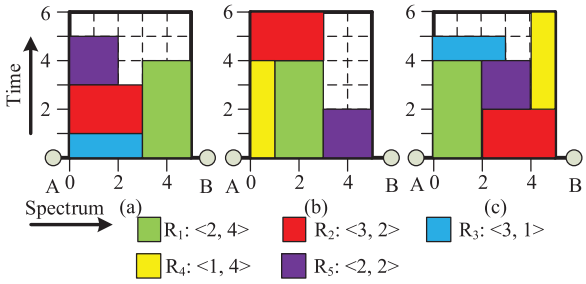
IEEE Photonics Journal
An IEEE Photonics Society Publication

Open Access

Bandwidth Reservation for Tenants in Reconfigurable Optical OFDM Datacenter Networks

Volume 10, Number 5, September 2018

Aijun Liu
Yongmei Sun
Yuefeng Ji, ***Senior Member, IEEE***



DOI: 10.1109/JPHOT.2018.2859275
1943-0655 © 2018 IEEE

IEEE photonics SOCIETY

IEEE

Bandwidth Reservation for Tenants in Reconfigurable Optical OFDM Datacenter Networks

Aijun Liu,[1] **Yongmei Sun**,[1]
and Yuefeng Ji,[1] ***Senior Member, IEEE***

[1]State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China

DOI:10.1109/JPHOT.2018.2859275

Manuscript received June 11, 2018; revised June 23, 2018; accepted July 19, 2018. Date of publication July 24, 2018; date of current version August 22, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61331008. Corresponding author: Yongmei Sun (e-mail: ymsun@bupt.edu.cn).

Abstract: The optical datacenter networks need periodical reconfiguration in response to traffic change. Before the networks reconfigure, the tenant requests are given and should be served within fixed transfer time and spectrum capacity. In this paper, the planning problem of serving the requests in optical orthogonal frequency division multiplexing datacenter networks is investigated. We introduce the knapsack-based spectrum and time allocation (KSTA) problem. The objective of this paper is to maximize the network throughput. We formulate the KSTA problem as an integer linear programming (ILP) model. However, ILP cannot find the optimal solution for large input requests within shorter time. To solve the problem, three fast heuristic algorithms, i.e., the most spectrum first, the most time first, and the most data volume first, are proposed to achieve suboptimal solutions. Furthermore, the simulated annealing (SA) algorithm is employed to yield a better suboptimal solution. The simulation results indicate that ILP provides an optimal solution for small input requests, whereas the three heuristic algorithms and SA can yield suboptimal solutions for large input requests. The results also show that the suboptimal solution to SA is better than those provided by the three heuristic algorithms.

Index Terms: Optical datacenter networks, reconfiguration, knapsack-based spectrum and time allocation, orthogonal frequency division multiplexing.

1. Introduction

With the exponential growth of global traffic, datacenters, which host many cluster-computing applications, are facing rapid increase in bandwidth demand. Today's datacenters with thousands of servers, typically employ the multi-hierarchical tree networks to provide bandwidth among the servers, which can cause the problem of bandwidth bottleneck. To overcome it, many research communities that leverage off-the-shelf commodity switches, have begun designing novel network architectures, including the Fat Tree network [1], the small-world network architecture [2], the uniform random graph [3], DCell [4] as well as BCube [5]. Although these network architectures provide good network capacity, none of them can satisfy the need of substantial traffic growth due to the limited transmission speed of electrical switching.

Compared to electrical switching, optical switching is a promising technology for improving network capacity, because it can provide huge transmission capacity. In recent years, many literatures

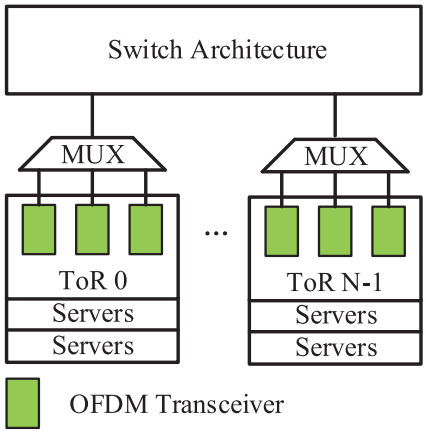


Fig. 1. Typical network architecture of OODCNs. The network architecture consists of switch architecture, MUXs, ToR switches, and servers.

have employed hybrid optical/electrical or all-optical interconnect architectures to build datacenters [6]–[12], with the high efficiency such as high throughput, low latency, and low power consumption. Many of them rely on optical switching architecture that is composed of micro-electro-mechanical systems (MEMS) or wavelength selective switch (WSS).

Although, wavelength division multiplexing (WDM) can be employed by these datacenter networks mentioned above to further increase network capacity, it can lead to inefficient capacity utilization due to its rigid and coarse granularity. Optical orthogonal frequency division multiplexing (OFDM), has been demonstrated in optical networks, to be able to serve traffic requests with fine granularity bandwidth assignment. For optical datacenter networks, optical OFDM is also a more promising modulation technology. Literature [23] has discussed the feasibility of using OFDM in data centers. Therefore, we expect optical OFDM datacenter networks (OODCNs) to provide more efficient support for cluster-computing applications. In OODCNs, a typical network architecture consists of a switch architecture, multiplexers (MUXs), top of rack (ToR) switches, and servers, as shown in Fig. 1 [15], [16]. Specifically, the switch architecture is composed of WSS or MEMS, which can provide ToR-to-ToR communication since the traffic are highly concentrated among a small number of ToR switches [40]. The ToR switches are equipped with OFDM transceivers.

OODCNs are a new type of optical networks. Although different types of optical networks have different features, one of the important factors is to improve the efficiency. For instance, in optical access networks, the bandwidth efficiency is pursued [17]; in all optical backbone networks, the spectrum efficiency is pursued [18]; the energy efficiency is pursued in all optical switching networks [19]. In OODCNs, the spectrum efficiency is also pursued.

Central to datacenters is their ability to share resources among multiple tenants [22]. Although both electrical and optical datacenters are designed for high capacity, at present cloud providers do not provide guaranteed network bandwidth to tenants [20], [21]. A tenant can involve many tasks and its application performance relies on the complete time of its last task. Because the tasks of different tenants compete for the shared bandwidth resource, the runtimes of tasks belonging to the same tenant vary significantly [20], [22]. The varying network performance of tenants makes their application performance unpredictable, and impedes the applicability of cloud to support various applications that depend on predictable performance, including user-facing web applications, data-parallel applications, and scientific computing applications [20]. To address this problem, literature [20] has proposed bandwidth reservation method that can guarantee network bandwidth for tenants in electrical datacenters, which can provide predictable application performance.

As is known, lightpaths for flow-oriented requests,[1] with reservation bandwidth and corresponding holding time, can offer guaranteed bandwidth for tenants in backbone networks, so they can also

[1] Requests can be divided into flow-oriented requests and data-oriented requests. In this paper, if not specified explicitly, requests refer to flow-oriented requests for simplification. For example, tenant requests refer to tenant flow-oriented requests.

provide guaranteed bandwidth in OODCNs. In OODCNs, the optical switches can only offer ToR-to-ToR communication. Thus, OODCNs require periodical reconfiguration to provide optical links for other ToR switch pairs in response to traffic change. Similar to other optical datacenter networks [6], [7], [10], the reconfiguration period of OODCNs tends to be from several seconds to tens of seconds. When tenant requests arrive, if optical switches do not offer optical links, the requests must wait until the establishment of their optical links. Thus, the traffic demand matrix for tenants is known in every reconfiguration, and we employ Kuhn-Munkres (KM) maximum weight matching algorithm to compute topology to provide optical links for ToR switch pairs. Although tenant requests are not served immediately, if they are served when their optical links establish, their application performance is guaranteed and predictable.

However, for these known traffic requests in every reconfiguration, the static routing spectrum allocation (RSA) problem [13] for setting up lightpaths in backbone networks does not apply to OODCNs. Because there are three significant differences in resource allocation between OODCNs and the backbone networks. First, the static RSA problem in backbone networks considers the routing, while the resource allocation problem in OODCNs does not consider it due to ToR-to-ToR communication. Second, the static RSA problem only considers spectrum demand and the spectrum consecutiveness must be satisfied. This is because the network scenario is static and traffic requests only have spectrum demand. While in OODCNs, the scenario is dynamic, the traffic requests have the demand of spectrum and holding time. To respond the traffic change, OODCNs must reconfigure. Thus, the life time of optical links is limited (the value of it is the configuration period or transfer time). An optical link, with fixed spectrum capacity and transfer time, can be shared by multiple tenants, which poses new challenges compared with the static RSA problem, i.e., spectrum and time must be considered simultaneously. Specifically, because the spectrum-time resources of optical links are given, these traffic requests should be served within the transfer time with arbitrary feasible starting spectrum and starting time. To ensure the guaranteed bandwidth for tenants, both spectrum consecutiveness and time consecutiveness constraints (i.e., using the same spectrum slots during the consecutive holding time) should be satisfied. Third, for the static RSA problem, all known traffic request should be served, and the objective is to minimize the used spectrum slots. While in OODCNs, the objective is to maximize the throughput. Because the resources of the optical links are limited, not all traffic requests can be accommodated. These traffic requests should be selectively served, i.e., a request may be served or rejected, which is called request selection constraint. Because of these constraints, we refer resource allocation problem in OODCNs to the knapsack-based spectrum and time allocation (KSTA) problem.

The KSTA problem is also different from the static data-oriented RSA (DORSA) problem in backbone networks [36]. The static DORSA problem considers the routing, spectrum, and holding time jointly. The significant differences can be reflected in four aspects. First, the KSTA problem considers flow-oriented requests that require reservation bandwidth and holding time. Their lightpaths can offer guaranteed bandwidth. The static DORSA problem considers data-oriented requests. The demand of a data-oriented request is the data volume to be transferred. The data-oriented request is valid only when its data is completely transferred from the arrival time to deadline. Its lightpath can not offer guaranteed bandwidth. Second, for our KSTA problem, if a flow-oriented request is served, its demand will be completely satisfied, which is refer to as complete transfer. While the static DORSA problem does not guarantee the complete transfer, i.e., allows part of the data to be transferred. Third, the static DORSA problem considers the routing, while the KSTA problem does not consider it. Fourth, the KSTA problem considers the time consecutiveness constraint, while the static DORSA problem does not consider it. This is because the bandwidth of the data-oriented request is not guaranteed during the data transfer. The data-oriented request allows the different spectrum slots to be used at different time, and even allows the data transfer to be suspended between the arrival time and deadline [36]. Thus the time consecutiveness constraint is not considered. Please note that the data-oriented requests are not suited for tenants in OODCNs, because no guaranteed bandwidth and even suspended data transfer, can not provide predictable application performance for tenants [20].

In this paper, we focus on the KSTA problem in OODCNs. The KSTA problem is the planning problem of such networks at the beginning of every reconfiguration period, because the request matrix is given. Every request involves spectrum demand and holding time demand. In addition to spectrum consecutiveness constraint, the time consecutiveness constraint must be considered simultaneously. Due to the limited spectrum-time resources of optical links, some requests might be rejected in every reconfiguration period, therefore, our objective is to maximize the network throughput to fully utilize the optical link resources. To the best of our knowledge, this is the first work on the KSTA problem in OODCNs. Our major contributions can be summarized as below.

- We introduce the KSTA problem formally;
- We formulate the KSTA problem as an integer linear programming (ILP) model;
- We propose three efficient heuristic algorithms with sub-optimal solutions to solve the ILP model. Furthermore, we employ the simulated annealing (SA) algorithm to get a better sub-optimal solution. We find that the solution to SA is more close to that of ILP and it outperforms the three heuristic algorithms.

The remainder of this paper is organized as follows. In Sec. 2, the related work is introduced. In Sec. 3, the steps of periodical reconfiguration is introduced. In Sec. 4, the problem of KSTA is formally stated. In Sec. 5, an ILP model is developed using mathematical formulations. In Sec. 6, three heuristic algorithms are proposed. Performance evaluations of the ILP model, the proposed algorithms, and the SA algorithm are conducted in Sec. 7, and we conclude the work in Sec. 8.

2. Related Work

OFDM has been envisioned as a promising modulation technique for elastic optical networks (EONs), thus the RWA problem is expanded to the RSA problem. Some exiting works on RSA only considered the spectrum consecutiveness constraint [13], [14], both of which satisfied it to make sure the on-demand allocation of bandwidth for requests. In EON, the advance reservation (AR) was considered by literatures [28], [29], and the deadline for multicast was considered by literature [30]. Each of them considered the routing, spectrum, and time allocation problem. The tenant requests should be severed between the arrival time and a preset deadline. Although these literatures considered the routing problem, compared to the KSTA problem, the routing spectrum and time allocation problem is simple. Because their requests are unknown in advance and their arrivals are random. Once a request arrives, the request must be considered immediately. That is, the number of request considered is one. Thus, we can easily search the available routing and spectrum for the request, to decide whether the request is served or rejected. In contrast, although our KSTA problem does not consider the routing problem, it is more challenging. Because the requests are known in advance and should be considered together. The KSTA problem is in essence a combination optimization problem and a NP-hard problem. To this end, an ILP model is usually formulated to achieve an optimal solution.

OFDM can also be used in optical datacenter and inter-datacenter networks. A software defined optical network for datacenter was discussed by literature [38]. However, the network employs a single tunable wavelength converter at each input port. The network capacity can be greatly increased by using multiple OFDM transceivers at each input port. In optical OFDM datacenter networks, the routing and spectrum/IT resource allocation was investigated based on anycast routing [37]. It considers both network and IT resources jointly. However, it does not consider the time. In multilayer optical OFDM inter-datacenter networks, literature [39] investigated the task scheduling. The task scheduling includes the establishment of lightpaths in inter-datacenter networks and the reservation of IT resources in datacenters. But the task scheduling considers data-oriented requests, thus the time consecutiveness constraint is not satisfied.

Space division multiplexing (SDM) had been investigated in order to enhance the capacity of optical networks [31]. In EON, literature [34] investigated the routing, spectrum and core allocation (RSCA) problem. Because of the inter-core crosstalk interference, the adjacent cores do not use the same spectrum, which is contrast to the time consecutiveness constraint from the KSTA problem. Literature [32] considered the combination of routing, spectrum, mode assignment (RSMA) and

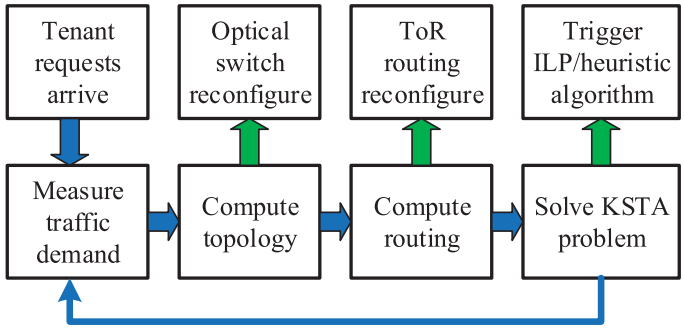


Fig. 2. The specific steps of periodical reconfiguration in OODCNs.

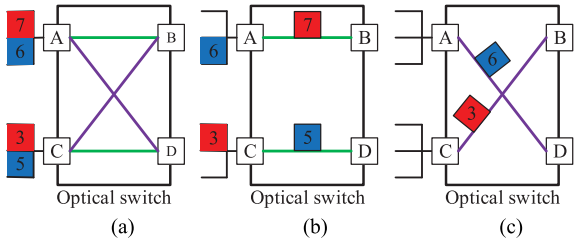


Fig. 3. Compute topology by KM algorithm. There are 7 units of data and 3 units of data colored by red toward to destination node B, while 6 units of data and 5 units of data colored by blue toward to destination node D. (a) The initial stage of the network. (b) In the first reconfiguration, the computed topology is 'A-B' and 'C-D'; (c) In the second reconfiguration, the computed topology is 'A-D' and 'C-B'.

RSCA, while the AR was considered by literature [33]. However, each of them is also different from the KSTA problem, owing to the inter-core crosstalk interference.

Recently, free-space optics (FSO) technology was employed to build wireless datacenter networks [24], [25]. Literature [24] addressed the problem of topology reconfiguration by extending the traditional Blossom algorithm and a greedy augmenting-path approach for flow routing. Literature [25] tackled the topology reconfiguration problem by constructing two topologies and proposing a two-tier scheduling algorithm. Literature [26] solved the problem by designing two topologies and considering coloring problem on a bipartite multigraph. Intermediate topologies were employed to achieve the reconfiguration progressively and reduce total traffic loss during the reconfiguration of topology [27]. However, these approaches of topology reconfiguration do not consider the bandwidth reservation for tenants, which can not provide predictable application performance for tenants [20]. Literature [16] considered revenue-oriented spectrum allocation in OODCNs, which can offer guaranteed spectrum for tenants. However, it does not consider the time due to the lacking of topology reconfiguration.

3. Periodical Reconfiguration

In this section, we present the steps of periodical reconfiguration in OODCNs. Our goal is to compute the optimal topology such that the network throughput is maximized for a given traffic demand matrix. The specific steps of periodical reconfiguration is shown in Fig. 2, mainly including four steps.

1. Measure traffic demand: The work of this module is to measure, update the traffic demand for ToR switch pairs, and provide the traffic demand matrix for the second module.

2. Compute the topology: For a given traffic demand matrix, we localize high volume ToR switch pairs communicating over the links of optical switch. This is accomplished by using KM maximum weight matching algorithm. For example, there is 2×2 optical switch with two possible topologies. The first is 'A-B' and 'C-D', which is specified by green lines shown in Fig. 3(a), while the other is

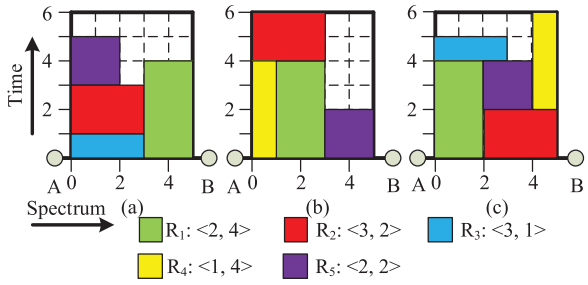


Fig. 4. Feasible KSTA in OODCNs. Requests R_1, R_2, R_3, R_4, and R_5 with different spectrum and time demands are colored in green, red, blue, yellow, and purple, respectively. (a) R_4 is rejected and the throughput is 21. (b) R_3 is rejected and the throughput is 22. (c) All requests are served and the throughput is 25.

'A-D' and 'C-B' depicted by purple lines. In addition, in Fig. 3(a), there are 7 units of data and 3 units of data colored by red toward to destination node B, while 6 units of data and 5 units of data colored by blue toward to destination node D. In the first reconfiguration, the topology computed by KM algorithm is 'A-B' and 'C-D', as shown in Fig. 3(b). For the next reconfiguration, the computed topology is 'A-D' and 'C-B', as depicted in Fig. 3(c).

3. Compute the routing: Once we have optical links, the optical switch configuration is known. We proceed to use the standard shortest path routing schemes such as Dijkstra algorithm to compute routing. Note that some of the optical switches are single-hop connection, such as MEMS. Hence, the routing is optical link itself.

4. Compute the spectrum: Given the traffic demand and associated routing between any pair of ToR switches, we proceed to compute spectrum to serve them by establishing individual lightpath. However, computing the spectrum is in essence to solve the KSTA problem. It can be formulated by ILP and quickly solved by heuristic algorithms. We will discuss the KSTA problem in detail in the following three sections. Once the requests have been considered and the transfer time is over, the next reconfiguration will proceed until all requests have been considered.

4. Knapsack-Based Spectrum and Time Allocation in OODCNs

In this section, we first introduce an example of KSTA in OODCNs, and then introduce the KSTA problem formally.

An example of KSTA in OODCNs is shown in Fig. 4. In this case, the OODCN is consisted of four nodes and two links. For each link, we divide spectrum (horizontal axis) into spectrum slots and divide time (vertical axis) into time slots. Each link have 5 spectrum slots and 6 time slots (the transfer time is 6). For simplification, only one link, 'A-B' is considered. Because the resource demand of a request includes both spectrum and time dimensions, it can be exhibited by a rectangle. From Fig. 4, there are five requests R_1, R_2, R_3, R_4, and R_5 from the source node A to the destination node B; R_1 colored in green has 2 spectrum slots and 4 time slots, while R_2, R_3, R_4, and R_5 are colored in red, blue, yellow and purple, respectively. To be a feasible KSTA, any two rectangles should not overlap. The horizontal length of a rectangle guarantees the spectrum consecutive constraint, while the time consecutive constraint is specified by the vertical length of the rectangle. Figure 4(a)–(c) are the feasible KSTA instances. However, not all requests are served under the limited spectrum-time resources of the optical link. In Fig. 4(a), R_4 is rejected, while R_3 is rejected in Fig. 4(b). Only Fig. 4(c) guarantees all the requests are served. Therefore, the throughput for Fig. 4(a)–(c) is 21, 22, and 25, respectively. The throughput of a request is its data volume, i.e., the product of spectrum demand and time demand.

Then, we introduce the KSTA problem formally. We consider an OODCN topology as $G(N, E, F, T)$, where N is the node set, E is the set of bidirectional links. Because the OODCN topology only provides ToR-to-ToR communication, the number of nodes is two times that of links, i.e., $|N| = 2|E|$. In addition, F is the capacity of spectrum slots on each link, and T is the size of

time slots on each link. A request R_{ij} is a tuple $\langle F_{ij}, T_{ij}\rangle$ that denotes the size (in terms of the number of spectrum slots) of spectrum demand between the source node i and the destination node j, and the size (in terms of the number of time slots) of time demand.

Definition: The KSTA problem - given a OODCN topology $G(N, E, F, T)$, a predefined set of requests. For any request R_{ij} with $\langle F_{ij}, T_{ij}\rangle$, is it possible to establish lightpaths for the requests using both consecutive spectrum slots and consecutive time slots while maximizing the network throughput?

Theorem: The KSTA problem is NP-hard.

Proof: The NP-hard bin packing problem [35] is reduced to the KSTA problem. A set of numbers and a fixed set of bins are given, and each bin has the same capacity. An instance of the bin packing problem is to allocate each number to one of bins, with the constraint of the sum of numbers in each bin not exceeding the bin capacity.

Now, an optical link of OODCN is considered and an equivalent instance of the KSTA problem is generated. For the number in bin packing problem, we generate a rectangle (tenant's lightpath) with unit width and its height is the value of the number. Furthermore, we also generate an optical link (i.e., an enclosing rectangle) whose height is the bin capacity, and whose width is the number of bins. Thus, each bin maps to a vertical strip in the optical link. In the KSTA problem, each rectangle is allocated to a column (bin) of the optical link, such that the sum of heights (numbers) of the rectangles allocated to each column (bin) does not exceed the height (the capacity of the bin) of the optical link. Thus, the KSTA problem is equivalent to the bin packing problem. Because the bin packing problem is NP-hard, the KSTA problem is NP-hard as well. ■

5. Integer Linear Programming Model

In this section, we formulate the KSTA problem as an ILP model, and the objective of it is to maximize the network throughput.

5.1 Notations and Variables

- $\mathbf{D}$: The set of tenant requests; $\mathbf{D} = \bigcup_{ij} \mathbf{D_{ij}}$;
- $\mathbf{D}_{ij}$: The set of requests that has the same source and destination nodes, i.e., these requests share a common link 'i-j'; in this paper, we consider the ToR-to-ToR communication provided by MEMS, thus the link 'i-j' is its routing; $\mathbf{D_{ij}} = \{R_{ij}\}$, where $R_{ij} = \langle F_{ij}, T_{ij}\rangle$ denotes that the request R_{ij} requires F_{ij} spectrum slots and T_{ij} time slots, respectively;
- $\tilde{R}_{ij}$: The request is from $\mathbf{D_{ij}}$ and requires $\tilde{F}_{ij}$ spectrum slots and $\tilde{T}_{ij}$ time slots; it differs from R_{ij};
- F: A constant that denotes the capacity of all spectrum slots in each link;
- T: A constant that denotes the size of the transfer time in each link, i.e., the number of time slots;
- f_{ij}: Integer variable that represents the starting frequency for the request R_{ij}; the starting frequency f_{ij} is relative to Y-axis in each link and satisfies $0 \leq f_{ij} < F$;
- t_{ij}: Integer variable that denotes the starting time for the request R_{ij}; t_{ij} is relative to X-axis in each link and satisfies $0 \leq t_{ij} < T$;
- $\delta_{ij,mn}$: Boolean variable that denotes the relative position between the requests R_{ij} and R_{mn} in the spectrum dimension; Please note that to formulate the KSTA problem from the perspective of OODCNs, we allow R_{mn} to come from D_{ij} and share the link 'i-j'; then R_{mn} is identical to $\tilde{R}_{ij}$ and the variable $\delta_{ij,mn}$ becomes $\delta_{ij,\tilde{ij}}$;
- $o_{ij,mn}$: Boolean variable that denotes the relative position between the requests R_{ij} and R_{mn} in the time dimension;
- $c_{ij,mn}$: Constant that equals to 1 if requests R_{ij} and R_{mn} share the link 'i-j' (i.e., $R_{mn} = \tilde{R}_{ij}$ and $c_{ij,mn} = c_{ij,\tilde{ij}} = 1$), and 0 otherwise;
- s_{ij}: Boolean variable that equals to 1 if request R_{ij} is served, and 0 is rejected;

5.2 Objective and Constraints of the KSTA Problem

Because the transfer time and spectrum capacity of each optical link and tenant requests are given, some requests should be served within the transfer time, while others may be rejected. Therefore, our objective is to maximize the network throughput, as shown in Eq. (1).

$$Maximize \quad \sum_{ij} s_{ij} * F_{ij} * T_{ij}, \forall R_{ij} \in \mathbf{D} \tag{1}$$

subject to the following constraints:

- Transfer time constraint:

$$T \geq t_{ij} + T_{ij}, \forall R_{ij} \in \mathbf{D} \tag{2}$$

  For each request, it should be served within the fixed transfer time, as specified by Eq. (2).
- Spectrum capacity constraint:

$$F \geq f_{ij} + F_{ij}, \forall R_{ij} \in \mathbf{D} \tag{3}$$

  For each request, Eq. (3) denotes that it should be served within the spectrum capacity.
- Link spectrum-time constraint:

$$\sum_{ij} s_{ij} * F_{ij} * T_{ij} \leq F * T, \forall R_{ij} \in \mathbf{D_{ij}} \tag{4}$$

  For some requests that share a common link, the served requests do not exceed the capacity of spectrum-time resources of the link, as specified by Eq. (4).
- Spectrum consecutiveness constraints, time consecutiveness constraints, non-overlapping constraints, and request selection constraints:

$$f_{ij} + F_{ij} - f_{mn} \leq F(3 + \delta_{ij,mn} + o_{ij,mn} - c_{ij,mn} - s_{ij} - s_{mn}) \quad \forall R_{ij}, R_{mn} \in \mathbf{D} \tag{5}$$

$$f_{mn} + F_{mn} - f_{ij} \leq F(4 + \delta_{ij,mn} - o_{ij,mn} - c_{ij,mn} - s_{ij} - s_{mn}) \quad \forall R_{ij}, R_{mn} \in \mathbf{D} \tag{6}$$

$$t_{ij} + T_{ij} - t_{mn} \leq T(4 - \delta_{ij,mn} + o_{ij,mn} - c_{ij,mn} - s_{ij} - s_{mn}) \quad \forall R_{ij}, R_{mn} \in \mathbf{D} \tag{7}$$

$$t_{mn} + T_{mn} - t_{ij} \leq T(5 - \delta_{ij,mn} - o_{ij,mn} - c_{ij,mn} - s_{ij} - s_{mn}) \quad \forall R_{ij}, R_{mn} \in \mathbf{D} \tag{8}$$

  For any two requests R_{ij} and R_{mn}, if they do not share the link 'i-j' (i.e., do not come from the same D_{ij}, and $c_{ij,mn} \neq 1$), Eqs. (5)–(8) are not considered since their values of the left-hand side are always less than those of the right-hand side. Otherwise, if they share the link 'i-j' (i.e., $R_{mn} = \tilde{R}_{ij}$ and $c_{ij,mn} = c_{ij,\tilde{ij}} = 1$), these constraints are activated and reduced to:

$$f_{ij} + F_{ij} - \tilde{f}_{ij} \leq F(2 + \delta_{ij,\tilde{ij}} + o_{ij,\tilde{ij}} - s_{ij} - \tilde{s}_{ij}) \quad \forall R_{ij}, \tilde{R}_{ij} \in \mathbf{D_{ij}} \tag{9}$$

$$\tilde{f}_{ij} + \tilde{F}_{ij} - f_{ij} \leq F(3 + \delta_{ij,\tilde{ij}} - o_{ij,\tilde{ij}} - s_{ij} - \tilde{s}_{ij}) \quad \forall R_{ij}, \tilde{R}_{ij} \in \mathbf{D_{ij}} \tag{10}$$

$$t_{ij} + T_{ij} - \tilde{t}_{ij} \leq T(3 - \delta_{ij,\tilde{ij}} + o_{ij,\tilde{ij}} - s_{ij} - \tilde{s}_{ij}) \quad \forall R_{ij}, \tilde{R}_{ij} \in \mathbf{D_{ij}} \tag{11}$$

$$\tilde{t}_{ij} + \tilde{T}_{ij} - t_{ij} \leq T(4 - \delta_{ij,\tilde{ij}} - o_{ij,\tilde{ij}} - s_{ij} - \tilde{s}_{ij}) \quad \forall R_{ij}, \tilde{R}_{ij} \in \mathbf{D_{ij}} \tag{12}$$

  Then, we analyse the request selection constraints. If either of them is not served ($s_{ij} \neq 1$ or $\tilde{s}_{ij} \neq 1$). Eqs. (9)–(12) are not considered since their values of the left-hand side are always less than those of the right-hand side. In contrast, only when they are served ($s_{ij} = 1$ and $\tilde{s}_{ij} = 1$), these constraints are activated and reduced to:

$$f_{ij} + F_{ij} - \tilde{f}_{ij} \leq F(\delta_{ij,\tilde{ij}} + o_{ij,\tilde{ij}}) \tag{13}$$

$$\tilde{f}_{ij} + \tilde{F}_{ij} - f_{ij} \leq F(1 + \delta_{ij,\tilde{ij}} - o_{ij,\tilde{ij}}) \tag{14}$$

$$t_{ij} + T_{ij} - \tilde{t}_{ij} \leq T(1 - \delta_{ij,\tilde{ij}} + o_{ij,\tilde{ij}}) \tag{15}$$

$$\tilde{t}_{ij} + \tilde{T}_{ij} - t_{ij} \leq T(2 - \delta_{ij,\tilde{ij}} - o_{ij,\tilde{ij}}) \tag{16}$$

To prevent the overlap of any two requests in both spectrum and time dimensions, we introduce two boolean variables $\delta_{ij,\tilde{ij}}$ and $o_{ij,\tilde{ij}}$. The combination of them can indicate the relative positions of the two requests. There are four values for the combination of the two variables, i.e., $(\delta_{ij,\tilde{ij}}, o_{ij,\tilde{ij}}) = (0,0), (0,1), (1,0), or (1,1)$.

For each of the four values of $(\delta_{ij,\tilde{ij}}, o_{ij,\tilde{ij}})$, only one inequality for Eqs. (13)–(16) can be further activated. For instance, when $(\delta_{ij,\tilde{ij}}, o_{ij,\tilde{ij}}) = (0,1)$, Eq. (14) is activated and reduced to be

$$\tilde{f}_{ij} + \tilde{F}_{ij} - f_{ij} \leq 0 \tag{17}$$

Equation. (17) denotes that the request R_{ij} is on the right of $\tilde{R}_{ij}$, resulting in no overlap between the two requests. Moreover, Eq. (17) guarantees that $\tilde{R}_{ij}$ occupies $\tilde{F}_{ij}$ units of spectrum slots. However, in this case, the other three equations can not be further activated, because they are always satisfied. For example, for Eq. (13), it is deactivated and becomes

$$f_{ij} + F_{ij} - \tilde{f}_{ij} \leq F \tag{18}$$

Equation. (18) always holds irrespectively of f_{ij} and $\tilde{f}_{ij}$, because the difference of f_{ij} and $\tilde{f}_{ij}$ is always less than F.

For the same reason, $(\delta_{ij,\tilde{ij}}, o_{ij,\tilde{ij}}) = (1,1)$ can activate Eq. (16) and reduce it to be

$$\tilde{t}_{ij} + \tilde{T}_{ij} - t_{ij} \leq 0 \tag{19}$$

Equation. (19) indicates that the request R_{ij} is above $\tilde{R}_{ij}$ and the latter occupies $\tilde{T}_{ij}$ units of time slots.

In addition, it can be seen that (0, 0) can reduce Eq. (13) to $f_{ij} + F_{ij} \leq \tilde{f}_{ij}$ that specifies R_{ij} is on the left of $\tilde{R}_{ij}$, while (1, 0) can reduce Eq. (15) to $t_{ij} + T_{ij} \leq \tilde{t}_{ij}$ that represents R_{ij} is under $\tilde{R}_{ij}$.

The above ILP model is mainly designed from two aspects. The one is when any two requests do not share a common link, the overlap is not considered. The other is how to prevent the overlap of any two requests when they are served. Specifically, Eqs. (5)–(8) are activated and reduced to Eqs. (9)–(12), respectively. Then these equations reduced are further reduced to Eqs. (13)–(16), respectively. Only one of Eqs. (13)–(16) can be activated further by the value of $(\delta_{ij,\tilde{ij}}, o_{ij,\tilde{ij}})$, which can determine their relative positions. Besides, the other inequalities can not be further activated and always be satisfied to guarantee no overlap in their spectrum and time dimensions.

The technical process of the ILP model is to find the starting frequency f_{ij} and the starting time t_{ij} for the request R_{ij} over its path that is composed of a single link. The non-overlapping time constraint specifies the time consecutiveness skillfully, which is similar to the spectrum consecutiveness constraint.

6. Heuristic Algorithms

ILP can only provide an optimal solution using the optimal algorithm within shorter time for small input requests. However, the computation complexity of the optimal algorithms are significant and ILP can not find the optimal solution for large input requests within shorter time; thus quick heuristic algorithms with sub-optimal solutions are resorted to tackle the ILP model for large input requests. The idea of heuristic algorithm is allocating the spectrum and time resources to the requests one-by-one sequentially along the spectrum dimension. Since the KSTA problem is in essence a problem of combination optimization, the ordering is quite important in the heuristics, and different orderings will lead to different throughput. In this section, three heuristic algorithms–the most spectrum first (MSF) algorithm, the most time first (MTF) algorithm, and the most data volume first (MDVF) algorithm–are proposed based on different ordering policies. In addition, we use the SA algorithm to find a better ordering based on MDVF.

6.1 MSF Algorithm

The MSF algorithm is based on a descending order of all requests that will be considered, according to their spectrum demand, which can fully utilize the spectrum slots of the optical links. Because of

Algorithm 1: MSF Algorithm.

Input: $G = (N, E, F, T)$, F_{ij}, T_{ij}, $i, j \in N$, $throughput = 0$;
Output: $throughput$

1: **Ordering policy of MSF**: Sort the requests according to their spectrum demand F_{ij} in the descending order in a queue Q; serve the first request that is at the top of Q, and it requires the highest number of spectrum slots.
2: **while** $Q \neq \emptyset$ **do**
3: **Searching the lowest starting coordinate**:
4: $R_{ij} \leftarrow$ the request at top of Q
5: Find the lowest (f_{ij}, t_{ij}) for R_{ij} :
6: $f_{ij} \leftarrow 0$, $t_{ij} \leftarrow 0$, counter=0
7: **for** t_{ij};$t_{ij} \leq T - T_{ij} + 1$;$t_{ij}++$ **do**
8: **for** f_{ij};$f_{ij} \leq F - F_{ij} + 1$;$f_{ij}++$ **do**
9: **if** From $t_{ij}.f_{ij}$ to $t_{ij}.(f_{ij} + F_{ij} - 1)$ is not available **then**
10: $f_{ij} \leftarrow f_{ij} + 1$; counter=0;
11: **else**
12: counter = counter+1;
13: **if** counter$==T_{ij}$ **then**
14: Obtain t_{ij} and f_{ij}; break; break;
15: **end if**
16: t_{ij}++;
17: **end if**
18: **end for**
19: **end for**
20: **if** t_{ij} and f_{ij} is not obtained **then**
21: Reject the request; go to Line. 25;
22: **end if**
23: **Spectrum allocation**: Assign the time and spectrum slots for the request.
24: $throughput = throughput + T_{ij} * F_{ij}$
25: Delete it from the queue Q, and update the spectrum, time and queue states in the network.
26: **end while**
27: return $throughput$

the nature of spectrum-time resources, only the combination of f_{ij} and t_{ij} can specify a valid resource allocation for a request. Thus, in this section, we introduce (f_{ij}, t_{ij}) that denotes the lowest staring coordinate for R_{ij}. We find (f_{ij}, t_{ij}) along the spectrum dimension from the low spectrum to the high, under the low t_{ij}. If we can not find the (f_{ij}, t_{ij}), we will search the spectrum dimension of higher t_{ij}. For example, in Fig. 4(a), we first serve R_3, then serve R_2, R_1, and finally serve R_5. The lowest staring coordinate of R_3 is $(0, 0)$. For R_2, R_1 and R_5, their lowest staring coordinate are $(0, 1)$, $(3, 0)$ and $(0, 3)$, respectively.

The specific process of MSF is depicted in Algorithm 1. Sort the requests according to their spectrum demand F_{ij} in the descending order in queue Q. Then we find the lowest starting coordinate (f_{ij}, t_{ij}) for each request. That is, we scan available spectrum-time resource of a rectangle with length of T_{ij} and width of F_{ij} from coordinate $(0, 0)$, as shown in lines $7-19$. Specifically, along with horizontal axis, we find the available spectrum slots line-by-line. We introduce a *counter*, and it adds one after finding F_{ij} units of spectrum slots under higher t_{ij}. If we can find these available spectrum slots under consecutiveness starting time from t_{ij} to $t_{ij} + T_{ij} - 1$, i.e., $counter == T_{ij}$, the lowest starting coordinate (f_{ij}, t_{ij}) is obtained. Obviously, the other three coordinates of the rectangle are $(f_{ij} + F_{ij} - 1, t_{ij})$, $(f_{ij}, t_{ij} + T_{ij} - 1)$, and $(f_{ij} + F_{ij} - 1, t_{ij} + T_{ij} - 1)$, respectively. In contrast, if the lowest

starting coordinate is not obtained, the optical link can not accommodate the request. Hence, the request is rejected and next request is considered.

Line. 23 is the spectrum allocation algorithm using the first fit policy. The variable $throughput$, which stores the current throughput, is updated by the next request if possible. We repeat the operations of searching the lowest starting coordinate and spectrum allocation algorithm for other requests until all of them have been considered.

The computation complexity of MSF is bounded by $O(|\mathbf{R}|^2 + |\mathbf{R}||N|^2)$ in the worst case.

6.2 MTF Algorithm

Although MSF can make the best of the spectrum resource, with the execution of MSF, it can greatly reject the requests with small spectrum demand but large time demand. This is because that MSF allocates the spectrum slots along with the horizontal axis, making the residual transfer time of the link shorter. The shorter residual transfer time do not satisfy the requests with larger time demand.

To overcome the shortcoming of MSF. We propose the MTF algorithm that employs a descending order of requests, according to their demands of time slots. The remaining process are the same as MSF, so we omit them. MTF preferential serves the request that requires the higher number of time slots. Besides, although the requests with large spectrum demand can be considered latter, they also have larger chance of being served. This is because that the spectrum capacity of the link is a constant F. An instance of MTF is shown in Fig. 4(b), and it is better than that of MSF shown in Fig. 4(a). From Fig. 4(b), the ordering is 'R_4-R_1-R_5-R_2'. Although R_2 is consider latter, it is also served.

The computation complexity of the MTF algorithm is also bounded by $O(|\mathbf{R}|^2 + |\mathbf{R}||N|^2)$ in the worst case.

6.3 MDVF Algorithm

Since the demand of request is two dimensional, a straightforward parameter is the data volume that is the product of time and spectrum. The data volume of a request is equal to its throughput. As shown in Fig. 4, the data volumes of requests R_1, R_2, R_3, R_4, and R_5 are 8, 6, 3, 4, and 4, respectively. The MDVF algorithm adopts a descending order based on the data volumes of requests. An instance of MDVF is shown in Fig. 4(c), which is better than MSF and MTF, since it guarantees requests with larger data volume is considered first.

The MDVF algorithm has the same computation complexity as MSF and MTF, bounded by $O(|\mathbf{R}|^2 + |\mathbf{R}||N|^2)$ in the worst case.

6.4 Simulated Annealing Algorithm

SA can solve the problems of combination optimization [13], thus we utilize it to find a better ordering that achieves better network throughput based on MDVF. SA can find a better sub-optimal solution through probabilistic searching the state space of solution.

The SA heuristic is described in Algorithm 2. It uses the ordering of requests based on MDVF to initiate the current ordering $order_{cur}$ and the optimal ordering $order^*$. Their throughput are denoted by $throughput_{cur}$ and $throughput^*$, respectively. In addition, the threshold of iterators and the temperature are set by $Thr = 100$ (or $Thr = 1000$) and $Te = 100$, respectively. The algorithm searches in the search space (line 2–14) until the number of iterators exceeds the threshold. At each iteration, we use the $ComputerNeighbor$ to achieve a neighbor ordering $order_{nei}$ by interchanging R_{ij} and R_{mn} uniformly. Based on $order_{nei}$, we employ $ComputerThroughput$ to achieve its throughput $throughput_{nei}$ by serving requests one-by-one. A probabilistic function P is used to decide whether to transit from the current ordering to the neighbor ordering. Such P is defined as follows: if the throughput of neighbor ordering is higher than that of current ordering, the probability is 1; otherwise, the probability is $e^{(throughput_{cur} - throughput_{nei})/Te}$. Note that because more iterations means more running time, we only consider the iterations at one temperature.

Algorithm 2: SA Algorithm.

Initialization: $order^* \leftarrow order_{cur}$, $throughput^* \leftarrow throughput_{cur}$, $Thr \leftarrow 100$ or $Thr \leftarrow 1000$, $Te \leftarrow 100$;
while $Thr > 0$ **do**
  $order_{nei} \leftarrow ComputerNeighbor(order_{cur})$
  $throughput_{nei} \leftarrow ComputerThroughput(order_{nei})$
  if $throughput_{nei} > throughput^*$ **then**
    $throughput^* \leftarrow throughput_{nei}$;
    $order^* \leftarrow order_{nei}$;
  end if
  if $P(throughput_{cur}, throughput_{nei}, Te) > rand(0, 1)$ **then**
    $throughput_{cur} \leftarrow throughput_{nei}$;
    $order_{cur} \leftarrow order_{nei}$;
  end if
  $Thr \leftarrow Thr - 1$;
end while
return $throughput^*$

7. Performance Evaluation

In this section, we present the optimal solution for the ILP model and the sub-optimal solutions to the three proposed algorithms and the SA algorithm. They are running on a machine with a 3.30 GHz processor and 32 GB RAM. The ILP model is implemented using CPLEX 12.5, while the three proposed algorithms and the SA algorithm are implemented under the Visual Studio 2013 C++ simulation platform.

We assume that source ToR switch and destination ToR switch are uniformly generated and they are interconnected by MEMS. The available spectrum capacity of each MEMS optical link is 20 spectrum slots, and the transfer time is 40 units of time. Such transfer time means that every 40 units of time, the KM algorithm is employed to set up optical links for some ToR switch pairs. The weight of a link is the data volume between the ToR switches. The arrival of tenant request follows Poisson process [20], [22], [24], [25] and their time demand follows a negative exponential distribution with mean of 0.2 units of time. Traffic demand for each ToR switch pair in terms of spectrum slots number is randomly generated between 3 to spectrum granularity (SG) [41], [42]. SG is the upper bound of spectrum demand for any request ($S_{ij} \leq SG$).

7.1 ILP and Heuristic Algorithms for Small Input Requests in Small Topologies

Since the KSTA problem is NP-hard, the ILP model can find an optimal solution within shorter time only for small input instances in small networks. Thus, to evaluate the gap between the optimal solution of the ILP model and the proposed heuristic algorithms as well as the SA algorithm, they are implemented on a 8×8 MEMS which includes 16 ToR switches and eight bidirectional links in each reconfiguration period. The requests are launched and terminated by the ToR switches. However, with the execution of simulation, the number of input requests increases in the following reconfiguration periods, and thus the optimal solution can not be found within shorter time. To this end, we only consider 50 input requests at a reconfiguration period. Hence, the network topology is fixed, and ToR i is connected to ToR $i + 8$, where $0 \leq i \leq 7$. For comparison, a scheme of first-come-first-served (FCFS) is considered as the baseline. FCFS considers a ascending order of requests, according to their index number. Smaller index number indicates the corresponding request coming earlier. The metric here is the network throughput and the running time.

The network throughput with varying spectrum granularity is investigated, and the results are shown in Fig. 5(a). It is clear that the throughput decreases when spectrum granularity increases. Because the increase of spectrum granularity gives rise to the growth of network load. The limited

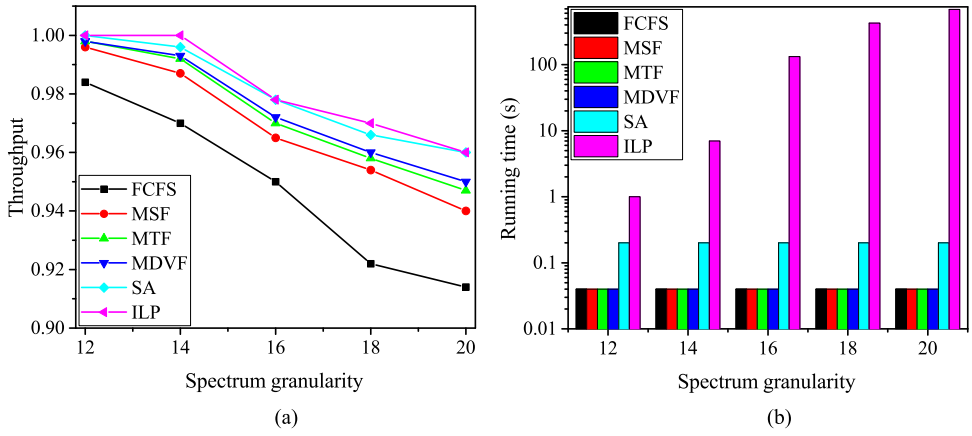


Fig. 5. Network performance versus spectrum granularity in a 8×8 MEMS with fixed topology. (a) Throughput. (b) Running time.

spectrum-time resources of MEMS links can not accommodate all the request, leading to the rejection of more and more requests. Thus, these rejected requests result in the decrease of the throughput. From Fig. 5(a), MSF is better than FCFS that performs badly, because the former can fully utilize the spectrum resources. Compared to MSF, MTF possesses better sub-optimal solution, since it guarantees that the requests with larger time demand are served first. In addition, it also ensures the requests with larger spectrum have large chance to be served in the latter phase of the algorithm execution. From Fig. 5(a), MDVF is better than MTF. The reason is that MDVF can jointly take advantage of spectrum and time, and ensure that requests with bigger data volume are served first. SA obtains the best sub-optimal solution, since it carries out many iterators (the threshold of the iterations is 100) to find a better ordering by probabilistic searing. Moreover, the performance of SA is more close to that of ILP because of small input requests. As is known, ILP provides an optimal solution since the optimal algorithm, branch-and-bound is employed. When $SG = 20$, both ILP and SA outperform FCFS by up to 5%, while MSF, MTF, and MDVF can increase the network throughput compared to FCFS by up to 2.8%, 3.6%, and 3.9%, respectively.

With the growth of spectrum granularity, the trend of the running time for all algorithms and ILP is exhibited by Fig. 5(b). For our proposed heuristic algorithm and SA algorithm, the running time is a constant as the spectrum granularity increases, owing to the constant number of input requests. All the heuristic algorithm has the same running time of around 40 milliseconds due to the same computation complexity. Compared to them, SA consumes more time with around 200 milliseconds because of iterations. As expected, from Fig. 5(b), we also discovery that ILP requires the most time, more than 600 seconds at SG of 20. Because when network load is high, the spectrum-time resources of links can not accommodate all the requests, and thus the request selection constraint plays an important role such that the complexity of the inequality constraints increases sharply.

7.2 Heuristic Algorithms for Large Input Requests in Large Topologies

To evaluate the proposed algorithms and SA further, we employ a 16×16 MEMS composed of 32 nodes and 16 bidirectional links as a large OODCN topology. 10000 tenant requests are considered. The metric here is network throughput. However, due to many reconfiguration periods involved in the simulation, each reconfiguration has different input requests and different throughput. Therefore, to simplify statistics, we normalize the throughput, i.e., the ratio of the sum of served data volume and the sum of total input data volume, which is formulated by $\frac{\sum_{ij} s_{ij} * F_{ij} * T_{ij}}{\sum_{ij} F_{ij} * T_{ij}}$. The other metrics here including bandwidth occupation ratio and the running time, are considered.

With the varying traffic load, the trend of throughput is investigated by setting $SG = 17$, and the results are shown in Fig. 6(a). For all algorithms, the increasing $Erlang$ can result in the decrease of

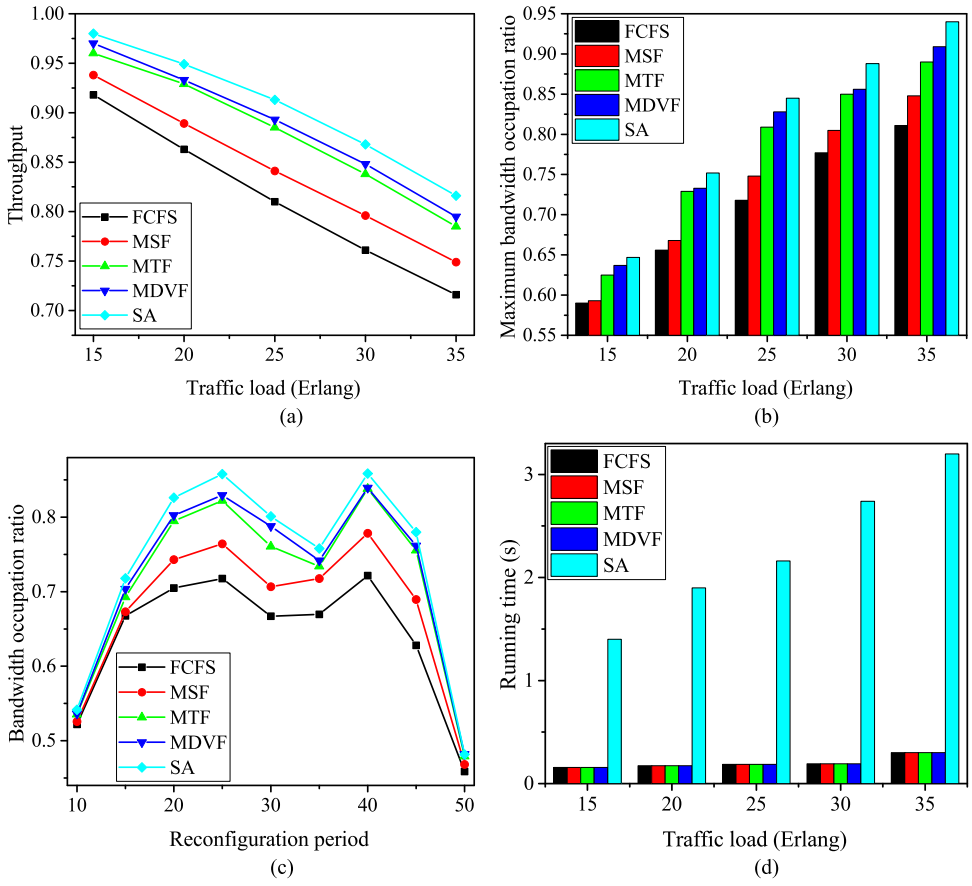


Fig. 6. Network performance in a 16 × 16 MEMS. (a) Throughput versus traffic load. (b) Maximum bandwidth occupation ratio versus traffic load. (c) Bandwidth occupation ratio versus reconfiguration period. (d) Running time versus traffic load.

the throughput. This is because that in each reconfiguration period, the number of tenant requests increases along with the network load increases. The limited spectrum-time resources of MEMS links can not accommodate all the request, and more and more requests are rejected. As a result, the throughput decreases. From Fig. 6(a), FCFS performs badly. MSF is better than FCFS because the former considers the requests with larger spectrum demand first. Compared to MSF, MTF performs better, since it serves the requests with larger time demand first. Besides, MDVF is better than MTF, because the former ensures the requests with larger data volume are served first. It is obvious that SA performs best, since it carries out many iterators (1000) to find a better ordering by probabilistic searing. When the network load is high at *Erlang* of 35, SA outperforms FCFS by up to 14%, whereas MSF, MTF, and MDVF outperform that by up to 4.6%, 9.6%, and 11%, respectively.

Due to the simulation involving many reconfiguration periods, each period has different input requests and different bandwidth occupation ratio. Thus, the maximum bandwidth occupation ratio is selected and investigated as traffic load increases, and the results are presented by Fig. 6(b). As traffic load increases, the number of input requests increases accordingly, resulting in the fully utilization of spectrum-time resources of MEMS links. Thus, the maximum bandwidth occupation increases according. The discrepancy between all the algorithms are similar to those shown in Fig. 6(a), which further validates that all our proposed algorithms with sub-optimal solutions can outperform FCFS obviously. Because each of them utilizes the same spectrum-time resources of MEMS links to serve more requests, resulting in a associated increase in bandwidth efficiency. It is reasonable that SA possesses highest bandwidth occupation ratio because of the highest throughput. We can also discovery that when the network load is high with $Erlang = 35$, MSF, MTF,

and MDVF can increase the maximum bandwidth occupation ratio in comparison to FCFS by up to 4.5%, 9.7%, and 12%, respectively, whereas SA can increase that by up to 16%.

Figure 6(c) reveals the trend of bandwidth occupation in different reconfiguration periods by setting $Erlang = 30$. The tread mainly involves three phases. First, with the input requests arrival, more and more request are served and bandwidth occupation ratio increases, which can be revealed by Fig. 6(c) from 10∼15 reconfiguration periods. Then, at reconfiguration periods of 15∼45, because of the arrival and handling of requests, the bandwidth occupation ratio steps in a stable stage, which fluctuates at higher bandwidth occupation ratio. Last, no input requests arrive and the bandwidth occupation ratio decreases sharply, as shown in Fig. 6(c) at reconfiguration periods of 45∼50.

The average running time of each reconfiguration period for all algorithms are also considered under different traffic load, and the results are exhibited by Fig. 6(d). For FCFS, MSF, MTF, and MDVF, the running time increases slightly as the traffic load increases, owing to the increase of the number of input requests. In addition, they consume almost the same time, due to the same computation complexity. For instance, they consume around 300 milliseconds at $Erlang$ of 35. As expected, SA consumes more time because of iterations, and it requires around 3.2 seconds when $Erlang = 35$.

8. Conclusion

In this paper, the problem of KSTA was formally stated for the OODCNs reconfiguration in response to the change of traffic. The objective of the KSTA problem was to maximize the network throughput under the given tenant requests. Compared to the static RSA problem, the KSTA problem considered the constraints of spectrum consecutiveness, time consecutiveness, and request selection. An ILP model was formulated for KSTA problem. ILP could achieve an optimal solution using the optimal algorithms. However, the computation complexity of the optimal algorithms were significant and ILP could not find the optimal solution for large input requests within shorter time. To address the problem, three heuristic algorithms with sub-optimal solutions, i.e., MTF, MSF as well as MDVF, were proposed. Among the three algorithms, MTF outperformed MSF, while MDVF was better than MTF. In addition, SA was used to find a better sub-optimal solution. The reconfiguration in future OODCNs provides new challenges on resource allocation. For the KSTA problem, our ILP model offered the accurate mathematical abstraction for such problem. In addition, our proposed algorithms offered promising solutions to the problem.

References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM Conf.*, 2008, vol. 38, no. 4, pp. 742–758.
[2] J-Y. Shin, B. Wong, and E. G. Sirer, "Small-world datacenters," in *Proc. 2nd ACM Symp. Cloud Comput. Conf.*, 2011, pp. 2:1–2:13.
[3] A. Singla, C. Hong, L. Popa, and P. Godfrey, "Jellyfish: Networking data centers randomly," in *Proc. Symp. Netw. Syst. Des. Implementation Conf.*, 2012, pp. 225–238.
[4] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: A scalable and fault-tolerant network structure for data centers," in *Proc. ACM SIGCOMM Conf.*, 2008, vol. 38, no. 4, pp. 75–86.
[5] C. Guo *et al.*, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. ACM SIGCOMM Conf.*, 2009, vol. 39, no. 4, pp. 63–74.
[6] G. Wang *et al.*, "c-Through: Part-time optics in data centers," in *Proc. ACM SIGCOMM Conf.*, 2010, vol. 41, no. 4, pp. 327–338.
[7] N. Farrington *et al.*, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proc. ACM SIGCOMM Conf.*, 2010, vol. 41, no. 4, pp. 339–350.
[8] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: A topology malleable data center network," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, 2010, pp. 1–6.
[9] G. Porter *et al.*, "Integrating microsecond circuit switching into the data center," in *Proc. ACM SIGCOMM Conf.*, 2013, vol. 43, no. 4, pp. 447–458.
[10] K. Chen *et al.*, "OSA: An optical switching architecture for data center networks with unprecedented flexibility," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 498–511, Apr. 2014.

[11] K. Chen *et al.*, "WaveCube: A scalable, fault-tolerant, high-performance optical data center architecture," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1903–1911.
[12] K. Ueda, Y. Mori, H. Hasegawa, and K. Sato, "Large-scale optical switch utilizing multistage cyclic arrayed-waveguide gratings for intra-datacenter interconnection," *IEEE Photon. J.*, vol. 9, no. 1, Feb. 2017, Art. no. 7800412.
[13] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Routing and spectrum allocation in OFDM-based optical networks with elastic bandwidth allocation," in *Proc. IEEE Globecom*, 2010, pp. 1–6.
[14] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," in *Proc. IEEE Conf. Comput. Commun.*, 2010, pp. 1503–1511.
[15] C. Kachris and I. Tomkos, "Energy-efficient bandwidth allocation in optical OFDM-based data center networks," in *Proc. Opt. Fiber Commun. Conf.*, 2012, Paper JTh2A-34.
[16] Y. Li, S. Dai, and J. You, "Revenue-oriented bandwidth allocation in optical OFDM intra data center networks," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.
[17] Y. Ji, X. Wang, S. Zhang, R. Gu, T, Guo, and Z. Ge, "Dual-layer efficiency enhancement for future passive optical network," *Sci. China Inf. Sci.*, vol. 59, no. 2, 2016, Art. no. 022313.
[18] Y. Ji, J. Zhang, Y. Zhao, X. Yu, J, Zhang, and X. Chen, "Prospects and research issues in multi-dimensional all optical networks," *Sci. China Inf. Sci.*, vol. 59, no. 10, 2016, Art. no. 101301.
[19] Y. Ji *et al.*, "All optical switching networks with energy-efficient technologies from components level to network level," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 8, pp. 1600–1614, Aug. 2014.
[20] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *Proc. ACM SIGCOMM Conf.*, 2011, vol. 41, no. 4, pp. 242–253.
[21] A Greenberg *et al.*, "VL2: A scalable and flexible data center network," in *Proc. ACM SIGCOMM Conf.*, 2009, vol. 54, no. 3, pp. 95–104.
[22] J. Lee *et al.*, "Application-driven bandwidth guarantees in datacenters," in *Proc. ACM SIGCOMM Conf.*, 2014, vol. 44, no. 4, pp. 467–478.
[23] Y. Benlachtar *et al.*, "Optical OFDM for the data center," in *Proc. Int. Conf. Transp. Opt. Netw.*, 2010, pp. 1-4.
[24] N. Hamedazimi *et al.*, "FireFly: A reconfigurable wireless data center fabric using free-space optics," in *Proc. ACM SIGCOMM Conf.*, 2014, vol. 44, no. 4, pp. 319–330.
[25] M. Ghobadi *et al.*, "ProjecToR: Agile reconfigurable data center interconnect," in *Proc. ACM SIGCOMM Conf.*, 2016, vol. 46, no. 4, pp. 216–229.
[26] L. Chen *et al.*, "Enabling wide-spread communications on optical fabric with megaswitch," in *Proc. Symp. Netw. Syst. Des. Implementation Conf.*, 2017, pp. 577–593.
[27] Y. Zhao *et al.*, "Dynamic topology management in optical data center networks," *J. Lightw. Technol.*, vol. 33, no. 19, pp. 4050–4062, Oct. 2015.
[28] H. Chen *et al.*, "Time-spectrum consecutiveness based scheduling with advance reservation in elastic optical networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 70–73, Jan. 2015.
[29] B. H. Ramaprasad, T. Schondienst, and V. M. Vokkarane, "Dynamic continuous and non-continuous advance reservation in SLICE networks," in *IEEE Proc. Int. Conf. Commun.*, 2014, pp. 3319–3324.
[30] M. Markowski, "Utilization balancing algorithms for dynamic multicast scheduling problem in EON," *Int. J. Electron. Telecommun.*, vol. 62, no. 4, pp. 363–370, 2016.
[31] G. M. Saridis, D. Alexandropoulos, G. Zervas, and D. Simeonidou, "Survey and evaluation of space division multiplexing: From technologies to optical networks," *IEEE Commun. Surveys Tut.*, vol. 17, no. 4, pp. 2136–2156, Oct.–Dec. 2015.
[32] H. Tode and Y. Hirota, "Routing, spectrum, and core and/or mode assignment on space-division multiplexing optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. A99–A113, Jan. 2017.
[33] S. Sugihara, Y. Hirota, S. Fujii, H. Tode, and T. Watanabe, "Dynamic resource allocation for immediate and advance reservation in space-division-multiplexing-based elastic optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 3, pp. 183–197, Mar. 2017.
[34] A. Muhammad, G. Zervas, D. Simeonidou, and R. Forchheimer, "Routing, spectrum and core allocation in flexgrid SDM networks with multi-core fibers," in *Proc. Opt. Netw. Des. Model.*, 2014, pp. 192–197.
[35] 2018. [Online]. Available: https://en.wikipedia.org/wiki/Bin_packing_problem
[36] W. Lu, Z. Zhu, and B. Mukherjee, "Optimizing deadline-driven bulk-data transfer to revitalize spectrum fragments in EONs," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 7, no. 12, pp. B173–B183, Dec. 2015.
[37] L. Peng, M. Chen, K. Park, and C. H. Youn, "Virtual-pod-assisted routing and resource assignment in elastic all-optical intra-datacenter networks," *IEEE Access*, vol. 5, pp. 406–420, 2017.
[38] Y. Yin, L. Liu, R. Proietti, and S. J. B. Yoo, "Software defined elastic optical networks for cloud computing," *IEEE Netw.*, vol. 31, no. 1, pp. 4–10, Jan./Feb. 2017.
[39] P. Lu and Z. Zhu, "Data-oriented task scheduling in fixed-and flexible-grid multilayer inter-DC optical networks: A comparison study," *J. Lightw. Technol.*, vol. 35, no. 24, pp. 5335–5346, Dec. 2017.
[40] S. Kandula, J. Padhye, and P. Bahl, "Flyways to de-congest data center networks," in *Proc. ACM Workshop Hot Topics Netw.*, 2009, pp. 1–6.
[41] G. Zhang, M. D. Leenheer, and B. Mukherjee, "Optical traffic grooming in OFDM-based elastic optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 4, no. 11, pp. B17–B25, Nov. 2012.
[42] G. Zhang, M. D. Leenheer, and B. Mukherjee, "Optical grooming in OFDM-based elastic optical networks," in *Proc. Opt. Fiber Commun. Conf.*, 2012, Paper OTh1A.