IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 November 2022, accepted 30 November 2022, date of publication 7 December 2022,
date of current version 13 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227438

RESEARCH ARTICLE

Dynamic Routing and Spectrum Assignment Based on the Consecutive Sub-Channels in Flexible-Grid Optical Networks

VARSHA LOHANI, (Member, IEEE), ANJALI SHARMA, (Member, IEEE),
AND YATINDRA NATH SINGH, (Senior Member, IEEE)
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India

Corresponding author: Varsha Lohani (lohani.varsha7@gmail.com)

This work was supported in part by the Indian Institute of Technology Kanpur and MHRD.

ABSTRACT Variable bandwidth channels can be created in Flexible Grid Optical Networks using Optical Orthogonal Frequency Division Multiplexing (O-OFDM). This allows more efficient spectrum use by allocating integral multiple of basic bandwidth slots (sub-channels) to the lightpath requests. In these networks, the constraint of keeping all the allocated slots together is added when deciding routes for the requests. This makes the Routing and Spectrum Assignment (RSA) algorithms in Flexible Grid Optical Networks more challenging. In any network, the lightpath requests will arrive and depart dynamically, leading to spectrum fragmentation. It leads to a reduction of maximum possible utilization as well as an increase in blocking probability. In this paper, we have presented an improvised RSA algorithm using consecutive spectrum slots as an adaptive parameter, that leads to reduced fragmentation. It is evident from the results that the presented RSA algorithm using adaptive parameters reduces the blocking probability and fragmentation more effectively than the shortest path and k-shortest path algorithms reported in the earlier works.

INDEX TERMS Flexible-grid optical networks, routing and spectrum assignment, dynamic algorithms, fragmentation, blocking probability, spectrum utilization.

I. INTRODUCTION

With the innovations continuously improving the performance of network endpoint devices, communication network capacities need to be continuously improved to meet the demands. All-Optical networks are used to provide the much needed increased capacity. The signal that traverses from a source to a destination node in these networks, remains in the optical domain [1]. These networks contain routing nodes interconnected by optical fiber links. The resources used in these links can be either fixed-width wavelength slots or flexible spectrum slots. Fixed width wavelength slots are generally based on the Dense Wavelength Division Multiplexing technique. The bandwidth of each channel within the network link, is fixed as either 50 GHz or 100 GHz according to ITU-T G.694.1 [2] specifications. High data-rate connection cannot be established due to limit posed by the slot width. At the same time, a very low data-rate lightpath requests will lead to bandwidth wastage. Jinno et al. [3] proposed the spectrum sliced elastic optical networks to resolve this problems.

Use of Optical-Orthogonal Frequency Division Multiplexing (O-OFDM), allows the further reduction of the channel size to 12.5 GHz thereby improving the granularity of bandwidth allocation with flexibility. The multiple adjacent sub-channels can be used together to accommodate high data-rate connection demands. Such networks are called Flexible-grid Optical Networks. They provide a scalable network architecture using Bandwidth Variable-Wavelength Cross-connects (BV-WXC) and Bandwidth Variable Transponders (BVT). An overview of Flexible-grid Optical Network and the related design issues can be found in [4] and [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chatterjee.

One of the research problems in Flexible-grid Optical Networks is Routing and Spectrum Assignment (RSA). The routing of a lightpath request implies finding an end-to-end path between source and destination nodes using a suitable algorithm. Spectrum assignment implies finding out the link resources for the path setup request while following the relevant constraints. This paper aims to improve the RSA algorithms for flexible grid optical networks to reduce the blocking probability, to cater for more connection requests with reduced computational complexity. However, due to spectrum elasticity, the Flexible-grid also adds contiguity constraints to the RSA problem. This additional constraint and the fluctuating traffic results in the fragmentation of the spectrum. As a consequence, there is an increase in the blocking probability of lightpath requests.

The paper is organised as follows. The basic concept of Routing and Spectrum Allocation is explained through examples in Section II. In Section III, the problem being investigated in this paper is stated. The proposed solution is mentioned in Section IV. In Section IV, network definition, tackling unequal capacity on edges, and different proposed RSA algorithms have also been presented. Numerical results are shown for two example networks in Section V, where different network settings are used to understand the performance of the proposed algorithms. Finally, we conclude the work in Section VI.

II. ROUTING AND SPECTRUM ASSIGNMENT (RSA)

The objective of RSA is to find routes [1] over the network [6] and allocate spectrum resources to the lightpath requests while using minimum resources and accommodating maximum lightpath requests.

In RSA, there are constraints for spectrum assignment, which need to be satisfied as explained with the help of Figure 1. A lightpath request arrives for three slots connection assignment from source node *A* to destination node *C*. **Spectrum Contiguity Constraint:** The assigned spectrum slots indexes 2, 3, 4 or 6, 7, 8 for link A-B and 3, 4, 5 or 4, 5, 6 or 5, 6, 7 or 6, 7, 8 can be used for link B-C as the possible group of three slots sharing boundary with each other. This is contiguity constraint. **Spectrum Continuity Constraint:** The assigned continuous spectrum slots indexes 6, 7, 8 for a lightpath request A-B-C is same throughout the path. **Non-Overlapping Constraint:** The assigned indexes to different requests cannot overlap with one other. It is consequence of capability of a slot to carry one signal at a time. Therefore, the slot indexes 6, 7, and 8 are assigned as it satisfies the requirements.

Finding a suitable RSA algorithm lies at the core of the Flexible-grid optical networks design. It is supposed to achieve efficient spectrum utilization while accommodating maximum number of lightpath requests. There exist various RSA algorithms in the literature [7], [8], [9], [10], [11], [12], which envisage to reduce the blocking of arriving lightpath requests. In these papers, various heuristics for RSA have been proposed to reduce the blocking probability of lightpath requests. The problem has been attempted using many methods like employing different routing schemes, devising spectrum assignment algorithms based on the cost of paths, and distance-adaptive modulation level manipulation. In [13], the modulation format as another dimension for manipulating the bandwidth in different sections of a lightpath, is introduced in the Elastic optical network. The authors in [14] proposed heuristics for Dynamic routing and spectrum (re)allocation, where the allocated lightpath requests are re-allocated the optical spectrum to make room for new lightpath requests. Few research works [10] and [15] also considered multipath routing for RSA in EON to accommodate more lightpath requests as well as to reduce the blocking probability. Fixed parameters e.g., distance (in km), hops, etc., are used for deciding the optimal route in most of these works. In the current work, we are considering adaptive parameters (which change with network conditions) for RSA. The spectrum resources are allotted at the connection setup time and released only when the connection is dismantled.

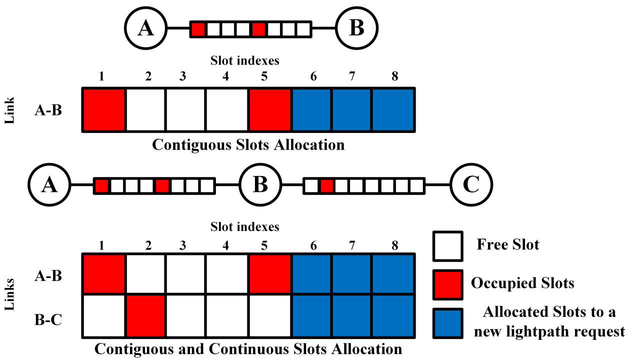


FIGURE 1. Example of constraints used for routing and spectrum allocation of lightpath request.

One of the problems due to the spectrum assignment constraints is Spectrum Fragmentation. It can be explained with the help of an example shown in Figure 2. A lightpath request for four slots arrives from the source node *A* to the destination node *C*. Although the four slots are available but they cannot be allocated, as we have to follow spectrum assignment constraints (Figure 1). It leads to the blocking of the lightpath request due to fragmentation. If spectrum converters are present at each node, the fragmentation can be mitigated to some extent, but converters are expensive to deploy.

Chatterjee et al. [16] have extensively studied the various forms of fragmentation. The survey paper presents various types of fragmentation metrics and de-fragmentation strategies. They also discussed various RSA strategies based on the fragmentation metrics without any de-fragmentation procedures. Also, numerous de-fragmentation algorithms have been proposed to reduce the spectrum fragmentation periodically. In [17], the authors have presented a path-based

---

[1] The terms route and path are used interchangeably.

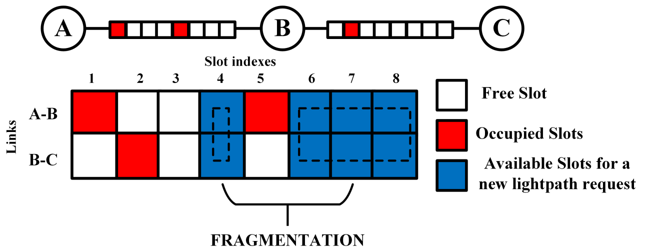


FIGURE 2. **Spectrum fragmentation example.**

method to calculate fragmentation level and then to employ it in RSA decision making. [18], [19] further explored fragmentation-aware routing and spectrum allocation, considering both contiguity and continuity aspects. Basically, various parameters of the spectrum in the network links are observed, and based on their state; the routing decisions are made. These parameters may or may not directly affect the fragmentation level. The contiguity of the spectrum slices is one of the most prominent indicators of fragmentation level. Therefore, it is also possible to use suitable RSA algorithms to minimize contiguity fragmentation during the network operation. De-fragmentation procedures, whenever invoked, disrupts the existing traffic in the network for the period required for reconfiguration, which leads to another inefficiency in network performance.

III. PROBLEM STATEMENT

The RSA algorithms can be further classified into *offline RSA* where the lightpath requests are known in advance, and *online RSA* in which the lightpath requests arrive and get released as time progresses. The later scenario is more realistic. In this paper, we are considering online RSA without any periodic de-fragmentation. The RSA problem is further sub-classified based on whether the static or adaptive parameters are considered for routing. The static parameters for routing are independent of the changes within the network; while the adaptive parameters for routing change continuously with the changes in the network conditions.

Many research works discussed in Section II, consider static parameters such as fixed distance and hops[2] for routing of lightpath requests from source to destination. Some of them also consider adaptive parameters for Routing and Spectrum Assignment. References [20], [21], [22] use relative cost parameter (e.g. link congestion) which changes during the network operation while performing Routing and Spectrum Assignment for incoming lightpath requests. For various cases, the algorithms have been analyzed. In essence, the algorithms select the route/spectrum set with the least relative cost. Y. Zhou *et. al* [23] computes the link-state based on Chromatic Dispersion and Optical Signal to Noise Ratio (OSNR). This link-state is used for routing purposes. Another adaptive parameter that has been discussed in the literature is crosstalk [24], [25], [26]. The crosstalk occurs within the multi-core fibres and has a bearing on Routing, Core and Spectrum Assignment in Space Division Multiplexed Elastic Optical Networks. In [27], authors studied the problem of spectrum de-fragmentation in crosstalk aware RCSA.

TABLE 1. **Notations used.**

G	Graph
V	Set of vertices in G
E	Set of edges in G
B_e	Usable bandwidth on each edge $e \in E$
$\{\Delta_e\}$	Bitmap (sequence of 1s and 0s) on an edge $e \in E$ to model the availability status of the spectrum slots
F	Set of spectrum slots
f_s	$f_s \in \{\Delta_e\}$ is either 0 or 1 depending on the condition whether the s^{th} frequency slot is busy or free
LR	Lightpath Request
s	Source Node
d	Destination Node
B^r	Required Bandwidth for an $s-d$ pair
$\lvert\Delta^r\rvert$	Required contiguous spectrum slots for an $s-d$ pair
k	The maximum number of paths to be computed by RSA
m	The number of bits per symbol
Δ_p	The bitmap of available spectrum slots in path p from s to d.
$(u, v) \in E$	The edge joining the pair of vertices, where $u \in V$ is the starting (head) node and $v \in V$ is the ending (tail) node

IV. PROPOSED SOLUTION

One of the adaptive parameters that can be effective for RSA is consecutive spectrum slots on each network link. One can use this parameter for routing purposes in conjunction with the works mentioned earlier (Section III). In the present work, our objective is to find suitable routes and spectrum slots for incoming lightpath requests using an adaptive parameter called **Link Spectrum Consecutiveness** in dynamic traffic scenario. At the same time, we are minimizing the fragmentation within the network without performing any de-fragmentation strategies so that we can accommodate the maximum number of requests. We are not using any de-fragmentation strategy as it causes interruption of active lightpath requests.

A. NETWORK MODEL AND NOTATIONS USED

In table 1, various notations used in the paper is summarized. The detailed description is given here.

---

[2] Independent of the network conditions.

- $G(V, E, \{\Delta_e\})$: We represent an optical network as a graph $G(V, E)$ where G is defined as a set of optical vertices (nodes) V, indexed by v and set of optical fiber edges (links) E, indexed by e. Each edge is connected to a pair of vertices e.g., $(i, j) \in E$, where i and $j \in V$. Each edge $e \in E$ has usable bandwidth, B_e. The B_e is partitioned into multiple spectrum slots in order to be used efficiently. We define a bitmap, Δ_e (sequence of 1s and 0s) on an edge $e \in E$ to model the availability status of the spectrum slots. The number of the possible spectrum slots on an edge e, $|\Delta_e|$ is represented by

$$|\Delta_e| = \left\lfloor \frac{\text{Total Usable Bandwidth}(B_e)}{\text{Grid Size}} \right\rfloor. \tag{1}$$

Suppose each edge has same set (F) of spectrum slots (f_s), then the bitmap for an edge can be represented as $\Delta_e = [f_1, \ldots ., f_{|F|}]$ where f_s can be either 0 or 1 depending on the condition whether the s^{th} frequency slot is busy or free respectively. The bit s in Δ_e is represented as

$$\Delta_e[s] = \begin{cases} 1, & \text{if } s^{th} \text{ slot on edge } e \text{ is free,} \\ 0, & \text{if } s^{th} \text{ slot on edge } e \text{ is occupied.} \end{cases} \tag{2}$$

The slot availability status can be easily propagated for a destination through the neighbours by simply performing AND operation of the current status with the slot availability status in the links to neighbour.

- $LR(s, d, |\Delta^r|, k)$ is a Lightpath Request where s is the source node, $s \in V$, and d is the destination node, $d \in V$. In a network $G(V, E)$, if a lightpath request arrives with a required bandwidth of B^r for an s-d pair, then the number of the required contiguous spectrum slots on any choosen path is computed with the eq. 3.

$$|\Delta^r| = \left\lceil \frac{\text{Required Bandwidth}(B^r)}{\text{Grid Size} * m} \right\rceil + \left\lceil \frac{\text{GB}}{\text{Grid Size}} \right\rceil. \tag{3}$$

Here m is the number of bits per symbol (modulation level) used depending on the path length of the lightpath. We are also considering an additional guard band (GB) such that no two lightpaths interfere if they are placed next to each other. The maximum number of paths to be computed by RSA is represented by k, a positive integer.

- Δ_p is the bitmap of available spectrum slots in path p from s to d. The availability bitmap for a path p, Δ_p can be achieved by intersecting or Bit-wise ANDing of bit maps on all the constituent edges of path p

$$\Delta_p = \{\Delta_p[l]\} = \{\bigcap_{i=1}^{L} \Delta_{e_i}[l]\}. \tag{4}$$

- $(u, v) \in E$ is the edge joining the pair of vertices (nodes), where $u \in V$ is the starting (head) node and $v \in V$ is the ending (tail) node.

B. CASES OF EQUAL AND UNEQUAL CAPACITIES ON THE NETWORK LINKS

Suppose the fibers are deployed between different pair of nodes in a network at different time instants; consequently the bandwidth capacity of some of the fiber links/edges may be different, subject to technological advancements at the time of deployment. Let the bitmap of edge 1 be Δ_{e_1} and edge 2 be Δ_{e_2} and suppose $|\Delta_{e_1}| > |\Delta_{e_2}|$. This is the case of non-uniform bandwidth, hence, the bitmap sequence, on the edges of the network. The $(|\Delta_{e_1}| - |\Delta_{e_2}|)$ slots are zero-padded in Δ_{e_2} to make uniform size of bitmap sequences for all the links in the network for the proposed algorithms. However, the zero-padded bit sequences remain unavailable for use by lightpath requests, independently of the network conditions. Now, $\Delta_e[s] = 0$ in eq. 2, when s^{th} slot on edge e is occupied or unavailable.

Eq. 4 holds for both the cases of uniform and non-uniform bandwidth provisioning on the edges in the network. Here, L is the number of hops for a path p.

1) Consider a uniform network with three link lightpath (p) A-B-C-D as shown in Figure 3a. Each link on a path has 8 slots. The slot values are represented with 1s and 0s based on the availability status. The value of Δ_{e_1} for edge 1 is 00111001, Δ_{e_2} for edge 2 is 11111001, and Δ_{e_3} for edge 3 is 10011001. Therefore,

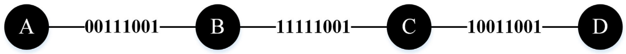
$$\Delta_p = \{\bigcap_{i=1}^{3} \Delta_{e_i}\} = 00011001.$$

2) Now consider the case when the bandwidths on the links are different as shown in Figure 3b. The values of Δ_{e_1} for edge 1 is 00111, Δ_{e_2} for edge 2 is 111110, and Δ_{e_3} for edge 3 is 100110011. The number of available slots on edge 1, $|\Delta_{e_1}|$, is 5 on edge 2, $|\Delta_{e_2}|$, is 6, and on edge 3, $|\Delta_{e_3}|$, is 9 i.e., $|\Delta_{e_1}| < |\Delta_{e_2}| < |\Delta_{e_3}|$. The number of slots on edge 3 is the highest so there is no zero-padding for it; however, zero-padding is needed for edge 1 and edge 2. After zero-padding, the number of slots on $|\Delta_{e_2}|$ changes to $|\Delta_{e_2}| + (|\Delta_{e_3}| - |\Delta_{e_2}|)$, and $|\Delta_{e_1}|$ changes to $|\Delta_{e_1}| + (|\Delta_{e_3}| - |\Delta_{e_1}|)$. The difference part is zero-padded to the edge 1 and edge 2 slots. Therefore, now the Δ_{e_1} for edge 1 is 001110000, and Δ_{e_2} for edge 2 is 111110000, and hence

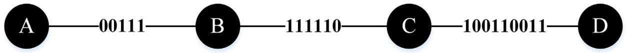
$$\Delta_p = \{\bigcap_{i=1}^{3} \Delta_{e_i}\} = 000110000.$$

C. PROPOSED ALGORITHMS

In this paper, we proposed three algorithms for the joint optimization of Routing and Spectrum Assignment, which use the availability of consecutive spectrum slots for routing. If a lightpath with required slots is available, the spectrum slots are allocated to the request and lightpath is established; otherwise, the request will be blocked.



(a) Uniform spectrum slots on links, and



(b) Non-uniform spectrum slots on links.

FIGURE 3. **Examples show a lightpath request with three links and their spectrum status.**

1) TYPE I: ROUTING AND SPECTRUM ASSIGNMENT BASED ON AVAILABLE CONSECUTIVE SLOTS (RSACS)

The Type I RSACS algorithm selects a path from the set of available paths. In this algorithm, whenever a new lightpath request $LR(s, d, |\Delta^r|)$ arrives, the algorithm traverses from the source node to its neighbouring nodes and so on, based on the available spectrum slots. If even one spectrum slot is not available through a link, the search through that link is terminated. The search continues through the other links to find all possible paths with at least one contiguous slot. The process continues until we reach the destination node. At the end of the algorithm, we will find all possible paths from source node to the destination node along with the number of available contiguous slots. The best path among all the found paths having the required number of contiguous slots, is allocated to the lightpath request based on the first-fit spectrum assignment as specified in the Algorithm 1. The set of k (where k is a positive integer) possible paths is computed using CandidatePaths() function (Algorithm 2). In this function, the search for the candidate paths starts from the source node s. Then in each iteration, based on the number of hops, the nodes are added. In other words, with the increase in the number of iteration, the number of hops for each path in the list keeps on increasing.

The search is terminated when the destination node is found for at most k paths or no more paths are possible. The paths which have slots greater or equal to the required slots are selected. The paths we found using the CandidatePaths() function, are sorted based on the number of hops. After finding candidate paths as a list of paths with available slots greater than or equal to one, allocation to requests is done. The first path may or may not have slots greater than or equal to the required number of slots. If the first path has the required number of slots, that path will be the best. Otherwise, we have to move on to the next path in the list. Whichever path fulfils the condition, we select that path for Routing and Spectrum Assignment. Based on the first fit, the slots are chosen from the set of slots present on the selected path.

In the Algorithm 2, *tmp* is the temporary matrix. It contains the list of those paths for which search have not arrived at the destination node. Once the destination node is found, the path will go to the *AllPath* list. Also, when the size of *AllPath* reaches k (any integer), we will purge out the paths in the *tmp*.

Although for single-path routing, this algorithm has higher time complexity, but it is still useful for multipath routing [10], [15] as the source node has the details of all the paths with the spectrum slots available in them. If there are no paths with required available slots, the source node can select multiple paths whose combined available slots are greater than or equal to the required slots. Though, for the feasibility of this scenario, capability to split the transmitted signal into multiple streams should exist at the source node. We can reduce the blocking probability of the lightpath requests with this method.

Algorithm 1 Type I: RSACS

```
Input: $G(V, E, \{\Delta_e\})$, $LR(s, d, |\Delta^r|, k)$
$AllPath \leftarrow$ list of $k$ candidate paths alongwith available slots from s to d using CandidatePaths($G(V, E, \{\Delta_e\})$, $s$, $d$, $k$)   %Algorithm 2
if $AllPath$ is not empty then
    for all the paths in $AllPath$ indexed by $i$ do
        if |MaxContg($\Delta_i$)| >= $|\Delta^r|$ then   %MaxContg returns the number of maximum contiguous slots available in bitmap.
            Return $BestPath = AllPath(i)$   %First one in the $AllPath$ which satisfy the constraint is picked. There can be other ways of choosing if more than one options are there.
        else
            $i++$
        end if
    end for
    if $BestPath$ is empty then
        Block the request
    end if
else
    Block the request
end if
```

2) TYPE II: ROUTING AND SPECTRUM ASSIGNMENT BASED ON REQUIRED CONSECUTIVE SLOTS

Algorithm 3 is second RSA algorithm. It is called Type II: Routing and Spectrum Assignment based on Consecutive Slots. It is another way of routing using consecutive spectrum slots. Each node checks the contiguous required slots with the slots present after Bit-wise ANDing of the bitmap of path from source node and the bitmap of the link from the current node to the neighbouring node. If the current node is the destination node, that path will be considered for RSA. But, if the required slots are not available from source to the current node, then no path is feasible via current node. In this method, computational time is reduced as we need to compute the candidate path only with minimum required contiguous slots.

The role of function CandidatePaths() (Algorithm 4) changes slightly and it finds best candidate paths with at least $|\Delta^r|$ contiguous slots. In this function, the search for the candidate path originates from the source node s. Then

Algorithm 2 CandidatePaths()

```
function CandidatePaths($G(V, E, \{\Delta_e\}), s, d, k$) %First $k$ paths during the search returned by the function.
    $AllPath = [\ ]$ %Contains the list of all the paths possible with slots available in them, at the end of function.
    $Path = [s]$ %A path is an ordered list of nodes with neighbouring nodes having a link between them. Path is initialized with one entry for source $s$.
    while $Path$ is not empty do
        $tmp = [\ ]$
        for all paths in $Path$ indexed by $i$ do
            $u = Path(i).end$ % $u$ is the end node as of now for the $i^{th}$ path entry
            $\{v\} = \mathrm{adj}(u))\backslash Path(i)$ %adj function returns set of all neighbours of $u$ excluding the ones already on $Path(i)$, \ sign indicate exclusion. All nodes in the $i^{th}$ path are excluded
            if $\{v\}$ is not empty then
                for all nodes in $\{v\}$ indexed by $j$ do
                    $\Delta_i \leftarrow \Delta_i \cap \Delta_{(u,v(j))}$
                    if sum($\Delta_i$) $\neq 0$ then % sum function gives the number of available slots in $Path$, $i$ with bitmap $\Delta_i$. $Path(i)$ should not be considered if sum is zero.
                        if $v(j) == d$ then
                            $AllPath = AllPath + [(Path(i), v(j)); \Delta_i]$ %add the $[(Path(i), v(j)); \Delta_i]$, as another path from $s$ to $d$ with $\Delta_i$ available slots.
                            if $size(AllPath) == k$ then
                                Return $AllPath$
                            end if
                        else
                            $tmp = tmp + [(Path(i), v(j)); \Delta_i]$ %add the path $[(Path(i), v(j)); \Delta_i]$ to $tmp$ path storage.
                        end if
                    end if
                end for
            end if
        end for
        $Path = tmp$ %All new paths learnt are stored in $Path$. All older values in $Path$ discarded.
    end while
    Return: $AllPath$
end function
```

Algorithm 3 Type II: RSACS

```
Input: $G(V, E, \{\Delta_e\}), LR(s, d, |\Delta^r|, 1)$
$AllPath \leftarrow$ A candidate path ($k = 1$) from $s$ to $d$ using CandidatePaths($G(V, E, \{\Delta_e\}), LR(s, d, |\Delta^r|, 1)$) %Algorithm 4
if $AllPath$ is not empty then
    $BestPath = AllPath(1)$
    Return $BestPath$
else
    Block the request;
end if
```

for each iteration, the number of hops for each path in the list keeps on increasing.

IsFeasible() function (Algorithm 5) checks the required number of slots $|\Delta^r|$ at each node, if available, then passes the information to the adjacent nodes for route discovery further. Else, the path with slots $\leq |\Delta^r|$ is discarded from the list. The search is stopped when the destination node is found for one of the paths ($k = 1$). Then, Algorithm 3 checks the *AllPath*. If it is not empty, then that path is used for RSA. Whereas, if the path is not found, then that lightpath request gets blocked.

This way of routing is best suited for single path routing i.e. $k = 1$.

3) TYPE III: ROUTING AND SPECTRUM ASSIGNMENT BASED ON REQUIRED CONSECUTIVE SLOTS AND SHORTEST PATH

The Algorithm 6 is Type III Routing and Spectrum Assignment based on Consecutive Slots with additional Shortest Path Constraint. The method for computing path is same as used for Type II Algorithm except now instead of single path, multiple candidate paths are maintained from the source node s to destination d using CandidatePaths function (Algorithm 4). In this function, the search for the candidate path originates from the source node s. Then for each iteration, the number of hops for each path in the list keeps on incrementing. Also, the paths with maximum contiguous slots $\leq |\Delta^r|$ are discarded during the search. The search is stopped when the destination node is found for at most k paths. Based on the distance, the shortest path from *AllPath* is chosen for Routing and Spectrum Assignment. If the path is not found, then the lightpath request gets blocked.

This method is the reverse process of finding k-shortest path first and then finding one with the required contiguous slots. Here we find path with required slots, and then choose the shortest one among them. This is also used for single path routing if the value of k is 1.

V. NUMERICAL RESULTS

The k-shortest path (k-sp) algorithm finds the route on the basis of the path containing least number of hops or shortest distance (km or miles). After that it checks whether the required slots are available or not on the found paths. This method might require extra computation. Therefore, we proposed algorithms in which the routes are found on the basis of contiguous and continuous spectrum slots availability. The performance of the proposed algorithms is compared with the k-shortest path algorithm and shortest path algorithm.

A. NETWORK SETTINGS

To evaluate the efficacy of our proposed algorithms, we operated a set of simulation experiments using MATLAB R2019b.

Algorithm 4 CandidatePaths() → Path With Contiguous Slots ≥ Δ^r

```
function CandidatePaths($G(V, E, \{\Delta_e\})$, $LR(s, d, |\Delta^r)|, k)$
    $AllPath = [\,]$
    $Path = [s]$
    while $Path$ is not empty do
        $tmp = [\,]$
        for all paths in $Path$ indexed by $i$ do
            $u = Path(i).end$
            $\{v\} = \mathrm{Adj}(u))\backslash Path(i)$
            if $\{v\} \neq [\,]$ then
                for all nodes in $\{v\}$ indexed by $j$ do
                    $\Delta_i \leftarrow \Delta_i \cap \Delta_{(u,v(j))}$
                    if IsFeasible($(\Delta_i, |\Delta^r|)$ == True then
                        if $v(j) == d$ then
                            $AllPath = AllPath + [Path(i), v(j)]$
                            if $size(AllPath) == k$ then
                                Return $AllPath$
                            end if
                        else
                            $tmp = tmp + [Path(i), v(j)]$
                        end if
                    end if
                end for
            end if
        end for
        $Path = tmp$
    end while
    Return: $AllPath$
end function
```

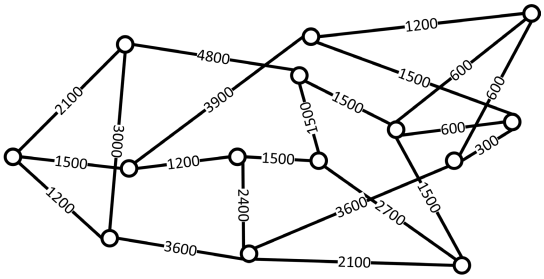
The simulations are done for 2,00,000 requests for multiple iterations. The performance of proposed algorithms - Type I: RSACS, Type II: RSACS and Type III: RSACSSP in Flexi-grid Optical Networks, are evaluated on the 14-nodes 22-links NSFNET with an average nodal degree (nodal degree, $n_d = \dfrac{2E}{V}$) 3.0, and 24 nodes 42 links USNET with an average nodal degree 3.5 as shown in Figure 4. We assume the fiber bandwidth in the frequency range of C-band to be 4 THz on each link of the network. Using O-OFDM technology, the whole bandwidth is divided into 12.5 GHz parallel channels. Therefore, there are 320 spectrum slots on each link of the network, calculated using eq. 1. The traffic demands for all the lightpath requests on each node pair are uniformly distributed. The bandwidth required for each lightpath is chosen randomly between 1 and B, where different B values are used as a parameter in simulations. In this paper, the values of B are 100 Gbps (8 spectrum slots, if the grid size is 12.5 GHz) and 200 Gbps (16 spectrum slots, if the grid size is 12.5 GHz). For spectrum allocation, we also considered an additional guard band (GB). The size of GB is considered to be 10 GHz.

Algorithm 5 IsFeasible()

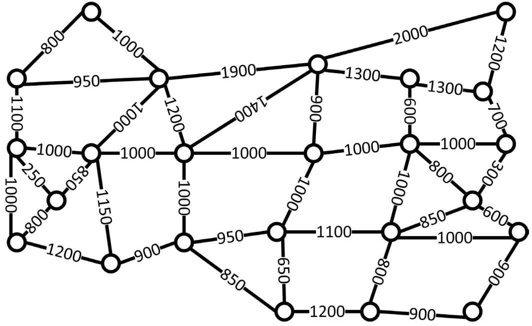
```
function IsFeasible($(\Delta_a, |\Delta_b|)$)    % IsFeasible function checks the $|\Delta_b|$ number of contiguous slots are available in the bitmap $\Delta_a$
    if $|\Delta_b|$ can be accommodated in $\Delta_a$ then
        Return True
    else
        Return False
    end if
end function
```

Algorithm 6 Type III: RSACSSP

```
Input: $G(V, E, \{\Delta_e\})$, $LR(s, d, |\Delta^r|, k)$
$AllPath$ ← list of candidate paths from s to d using CandidatePaths($G(V, E, \{\Delta_e\})$, $LR(s, d, |\Delta^r|, k)$) %Algorithm 4
if $AllPath$ is not empty then
    for all the candidate paths in $AllPath$ indexed by $i$ do
        Select the shortest path (distance in km) from all paths listed in $AllPath$ and store in $BestPath$
        Return $BestPath$
    end for
else
    The request is marked blocked.
end if
```



(a) 14 nodes, 22 links NSFNET.



(b) 24 nodes, 43 links USNET.

FIGURE 4. Networks topologies used.

The use of static traffic for simulation does not show the effectiveness of the proposed algorithms. Hence, lightpath requests were generated dynamically, i.e., we considered dynamic traffic scenario. The incoming lightpaths can be set up and released upon request. These are equivalent to setting up and releasing circuits in circuit-switched networks. The incoming lightpath requests arrive with an exponentially distributed inter-arrival time with the average of $\frac{1}{\lambda}$ seconds. Each connection is maintained for exponentially distributed holding time with average of $\frac{1}{\mu}$ seconds before being released. The offered load (ρ) in Erlang (E) is given by

$$\rho = \frac{\frac{1}{\mu}}{\frac{1}{\lambda}} = \frac{\lambda}{\mu}. \tag{5}$$

The values of offered load per node (ρ) between 2 to 30 Erlangs have been considered.

The pathfinding in the k-shortest path (k-sp) algorithm depends on the number of shortest paths (k) considered using distance first, and then the availability of the spectrum slots. On the other hand, the pathfinding in our proposed algorithms gives priority to the availability of the spectrum slots before the length of the path. The value of k chosen for pathfinding is taken as 10 for k-sp, Type I and Type III algorithms. As already discussed in pseudocodes, the shortest path and Type II algorithms are independent of the value k. The performance parameters are estimated based on the observations made during the steady-state condition (which is observed to happen after approximately three times the average holding time, i.e., $3 \times \frac{1}{\mu}$). All the plot results are within 95% confidence interval over multiple iterations.

B. PERFORMANCE METRICS

The performance of the proposed algorithms is evaluated on the basis of blocking probability of the new incoming requests, blocking probability of the required slots by those blocked requests, and the spectrum utilization with the gradual increments in demand rate (in Gbps):

- **Blocking Probability**: It is defined as the ratio of total number of blocked connections to the total number of arrived connections.
- **Bandwidth Blocking Probability**: It is defined as the ratio of the total amount of incoming bandwidth or slots which are blocked, to the total amount of bandwidth or slots required by all the incoming connections.
- **Spectrum Utilization**: The average ratio of total used bandwidth (or slots) to the total bandwidth (or slots) in the spectrum when observed for a time period.

C. WITHOUT DISTANCE-ADAPTIVE MODULATION TECHNIQUES

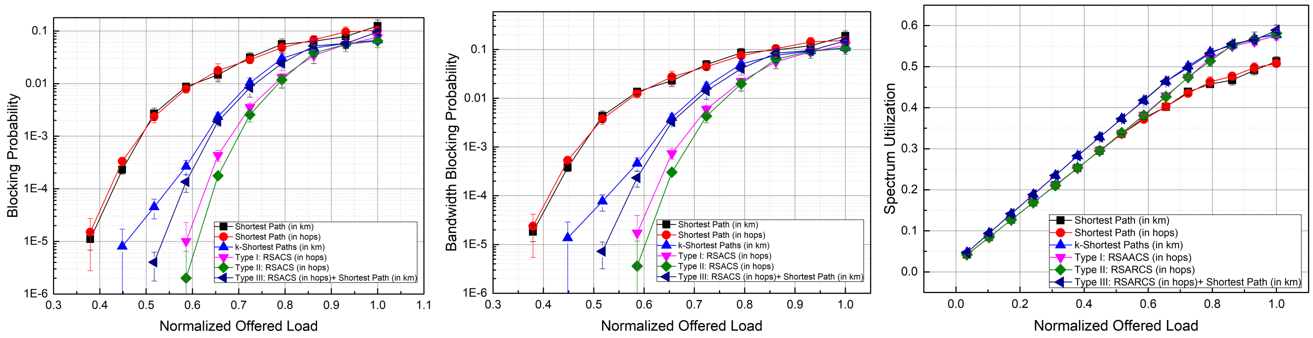
In this section, for figures 5 through 7, we evaluate the performance of the proposed algorithms, considering BPSK modulation format i.e., $m = 1$, for all source-destination pairs.

The Figures 5a and 5b are the plots of the blocking probability and bandwidth blocking probability of six pathfinding techniques with first-fit spectrum allocation policy. Intuitively, as the value of the offered load increases, (increasing the average arrival rate but keeping average holding time as constant) the blocking probability increases. The performance of consecutive slots-based pathfinding algorithms is better than the shortest path algorithms. Additionally, for low loads, Type II algorithm outperform other algorithms. Also, the Type III and k-shortest path algorithms are almost same after the normalized offered load of 0.6. But in Type III, the adaptive parameter is used first and then static parameter for routing. The performance of the Type III algorithm is better than the k-shortest path algorithm for lower loads. Intuitively, the shortest path algorithm finds a single route from source to destination based on the distance, not the spectrum slots available. Therefore, if the spectrum slots are not available at the time of spectrum assignment, then the connection request gets blocked. In shortest path case, other paths between source and destination might have required spectrum slots; but they are not used for the current connection request. Type-III algorithm takes all the possibilities in consideration and hence, performs better than the other algorithms in NSFNET.

The Type II algorithm finds the path based on required spectrum slots only. The path gets blocked at the routing time if no path is available with the required spectrum slots. In this, all the possible links with required spectrum slots are checked until the destination node is found. Therefore, the Type-II algorithm's time complexity for the worst case scenario is higher than the shortest path algorithm, as given in Table 2. Type II and shortest path algorithms are independent of the value of k. However, the performance of Type II is far better than the shortest path for all load conditions.

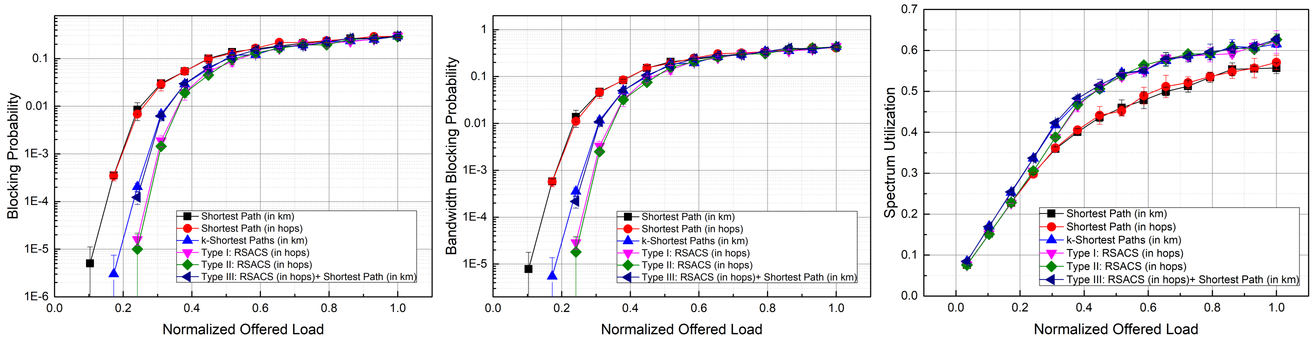
Figure 5c compares the performance in terms of spectrum utilization. The blocking probability of the connections for lower loads is almost negligible. Hence, the spectrum remains mostly underutilized. But as the offered load values increase, spectrum utilization also increases for all the algorithms and the shortest path algorithms for higher loads. Based on the observations, higher blocking of connections and lower spectrum utilization for the shortest path is due to the unavailability of the contiguous spectrum slots in a fragmented state. In contrast, for other cases where blocking probability is low, and spectrum utilization is high, we can say that the fragmentation is less impactful.

Figures 6a, 6b and 6c are also for NSFNET network with the value of k as 10 but the maximum incoming demand rate is 200 Gbps. For higher demand rates, the blocking probability keeps increasing as the availability of required spectrum slots becomes less than that for the case of 100 Gbps. The blocking performance comparison is similar to that for the demand rate of 100 Gbps. The spectrum performance is also similar, i.e. the spectrum utilization of the shortest path algorithms is worst as the offered load per node increases. Whereas, other four algorithms' spectrum utilization performance is almost same. We can notice the spectrum



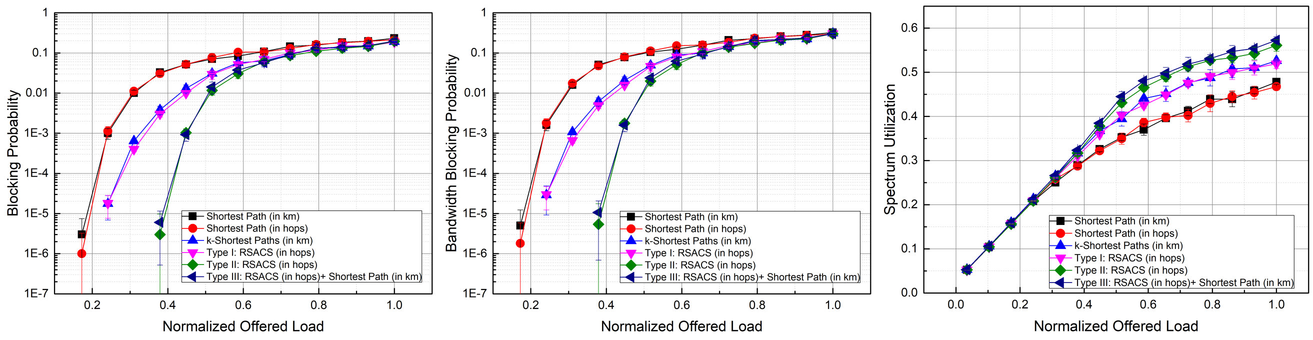
(a) (b) (c)

FIGURE 5. Various performance metric vs offered loads for NSFNET with a demand rate of 100 Gbps.



(a) (b) (c)

FIGURE 6. Various performance metric vs offered loads for NSFNET with a demand rate of 200 Gbps.



(a) (b) (c)

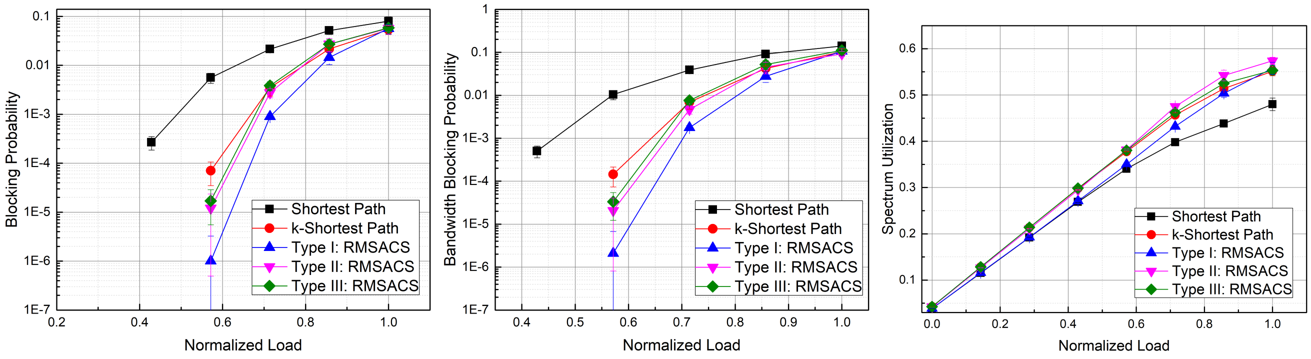
FIGURE 7. Various performance metric vs offered loads for NSFNET with a demand rate of 200 Gbps.

TABLE 2. Comparison of algorithms used for routing and spectrum assignment. Here, V is the set of vertices, E is the set of edges, Δ is the set of spectrum slots, and k is the maximum number of paths to be computed by RSA.

Parameters	Shortest Path	k-Shortest Path	Type I	Type II	Type III
Path finding	On the basis of distance or hops	On the basis of distance or hops	On the basis of available slots	On the basis of available slots	On the basis of available slots and distance
Routing and Spectrum Assignment Problem	Separate problem	Separate problem	Joint problem	Joint problem	Joint problem
Time Complexity	O(V.E)+O(E.Δ)	O(k.V.E)+O(E.Δ)	O(k.V.E.Δ)	O(V.E.Δ)	O(k.V.E.Δ)
Blocking of the request after path finding	If the required spectrum is not available, request can be blocked	If the required spectrum is not available, request can be blocked	If the required spectrum is not available, request can be blocked	No blocking	No blocking

utilization performance affected by fragmentation at offered load around 30%.

Alongside NSFNET, we used USNET network to determine the performance trends with topology change. Figures 7a, 7b and 7c are for USNET network where the value of k is 10 and the incoming demand rate is 100 Gbps. This observation is for large networks with 24 nodes and 43 links. The blocking performance of the lightpath requests



(a) (b) (c)

FIGURE 8. Various performance metric vs offered loads for NSFNET with a demand rate of 100 Gbps. Considering distance-adaptive modulation techniques.

and spectrum slots changes as compared to NSFNET. In USNET also the Type II and Type III algorithm outperforms the other strategies. In USNET again the shortest path algorithms perform worst in terms of blocking probability and utilization of the spectrum slots.

Now, as the network diameter changes, we can also observe the change in performance of the spectrum slots utilization for the other four algorithms. Type II and Type III have the higher spectrum utilization for the higher offered loads than Type I and k-shortest path algorithm, due to acceptance of more connection requests having higher incoming demand rates.

D. WITH DISTANCE-ADAPTIVE MODULATION TECHNIQUES

The performance evaluation of various proposed and baseline algorithms using distance adaption modulation techniques [28] is shown in figures 8a through 10c. Also, the proposed algorithm changes from Routing and Spectrum Assignment based on Consecutive Slots (RSACS) to Routing, Modulation, and Spectrum Assignment based on Consecutive Slots (RMSACS). Intuitively, RMSACS should perform better in all aspects as compared to RSACS and this is also observed in the simulation results.

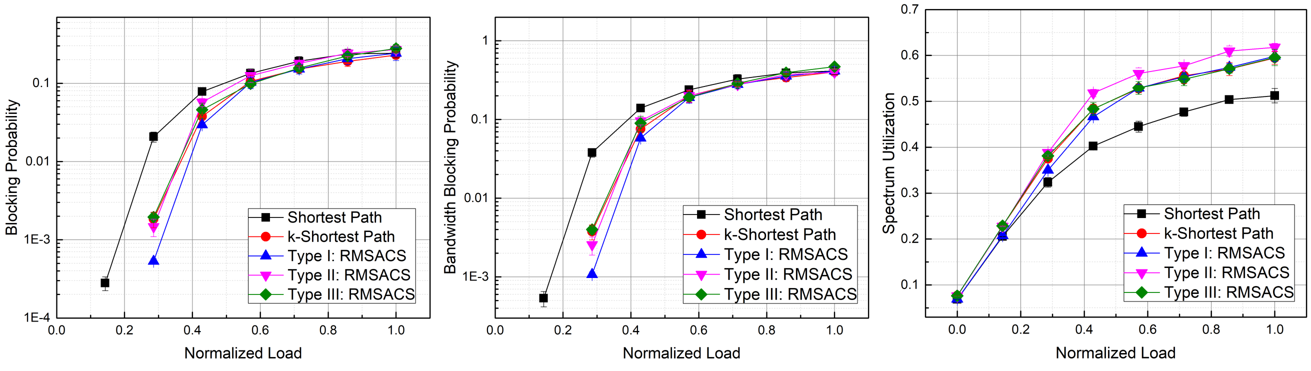
The figures 8a and 8b are plots of the blocking probability and bandwidth blocking probability of the algorithms for NSFNET with demand rate of 100Gbps. Here, we are using distance adaptive modulation and first-fit spectrum allocation techniques. From figures, we can see the performance of the proposed RMSACS algorithms is better than the traditional algorithms (also using distance adaptive modulation techniques). Additionally, for lower loads Type I: RMSACS outperforms other algorithms. As the load increases, the performance of all the algorithms converges towards almost same blocking probability values.

Figure 8c compares the performance of the proposed algorithms in terms of spectrum utilization. The blocking probability of the connections for lower loads is almost negligible. Hence, again the spectrum remains underutilized. But as the offered load values increase, spectrum utilization also increases for all the algorithms and the shortest path algorithms for higher loads. We again observed that higher blocking of connections and lower spectrum utilization for the shortest path is due to the unavailability of the contiguous spectrum slots in a fragmented state.

Figures 9a, 9b and 9c are also for NSFNET with the maximum demand rate of 200 Gbps. For higher demand rates, the early impact on connection blocking can be seen at lower loads, as the availability of required spectrum slots becomes less than that for the case of 100 Gbps. The connection blocking and bandwidth blocking comparison is similar to that for the demand rate of 100 Gbps. The spectrum performance is also similar, i.e. the spectrum utilization of the shortest path algorithms is worst as the offered load per node increases. Whereas, for other four algorithms', spectrum utilization performance is almost same.

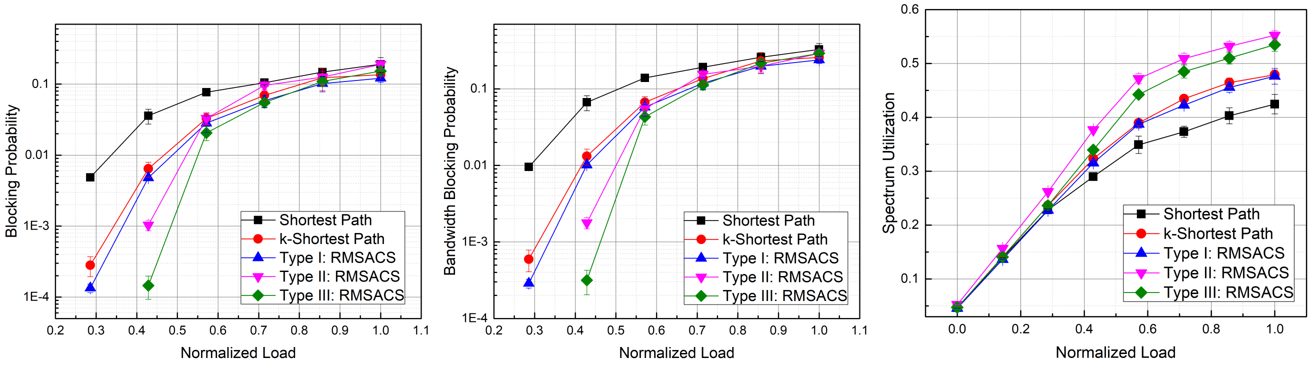
We have also used USNET network to determine the performance trends with topology change. Figures 10a, 10b and 10c are for USNET network where the value of the incoming demand rate is 100 Gbps. The blocking performance of the lightpath requests and spectrum slots changes as compared to NSFNET. In USNET, the Type II and Type III algorithm outperforms the other strategies. Here, again the shortest path algorithms perform worst in terms of blocking probability and utilization of the spectrum slots. When the network diameter changes, Type II and Type III have the higher spectrum utilization for the higher offered loads than Type I and k-shortest path.

Another set of observations are presented in Table 2. The time complexity for the worst-case scenario is lowest for Shortest Path. Whereas time complexity for k-Shortest Path and Type II are comparable depending upon the value of k. Because if $k = 1$, then the k-shortest path converges to the shortest path. However, if the value of k is too high, then time complexity keeps worsening. Even higher than Type II. Therefore, the time complexity for the worst-case scenario is in the following order Shortest Path $<$ k-Shortest Path, Type II $<$ Type I, Type III.



(a) (b) (c)

FIGURE 9. **Various performance metric vs offered loads for NSFNET with a demand rate of 200 Gbps. Considering distance-adaptive modulation techniques.**



(a) (b) (c)

FIGURE 10. **Various performance metric vs offered loads for USNET with a demand rate of 100 Gbps. Considering distance-adaptive modulation techniques.**

The Type II algorithm has better performance in blocking probability, spectrum utilization and time complexity for the worst-case scenario. Also, we know a priori blocking, i.e., at the routing time, so there is no need for spectrum assignment in this algorithm. Therefore, it can be used to lower the blocking of the lightpath requests. However, eliminating fragmentation is impossible; therefore, one can use the de-fragmentation strategy in addition to Type II algorithm, instead with the k-shortest path such that the traffic is disrupted fewer times.

VI. CONCLUSION

The Routing and Spectrum Assignment is a tedious process in Flexible grid Optical Networks. The main reason is dynamic arrivals and departures of the lightpath requests and contiguity constraint in lightpath setup, resulting in fragmentation within the spectrum of the network. In this paper, instead of using static parameters for RSA, we used adaptive parameters, i.e., available consecutive spectrum slots for routing. The consecutive spectrum slots keep changing with the network conditions. We performed the detailed analysis under different demand rates for NSFNET and USNET. The performance of the proposed algorithms is better than the existing strategies in terms of lower blocking of the lightpath requests. One of the reasons is the lower fragmentation of the spectrum slots. We also performed a detailed analysis of the proposed algorithms using distance-adaptive modulation formats. The Type II algorithm has lower blocking probability, higher spectrum utilization and lower time complexity.

In this work, we did not consider the fragmentation metric for RSA. We can further design fragmentation-aware joint RSA algorithms and compare their performance with the proposed fragmentation-aware k-shortest path algorithm. Another work that we can extend is the use of the multipath routing concept as discussed in section V.C.1.

REFERENCES

[1] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective*, 2nd ed. Amsterdam, The Netherlands: Elsevier, 2004.

[2] *Spectral Grids for WDM Applications: DWDM Frequency Grid*, document ITU-T G.694.1, 2021. [Online]. Available: https://www.itu.int/rec/T-REC-G.694.1/en

[3] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 66–73, Nov. 2009.

[4] O. Gerstel, M. Jinno, A. Lord, and S. J. B. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 12–20, Feb. 2012.

[5] V. Lohani and Y. N. Singh, "Migration to next generation elastic optical networks," in *Proc. ICEIT*. Delhi: MCNC, Feb. 2017, pp. 137–141.
[6] B. Mukherjee, "WDM optical communication networks: Progress and challenges," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1810–1824, Oct. 2000.
[7] M. Klinkowski and K. Walkowiak, "Routing and spectrum assignment in spectrum sliced elastic optical path network," *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 884–886, Aug. 2011.
[8] X. Zhou, W. Lu, L. Gong, and Z. Zhu, "Dynamic RMSA in elastic optical networks with an adaptive genetic algorithm," in *Proc. GLOBECOM*, Dec. 2012, pp. 2912–2917.
[9] M. Dallaglio, A. Giorgetti, N. Sambo, L. Velasco, and P. Castoldi, "Routing, spectrum, and transponder assignment in elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 22, pp. 4648–4658, Nov. 15, 2015.
[10] X. Chen, Y. Zhong, and A. Jukan, "Multipath routing in elastic optical networks with distance-adaptive modulation formats," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 3915–3920.
[11] B. C. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1776–1800, 3rd Quart., 2015.
[12] M. Salani, C. Rottondi, and M. Tornatore, "Routing and spectrum assignment integrating machine-learning-based QoT estimation in elastic optical networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, Apr. 2019, pp. 1738–1746.
[13] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]," *IEEE Commun. Mag.*, vol. 48, no. 8, pp. 138–145, Aug. 2010.
[14] A. Castro, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio, "Dynamic routing and spectrum (re)allocation in future flexgrid optical networks," *Comput. Netw.*, vol. 56, no. 12, pp. 2869–2883, Aug. 2012.
[15] F. Yousefi, A. G. Rahbar, and M. Yaghubi-Namaad, "Fragmentation-aware algorithms for multipath routing and spectrum assignment in elastic optical networks," *Opt. Fiber Technol.*, vol. 53, Dec. 2019, Art. no. 102019.
[16] B. C. Chatterjee, S. Ba, and E. Oki, "Fragmentation problems and management approaches in elastic optical networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 183–210, 1st Quart., 2018.
[17] F. Pederzolli, D. Siracusa, A. Zanardi, G. Galimberti, D. La Fauci, and G. Martinelli, "Path-based fragmentation metric and RSA algorithms for elastic optical networks," *J. Opt. Commun. Netw.*, vol. 11, no. 3, pp. 15–25, 2019.
[18] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, and S. J. B. Yoo, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks [Invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, p. A100, 2013.
[19] R. Zhu, Y. Zhao, H. Yang, X. Yu, J. Zhang, A. Yousefpour, N. Wang, and J. P. Jue, "Dynamic time and spectrum fragmentation-aware service provisioning in elastic optical networks with multi-path routing," *Opt. Fiber Technol.*, vol. 32, pp. 13–22, Dec. 2016.
[20] A. Alyatama, I. Alrashed, and A. Alhusaini, "Adaptive routing and spectrum allocation in elastic optical networks," *Opt. Switching Netw.*, vol. 24, pp. 12–20, Apr. 2017.
[21] A. Alyatama, "Adaptive spectrum allocation algorithm for elastic optical networks with multicasting," *J. High Speed Netw.*, vol. 24, no. 2, pp. 133–146, Mar. 2018.
[22] A. Alyatama, "Relative cost routing and spectrum allocation in elastic optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 12, no. 3, pp. 38–49, Mar. 2020.
[23] Y. Zhou, Q. Sun, and S. Lin, "Link state aware dynamic routing and spectrum allocation strategy in elastic optical networks," *IEEE Access*, vol. 8, pp. 45071–45083, 2020.
[24] R. Zhu, Y. Zhao, H. Yang, H. Chen, J. Zhang, and J. P. Jue, "Crosstalk-aware RCSA for spatial division multiplexing enabled elastic optical networks with multi-core fibers," *Chin. Opt. Lett.*, vol. 14, no. 10, pp. 100604–100608, 2016.
[25] Y. Lei, B. Chen, M. Gao, L. Xiang, and Q. Zhang, "Dynamic routing, core, and spectrum assignment with minimized crosstalk in spatial division multiplexing elastic optical networks," in *Proc. Asia Commun. Photon. Conf. (ACP)*, Hangzhou, Oct. 2018, pp. 1–3.
[26] Y. Zhao, L. Hu, R. Zhu, X. Yu, X. Wang, and J. Zhang, "Crosstalk-aware spectrum defragmentation based on spectrum compactness in space division multiplexing enabled elastic optical networks with multicore fiber," *IEEE Access*, vol. 6, pp. 15346–15355, 2018.
[27] Y. Zhao, L. Hu, R. Zhu, X. Yu, Y. Li, W. Wang, and J. Zhang, "Crosstalk-aware spectrum defragmentation by re-provisioning advance reservation requests in space division multiplexing enabled elastic optical networks with multi-core fiber," *Opt. Exp.*, vol. 27, no. 4, pp. 5014–5032, 2019.
[28] V. Lohani, A. Sharma, and Y. N. Singh, "Routing, modulation and spectrum assignment using an AI based algorithm," in *Proc. 11th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2019, pp. 266–271.

VARSHA LOHANI (Member, IEEE) received the Bachelor of Technology (B.Tech.) degree in electronics and communication engineering from Uttarakhand Technical University, Dehradun, India, in 2013, the Master of Technology (M.Tech.) degree in electronics and communication engineering from the BTKIT, Dwarahat, India, in 2015, and the Ph.D. degree from the Indian Institute of Technology Kanpur, Uttar Pradesh, in 2022. Her current research interests include optical networks, fibre optics, optical switch design, and optical communication.

ANJALI SHARMA (Member, IEEE) received the Bachelor of Technology (B.Eng.) degree in electronics and communication engineering from Panjab University, Chandigarh, India, in 2014, and the Master of Technology (M.Tech.) degree in electronics and communication engineering from HBTI (now HBTU), Kanpur, India, in 2016. She is currently pursuing the Ph.D. degree with the Indian Institute of Technology Kanpur, Uttar Pradesh. Her current research interests include optical networks, fibre optics, optical communication, and optical switch design.

YATINDRA NATH SINGH (Senior Member, IEEE) received the B.Tech. degree from REC Hamirpur and the M.Tech. and Ph.D. degrees from the IIT Delhi. He is currently a Professor with the Department of Electrical Engineering, Indian Institute of Technology Kanpur. He has supervised 19 Ph.D. students. His research interests include optical networks-protection and restoration of the packet and circuit switching, optical packet switching architectures, overlaid multicasting, technology development for e-learning and e-education, peer-to-peer networking, and optical communication systems and theory. He is a fellow of IETE and a Senior Member of ICEIT. He was given the AICTE Young Teacher Award, in 2002–2003. He had been the Chairman IEEE UP Section.

• • •