about few related works concentrating on the VONE problem over MCF-based networks. Zhu et al. [22], Liu et al. [23], and Chen et al. [24], [25] all proposed VONE algorithms over substrate EONs with MCF aiming to achieve different objectives, but they did not consider IC-XT. Zhu et al. [26] started the IC-XT-aware algorithms for VONE over EONs with MCF. A trench-assisted MCF was employed to construct the substrate network and a strict IC-XT value was calculated to implement the IC-XT-aware VONE algorithm. Xuan et al. [27] established an IC-XT-worstcase constrained optimization model and an efficient genetic algorithm to determine the optimal schemes of optical network mapping, core, and spectrum assignment. Cheng et al. [28] set up a dynamic distance adaptive IC-XT-worstcase VONE algorithm and Zheng et al. [29] proposed a spectrum-efficiency IC-XT-worstcase VONE approach used for SDM-based elastic optical data center networks. In terms of implementation, Bremes et al. [10] implemented the network

slicing and service orchestration over an experimental 7-core MCF architecture during the blueSPACE project, which encourages more studies on VONE algorithms over MCF considering IC-XT.

B. Our Motivation and Contributions

Except for the attempt of [26], the abovementioned algorithms are all software methods. Inspired by [26], in this paper, we focus on VONE over SDM-EON using HMCF accompanied with an IC-XT-avoid RSCA method. Compared with other hardware techniques, the HMCF is not only simple in the manufacturing process instead of producing complex x-assisted micro-structure, but also more practical than the homogeneous MCF thanks to the process errors and impurities. As for IC-XT, Saitoh et al. [30] revealed that the average IC-XT between neighboring cores after 100km propagation is a function of bending radius in the same core layout. The average IC-XT increases as increasing the bending radius in homogeneous MCFs. Moreover, in HMCF, the average IC-XT can be drastically reduced when a bending radius is larger than a specific value. When fibers are deployed, the bending radius will tend to infinity, resulting in the average IC-XT suppression up to 40dB. However, Saitoh et al. [30] also emphasized that it is difficult to select three or more kinds of cores in the same HMCF under the single-mode, large effective area, and low bending loss conditions, which are commonly met in real implementation. As a consequence, there are still homogeneous cores that may be adjacent to each other in HMCF, causing IC-XT. To address the IC-XT issue, we propose a novel IC-XT-avoid RSCA strategy for HMCF in the process of VONE. The reason for choosing the IC-XT-avoid RSCA strategy is that there is no need to calculate a specific IC-XT value by a complicated formula like [31] displaying. Instead, the strategy is to avoid utilizing the same center frequency between adjacent cores in the fiber links, which brings convenience to linearizing the problem model.

The contributions of this paper are summarized as follows:

- We design an HMCF structure used in the substrate EONs for the VONE to reduce IC-XT. The designed EONs with the HMCF provide more options for the selection of the VONE algorithms.
- We formulate some ILP models of the VONE process over EONs with HMCF. The formulated ILP models elaborate the NP-hard property of the VONE problem over EONs with HMCF and the optimal embedding solution which can be considered as the bounds of our proposed heuristic algorithm.
- To solve the problem in a short span of time, we propose a novel VONE algorithm supporting joint IC-XT-avoid cores and frequency slots assignment in EONs with HMCF. The proposed algorithm combines the advantages of the two IC-XT suppression approaches, which can improve the AFR and reduce the FR.

C. Paper Organization

The rest of this paper is organized as follows. In Section II, we introduce our design framework. Since the problem is NP-hard by nature, we design heuristic algorithms in Section III. We

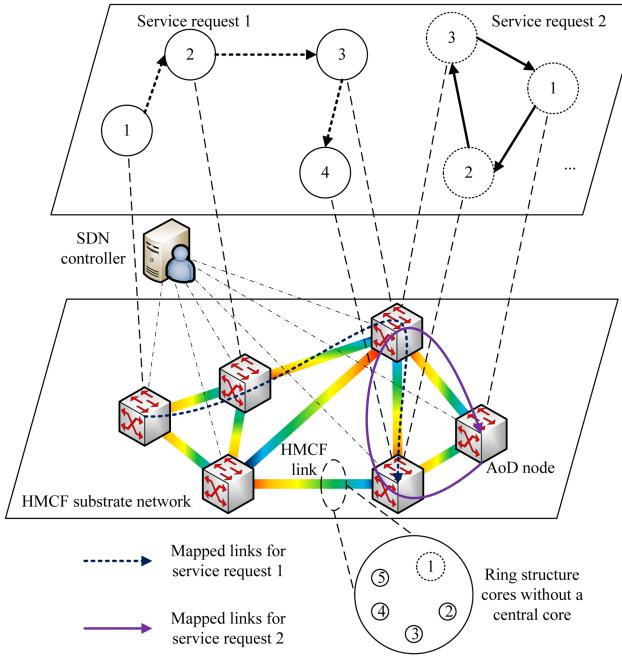


Fig. 2. VONE over substrate optical network with HMCF.

discuss simulation results in Section IV before concluding this paper in Section V.

II. PROBLEM DESCRIPTION AND ANALYSIS

In this section, we first introduce the network model and key notations. We then formulate our problem and discuss the NP-hard property of the problem by theory.

A. Network Model

Fig. 2 describes an example of the VONE scenario over an SDM-EON. The substrate optical network includes a set of architecture on demand (AoD) nodes [32] and a set of HMCF links can be abstracted as an undirected graph G . Note that such an SDM-EON can be usually multifunctional, which can allow multi-dimensional network resources such as spatial, time, and frequency domains to coexist simultaneously [32], hence, the substrate optical network should need the software-defined networking (SDN) controller as [33] claimed. Every node $V \in G$ has a computing capacity C_V and every edge $E \in G$ consists of two links in opposite directions. Each link has a set of ring structure cores C without a central core in which there are some heterogeneous cores defined as c_{Heter} and each core has the same frequency slots capacity B . The set of service requests is denoted as R . Each service request in R can be represented as a virtual directed graph G^r with a set of virtual nodes and a set of virtual links. The virtual nodes in a certain G^r request some computing resources C_v^r from nodes in G . Based on the transparent and opaque principle of the VONE problem [34], we assume that all service requests in R are transparent virtual optical network requests. Therefore, all virtual links in G^r should request the same frequency slots, denoted as Ω^r .

TABLE I
INPUT PARAMETERS

Parameter	Definition
B	The frequency slots capacity of each core in every link in G .
c	The c th core in set C .
C	The set of cores in fibers, and assume that each fiber has the same set of cores.
c_{adj}	The adjacent cores of core c .
c_{Heter}	The heterogeneous cores.
C_V	The computing capacity of the V th node in G .
C_v^r	The computing resources requested by the v th node in G^r .
e	The e th virtual link in set G^r .
e_s	The source node of the link e .
e_d	The destination node of the link e .
E	The E th edge in set G .
G	The topology of the HMCF substrate network.
G^r	The topology of the virtual network in the r th service request.
M	The ratio of the capacity of a heterogeneous core and that of a homogeneous core, for the sake of simplicity, assume that it is an integer.
N	The frequency slots utilized by the service request satisfy the condition of transmission through heterogeneous cores.
r	The r th service request in set R .
\hat{r}	The \hat{r} th service request different from the service request r .
R	The set of service requests.
(s, d)	s is the s th node in G . d is the d th node in G . (s, d) is the link from source node s to destination node d in G .
v	The v th virtual node in set G^r .
V	The V th node in set G .
Ω^r	The frequency slots requested by the r th service request.

When heterogeneous and homogeneous cores are fabricated with slight differences in radii [16], we supposed that the radii of heterogeneous cores are greater than those of homogeneous cores. According to the theory of fiber transmission in [35], the heterogeneous cores can support more spatial modes when propagating optical signals, which means the communication capacities of the heterogeneous cores can be increased by these additional spatial modes as [4] summarized. Note that there exist heterogeneous cores in C owning different capacities as mentioned above, here, we suppose a ratio described as M of the capacity between a heterogeneous core and a homogeneous. In addition, not all service requests can be transmitted through heterogeneous cores, in other words, some transmission conditions must be satisfied. Based on this fact, we also suppose that only the service request that requests N frequency slots could be transmitted through heterogeneous cores.

In summary, our problem is how to embed all graphs G^r , $\forall r \in R$ onto graph G with complicated fiber links consisting of cores set C optimally.

B. Problem Formulation

To facilitate discussion furtherly, we first list important notations in Tables I and II from two parts: parameters and decision variables. For each part, we list notations in alphabetic order.

Based on the network model and definitions, we then investigate the VONE over EONs with MCFs including homogeneous and heterogeneous cores, taking IC-XT into account or not, respectively. We formulate these problems by using the objective

TABLE II
DECISION VARIABLES

Variable	Definition
$f^r, f^{\hat{r}}$	An integer variable is defined for the constraints of frequency allocation. It indicates the start index of the frequency slots allocated to the r th or \hat{r} th service request (The frequency consistency of optical network claims that the start index of the frequency slots allocated to the same service request in different substrate link should be exactly alike. As a result, this variable is only associated with the index of the service request so that the frequency consistency is guaranteed naturally.).
Ms^r	An integer variable is defined for counting in the constraints of frequency allocation. It describes the maximum index of the frequency slots allocated to the r th service request.
$x_{e,(s,d)}^r$	A binary variable is defined for the constraints of link mapping. It indicates whether a link of a virtual network is mapped into a certain link of the substrate network or not. To be concrete, if the e th link of the r th service request is mapped into the link from node s to node d of the substrate network, this variable equals 1, otherwise, it is 0.
$X_{c,(s,d)}^r$, $X_{\hat{c},(s,d)}^r$	A binary variable is defined for the constraint of core assignment. It describes the selection of a core in a certain link of the substrate network. To be concrete, if the r th or \hat{r} th service request is assigned to the c th core in the link from node s to node d of the substrate network, this variable equals 1, otherwise, it is 0.
$y_{v,V}^r$	A binary variable is defined for the constraints of node embedding. It indicates if a node of a virtual network is embedded into a certain node of the substrate network or not. To be concrete, if the v th node of the r th service request is embedded into the V th node of the substrate network, this variable equals 1, otherwise, it is 0.

function (1).

$$\text{Minimize: } \text{Max}\{Ms^r\}, \quad \forall r \in R \quad (1)$$

Equation (1) describes the most efficient spectrum utilization of embedding all requested service requests. To formulate the problem, the above objective should satisfy several constraints.

1) *Node Embedding Constraints*: When we embed the nodes of a virtual network, the node embedding constraints (2)–(4) must be satisfied.

$$\sum_{V \in G} y_{v,V}^r = 1, \quad \forall r \in R, \forall v \in G^r \quad (2)$$

$$\sum_{v \in G^r} y_{v,V}^r \leq 1, \quad \forall r \in R, \forall V \in G \quad (3)$$

$$\sum_{r \in R} \sum_{v \in G^r} C_v^r \cdot y_{v,V}^r \leq C_V, \quad \forall V \in G \quad (4)$$

Equation (2) describes that each node in the topology of a service request should be only embedded into one node of the substrate network. Equation (3) indicates that each node of the substrate network could accept one node in the topology of a service request at most. Two constraints ensure that each virtual node of a service request is embedded into a unique substrate node. Equation (4) guarantees that an embedded substrate node has enough computing capacity to accommodate embedded requested computing resources.

2) *Link Mapping Constraints*: When we establish a lightpath from an embedded source to an embedded destination, the link

mapping constraints (5) and (6) must be satisfied.

$$x_{e,(s,d)}^r + x_{e,(d,s)}^r \leq 1, \quad \forall r \in R, \forall e \in G^r, \forall (s, d) \in G \quad (5)$$

$$\sum_{\forall (s,d) \in G} [x_{e,(s,d)}^r - x_{e,(d,s)}^r] = y_{e_s,s}^r - y_{e_d,s}^r, \quad \forall r \in R, \forall e \in G^r, \forall s \in G \quad (6)$$

According to the assumption in Section II-A, G^r is directed, thus, each link in the topology of a service request should be only embedded into one direction of a lightpath between two nodes in the substrate network, namely, the link-disjoint principle, as described in (5). Equation (6) is the flow conservation constraint, which ensures that the total number of the in-flows equals that of the out-flows on all substrate nodes except for the embedded source and destination substrate node.

3) *Core Assigning Constraint*: When we assign a core in a certain link to a service request, the core assigning constraint (7) must be satisfied.

$$\sum_{c \in C} X_{c,(s,d)}^r = \sum_{e \in G^r} x_{e,(s,d)}^r, \quad \forall r \in R, \forall (s, d) \in G \quad (7)$$

Equation (7) describes that each service request selects only a core on each link of the lightpath established by (5) and (6).

4) *Frequency Slots Allocating Constraints*: When we allocate frequency slots to a service request, the frequency allocating constraints (8)–(10) must be satisfied.

$$\begin{aligned} &\text{If } X_{c,(s,d)}^r + X_{\hat{c},(s,d)}^r = 2 \text{ Then} \\ &f^r - f^{\hat{r}} \geq \Omega^{\hat{r}} || f^{\hat{r}} - f^r \geq \Omega^r, \\ &\forall (r, \hat{r}) \in R, \forall c \in C, \forall (s, d) \in G \end{aligned} \quad (8)$$

$$\begin{aligned} &\text{If } X_{c,(s,d)}^r + X_{c_{adj},(s,d)}^r = 2 \text{ Then} \\ &f^r - f^{\hat{r}} \geq \Omega^{\hat{r}} || f^{\hat{r}} - f^r \geq \Omega^r, \\ &\forall (r, \hat{r}) \in R, \forall (c, c_{adj}) \in C, \forall (s, d) \in G \end{aligned} \quad (9)$$

$$f^r + \Omega^r - 1 = Ms^r, \quad \forall r \in R \quad (10)$$

$$f^r + \Omega^r - 1 \leq B, \quad \forall r \in R \quad (11)$$

Equation (8) claims that when two service requests r and \hat{r} are assigned to the same core, the frequency slots they utilize should not overlap, namely, the non-overlapping frequency allocating principle. In general, a guardband between two adjacent frequency slots allocated for two service requests should be added. In this paper, we do not consider it, but it can be easily included like [36]. Equation (9) named IC-XT-avoid constraint incorporates the IC-XT in the ILP formulation by employing the first approach in [21]. It indicates that the overlapping of the frequency slots two service requests r and \hat{r} utilizes should be avoided when the two service requests are assigned to core c and its adjacent core c_{adj} . Equation (10) describes the maximum index of the frequency slots, which represents the principle of

frequency continuity. Equation (11) guarantees a mapped substrate link has enough frequency slots capacity to accommodate mapped requested frequency slots.

5) *Core Assigning Constraint for the Heterogeneous Core:* (1)–(10) have modelled the problem of VONE over EONs with homogeneous MCF. As for heterogeneous cores, due to the different capacities, the allocated frequency slots will be corrected when a service request is assigned to a heterogeneous core. Especially, the core assigning constraint (7) will be corrected to assign the service request meeting heterogeneous core transmission condition, as (12) elaborates.

$$\begin{cases} X_{c_{Heter},(s,d)}^r = \sum_{e \in G^r} x_{e,(s,d)}^r, & \Omega^r = N \\ \sum_{c \in C, c \neq c_{Heter}} X_{c,(s,d)}^r = \sum_{e \in G^r} x_{e,(s,d)}^r, & \text{otherwise} \end{cases} \quad \forall r \in R, \forall (s, d) \in G \quad (12)$$

6) *Frequency Slots Allocating Constraints for the HMCF-Based Network:* When we convert some homogeneous cores into heterogeneous ones, the frequency allocating constraints will be corrected to (13)–(15).

If $X_{c,(s,d)}^r + X_{c,(s,d)}^{\hat{r}} = 2$ Then

$$\begin{cases} f^r - f^{\hat{r}} \geq \frac{\Omega^{\hat{r}}}{M} \mid |f^{\hat{r}} - f^r| \geq \frac{\Omega^r}{M}, \\ c = c_{Heter}, \Omega^r = N, \Omega^{\hat{r}} = N \\ f^r - f^{\hat{r}} \geq \Omega^{\hat{r}} \mid |f^{\hat{r}} - f^r| \geq \Omega^r, \\ c \neq c_{Heter}, \Omega^r \neq N, \Omega^{\hat{r}} \neq N \end{cases}, \quad \forall (r, \hat{r}) \in R, \forall c \in C, \forall (s, d) \in G \quad (13)$$

If $X_{c,(s,d)}^r + X_{c_{adj},(s,d)}^{\hat{r}} = 2$ Then

$$\begin{cases} f^r - f^{\hat{r}} \geq \Omega^{\hat{r}} \mid |f^{\hat{r}} - f^r| \geq \Omega^r, \\ \forall (r, \hat{r}) \in R, \forall (c, c_{adj}) \in C, \forall (s, d) \in G, \\ c \neq c_{Heter}, c_{adj} \neq c_{Heter}, \Omega^{\hat{r}} = \Omega^r \end{cases} \quad (14)$$

$$\begin{cases} f^r + \frac{\Omega^r}{M} - 1 = Ms^r, & \Omega^r = N \\ f^r + \Omega^r - 1 = Ms^r, & \text{otherwise} \end{cases}, \quad \forall r \in R \quad (15)$$

Equation (13) claims the non-overlapping frequency allocating principle in the HMCF. The difference is that when a service request meeting the heterogeneous transmission condition is assigned to a heterogeneous core c_{Heter} , Ω^r will be reduced to Ω^r/M due to the ratio M of capacity. (14) describes the IC-XT in the HMCF. The advantage is that IC-XT between heterogeneous core and homogeneous one is so tiny that can be ignored. Therefore, the IC-XT-avoid constraint could be removed when assigned core c or its adjacent core c_{adj} is heterogeneous.

By correcting constraints (7)–(10) to (12)–(15), we model the problem of VONE over EONs with HMCF. More cursorily, the IC-XT cannot be avoided, that is to say, constraints (9) and (14) might be removed from our formulations when the IC-XT could be ignored or other special cases (for example, simplify the problem to achieve less elapsed time). As a result, by removing

and replacing some constraints, our formulations can describe four scenarios more and more complicated: VONE over EONs with MCF with the IC-XT (VMXT), VONE over EONs with MCF avoiding the IC-XT (VMXTA), VONE over EONs with HMCF with the IC-XT (VHXT) and VONE over EONs with HMCF avoiding the IC-XT (VHXTA), respectively.

Theorem 1: Our formulations in four scenarios are all NP-hard.

Proof: Intuitively, although there exist nonlinear constraints of “If...Then” statements in some equations, the whole of them can be converted to linear constraints by adding more binary variables and some necessary constraints, as constraints (6~9) elaborate in [21]. In this paper, we formulate them in “If...Then” form instead of that of [21] so that we can apply them in IBM ILOG CPLEX Optimization Studio directly and efficiently to find their optimal solutions. Therefore, when these intuitively nonlinear constraints are converted to linear ones, all of our formulations are pure integer linear programming which has been proved to be NP-Hard. ■

To elaborate the proof more clearly, we take constraint (9) as an example to explain how to convert the constraint from the “If...Then” nonlinear form to linear form. Firstly, a new input parameter and two new decision variables should be added including:

- ξ : A large enough auxiliary constant.
- $\delta_{c,(s,d)}^{(r,\hat{r})}$: A binary decision variable is defined for the conversion. The variable equals to 1 if f^r is greater than $f^{\hat{r}}$ when the r th and \hat{r} th service request are assigned to the core c in the link from node s to node d of the substrate network simultaneously, otherwise, it is 0.
- $\Psi_{(c,c_{adj}),(s,d)}^{(r,\hat{r})}$: A integer decision variable is defined for the conversion. The variable equals to 2 if the r th service request is assigned to the core c and the \hat{r} th service request is assigned to c 's adjacent core c_{adj} in the link from node s to node d of the substrate network simultaneously; The variable equals to 1 if the r th service request is assigned to the core c or the \hat{r} th service request is assigned to c 's adjacent core c_{adj} in the link from node s to node d of the substrate network, otherwise, it is 0.

Then according to [21], the conversion from the “If...Then” nonlinear form to linear form can be formulated as (16–19). The elaboration of (16–19) can refer to [21], which we will not repeat here. Obviously, the converted equations are all linear constraints. Eqs. (8), (13), and (14) can be also converted by a similar method, as a result, all nonlinear constraints with the “If...Then” form can convert to linear form, which declares the proof more clearly.

$$\begin{aligned} X_{c,(s,d)}^r + X_{c_{adj},(s,d)}^{\hat{r}} &= \Psi_{(c,c_{adj}),(s,d)}^{(r,\hat{r})}, \\ \forall (r, \hat{r}) \in R, \forall (c, c_{adj}) \in C, \forall (s, d) \in G \end{aligned} \quad (16)$$

$$\begin{aligned} \Psi_{(c,c_{adj}),(s,d)}^{(r,\hat{r})} &\leq 2\delta_{c,(s,d)}^{(r,\hat{r})} + 2\delta_{c_{adj},(s,d)}^{(\hat{r},r)}, \\ \forall (r, \hat{r}) \in R, \forall (c, c_{adj}) \in C, \forall (s, d) \in G \end{aligned} \quad (17)$$

$$f^r - f^{\hat{r}} + \xi \left[\delta_{c,(s,d)}^{(r,\hat{r})} + \Psi_{(c,c_{adj}),(s,d)}^{(r,\hat{r})} \right] \leq 3\xi - \Omega^r,$$

TABLE III
NOTATIONS IN PROPOSED HEURISTIC ALGORITHM

Notation	Description
Italic symbol without subscript index	e.g. G^r , v_{sorted} , and R_{sorted} , describes a set of some certain network elements.
Italic symbol adding subscript index	e.g. v_i , $edge_j$ or c_k , describes the element in the corresponding set.
A set symbol not in italic	e.g. V_{sorted} , represents its programming data structure.
Dot operator (“.”)	e.g. <code>Vertex_EMBEDDING_Result.clear()</code> , uses “.” to call some attributes.

Input: G^r and G .

Output: the result of node embedding `Vertex_EMBEDDING_Result[|v|]`.

```

1:  $v_{sorted} \leftarrow$ 
   sort( $\forall v \in G^r$ , 'descending requested computing resources');
2:  $V_{sorted} \leftarrow$ 
   sort( $\forall V \in G$ , 'descending remaining computing resources');
3: for  $\forall v_i \in v_{sorted}$  do
4:   if  $V_{sorted}[v_i.vertex\_id].remaining\_computing\_resources \geq$ 
       $v_i.request\_computing\_resources$  then
5:     Vertex_EMBEDDING_Result[v_i.vertex_id] \leftarrow
         $V_{sorted}[v_i.vertex\_id].vertex\_id$ ;
6:      $V_{sorted}[v_i.vertex\_id].remaining\_computing\_resources \leftarrow$ 
         $V_{sorted}[v_i.vertex\_id].remaining\_computing\_resources -$ 
         $v_i.request\_computing\_resources$ ;
7:   else
8:     Block  $G^r$ ;
9:     Vertex_EMBEDDING_Result.clear();
10:    break;
11:   end if
12: end for
13: return Vertex_EMBEDDING_Result;

```

Fig. 3. Node embedding algorithm.

$$\forall (r, \hat{r}) \in R, \forall (c, c_{adj}) \in C, \forall (s, d) \in G \quad (18)$$

$$f^{\hat{r}} - f^r + \xi \left[\delta_{c_{adj}, (s, d)}^{(\hat{r}, r)} + \Psi_{(c, c_{adj}), (s, d)}^{(r, \hat{r})} \right] \leq 3\xi - \Omega^{\hat{r}},$$

$$\forall (r, \hat{r}) \in R, \forall (c, c_{adj}) \in C, \forall (s, d) \in G \quad (19)$$

III. EFFICIENT HEURISTIC ALGORITHMS

In Section II, with the consideration of IC-XT in heterogeneous cores, we have formulated the problem. Since the problem is NP-hard, in this section, we propose an efficient heuristic algorithm named VHXTAA for the most complicated scenario to solve it. The following notations in Table III are given.

Here, a base solution known as the two-stage virtual network embedding framework is adopted. Before the two-stage virtual network embedding, a sorted procedure should be executed to make the virtual networks with more nodes and more complex link connections be embedded into the substrate network as much as possible they can. This is because these virtual networks are difficult to be embedded potentially due to limited resources compared to those fewer nodes and fewer link connections. In this paper, we adopt a sort procedure of descending the number of nodes, then the number of links. The first stage names node embedding as Fig. 3 displays. The second stage names link mapping shown in Fig. 4.

In Fig. 3, `Vertex_EMBEDDING_Result` is a vector to store the node embedding result. The first element in it is the embedded

Input: `Vertex_EMBEDDING_Result[|v|]` of G^r and G .

Output: the result of link mapping `Edge_EMBEDDING_Result[|e|]`.

```

1: for  $\forall e_i \in G^r$  do
2:   Edge_EMBEDDING_Result[e_i.edge_id] \leftarrow
      Dijkstra(Vertex_EMBEDDING_Result[e_i.edge_source_id],
      Vertex_EMBEDDING_Result[e_i.edge_destination_id],
      ( $\forall E \in G$ ).allocated_frequency);
3:   if Edge_EMBEDDING_Result[e_i.edge_id].size() \leq 1 then
4:     Block  $G^r$ ;
5:     Edge_EMBEDDING_Result[e_i.edge_id].clear();
6:     Vertex_EMBEDDING_Result.clear();
7:     break;
8:   end if
9: end for
10: return Edge_EMBEDDING_Result;

```

Fig. 4. Link mapping algorithm.

node index of G indicating the embedding result of the start virtual node in G^r , the second element is the embedded node index of G indicating the embedding result of No. 2 virtual node in G^r and so on. In the process of node embedding, when we input the r th service request G^r after the sorted procedure, its virtual nodes will be sorted in descending order according to the requested computing resources at first, which prepares for the embedding order. Then we also sort the nodes of the substrate network G by the remaining computing resources in descending order. Finish the node embedding by embedding the sorted virtual nodes into the sorted substrate nodes one by one if the remaining computing resources can accommodate the request, otherwise, the r th service request will be blocked.

In Fig. 4, `Edge_EMBEDDING_Result` is a two-dimensional vector to record the link mapping result. The first element in it is the established lightpath connected by substrate nodes along the path, which indicates the mapping result of the start virtual link in G^r and the second element is the lightpath indicating the mapping result of No. 2 virtual link in G^r and the like. In the procedure to establish a lightpath, Dijkstra's algorithm [37] with the link weight of allocated frequency is adopted to find the shortest path according to `Vertex_EMBEDDING_Result` of each virtual link in G^r . If there is no path found, block G^r and clear the record of the `Vertex_EMBEDDING_Result`.

In general, at the end of the link mapping, the frequency slots of the substrate link should be allocated to the service request. However, HMCF makes things more complex. There should be other auxiliary algorithms to deal with these new scenarios, which are illustrated in Figs. 5, 7, and 9 named core priority, core classification, and core assignment, respectively.

In Fig. 5, to reduce the IC-XT between the homogeneous cores, the core priority algorithm has been corrected for HMCF from the *Predefined Core Prioritization* procedure for MCF in [38]. Core_Priority is a vector to define the priority of each core. The smaller index in Core_Priority, the higher priority the element is defined, which means the service request will be assigned preferentially. The difference between the two algorithms is that the core cost will be decreased rather than increased by 1 for the adjacent cores of the heterogeneous core. Then the core owning core cost of the value 0 will be defined to have a higher priority than that owning negative core cost instead of defining the core owning the minimum core cost. Fig. 6 provides an

```

Input:  $C$ .
Output: the result of core priority Core_Priority[|C|].
1: Initialize Core_Cost[|C|];
2: for  $p = 1$  to  $|C|$  do
3:   if  $p = 1$  then
4:     Core_Priority[ $p$ ]  $\leftarrow c_{Heter}.core\_id$ ;
5:     Core_Cost[ $c_{Heter}.adjacent\_core\_id$ ]  $\leftarrow -$ ;
6:     Core_Cost[ $c_{Heter}.core\_id$ ]  $\leftarrow +\infty$ ;
7:   else
8:      $c_{0\_cost} \leftarrow \text{find}(\text{Core_Cost.begin}(), \text{Core_Cost.end}(), 0)$ ;
9:     if  $c_{0\_cost} = \text{Core_Cost.end}()$  then
10:        $c_{min\_cost} \leftarrow \text{min\_element}(\text{Core_Cost})$ ;
11:        $c_{nearest} \leftarrow$ 
12:          $\text{min\_element}(\text{distance}(C[c_{min\_cost}],$ 
13:            $C[\text{Core_Priority}[p - 1]]))$ ;
14:     else
15:        $c_{nearest} \leftarrow$ 
16:          $\text{min\_element}(\text{distance}(C[c_{0\_cost}],$ 
17:            $C[\text{Core_Priority}[p - 1]]))$ ;
18:     end if
19:   end for

```

Fig. 5. Core priority algorithm.

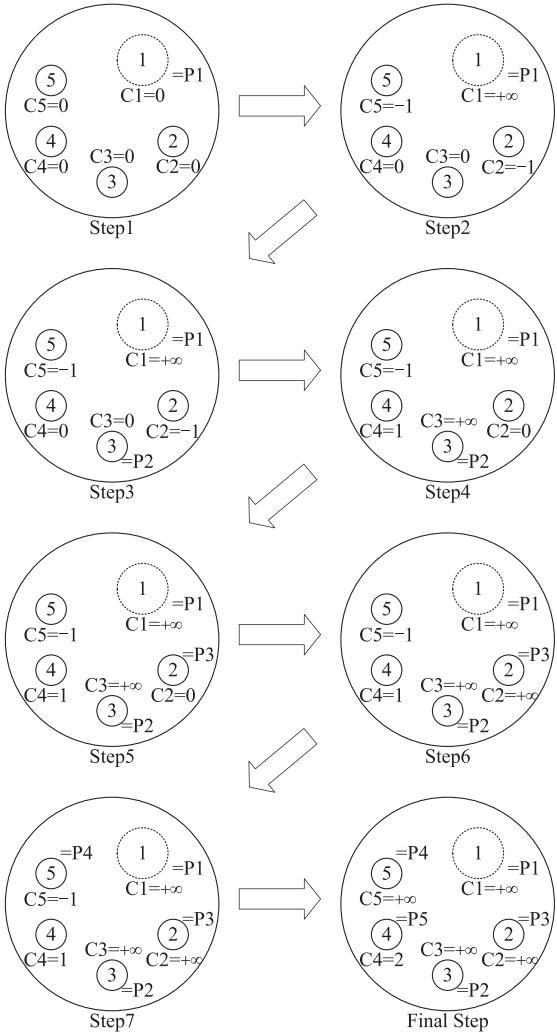


Fig. 6. An example of the core priority algorithm for a 5-core HMCF.

```

Input:  $C$ , Core_Priority[|C|], the list of service type Service_Type.
Output: the result of core classification Core_Classification[|C|].
1: Initialize index  $\leftarrow 0$ ;
2: for  $p = 1$  to  $|C|$  do
3:   if  $p = 1$  then
4:     Core_Classification [ $c_{Core\_Priority[p]}.core\_id$ ]  $\leftarrow$ 
        Service type meeting the heterogeneous transmission condition  $N$ ;
5:   else
6:     Current_Service_Type  $\leftarrow$ 
        Service_Type[index % Service_Type.size()];
7:     while Core_Classification [ $c_{Core\_Priority[p]}.adjacent\_core\_id$ ]  $=$ 
        Current_Service_Type do
8:       index  $\leftarrow$ 
9:       Current_Service_Type  $\leftarrow$ 
        Service_Type[index % Service_Type.size()];
10:      end while
11:      Core_Classification[p]  $\leftarrow$  Current_Service_Type;
12:    end if
13:  end for

```

Fig. 7. Core classification algorithm.

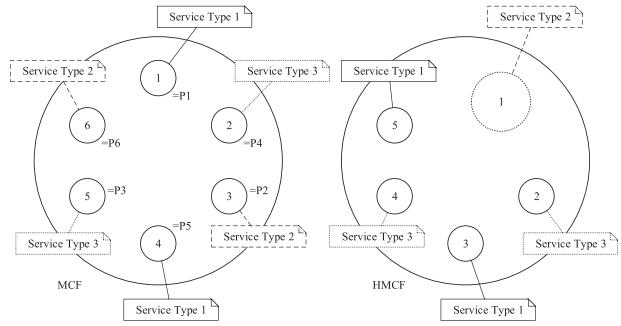


Fig. 8. An example of the core classification algorithm for three service types.

example for a 5-core HMCF. After the core priority algorithm, the core priority order is $1 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 4$.

In Fig. 7, to reduce the fragmentation after the frequency slots allocation, the core classification algorithm has been redesigned based on [38]. Core_Classification is a vector to classify which type of service request can be assigned according to the core index. The algorithm tends to assign different types of service requests to adjacent cores and the same type of service request to the same core so that the IC-XT could be avoided and reduce fragmentation. Fig. 8 shows an example of the core classification for MCF and HMCF. It proves different types of service requests are assigned between adjacent cores, which avoids the IC-XT.

According to the results of core priority and core classification, in Fig. 9, the core assignment algorithm is proposed based on the load-balance strategy in spectrum-first order [4], [39], [40], [41] so that the index of allocated frequency slots could be squeezed. The load-balance strategy always finds the core with the minimum index of unallocated frequency slots to accommodate the next service request, which could achieve the objective to minimize $M s^r$. Core_Assignment_Record is a vector to describe the core index successively along with the substrate link mapped by each virtual link in order. Start_Frequency_Index is an integer for all mapping links to indicate the start index of

Input: Vertex_EMBEDDING_Result[v] and Edge_EMBEDDING_Result[e] of G^r , C , Core_Priority[$|C|$], Core_Classification[$|C|$] and G .
Output: the result of core assignment Core_Assignment_Result in all mapped links, the initial sequence number of the frequency slots Start_Frequency_Index.

```

1: Initialize Start_Frequency[ $|C|$ ], Core_Assignment_Record;
2: if  $\Omega^r = N$  then
3:    $\Omega^r \leftarrow \Omega^r / M$ ;
4: end if
5: for  $p = 1$  to  $|C|$  do
6:   c1  $\leftarrow$  Core_Priority[ $p$ ];
7:   if Core_Classification[c1] = Request_Type( $G^r$ ) then
8:     Initialize Core_Assignment_tmp  $\leftarrow \emptyset$ ;
9:     for start  $\leftarrow 1$ : $E$ .frequency[1].size() do
10:    if Core_Assignment_tmp  $\neq \emptyset$  then
11:       $E$ .frequency[:] [ $E$ .frequency [:] < 0]  $\leftarrow 0$ ;
12:      Core_Assignment_tmp.clear();
13:    end if
14:    First_Edge  $\leftarrow$  find(* (Edge_EMBEDDING_Result[1].begin()), * (Edge_EMBEDDING_Result[1].begin() + 1))  $\in G$ ;
15:    if First_Edge.frequency[c1][start:start +  $\Omega^r - 1$ ] = 0 then
16:      First_Edge.frequency[c1][start:start +  $\Omega^r - 1$ ]  $\leftarrow -1$ ;
17:      Core_Assignment_tmp.push_back(c1);
18:      Initialize following_mappable_flag  $\leftarrow$  true;
19:      for  $\forall edge_j \in Edge_EMBEDDING_Result$  do
20:        if  $edge_j = First_Edge$  then
21:          continue;
22:        else
23:          Initialize core assignable_flag  $\leftarrow$  false;
24:          for  $q = 1$  to  $|C|$  do
25:            c2  $\leftarrow$  Core_Priority[ $q$ ];
26:            if Core_Classification[c2] = Request_Type( $G^r$ ) then
27:              if  $edge_j$ .frequency[c2][start:start +  $\Omega^r - 1$ ] = 0 then
28:                 $edge_j$ .frequency[c2][start:start +  $\Omega^r - 1$ ]  $\leftarrow -1$ ;
29:                Core_Assignment_tmp.push_back(c2);
30:                core assignable_flag  $\leftarrow$  true;
31:                break;
32:              end if
33:            end if
34:          end for
35:          if core assignable_flag = true then
36:            continue;
37:          else
38:            following_mappable_flag  $\leftarrow$  false;
39:            break;
40:          end if
41:        end if
42:      end for
43:      if following_mappable_flag = false then
44:        continue;
45:      else
46:        Core_Assignment_Record.
47:        push_back(Core_Assignment_tmp);
48:        Start_Frequency[c1]  $\leftarrow$  start;
49:         $E$ .frequency[:] [ $E$ .frequency [:] < 0]  $\leftarrow 0$ ;
50:        Core_Assignment_tmp.clear();
51:      end if
52:    else
53:      continue;
54:    end if
55:  end for
56: end for
57: if Core_Assignment_Record =  $\emptyset$  then
58:   Block  $G^r$ 
59:   Edge_EMBEDDING_Result.clear();
60:   Vertex_EMBEDDING_Result.clear();
61: else
62:   Start_Frequency_Index  $\leftarrow$  min_element(Start_Frequency[Core_Assignment_Record[:][1]]);
63:   Core_Assignment_Result  $\leftarrow$  Core_Assignment_Record[Start_Frequency = Start_Frequency_Index];
64: end if

```

Fig. 9. Core assignment algorithm.

Input: G^r , Edge_EMBEDDING_Result[e], Core_Assignment_Result and Start_Frequency_Index.
Output: the result of frequency allocating Frequency_Allocating_Result[$|E|$], the maximum sequence number of the frequency slots allocated to the r th service requirement M^r .

```

1: for  $\forall edge_i \in Edge_EMBEDDING_Result$  do
2:   Frequency_Allocating_Result[edge_i.edge_id]
3:   [Core_Assignment_Result[i]] [Start_Frequency_Index +  $\Omega^r - 1$ ]  $\leftarrow$  Request_Type( $\Omega^r$ );
4:    $M^r \leftarrow edge_i$ .frequency.size() -
5:   min_element(find(flipr(edge_i.frequency > 0))) + 1;
6:   edge_i.allocated_frequency  $\leftarrow M^r$ ;
5: end for

```

Fig. 10. Frequency allocating algorithm.

Input: R , G , C and Service_Type.

Output: the result of core priority Core_Priority[$|C|$], the result of core classification Core_Classification[$|C|$], the result of node embedding Vertex_EMBEDDING[$|R|$], the result of link embedding Edge_EMBEDDING[$|R|$], the result of core assignment Core_Assignment, the list of the initial sequence number of the frequency slots Start_Frequency_List, the result of frequency allocating Frequency_Allocating_Result[$|E|$], the maximum sequence number of the frequency slots M^r .

```

1: Initialize  $M^r \leftarrow 0$ ;
2: Core priority algorithm;
3: Core classification algorithm;
4:  $R_{sorted} \leftarrow$  sort( $R$ , 'descending vertex number, then edge number');
5: for  $\forall G^r \in R_{sorted}$  do
6:   Initialize Vertex_EMBEDDING_Result[ $\forall v \in G^r$ ],
7:   Edge_EMBEDDING_Result[ $\forall e \in G^r$ ], Core_Assignment_Result,
8:   Start_Frequency_Index,  $M^r$ ;
9: Node embedding algorithm;
10: Vertex_EMBEDDING.push_back(Vertex_EMBEDDING_Result);
11: Link mapping algorithm;
12: Edge_EMBEDDING.push_back(Edge_EMBEDDING_Result);
13: Core_Assignment.push_back(Core_Assignment_Result);
14: Start_Frequency_List.push_back(Start_Frequency_Index);
15: Frequency allocating algorithm;
16:  $M^r \leftarrow M^r$ ;
16: end for

```

Fig. 11. VONE algorithm over EONs with HMCF.

allocated frequency slots in each substrate link due to the constraints of the frequency consistency in the transparent optical network. In addition, the algorithm should also assign a suitable core index to meet the non-overlapping frequency allocating and continuity principle for the current service request.

With the results of Core_Assignment_Record and Start_Frequency_Index, the frequency slots could be easily allocated as the algorithm shown in Fig. 10. We need only to allocate the frequency slots consecutively for each service request to be subject to the frequency continuity constraints.

Finally, by integrating all algorithms in Figs. 3–11 gives the complete algorithm framework of our proposed VHXTAA. Except for it, by employing different core priority and core classification, different heuristic algorithms of other scenarios in Section II can be achieved which are called VMXT algorithm (VMXTA), VMXTA algorithm (VMXTAA), and VHXT algorithm (VHXTA), respectively. Fig. 12 lists the results of allocated frequency slots in a substrate link by employing the four algorithms. Here, we can see the contrast among

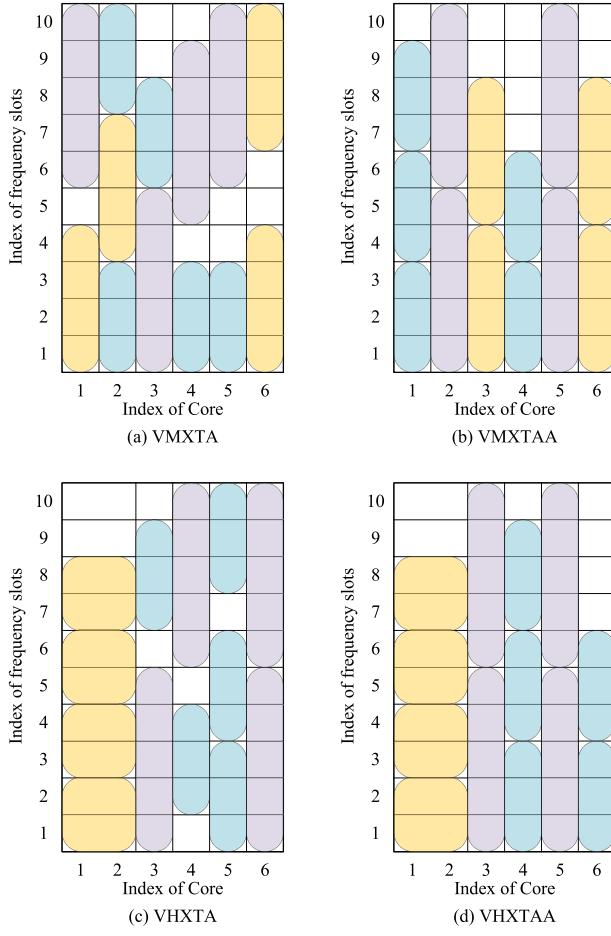


Fig. 12. An example of allocated frequency slots for four algorithms.

the four algorithms. In subfigure (b), compared with VMXTA in subfigure (a), the same type of service request seems to be assigned to a special nonadjacent core channel to be accommodated, which could avoid IC-XT and reduce fragmentation. In subfigure (c), compared with VMXTA, only the particular service request could be assigned to the heterogeneous core, which degrades the probability of the appearance of the same type of service request in adjacent cores, i.e., reduces the IC-XT. In subfigure (d), compared with VHXTA in subfigures (c), the employment of core priority and classification could further avoid IC-XT and reduce fragmentation. That employment completes the whole heuristic algorithm we proposed for the VONE problem over EONs with HMCF.

A. Time Complexity

We will list the time complexity of each algorithm from Figs. 3–10 one by one, and then obtain the total time complexity of Fig. 11 by integrating these time complexity. For **Node embedding** algorithm in Fig. 3, the main time-consuming part is the sort step of substrate nodes in line 2. We implement the sort step by the sort function in C++ STL, which the time complexity is approximately $O(|V| \log_2 |V|)$. For **Link mapping** algorithm in Fig. 4, the main time-consuming part

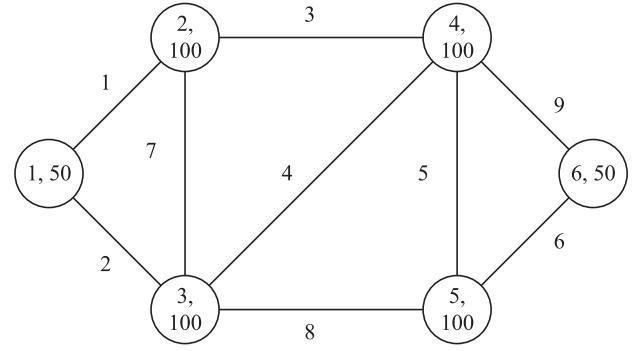


Fig. 13. Six-node experimental Topology.

is Dijkstra's algorithm with the outer for-loop in lines 1-2. We implement the Dijkstra's algorithm by the adjacency matrix, hence, the time complexity is approximately $O(|e||V|^2)$. For **Core priority** algorithm in Fig. 5, the main time-consuming part is the find or min_element function in C++ STL with the outer for-loop in lines 2, 8, 10, 11, and 13. The time complexity is about $O(|C|^2)$. For **Core classification** algorithm in Fig. 7, the time complexity is simply $O(|C|)$. For **Core assignment** algorithm in Fig. 9, the time complexity is more complicated. The main time-consuming parts include four for-loops in lines 5, 9, 19, and 24. Since the total number of each substrate edge's frequency slots defined as $|FS| = E.\text{frequency}[1].\text{size}()$ meets the condition $|FS| \gg \Omega^r$, the worst time complexity is approximately $O(|C|^2|FS|^2|e||E|)$. For **Frequency allocating** algorithm in Fig. 10, the main time-consuming part is the for-loop in line 1. The worst time complexity is $O(|e||E||FS|)$ thanks to $|FS| \gg \Omega^r, |C|$. Finally, considering $|R| \gg |C|, |C|^2$, the total time complexity of Fig. 11 is mainly composed of two parts: one is the sort step of R in line 4 with the time complexity $O(|R| \log_2 |R|)$, the other one is the for-loop in line 5 with the worst time complexity approximately $O[(|V|^2 + |C|^2|FS|^2|E|) \cdot \sum_{G^r \in R} \sum_{e \in G^r} |e|]$. As a result, the total time complexity of Fig. 11 is approximately the sum of the two, which is the polynomial time.

IV. SIMULATION AND DISCUSSION

In this section, we introduce simulation settings and discuss extensive simulation results.

A. Simulation Settings

Firstly, we evaluate the results of the formulations by solving the proposed ILP models over a simple six-node experimental topology [40] with a few service requests as Fig. 13 shows. Each circle describes a node of the topology. There are two values in a circle. The first value is the index of the node and the second is the node's computing capacity C_V . The number beside each link is the index of the link and suppose the number of frequency slots in each link is 50, i.e., $B = 50$ of the simple topology. To be fair, the four algorithms are based on the same topology. Next, as shown in Fig. 14, a modified large-scale topology named Future Internet Technology Infrastructure (FITI) [42] by

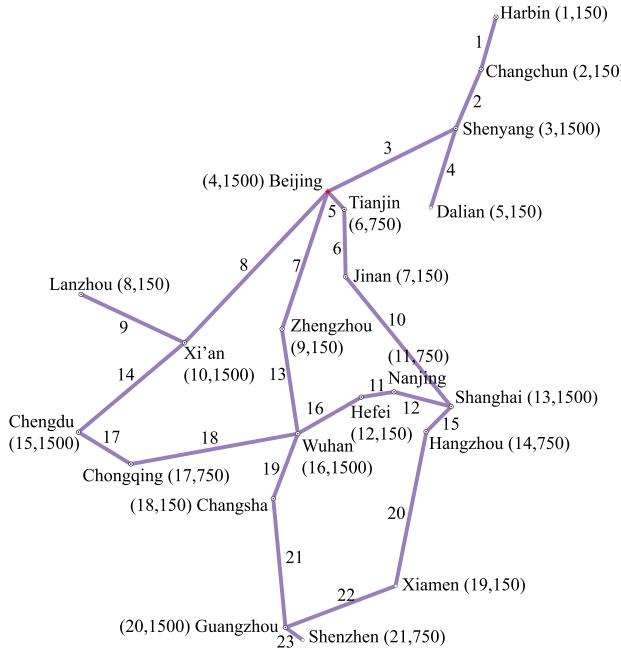


Fig. 14. FITI Topology.

abstracting all nodes in the same city for a bigger node and only reserving the 100Gbps links is applied to estimate more metrics of our proposed heuristic algorithms. The number associated with each node or link represents the index. The computing capacity C_V follows the node index. For two kinds of fibers, we set the width of the frequency slot at 12.5GHz and the total spectrum resource per core is 4THz in C-band. Therefore, the number of frequency slots per core is 320 [38], i.e., $B = 320$ of the large-scale topology. The number of cores for MCF and HMCF is 6 and 5, respectively, and the first core of HMCF is heterogeneous.

All virtual nodes and links in a service request are generated randomly. The number of virtual nodes for each service request is assumed to be a uniformly distributed random variable over range [2, 4]. The requested computing resources C_v^r is assumed to range within [1, 5] units. The number of virtual links for each service request is also assumed to be a uniformly distributed random variable with maximum range $|v|(|v| - 1)/2$. Since the number of virtual links has to guarantee that all virtual vertexes are connected at least, the minimum range of the number of virtual links is $|v| - 1$. The number of requested frequency slots of each service request is uniformly distributed from [3, 5] according to [43].

To improve the solving efficiency, we implement the four ILP models by employing the C++ language interface of the commercial software IBM ILOG CPLEX Studio 12.8.0 running on a 64-bit Windows machine with a 3.20GHz CPU Intel(R) Core(TM) i7-8700 and 16G memory. For the consistency of comparisons, the heuristic algorithm for each scenario is also implemented in the C++ language.

To evaluate the performance, the following metrics are measured:

- *Time*: This metric describes the solving time of an ILP model or a heuristic algorithm, with the unit of milliseconds (ms).
- *Ms*: This metric is the maximum index of allocated frequency slots as defined in Fig. 11.
- *Fragmentation Ratio (FR)*: One core's fragmentation describes all the unallocated frequency slots before the maximum index of allocated frequency slots under the limitation of frequency consistency and continuity. This metric is defined as the ratio of the sum of all cores' fragmentation over all links (defined as $\sum F$) of the substrate network and Ms multiplied by the number of cores and links as (20) displays.

$$FR = \frac{\sum F}{2Ms \cdot |C| \cdot |E|} \quad (20)$$

- *IC-XT Ratio (IC-XTR)*: One simple way to describe IC-XT is the number of frequency slots influenced by the IC-XT, however, the result may fluctuate greatly when the random service requests change. Here, this metric is defined as the ratio of the total number of IC-XT (defined as $\sum IC - XT$) and Ms multiplied by the number of cores and all links so that IC-XT could be associated with the service request details as (21) displays.

$$IC-XTR = \frac{\sum IC - XT}{2Ms \cdot |C| \cdot |E|} \quad (21)$$

- *Frequency Utilization Ratio (FUR)*: To describe the utilization of frequency slots in the substrate network, the metric is defined as the ratio of the sum of allocated frequency slots in all cores over all links (defined as $\sum FU$) and Ms multiplied by the number of cores and links as (22) displays.

$$FUR = \frac{\sum FU}{2Ms \cdot |C| \cdot |E|} \quad (22)$$

- *Available Frequency Ratio (AFR)*: This metric is defined as the ratio of the sum of frequency slots between each core's maximum index of allocated frequency slots and Ms (defined as $\sum AF$) and Ms multiplied by the number of cores and all links as (23) displays, which describes the expansion capability to accommodate new more service requests. It is easy to prove that $FR + FUR + AFR = 1$.

$$AFR = \frac{\sum AF}{2Ms \cdot |C| \cdot |E|} \quad (23)$$

B. Comparisons for ILP Models and Heuristic Algorithms Over the Simple Topology

Table IV displays the comparison of Time and Ms for ILP models and the heuristic algorithms over the simple topology in Fig. 13. When the number of service requests increases from 5 to 25, decision variables and constraints in ILP models will be increased quickly. Therefore, the solving time of ILP models will also be increased rapidly. Moreover, when $|R| = 30$ for VMXTA, VHXT, and VHXTA, the ILP models even cannot obtain the optimal solution in a reasonable time period (the

TABLE IV
RESULTS BETWEEN ILP FORMULATION AND HEURISTIC ALGORITHM

$ R $	VMXT			
	ILP		Heuristic	
	Time (ms)	M_s	Time (ms)	M_s
5	359	5	1	5
10	2209	5	2	5
15	5279	5	2	6
20	16256	5	4	8
25	56901	5	4	11
30	289530	5	7	13
$ R $	VMXTA			
	ILP		Heuristic	
	Time (ms)	M_s	Time (ms)	M_s
5	605	5	2	5
10	4688	5	2	5
15	26731	5	2	6
20	337399	5	3	10
25	5186180	5	4	20
30	-	-	-	-
$ R $	VHXT			
	ILP		Heuristic	
	Time (ms)	M_s	Time (ms)	M_s
5	114	5	2	5
10	485	5	2	5
15	1635	5	2	8
20	4286	5	4	8
25	23208	5	4	14
30	-	-	-	-
$ R $	VHXTA			
	ILP		Heuristic	
	Time (ms)	M_s	Time (ms)	M_s
5	154	5	1	5
10	1078	5	1	5
15	7086	5	2	6
20	53886	5	3	10
25	3379870	5	4	20
30	-	-	-	-

solving time has exceeded 24 hours), which proves that solving ILP models are very time-consuming. By contrast, the solving time of heuristic algorithms only takes 99% less than that of ILP models, which proves the time efficiency of the heuristic algorithms. However, time efficiency is at the expense of accuracy. From Table IV, M_s of the heuristic algorithms is bigger and bigger when $|R|$ increases, but the solutions are still feasible. The reason is that the heuristic algorithms decrease the solution space by cutting some candidate solutions during the process of link mapping in Fig. 4 and core assignment in Fig. 9. In terms of saving time, such accuracy is sufficient for the larger-scale network and more service requests.

C. Comparisons for Heuristic Algorithms Over the Large-Scale Topology

In this section, we discuss the metrics defined in Section IV-A of the four scenarios as Figs. 15–25 illustrate over the large-scale topology in Fig. 14.

Fig. 15 shows the FR achieved by four different algorithms. The solid curves of VHXTA and VHXTAA tend to be under the dash curves of VMXTA and VMXTAA. It indicates that the FR is reduced when a heterogeneous core is introduced, whether the IC-XT is avoided or not. When the IC-XT is avoided, the FR could be reduced whether the heterogeneous core is introduced as the line of mark asterisk and diamond illustrate in Fig. 15.

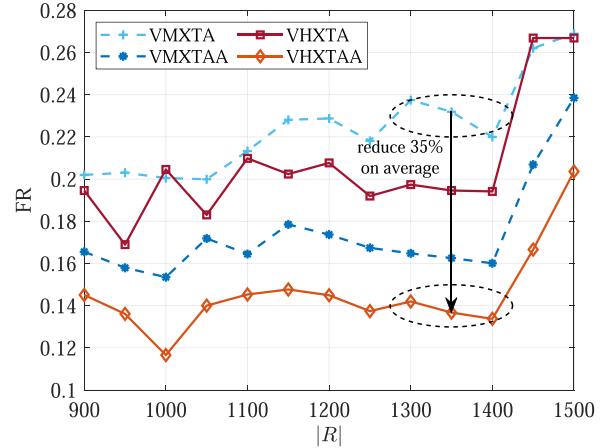


Fig. 15. FR vs. $|R|$.

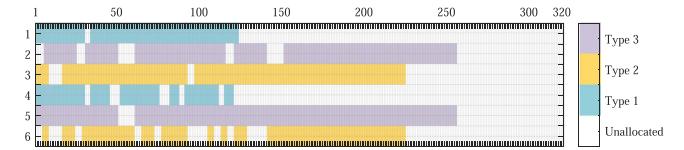


Fig. 16. MCF Frequency Allocating.

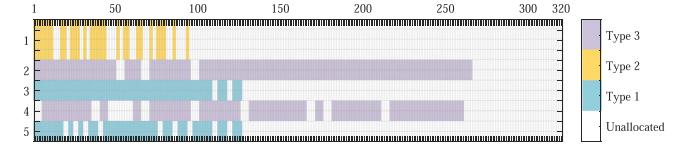


Fig. 17. HMCF Frequency Allocating.

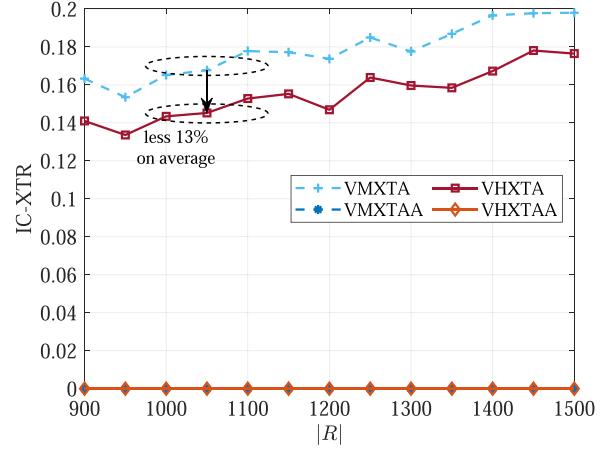


Fig. 18. IC-XTR vs. $|R|$.

Therefore, our proposed algorithm VHXTAA can dramatically reduce the FR 35% on average compared with the conventional VMXTA.

Fig. 18 displays the IC-XTR. Obviously, the IC-XT could be avoided completely according to VMXTAA and VHXTAA. At $|R| = 900$, Figs. 16 and 17 give the results of allocated frequency

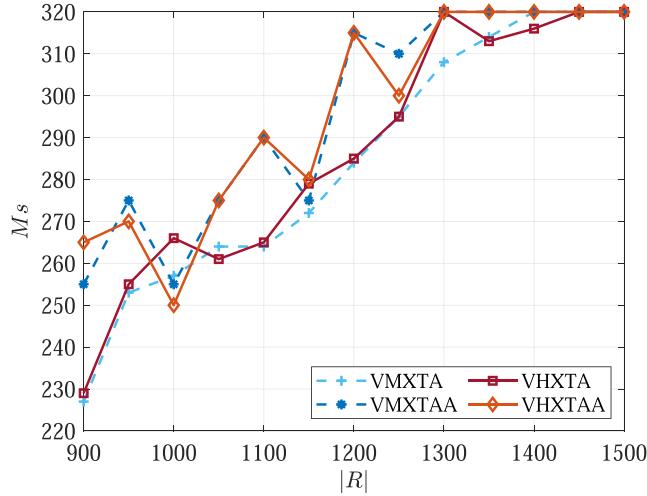
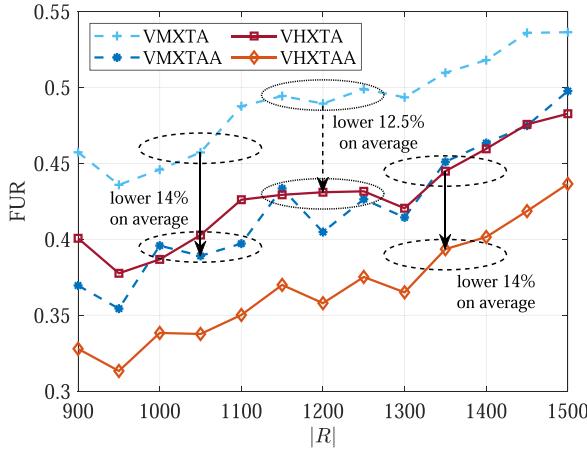
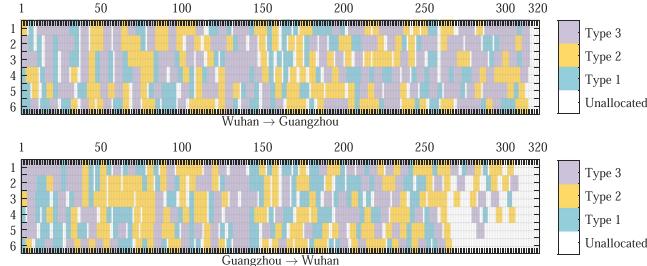
Fig. 19. M_s vs. $|R|$.Fig. 20. FUR vs. $|R|$.

Fig. 21. Frequency allocating result after VMXTA.

slots in the edge 3 (from Shenyang to Beijing) for VMXTAA and VHXTAA, respectively. The two algorithms only allow one type of service request accommodated in a core, and the adjacent cores assign different types of service requests. Therefore, one type of service request seems to own a private core to transmit, which avoids the IC-XT and reduces the FR simultaneously. Note that even if the IC-XT is not avoided, the IC-XTR of VHXTA is still less than 13% on average than that of VMXTA

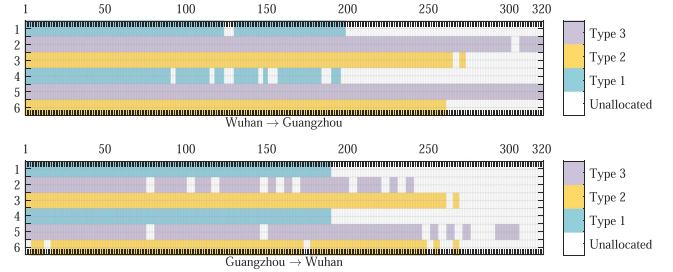


Fig. 22. Frequency allocating result after VMXTAA.

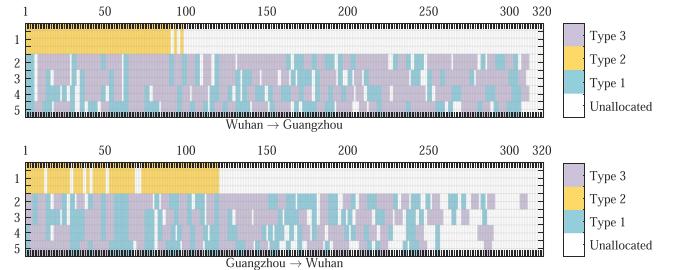
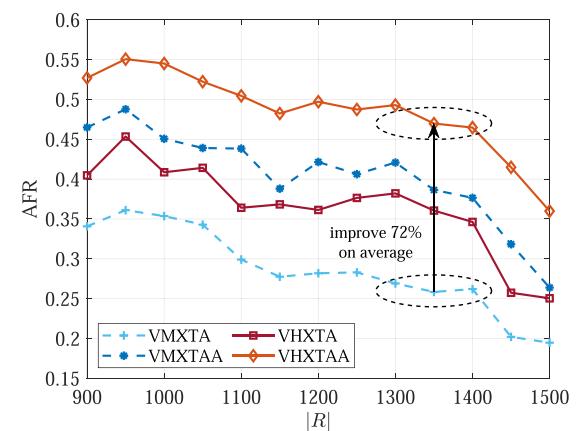


Fig. 23. Frequency allocating result after VHXTA.



Fig. 24. Frequency allocating result after VHXTAA.

Fig. 25. AFR vs. $|R|$.

thanks to the introduction of a heterogeneous core, which proves the advantage of VONE over EONs with HMCF.

Fig. 19 shows the M_s results of four different algorithms. We can see M_s of VMXTA and VHXTA trends to be the same and smaller than M_s of the other two algorithms. It is easy to understand that the IC-XT-avoid strategy sacrifices some

available frequency slots to avoid the IC-XT, hence, M_s of VMXTAA and VHXTAA not only tend to be the same but also bigger than the other two algorithms above mentioned, which indicates that VMXTAA and VHXTAA may have a lower FUR as Fig. 20 shows.

In Fig. 20, because of the IC-XT-avoid consideration, the FUR of VMXTAA and VHXTAA is both 14% on average lower than that of VMXTA and VHXTA. The result reveals that the IC-XT-avoid consideration sacrifices 14% FUR to avoid the IC-XT and reduce the FR. However, the FUR of VMXTAA and VHXTA trends to the same (lower 14% and 12.5% on average compared with VMXTA, respectively). The VHXTA result hits us that the introduction of a heterogeneous core without any other strategies does not seem to degrade the performance too much compared with the strict IC-XT-avoid consideration. When an overall consideration needs to be balanced, VHXTA could also be a potential choice.

Finally, although M_s of VMXTAA and VHXTAA is bigger than that of other algorithms, it does not mean that the two algorithms could not accommodate more service requests. Figs. 21–24 give the allocating results of frequency slots in edge 21 (between Wuhan and Guangzhou) at $|R| = 1350$ solving by four algorithms, respectively. For Figs. 22 and 24, we can see even if M_s of the link from Wuhan to Guangzhou achieves the maximum in some cores, there are still enough unallocated frequency slots in other cores, which is potential to be allocated when more service requests are required. While for Figs. 21 and 23, each core has been almost assigned because of the load-balance strategy in the process of core assignment shown in Fig. 9. Few unallocated frequency slots are enough to accommodate new service requests compared with Figs. 22 or 24. Therefore, in Fig. 25, the AFR of VMXTAA and VHXTAA is larger than that of VMXTA and VHXTA, and 72% on the average improvement of VHXTAA has been achieved compared to conventional VMXTA, which proves the expansion capability to accommodate new more service requests of VHXTAA.

V. CONCLUSION

In this paper, we have designed an HMCF structure for EONs at first. Then some ILP models are established to formulate the process of VONE over EONs with HMCF and prove the NP-hard property of the problem. Next, to avoid the high time complexity of NP-hard ILP, we propose a novel VONE algorithm supporting joint IC-XT-avoid cores and frequency slots assignment in EONs with HMCF. Finally, simulation results demonstrate that our proposed algorithm can significantly reduce the IC-XT when virtual networks are embedded and effectively improve 72% AFR and reduce 35% FR compared with VMXTA, which squeezes more bandwidth resources. For the selection of algorithms, it does not have to choose a strict algorithm like VHXTAA. When an overall consideration needs to be balanced, VHXTA could also be a potential choice to maintain the M_s and FUR as well as avoid the IC-XT.

REFERENCES

- [1] ETSI, “Fifth generation fixed network (F5G),” 2020. [Online]. Available: <https://www.etsi.org/technologies/fifth-generation-fixed-network-f5g>
- [2] European Telecommunications Standards Institute (ETSI), “Fifth generation fixed network (F5G); F5G network architecture,” 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/F5G/001_099/004/01.01.01_60/gs_F5G004v010101p.pdf
- [3] R. Nejabati, E. Escalona, S. Peng, and D. Simeonidou, “Optical network virtualization,” in *Proc. IEEE 15th Int. Conf. Opt. Netw. Des. Model.*, 2011, pp. 1–5.
- [4] M. Klinkowski, P. Lechowicz, and K. Walkowiak, “Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking,” *Opt. Switching Netw.*, vol. 27, pp. 58–78, 2018.
- [5] B. J. Puttnam et al., “Characteristics of homogeneous multi-core fibers for SDM transmission,” *APL Photon.*, vol. 4, no. 2, 2019, Art. no. 022804.
- [6] J. M. D. Mendenuta, S. Shinada, Y. Hirota, H. Furukawa, and N. Wada, “High-capacity super-channel-enabled multi-core fiber optical switching system for converged inter/intra data center and edge optical networks,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 4, pp. 1–13, Jul./Aug. 2020.
- [7] F. Bao et al., “DSP-free single-wavelength 100 Gbps SDM-PON with increased splitting ratio using 10G-class DML,” *Opt. Exp.*, vol. 27, no. 23, pp. 33916–33925, 2019.
- [8] A. Sano, H. Takara, T. Kobayashi, and Y. Miyamoto, “Petabit/s transmission using multicore fibers,” in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2014, pp. 1–3.
- [9] B. J. Puttnam et al., “2.15 Pb/s transmission using a 22 core homogeneous single-mode multi-core fiber and wideband optical comb,” in *Proc. IEEE Eur. Conf. Opt. Commun.*, 2015, pp. 1–3.
- [10] J. Brenes et al., “Network slicing architecture for SDM and analog-radio-over-fiber-based 5G fronthaul networks,” *J. Opt. Commun. Netw.*, vol. 12, no. 4, pp. B33–B43, 2020.
- [11] D. M. Taylor, C. R. Bennett, T. J. Shepherd, L. F. Michaille, M. D. Nielsen, and H. R. Simonsen, “Demonstration of multi-core photonic crystal fibre in an optical interconnect,” *Electron. Lett.*, vol. 42, no. 6, pp. 331–332, 2006.
- [12] K. Saitoh, T. Matsui, T. Sakamoto, M. Koshiba, and S. Tomita, “Multi-core hole-assisted fibers for high core density space division multiplexing,” in *Proc. IEEE 15th Optoelectron. Commun. Conf.*, 2010, pp. 164–165.
- [13] K. Takenaga et al., “Reduction of crosstalk by trench-assisted multi-core fiber,” in *Proc. Opt. Fiber Commun. Conf. Expo. Nat. Fiber Optic Engineers Conf.*, 2011, pp. 1–3.
- [14] H. Yuan, A. Saljoghei, A. Peters, and G. Zervas, “Comparison of SDM-WDM based data center networks with equal/unequal core pitch multicore fibers,” in *Proc. IEEE Opt. Fiber Commun. Conf. Expo.*, 2018, pp. 1–3.
- [15] Y. Amma, K. Takenaga, T. Fujisawa, K. Saitoh, M. Koshiba, and K. Aikawa, “Dependence of cladding diameter on inter-core crosstalk in heterogeneous multi-core fibers,” in *Proc. IEEE 23rd Opto-Electron. Commun. Conf.*, 2018, pp. 1–2.
- [16] M. Koshiba, K. Saitoh, and Y. Kokubun, “Heterogeneous multi-core fibers: Proposal and design principle,” *IEICE Electron. Exp.*, vol. 6, no. 2, pp. 98–103, 2009.
- [17] T. Ito, E. L. T. de Gabory, M. Arikawa, Y. Hashimoto, and K. Fukuchi, “Reduction of influence of inter-core cross-talk in MCF with bidirectional assignment between neighboring cores,” in *Proc. Conf. Opt. Fiber Commun./Nat. Fiber Optic Engineers Conf.*, 2013, pp. 1–3.
- [18] R. S. Luis, G. Rademacher, B. J. Puttnam, Y. Awaji, and N. Wada, “On the use of high-order MIMO for crosstalk mitigation in multi-core fiber based SDM systems,” in *Proc. IEEE SBFoton Int. Opt. Photon. Conf.*, 2018, pp. 1–4.
- [19] M. Yang, Y. Zhang, and Q. Wu, “Routing, spectrum, and core assignment in SDM-EONs with MCF: Node-arc ILP/MILP methods and an efficient XT-Aware heuristic algorithm,” *J. Opt. Commun. Netw.*, vol. 10, no. 3, pp. 195–208, 2018.
- [20] H. Yang, Q. Yao, A. Yu, Y. Lee, and J. Zhang, “Resource assignment based on dynamic fuzzy clustering in elastic optical networks with multi-core fibers,” *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3457–3469, May 2019.
- [21] A. Muhammad, G. Zervas, D. Simeonidou, and R. Forchheimer, “Routing, spectrum and core allocation in flexgrid SDM networks with multi-core fibers,” in *Proc. IEEE Int. Conf. Opt. Netw. Des. Model.*, 2014, pp. 192–197.
- [22] R. J. Zhu et al., “Dynamic virtual optical network embedding in spectral and spatial domains over elastic optical networks with multicore fibers,” *Opt. Eng.*, vol. 55, no. 8, 2016, Art. no. 086108.
- [23] H. Liu, H. Hu, Y. Chen, and L. Du, “Virtual optical network mapping method for fragment sensing in the space division multiplexed elastic optical network,” CN Patent, CN 111162865A, 2020.

- [24] Q. Chen et al., "Profit-aware virtual optical network mapping in space-division-multiplexing elastic optical networks," in *Proc. Asia Commun. Photon. Conf./Int. Conf. Inf. Photon. Opt. Commun.*, 2020, pp. 1–3.
- [25] Q. Chen et al., "Virtual optical network mapping approaches in space-division-multiplexing elastic optical data center networks," *J. Lightw. Technol.*, vol. 40, no. 12, pp. 3515–3529, Jun. 2022.
- [26] R. Zhu et al., "Crosstalk-aware virtual optical network embedding (VONE) in spatial division multiplexing enabled elastic optical networks with multi-core fibers," in *Proc. IEEE 42nd Eur. Conf. Opt. Commun.*, 2016, pp. 1–3.
- [27] H. Xuan, Y. Wang, Z. Xu, S. Hao, and X. Wang, "Virtual optical network mapping and core allocation in elastic optical networks using multi-core fibers," *Opt. Commun.*, vol. 402, pp. 26–35, 2017.
- [28] L. Cheng et al., "Dynamic virtual network embedding with distance adaptive RSA over SDM-Based elastic optical networks," in *Proc. Conf. Lasers Electro-Opt. Pacific Rim*, 2018, pp. 1–2.
- [29] W. Zhen et al., "Virtual optical network mapping approaches with inter-core crosstalk in space division multiplexing elastic optical data center networks," in *Proc. IEEE 19th Int. Conf. Opt. Commun. Netw.*, 2021, pp. 1–4.
- [30] K. Saitoh, M. Koshiba, K. Takenaga, and S. Matsuo, "Homogeneous and heterogeneous multi-core fibers," in *Proc. IEEE Photon. Soc. Summer Topical Meeting Ser.*, 2012, pp. 210–211.
- [31] F. Tang, Y. Li, G. Shen, and G. N. Rouskas, "Minimizing inter-core crosstalk jointly in spatial, frequency, and time domains for scheduled lightpath demands in multi-core fiber-based elastic optical network," *J. Lightw. Technol.*, vol. 38, no. 20, pp. 5595–5607, Oct. 2020.
- [32] X. Zhang et al., "On throughput optimization in software-defined multidimensional space division multiplexing optical networks," *J. Lightw. Technol.*, vol. 39, no. 9, pp. 2635–2651, May 2021.
- [33] B. Chen, J. Zhang, W. Xie, J. P. Jue, Y. Zhao, and G. Shen, "Cost-effective survivable virtual optical network mapping in flexible bandwidth optical networks," *J. Lightw. Technol.*, vol. 34, no. 10, pp. 2398–2412, May 2016.
- [34] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, no. 3, pp. 450–460, Feb. 2014.
- [35] G. P. Agrawal, *Nonlinear Fiber Optics*, 5th ed. New York, NY, USA: Academic Press, 2013.
- [36] K. Christodouloupolos, I. Tomkos, and E. A. Varvarigos, "Routing and spectrum allocation in OFDM-based optical networks with elastic bandwidth allocation," in *Proc. IEEE Glob. Telecommun. Conf.*, 2010, pp. 1–6.
- [37] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [38] S. Fujii, Y. Hirota, H. Tode, and K. Murakami, "On-demand spectrum and core allocation for reducing crosstalk in multicore fibers in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 6, no. 12, pp. 1059–1071, 2014.
- [39] H. Tode and Y. Hirota, "Routing, spectrum, and core and/or mode assignment on space-division multiplexing optical networks invited," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. A99–A113, 2017.
- [40] Y. Zhang, Q. Zhang, X. Cai, and L. Guo, "Resource allocation for heterogeneous multi-core fiber supporting Big Data transmission," in *Proc. IEEE/CIC Int. Conf. Commun. China*, 2017, pp. 1–5.
- [41] M. Klinkowski and G. Zalewski, "Dynamic crosstalk-aware lightpath provisioning in spectrally-spatially flexible optical networks," *J. Opt. Commun. Netw.*, vol. 11, no. 5, pp. 213–225, 2019.
- [42] Institute for Network Sciences And Cyberspace of Tsinghua University, "Future network test facilities of national major science and technology infrastructure: Future internet technology infrastructure (FITI)" 2021. [Online]. Available: https://www.insc.tsinghua.edu.cn/sypt/CNGI_CERNET21.htm
- [43] K. Morita and K. Hirata, "Dynamic spectrum allocation method for reducing crosstalk in multi-core fiber networks," in *Proc. IEEE Int. Conf. Inf. Netw.*, 2017, pp. 686–688.

Qihan Zhang (Member, IEEE) received the M.S. degree in communication and information systems in 2018, from the School of Computer Science and Engineering, Northeastern University, Shenyang, China, where he is currently working toward the Ph.D. degree in communication and information systems. His research interests include fiber nonlinear, optical signal processing, and secure optical transmission system.

Xu Zhang (Member, IEEE) received the B.Eng. and Ph.D. degrees from Northeastern University, Shenyang, China, in 2014 and 2019, respectively. From 2017 to 2018, he conducted Academic Research with the University of Tennessee, Knoxville, TN, USA. He is currently a Lecturer with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or coauthored more than 20 technical papers in international journals and conferences. His research interests include software-defined networking, optical networks, resilient communication, traffic engineering, and network optimization. Dr. Zhang was the recipient of the Best Paper Award of Qshine in 2017. He is the Member of OPTICA.

Xiaoxue Gong (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Northeastern University, Shenyang, China, in 2011, 2013, and 2017, respectively. She is currently an Associate Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests include PON systems, optical OFDM techniques, and secure optical communication.

Lei Guo (Member, IEEE) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Professor with the Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or coauthored more than 200 technical papers in international journals and conferences, such as the IEEE TRANSACTIONS IN COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE/OSA JOURNAL OF LIGHTWAVE TECHNOLOGY, IEEE/OSA JOURNAL OF OPTICAL COMMUNICATIONS AND NETWORKING, IEEE GLOBECOM, and IEEE ICC. His research interests include communication networks, optical communications, and wireless communications. He is a member of OSA and also Senior Member of CIC. He is currently an Editor of five international journals.