Optical Fiber Technology 66 (2021) 102655

Contents lists available at ScienceDirect

Optical Fiber Technology

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

Novel fragmentation-aware algorithms in space Division Multiplexing Elastic Optical Networks

Elham Pourkarimi, Akbar Ghaffarpour Rahbar *

Computer Networks Lab, Faculty of Electrical Engineering, Sahand University of Technology, Tabriz, Iran

ARTICLE INFO

Keywords:
Elastic optical networks
Space division multiplexing
Fragmentation
Multi-core Fiber

ABSTRACT

Space Division Multiplexing Elastic Optical Networks (SDM-EONs) has been presented to provide flexibility and high speed transmission to increase network capacity and optimal use of network resources. Dynamic establishment and release of connections causes fragmentation in EONs, thus increasing blocking probability as a result of increased fragmentation. In this paper, we present three novel algorithms that focus on how to allocate resources in SDM-EONs under Multi-Core Fiber (MCF) in order to reduce blocking probability by reducing bandwidth fragmentation. The Core Classification Aware of Fragmentation (CCAF) is a method based on the core classification mechanism that uses the concept of demand splitting and a cost function to determine the appropriate spectral space for a new request. The Fragmentation Aware Spectrum and Core Assignment (FASCA-without priority) is a routing and spectrum allocation method that uses a cost function to specify the appropriate spectral space to improve spectrum efficiency and reduce fragmentation. The Fragmentation Aware Spectrum and Core Assignment (FASCA-with priority) is a routing and spectrum allocation method that considers priority of requests to improve spectrum efficiency and reduce fragmentation. Simulation results show that the proposed CCAF algorithm can achieve better performance compared with the FASCA-without priority and FASCA-with priority in terms of blocking probability and spectrum utilization. Moreover, FASCA-without priority can provide better performance than FASCA-with priority.

1. Introduction

Technology developments in communication and networking are essential due to the increasing Internet demands [1]. Therefore, we require optical networks to transmit data at the speed of Tb/s [2]. Nowadays, optical networks are based on Fixed Wavelength Division Multiplexing (WDM) technologies. Because of its fixed bandwidth allocation, WDM has some disadvantages such as inflexibility, inefficient spectrum utilization, requirement for full allocation of a wavelength, and therefore, it cannot provide enormous bandwidth for connection requests [3]. The main problems for high speed and elastic transmission by optical networks are attenuation, cross-talk and dispersion [4].

Elastic Optical Networks (EONs) can allocate resources elastically, and therefore, they are better solution than WDM networks [5]. Spatial Division Multiplexing (SDM) by using the Multi-Core Fiber (MCF) or Multi-Mode Fiber (MMF) architecture can recover the transmission capacity of single fiber link. The Routing and Spectrum Assignment (RSA) is one of the problems of EONs which is divided into two sub-problems, routing and spectrum allocation. For a given connection request, RSA selects the possible path between its source and destination, and allocates bandwidth based on its required bandwidth. The most important constraints in EONs are the spectrum continuity and spectrum contiguity constraints [6]. According to the spectrum continuity constraint, required Frequency Slots (FSs) must be allocated in the same FS number over all the links of a path. The contiguity constraint ensures that the needed FSs are allocated in adjacent FSs. Spectrum fragmentation is caused by frequent establishment and release of dynamic connections [7]. In this paper, we suggest three spectrum and core classification methods to reduce fragmentation problem by considering the cost function under SDM scenarios.

1.1. Related work

In [8], RSA and Routing, Modulation and Spectrum Allocation (RMSA) methods have been introduced by considering the Quality of Transmission (QoT) and the complexity of computations. In [9,10], the multipath RSA has been presented in which if an arriving connection could not be allocated in the selected path, the network tries to allocate

* Corresponding author.
E-mail address: ghaffarpour@sut.ac.ir (A. Ghaffarpour Rahbar).

https://doi.org/10.1016/j.yofte.2021.102655
Received 15 April 2021; Received in revised form 16 July 2021; Accepted 26 July 2021
Available online 6 August 2021
1068-5200/© 2021 Elsevier Inc. All rights reserved.

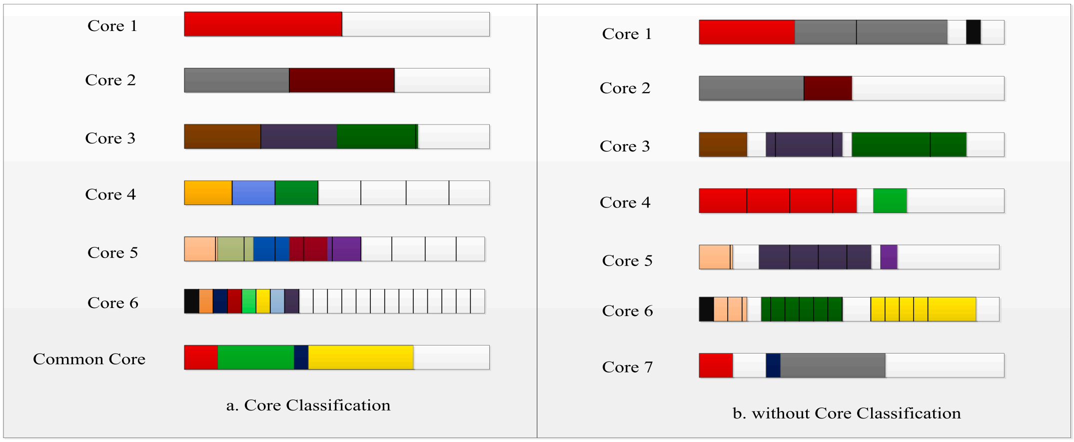

Fig. 1. An example of spectrum utilization with core classification and without core classification.

it in different paths. A Fragmentation-Aware method with Routing and Spectrum Assignment (FA-RSA) by considering external fragmentation measurement has been suggested in [11]. In [12], a review of resource allocation schemes and algorithms in a flexible spectral-spatial optical network has been introduced. In [13,14], two algorithms have been presented to improve QoS parameters such as average number of waiting calls and blocking probability. In [15], a method has been expressed for routing, spectrum assignment and core allocation in EONs with multi-cores and with maximum index used frequency slots.

In [16], the Fragmentation-Aware Best Splits (FABS) algorithm uses fragmentation criteria, and the Best Split and Crosstalk-Aware Best Split (CABS) algorithm as a new split technique. These two algorithms can improve blocking probability and increase spectrum utilization.

In [17], two methods have been presented in SDM-EONs, called CMDE-RSCA (crosstalk-aware) algorithm and FMDE-RSCA (fragmentation-aware) algorithm, in order to improve fragmentation and inter-core crosstalk.

In [18], the fragmentation-aware and time-based algorithms based on holding time and variable CVM coefficient have been introduced to reduce blocking probability and fragmentation in SDM-EONs. In [19], the methods for reducing crosstalk and blocking probability have been introduced in SDM-EONs by considering the security level of the physical layer.

In [20], a method has been presented for routing, spectrum assignment and core allocation based on the Multiple Criteria Decision Making (MCDM) methods in EONs with multi-cores, where cores are classified based on the holding time of an input request.

The authors in [21] have presented the Traffic Awareness Cross-talk Interference Avoidance (TACIA- RMSCA), which is a Routing, Modulation, Spectrum, and Core Allocation (RMSCA) algorithm, that overcomes inter-core crosstalk (XT) and calculates the performance of MCF-SDM-EONs for peak load of the fluctuating traffic.

The main purpose of [22] is to investigate the issue of time-spectrum defragmentation in SDM-EONs. In this paper, the Time-dimensional Spectrum Compactness (TSC) metric is presented. Using this metric, a Crosstalk-aware Re-Provisioning (CRP) algorithm has been introduced with two strategies to re-provision Advance Reservation) AR (requests and increase spectrum efficiency in SDM-EONs.

In [23], a Routing, Modulation, Spectrum, and Core Allocation (RMSCA) algorithm, called Distance-adaptive Energy-aware Resource Allocation (DERA) has been presented to reduce the dynamic XT effects in SDM-EONs by considering the concept of the survival multipath scheme and Spectrum Compactness (SC) during resource allocation.

In [24], a Routing, Modulation and Spectrum Assignment (RMSA) method in elastic optical networks has been presented, which selects an appropriate block for each request that has a minimum sum of weighted resource reductions.

The work in [25] reduces spectrum fragmentation by considering core classification in SDM-EONs. In [26,27], a method has been presented to classify the cores based on the number of requested frequency slots. Fig. 1(a) presents spectrum utilization with core classification. The core classification method is based on the number of cores C, where each of the Cores 1 to C-2 is classified only for one region. These regions are organized based on the number of cores in the network and prime numbers (2, 3, 5, 7, 11, 13…). If an input request cannot be allocated in Cores 1 to C-1, the common core (the C-th core) will be used. The common core is the last core that is searched for allocation. Note that any request with any number of frequency slots can be allocated in the C-th core.

The (C-1)-th core is organized for the region of one FS and the other cores are classified based on the prime numbers. Fig. 1(b) depicts spectrum utilization without core classification. Therefore, fragmentation decreases when the same bandwidth requests are allocated to specific cores.

In [9], a spectrum and core classification method has been presented for SDM-EONs under MCF, called Spectrum Block Multipathing per Core (SBMC). This method reduces blocking probability by using multipath mechanism and considering core classification. In [28], the KSP-FASA routing and spectrum allocation algorithm has been introduced to reduce spectrum fragmentation and increase spectrum efficiency. In this algorithm, a block cost function is specified to evaluate candidate spectrum blocks for a given path. The block cost function is investigated based on the status of adjacent frequency slots in order to reduce fragmentation after spectrum allocation.

The First Core Fit (FCF) algorithm [18] is a method of spectrum allocation and routing for SDM-EONs under MCF, that selects the first empty block appropriate to the request for allocation and does not consider fragmentation.

1.2. Objective and contribution

Our objective in this paper is to propose three new algorithms, which try to improve blocking probability by considering core classification and calculation of the block cost function in order to reduce spectrum fragmentation in SDM-EONs.

In the Core Classification Aware of Fragmentation (CCAF) algorithm, which is a method based on core classification mechanism under MCF using the concept of demand splitting and considering of block cost function, the appropriate spectral block for a new request is determined. The block cost function means that the appropriate empty block is selected for the allocation of the requested bandwidth, which has the lowest cost among the empty blocks, thus leading to the least fragmentation. The cost of each block is calculated based on the status of the neighboring slots. Under the proposed CCAF algorithm, by considering the cost functions of the available blocks in a candidate optical path, the spectrum fragmentation is improved compared to the SBMC algorithm described in [9]. In SBMC, the first empty block is selected for allocation according to the connection request size. However, in CCAF, the block that has the lowest cost from the available blocks is selected, thus the proposed CCAF algorithm causes the least fragmentation compared with SBMC.

In the Fragmentation Aware Spectrum and Core Assignment (FASCA-without priority), which is a routing and spectrum allocation method for SDM-EONs under MCF, which considers a block cost function, the appropriate spectral block is specified in order to improve spectrum efficiency and reduce fragmentation. In the Fragmentation Aware Spectrum and Core Assignment (FASCA-with priority), that is a routing and spectrum allocation method for SDM-EONs under MCF, which considers priority of requests and block cost function, the appropriate spectral block is determined to improve spectrum efficiency and reduce fragmentation.

Our contribution is to propose three novel fragmentation-aware algorithms to reduce blocking probability in SDM-EONs. The proposed algorithms can achieve this goal by using multi-core fiber and considering block cost function. All three proposed algorithms can improve the utilization of the spectrum and reduce blocking probability by assigning each connection in the corresponding empty block, according to the cost of the empty blocks in the candidate path in order to select the best block that causes the least fragmentation. Briefly, the contribution of this paper is as follows:

- Presenting a routing, spectrum allocation and core classification method in SDM-EONs under MCF, called CCAF that calculates the cost of appropriate empty spectral blocks in the candidate path and selects the block that causes the least fragmentation to be assigned to an arriving request. In this way, the CCAF algorithm can reduce the blocking probability.
- Proposing the FASCA-with priority and FASCA-without priority algorithms, that are routing and spectrum allocation methods in SDM-EONs under MCF to improve the spectrum fragmentation. These two algorithms reduce the blocking probability by considering the cost function.
- Evaluating the performance of three proposed algorithms and compare them in terms of blocking probability and spectrum utilization and compare them with benchmark algorithms.

1.3. Organization

The organization of this paper is as follows. In Section 2, the network model and definitions are expressed. In Section 3, the CCAF, FASCA-without priority and FASCA-with priority algorithms are introduced. In Section 4, performance evaluation results for the proposed algorithms are presented. Section 5 concludes the paper and finally Section 6 presents future works.

2. Network model and definitions

The SDM-EON is formulated as graph $G = (N, L, C)$ where N specifies the set of nodes, L is the set of fiber links and C denotes cores in the network. Each link contains a core set C and each core has a set of frequency slots. Matrix M_L is provided to indicate the status of each frequency slot on each fiber link L,

$$M_L = \begin{bmatrix} M_{1,1}M_{1,2}\cdots M_{1,c} \\ M_{2,1}M_{2,2}\cdots M_{2,c} \\ M_{3,1}M_{3,2}\cdots M_{3,c} \\ \cdots\cdots\cdots \\ M_{f,1}M_{f,2}\cdots M_{f,c} \end{bmatrix}$$

Where the matrix rows specify the number of cores C in link L and matrix columns specify the number of frequency slots in each core. For example, if $M_{f,c} = 0$ it means that the f-th slot in core C is occupied and if $M_{f,c} = 1$ it means that the f-th slot in core C is empty. The candidate path is selected using the shortest path algorithm. The number of requested frequency slots is calculated based on the requested bandwidth using the following Eq.(1). Parameter *Guardband* is defined as the minimum frequency range which separates two contiguous lightpaths in a common link.

$$\text{Number of requested FSs} = \frac{Requested\ bandwidth(GHz) + Guardband(GHz)}{FS\ width(GHz)} \tag{1}$$

Inter-core crosstalk is one of the important physical constraint that reduces the signal quality during transmissions in SDM-EONs. According to Eq.(2), the statistical mean inter-core crosstalk for a MCF can be evaluated. The mean inter-core crosstalk can be calculated as shown in Eq.(3) using the coupled-power theory.

$$h = \frac{2k^2 r}{\beta w_{th}} \tag{2}$$

$$XT = \frac{n - n \times \exp[-(n+1) \times 2hl]}{1 + n \times \exp[-(n+1) \times 2hl]} \tag{3}$$

Here k, r, β and $w_{th}$are the coupling coefficient, bend radius, propagation constant and core pitch. Parameter $l$specifies the length of the fiber link and n is the total number of neighboring cores. Bend radius, which is measured to the inside bend, is the minimum radius one can bend a fiber without damaging it. The smaller bend radius represents the greater material flexibility. In our proposed algorithms, XT is not considered [29].

3. The proposed algorithms

In this section, three algorithms are detailed, called Core Classification Aware of Fragmentation (CCAF), Fragmentation Aware Spectrum and Core Assignment (FASCA-without priority) and Fragmentation Aware Spectrum and Core Assignment (FASCA-with priority). In all three proposed algorithms, the cost of appropriate empty spectral blocks according to the *LowestCostBlock* (*Path, Required_FS, Core*) function that will be presented below is calculated to find the empty block with the lowest cost. The block cost function means that the appropriate empty block with the least fragmentation is selected for allocation to an arriving request in SDM-EONs under MCF, which has the lowest cost among the empty blocks. The cost of each block is calculated based on the status of the neighboring slots. The block with the lowest cost means a block that by assigning the request to it, there will be the least fragmentation in the spectrum.

The CCAF algorithm is a method based on core classification mechanism in SDM-EONs under MCF using the concept of demand splitting and considering of cost function. The FASCA-without priority and FASCA-with priority algorithms are the routing and spectrum allocation methods for SDM-EONs under MCF. Considering a block cost function, the appropriate spectral block is specified in order to improve spectrum efficiency and reduce fragmentation. The only difference between the FASCA-without priority and FASCA-with priority algorithms is that in the FASCA-with priority algorithm the priority of requests is considered. However, in the FASCA-without priority algorithm, the priority of requests is not considered.

In Algorithm 1, the *LowestCostBlock* (*Path, Required_FS, Core*) function is shown, which is used in CCAF, FASCA-without priority and FASCA-with priority algorithms. We now detail the operation of this function. In this function, the total number of frequency slots in each link is indicated by n and according to the requested FSs, the block that causes the least fragmentation is selected from the available blocks in the path. The first index of this block is considered to be *First_index* and

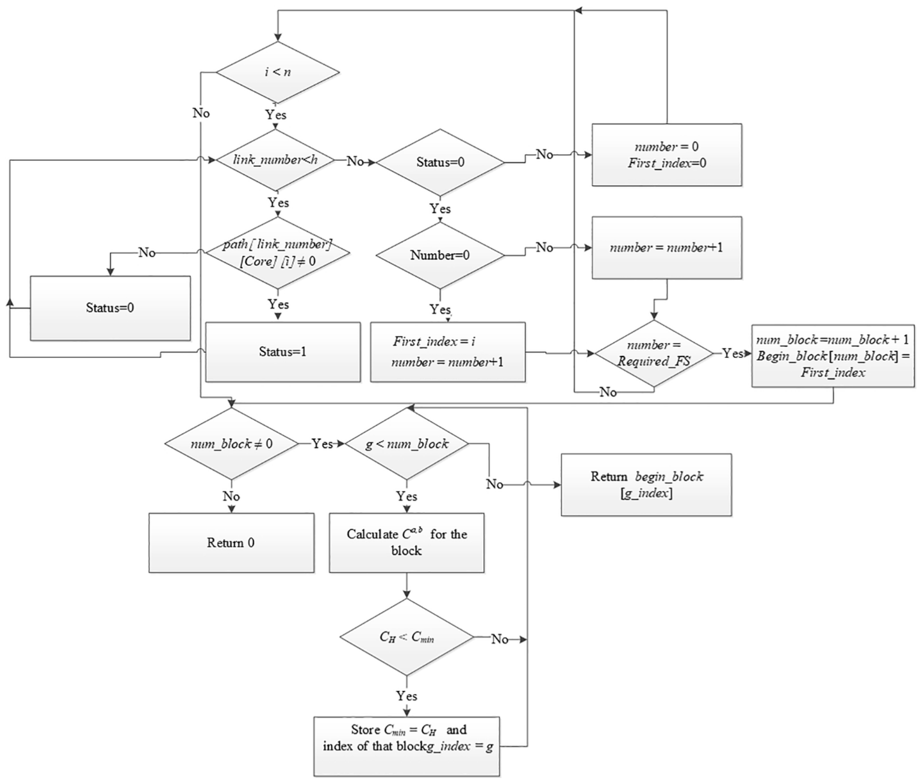

Fig. 2. Flowchart of the *LowestCostBlock* (*Path, Required_FS, Core*) function.

parameter *number* specifies the number of FSs in each block. According to Line 30 of the *LowestCostBlock* function, if all the FSs in the spectrum are checked and at least one available block appropriate to the request is found, according to Lines 31 and 32 of the *LowestCostBlock* function, the block cost ($C^{a,b}$) must be calculated for all available blocks that have been searched. According to Eq.(4), a FS has two statuses: 0(busy) or 1 (idle). Parameter $S^{a,b}(i)$specifies the status of the i-th slot. If$S^{a,b}(i) = 0$, it means that this slot is busy and it cannot be used to assign to a request. However, if$S^{a,b}(i) = 1$, the slot is idle and it can be used to assign to a request. Base on the Eq.(5), the cost of the leftmost block is calculated based on the status of the right neighboring slot, the cost of the rightmost block is calculated based on the status of the left neighboring slot, and the cost of the middle block is computed based on the status of right and left neighboring slots. Based on Line 38, the first available block with the lowest cost is selected for allocation.

In Algorithm 1, the degree of complexity for Lines 1 to 29 is O($n \times D$) and for Lines 30 to 41 is O($n/2$) in the worst case since the maximum number of available blocks is n/2. Therefore, the complexity of *LowestCostBlock* (*Path, Required_FS, Core*) is O($n \times D + n/2$) = O($n \times D$), where n is the total number of frequency slots in each link and D is the network diameter.

Assume a connection request needs m FSs along its path. First, in Core 1, empty available blocks corresponding to m FSs are searched. Then, according to Eq.(5), the block cost along the path needs to be calculated to choose a block of spectrum that has the lowest cost, since the block that has the lowest cost causes the least fragmentation. If the request cannot be allocated in Core 1, the remaining cores are searched.

In the CCAF, FASCA-without priority and FASCA-with priority algorithms, using the following Eq.(5), an available block according to the request that has the lowest cost is selected for allocation from the empty blocks in the path. This selected block causes the least fragmentation because according to Eq.(5), for each block, based on the status of the neighboring slots, its cost is checked and the block with the lowest cost is selected for allocation. Note that Eq.(5) is used to determine the most appropriate block with the least fragmentation in all three proposed algorithms to assign to a request.

$$S^{a,b}(i) = \begin{cases} 0 & (busy) \\ 1 & (idel) \end{cases} \tag{4}$$

$$C^{a,b}(i,m) = \begin{cases} S_R^{a,b}(i+m), & i = 1 \\ S_R^{a,b}(i+m) + S_L^{a,b}(i-1), & 1 < i < n-m+1 \\ S_L^{a,b}(i-1), & i = n-m+1 \end{cases} \tag{5}$$

Where we have:

s: Source node
d: Destination node
k: The number of shortest paths
a: Node *a*
b: Node *b*
n: Total number of FSs in the fiber
m: Number of requested FSs
L: Left neighboring slot
R: Right neighboring slot

$C_{k,H}^{s,d}(i,m)$: Cost of the H^{th} block from index i to i + m-1 along the candidate path k

$C^{a,b}(i,m)$: Cost of the block from index i to i + m-1 between node a and node b

$S^{a,b}(i)$: Cost of the i-th slot between node a and b

$S_R^{a,b}(i+m)$:Cost of right neighboring slot R for $C^{a,b}(i,m)$

$S_L^{a,b}(i-1)$: Cost of left neighboring slot L for $C^{a,b}(i,m)$

In Lines 31 to 38, according to the number of available blocks, the cost of the first block is checked and then if the cost is less than C_{min}, the index of that block is stored (i.e., *g_index* = 1). Then, the second block is checked. If the cost of the second block is less than the first block, (i.e., *g_index* = 2) is saved. Finally, after checking all the blocks, the first slot index of the block with the lowest cost is returned.

Input: *Path*: a path between source and destination, *Required_FS*: number of FSs needed for the connection, *Core*: index of core.
Output: the lowest cost block is found

1. $number = 0$
2. h = number of hops for Path
3. $num\_block = 0$
4. $C_{min} = \infty$
5. C_H = cost of the H-th block
6. **For** $i = 1$ **to** n //search all n slots in the links of the path
7. $Status = 0$//initially, the FS status is empty
8. **For** $link\_number = 1$ **to** h //Check all links of Path
9. **If** ($path$ [$link\_number$] [$Core$] [i] $\neq 0$) //FS status in core and specified path
10. $Status = 1$ //1 means busy frequency slot
11. **Break**//go to line 14
12. **End If**
13. **End For**
14. **If** ($Status = 0$) //if the i-th FS is empty on all links of Path
15. **If** ($number = 0$)
16. $First\_index = i$
17. $number = number + 1$
18. **Else**
19. $number = number + 1$
20. **If** ($number = Required\_FS$)
21. $num\_block = num\_block + 1$
22. $Begin\_block$ [num_block] $= First\_index$ //the first index slot of the block is stored
23. **End if**
24. **End if**
25. **Else**
26. $number = 0$
27. $First\_index = 0$
28. **End if**
29. **End For**
30. **If** ($num\_block \neq 0$)
31. **For** $g = 1$ **to** num_block
32. Calculate $C^{a,b}$ for the block//compute cost
33. **If** ($C_H < C_{min}$)
34. $C_{min} = C_H$ //store the H-th block
35. $g\_index = g$
36. **End If**
37. **End For**
38. *Return begin_block* [*g_index*]
39. **Else**
40. *Return* 0//no block found
41. **End If**

Algorithm 1: The *LowestCostBlock (Path, Required_FS, Core)* function

Fig. 2 shows the flowchart of the *LowestCostBlock* function. Firstly, the statuses of the FSs in all links of the selected path are checked. If the *i-th* slot is empty and this slot is the first slot of the block suitable for the request, the index of that slot is saved and then it is checked whether the number of requested slots is equal to the size of the block or not. If it is equal, the found block will be added to the number of allocable blocks. Otherwise, the next slots will be checked for allocation to the request. Finally, for all allocable blocks, the cost is calculated and the index of the first slot of the block with the lowest cost is returned.

3.1. Core Classification-Aware fragmentation (CCAF)

In this algorithm, a spectrum and core allocation method for SDM-EONs under MCF is proposed to improve the bandwidth fragmentation; thus reducing the blocking probability. In this method, the cores are classified only for one region, which is based on the number of cores C in the network and prime numbers (2, 3, 5, 7, 11, 13, 17, 19, 23…).

If an input request cannot be allocated in Cores 1 to C-1, the common core (the C-th core) will be checked. The (C-1)-th core is organized for region of one FS and the other cores are classified based on the prime numbers. For example, in a network with $C = 11$ cores, Core 11 is called the common core and Core 10 is classified to 1 FS Slot Block (SB). Therefore, Core 1 is classified for 23SBs, Core 2 is classified for 19SBs, Core 3 is classified for 17SBs, Core 4 is classified for 13SBs, Core 5 is classified for 11SBs, Core 6 is classified for 7SBs, Core 7 is classified for 5SBs, Core 8 is classified for 3SBs and Core 9 is classified for 2SBs.

In this algorithm, the first core (i.e., $c = 1$) is searched for a number of blocks corresponding to $M = E\times SB[c]$ FSs in the spectrum, where $SB[c]$ specifies the core region and E is an integer that is calculated by dividing the number of remaining requested slots by SB [c]. In this way, the available block is selected for allocation, which has the lowest cost among the available empty blocks. If the remaining requested slots are not equal to zero, the second core is searched. This process continues until Core C-1. If the request cannot be allocated on Cores 1 to C-1, the common core will be searched. If the request cannot be allocated in the common core, the request will be blocked.

In Algorithm 2, the CCAF algorithm is described step by step. In Lines 1 to 8, cores are classified based on prime numbers and the number of cores. In Lines 9 to 25, the requested bandwidth is attempted to be allocated in different cores. At the end, according to Line 26, if the remaining bandwidth request is equal to zero, this request connection can be allocated successfully. Otherwise, according to Lines 28 to 29, the requested connection will be blocked. In CCAF algorithm, due to considering the cost function of available blocks in candidate path, the fragmentation is improved.

The time complexity of Lines 1 to 8 is O(C). The time complexity of Lines 9 to 25 is O($C \times k$) × O($n \times D$), where the second term is the complexity of the *LowestCostBlock* function. The last lines are performed in at most O(D), where the works relevant to connection setup happen in at most D hops. Therefore, the time complexity of CCAF is O($n \times D \times C \times k$), where n is total number of frequency slots in each link in any direction, D is the network diameter, C specifies the number of cores in the network, and k specifies the number of shortest lightpaths.

Inputs: *Required_FS*: number of FSs needed for a new connection, *C*: number of cores, *SB* [*i*]: specifies the region of core *i*
Outputs: Establish connection

1. $Prime\_numbers$ = first C-2 prime numbers are stored in descending order in this list
2. **For** $c = 1$ **to** C
3. **If** ($c = C$ **or** $c = C$-1)
4. $SB[c] = 1$
5. **Else**
6. $SB[c] = prime\_numbers[c]$ //take prime number
7. **End if**
8. **End For**
9. **For** $j = 1$ **to** k
10. $Path$ = get the j-th path
11. **For** $c = 1$ **to** C //search the C cores
12. **If** ($Required\_FS \geq SB[c]$)
13. $E =$ **Int** ($Required\_FS/SB[c]$) //Give the result an integer
14. $M = E \times SB[c]$ //Give the number of FSs to be allocated in the specified Core c
15. $Block = LowestCostBlock$ ($Path, S, c$) //$Block$ = index of the lowest cost block on path and core c
16. **If** ($Block \neq 0$)
17. Allocate the lowest cost block found from the *Block* on Core c
18. $Required\_FS = Required\_FS - M$ //Find the number of remaining frequency slots
19. **If** ($Requirted\_FS = 0$)
20. **Break** //The input connection can be assigned. Go to line 25

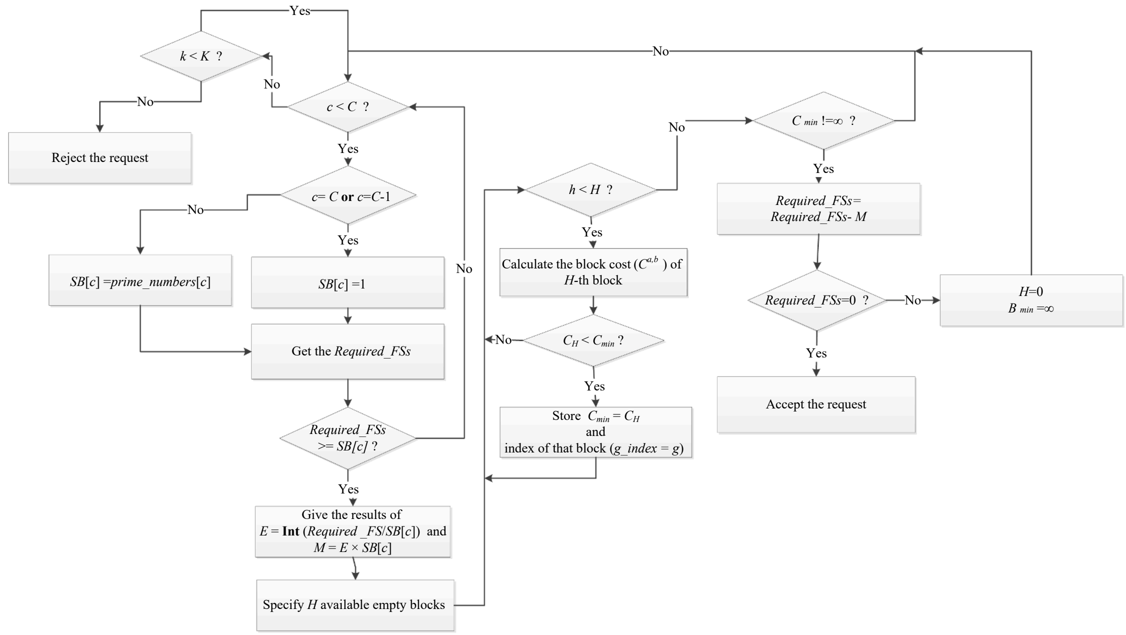

Fig.3. Flowchart of the CCAF algorithm.

21. **End if**
22. **End if**
23. **End if**
24. **End For**
25. **End For**
26. **If** (*Required_FS* = 0)
27. Setup the connection
28. **Else**
29. Block the connection
30. **End If**

Algorithm 2: The CCAF algorithm

According to the flowchart of the CCAF algorithm, which is shown in Fig. 3, firstly, the region of each core is specified, and then each core is searched for a number of blocks corresponding to M FSs in the spectrum. The requested bandwidth is attempted to be allocated in different cores. Firstly, the first core (i.e., $c = 1$) is searched for a number of blocks corresponding to M FSs and then the empty block with the lowest cost is selected for allocation. If the remaining requested slots are not equal to zero, the second core is searched. This process continues until Core C-1. If the request cannot be allocated on Cores 1 to C-1, the common core will be searched. If the request cannot be allocated in the common core, the request will be blocked.

Example 1:. *Assume we have $C = 11$ cores. Core 11 is the common core and Core 10 is classified for one FS. Then, the remaining cores are classified based on prime numbers as: $SB[1] = 23$, $SB[2] = 19$, $SB[3] = 17$, $SB[4] = 13$, $SB[5] = 11$ and $SB[6] = 7$. Parameter Required_FS must be updated for each core. For example, if a new connection arrives with 87 FSs, based on Lines 12 to 15 of the CCAF algorithm, this connection is tried to be allocated in the following order.*

Core 1:

$E = \text{Int}(Required\_FS/SB[1]) = \text{Int}(87/23) = 3$

$M = E\times SB[1]$: $S = 3 \times 23 = 69$

Since we have $M = 69$, available blocks as large as 69 FSs are searched in Core 1 and the block that causes the least fragmentation is selected from available blocks. According to Line 18, *Requeired_FS* is 87–69 = 18. Based on Line 12, we cannot assign the connection on Core 2. Therefore, Core 3 must be searched.

Core 3:

$E = \text{Int}(Required\_FS/SB[3]) = \text{Int}(18/17) = 1$

$M = E\times SB[3]$: $S = 1 \times 17 = 17$

Here, suppose there is no available blocks in Core 3 for 17 frequency slots.

Core 4:

$E = \text{Int}(Required\_FS/SB[4]) = \text{Int}(18/13) = 1$

$M = E\times SB[4]$: $S = 1 \times 13 = 13$

Here again, it is supposed that there is no available block in Core 4 to allocate 13 FSs. Therefore, Core 5 will be searched. Again, suppose that there is no available block in Core 5, and therefore, Core 6 must be investigated.

Core 6:

$E = \text{Int}(Required\_FS/SB[6]) = \text{Int}(18/7) = 2$

$M = E\times SB[6]$: $S = 2 \times 7 = 14$

If there are two available blocks in Core 6 to allocate 14 FSs, the available block with the lowest cost is selected for allocation. Here, we have *Required_FS* = 18–14 = 4 according to Line 12, and therefore, Core 7 cannot be used to allocate 4 FSs. If the remaining 4 FSs can be allocated in Cores 8 to 11, the request can be successfully established; otherwise, the connection will be blocked.

Inputs: *Required_FS*: number of FSs needed by the new connection, C: number of cores

Output: Establish connection

1. **For** $j = 1$ **to** k
2. *Path* = get the j-th Path//*Path*: the j-th path between source and destination
3. **For** $c = 1$ **to** C //search the C cores
4. *Block* = *LowestCostBlock* (*Path, Required_FS, c*) //*Block* = index of the lowest cost block on path and core c
5. **If** (*Block* ≠ 0)
6. Allocate the lowest cost block found from the *Block* on Core c
7. **Break**//go to line 10
8. **End if**
9. **End For**
10. **If** (*Block* ≠ 0) //The input connection can be assigned
11. **Break**//go to line 14
12. **End If**
13. **End For**
14. **If** (*Block* ≠ 0)//The input connection can be assigned

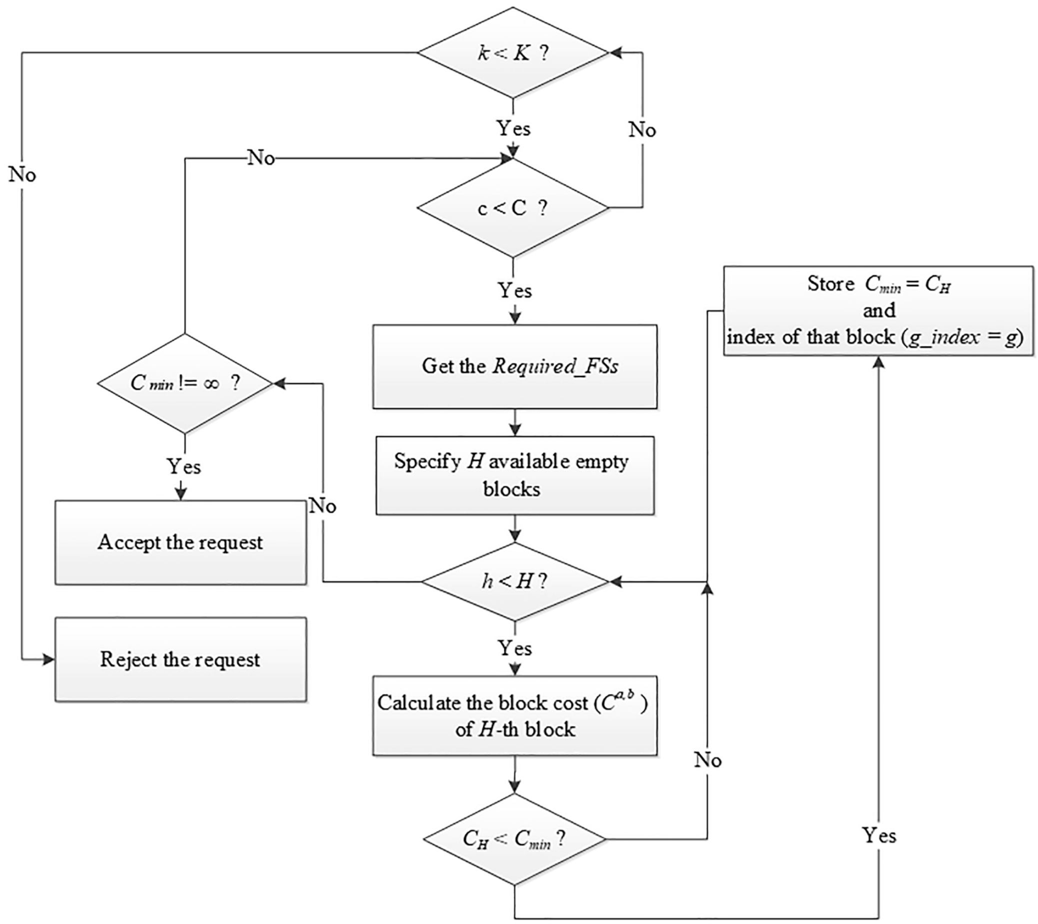

Fig. 4. Flowchart of the FASCA-without priority algorithm.

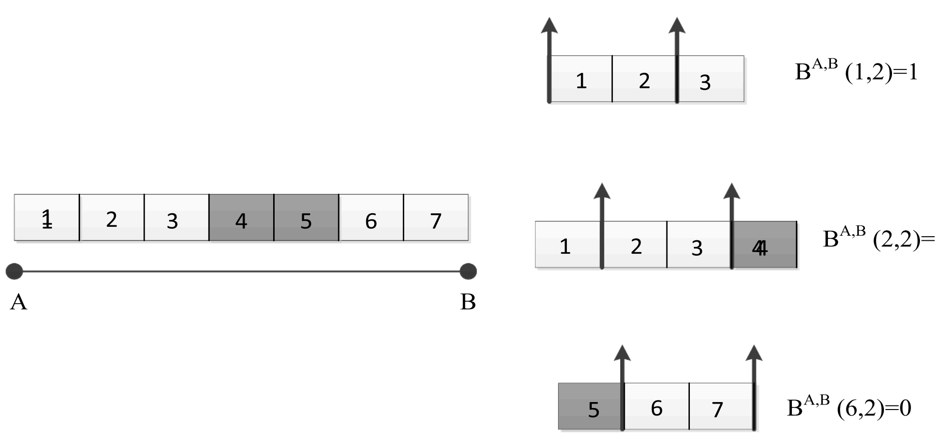

Fig.5. Calculation of block cost for Example 2.

15. Setup the connection
16. **Else**
17. Block the connection
18. **End If**

Algorithm 3: The FASCA-without priority algorithm

3.2. *Fragmentation-Aware spectrum and Core Assignment (FASCA without priority)*

In this section, we propose a spectrum and core allocation algorithm for SDM-EONs under MCF to improve fragmentation. In Algorithm 3, the cost functions of all available blocks on a candidate path are evaluated. The costs of available blocks are calculated according to Eq. (5) and the block that leads to the lowest cost is selected from the available blocks for allocation. In Lines 1 to 2 of the FASCA without priority algorithm

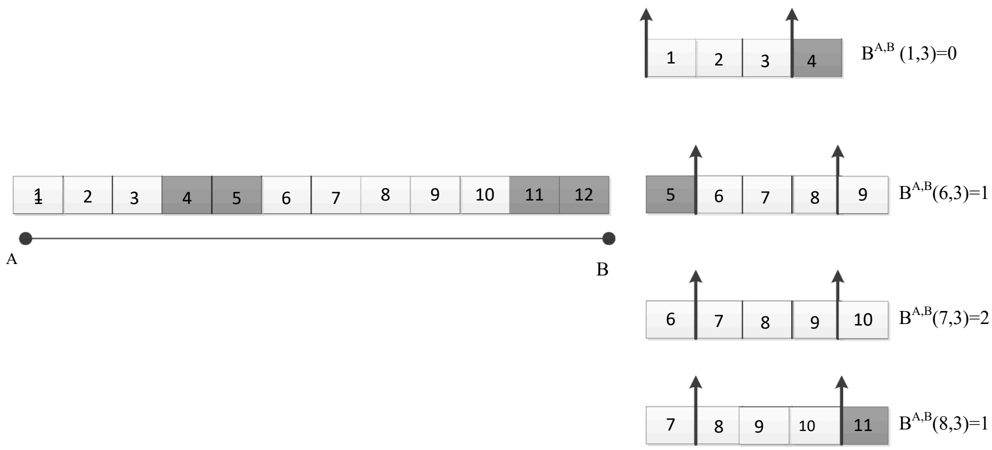

Fig.6. Calculation of block cost for Example 3.

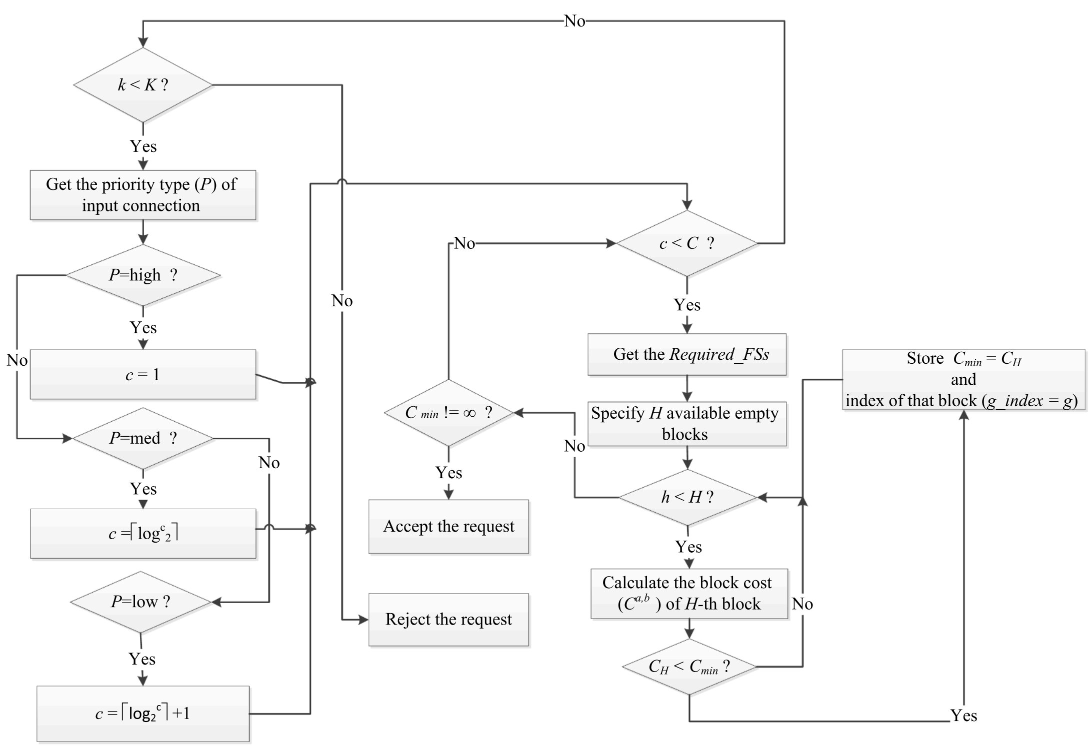

Fig.7. Flowchart of the FASCA-with priority algorithm.

(see Algorithm 3), the first shortest path is selected. In Line 3, all the cores will be checked for allocation. Firstly, Core 1 is searched. According to Line 4 of the algorithm, the first index of empty available block corresponding to the request with the lowest cost is stored in *Block*. Then, in Line 5, it is checked that if *Block*$\neq 0$ and then According to Line 6, the input request can be allocated to that block in the corresponding core. However, *Block* $= 0$ means that no available block is found in that core, and therefore, the remaining cores must be searched. In Line 14, the input request can be assigned successfully if *Block*$\neq 0$. Otherwise, according to Line 17, the request will be blocked. The time complexity of the FASCA without priority algorithm is $O(n \times D \times C \times k)$.

Fig. 4 shows the flowchart of the FASCA-without priority algorithm. First of all, Core 1 is searched for allocation, and for all allocable blocks, the cost of the block is evaluated. The block with the lowest cost is selected. Firstly, Core 1 is searched. According to Fig. 4, the first index of the empty available block corresponding to the request with the lowest cost is stored ($g\_index = g$). Then, it is checked whether $Cmin$!= ∞. In other words, if there is at least one appropriate empty block for the request, the input request can be allocated to that block in the corresponding core. However, $Cmin = \infty$ means that no available block is found in that core, and therefore, the remaining cores must be searched. The input request can be assigned successfully if $Cmin$!= ∞. The request will be blocked after all C cores have been searched and at least one empty block corresponding to the request is not available.

Example 2:. *Suppose a request enters the network that requires* 2 *FSs from source A to destination B (see* Fig. 5*). It is assumed that each link has* 7 *FSs and* $C = 7$*. Firstly, Core* 1 *is searched and if there are three available blocks in Core* 1 *to allocate to these* 2 *FSs, the cost of these three blocks must be*

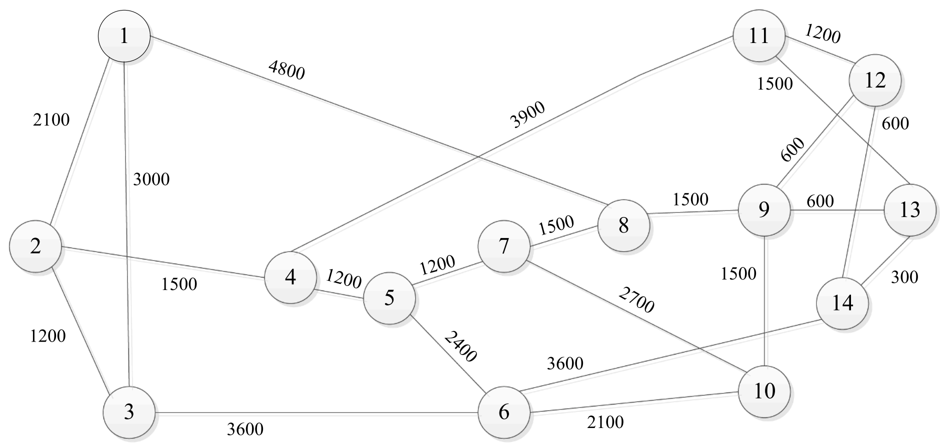

Fig.8. The NSFNET topology.

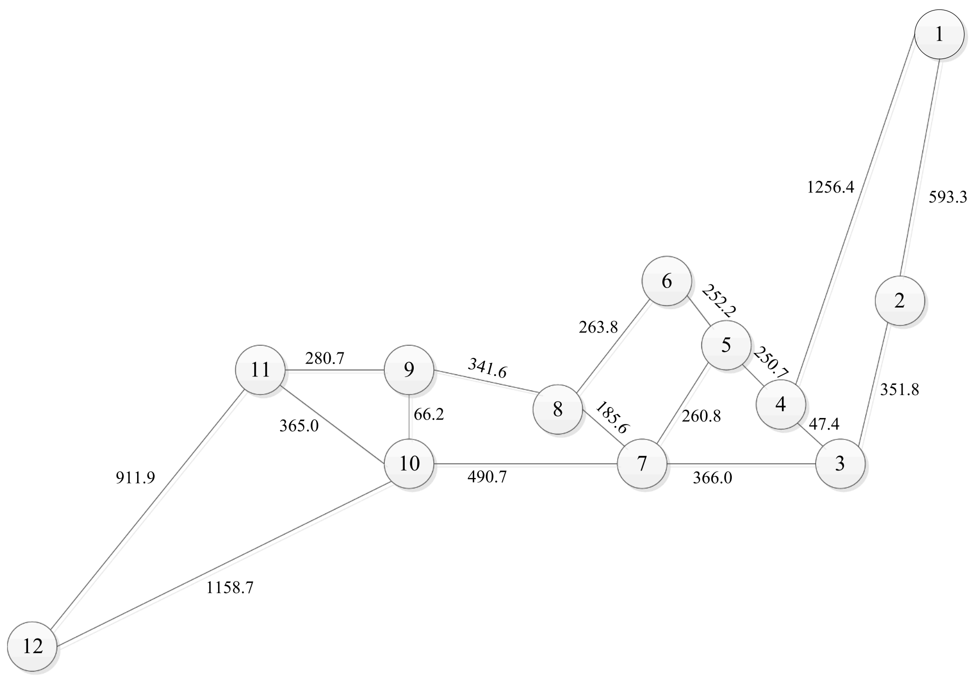

Fig.9. The JPN-12 topology.

calculated according to Eq.(5). The cost of these blocks are specified as follows: 1, 1 *and* 0. *Therefore, the third candidate block is selected for allocation to the requested connection because it has the lowest cost. If there is not enough FSs in Core* 1, *the next core is searched and thus this process continues until Core* 7. *If there is at least one available block in each of these cores, the request can be allocated; otherwise, the connection will be blocked.*

Example 3:. *In our proposed algorithms, Guardband is not considered (as our benchmark algorithms do not consider it). However, we consider an example with Guardband* = 1*FS. It is assumed that a connection arrives in the network with the bandwidth request of* 2 *FSs, C* = 7 *and each link has* 12 *FSs (see Fig. 6). Firstly, available blocks are searched to allocate* 3 *FSs (*2 *requested FSs and* 1 *FS for Guardband) in Core* 1. *According to Fig. 6, four empty blocks are available for allocation to these* 3 *FSs in Core* 1. *The cost of these four blocks must be calculated according to Eq.(5). The costs of these blocks are specified as follows*: 0, 1, 2, *and* 1. *Therefore, the first candidate block is selected for allocation to the requested connection because it has the lowest cost. If there is not enough FSs in Core* 1, *the next core is searched and thus this process continues until Core* 7. *If there is at least one available block in each of these cores, the request can be allocated*; *otherwise, the connection will be blocked.*

Example 4:. *Assume that a connection enters the network with bandwidth request of* 50 *FSs and C* = 7. *Firstly, Core* 1 *is searched for* 50 *FSs. If there are two available blocks in Core* 1, *according to Eq.(5), the block with the lowest cost is selected for allocation. However, if there is no available block for allocation in Core* 1, *the remaining cores are searched. According to Line* 4 *of the FASCA-without priority algorithm, the index of the first block that has the lowest cost is stored in Block. Then, in Line* 5, *it is checked whether the Block value is not equal to* 0. *Then, according to Line* 6, *the candidate block can be assigned to the connection. Otherwise, the remaining cores should be searched.*

3.3. Fragmentation-Aware Spectrum and Core Assignment (FASCA with priority)

In this section, we propose a spectrum and core allocation algorithm for SDM-EONs under MCF based on the priority of requests to improve

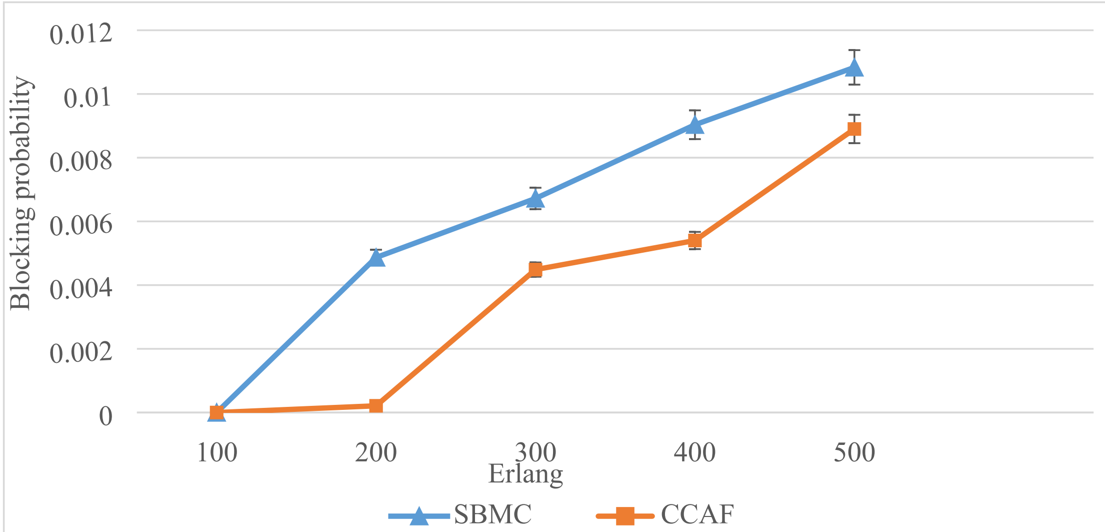

Fig. 10. Blocking Probability of Spectrum Block Multi-pathing Per Cores (SBMC) and CCAF under the NSFNET topology.

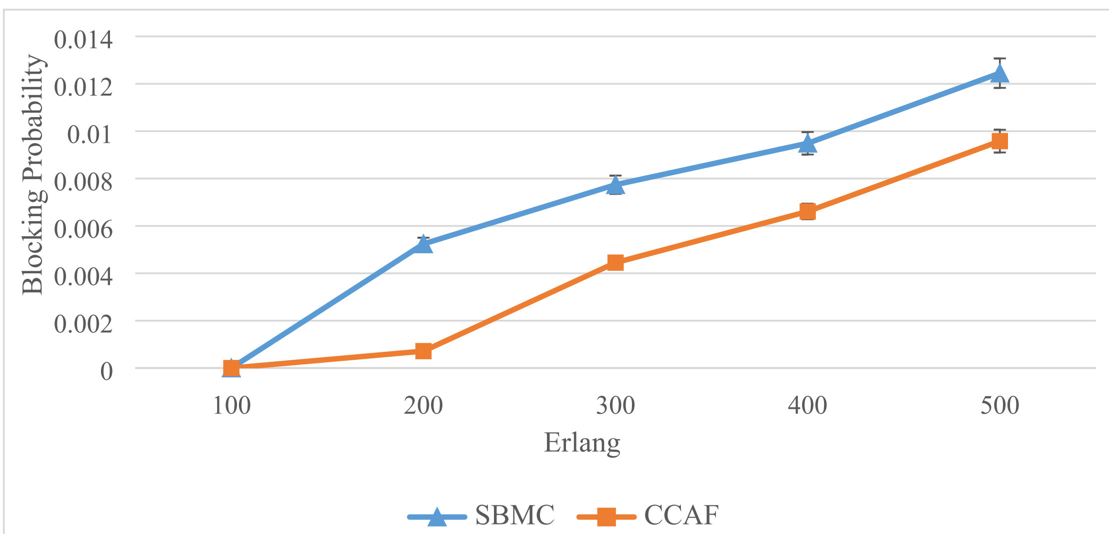

Fig.11. Blocking Probability of Spectrum Block Multi-pathing Per Cores (SBMC) and CCAF under the JPN-12 topology.

fragmentation. In this method, connection requests have three priorities: high-priority, medium-priority, and low-priority. In this algorithm (see Algorithm 4), the cost functions of all available blocks in a candidate path are evaluated. The costs of available blocks are calculated according to Eq.(5) and the block that results in the lowest cost is selected from the available blocks for allocation.

If the number of cores is considered to be $C = 7$, according to Algorithm 4, in Line 3 of the FASCA-with priority algorithm, the priority of input connection is checked first. If $pk\_type = 1$, based on Line 4, the connection priority is high and $t = 1$. In other words, in Line 11, all cores must be searched for allocation. However, if we have $pk\_type = 2$, based on Line 6, the connection priority is medium and $t = \lceil log_2^C \rceil$. In other words, in Line 11, the search is carried out starting from Core 3. Eventually, if $pk\_type = 3$, based on Line 8 of the algorithm, the connection priority is low and $t = \lceil log_2^C \rceil + 1$. In other words, in Line 11, the search is started from Core 4.

Inputs: *Required_FS*: number of FSs needed by new connection, *C*: the number of cores

Output: Establish connection

1. **For** $j = 1$ **to** k
2. $Path$ = get the j-th Path //get the j-th path between source and destination,
3. Pk_type = get the priority type of input connection
4. **If** ($Pk\_type = 1$) //High-priority request
5. $t = 1$//Start the search from Core 1
6. **Else if** ($Pk\_type = 2$) //Medium-priority request
7. $t = \lceil log_2^C \rceil$//Start the search from Core $\lceil log_2^C \rceil$
8. **Else** //Low-priority request
9. $t = \lceil log_2^C \rceil + 1$//Start the search from Core $\lceil log_2^C \rceil + 1$
10. **End if**
11. **For** $c = t$ **to** C //search cores starting from the t-th core
12. *Block* = *LowestCostBlock* (*Path*, *Required_FS*, *c*) //*Block* = index of the lowest cost block on path and core
13. **If** ($Block \neq 0$)

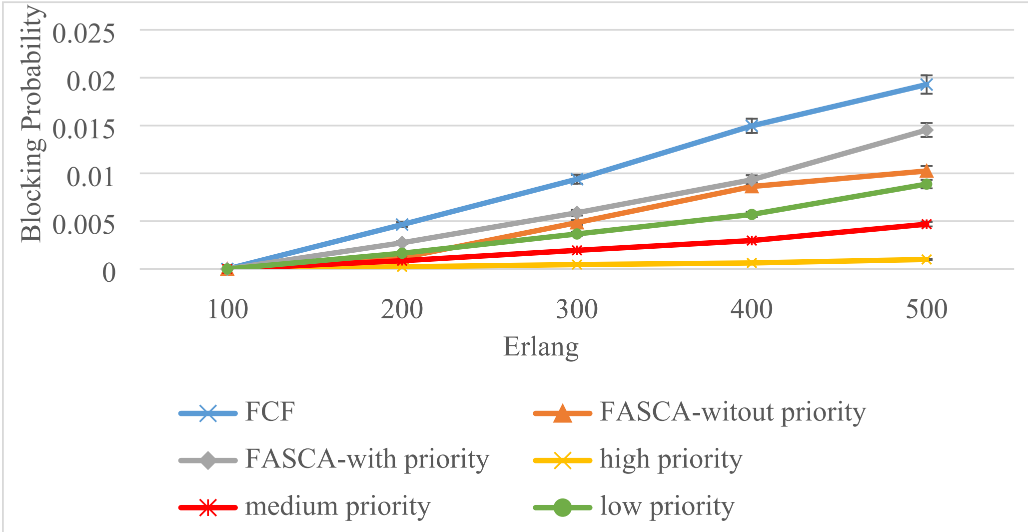

Fig. 12. Blocking Probability of First Core Fit (FCF), Fragmentation Aware Spectrum and Core Assignment (FASCA-without priority) and Fragmentation Aware Spectrum and Core Assignment (FASCA-with priority) under the NSFNET topology.

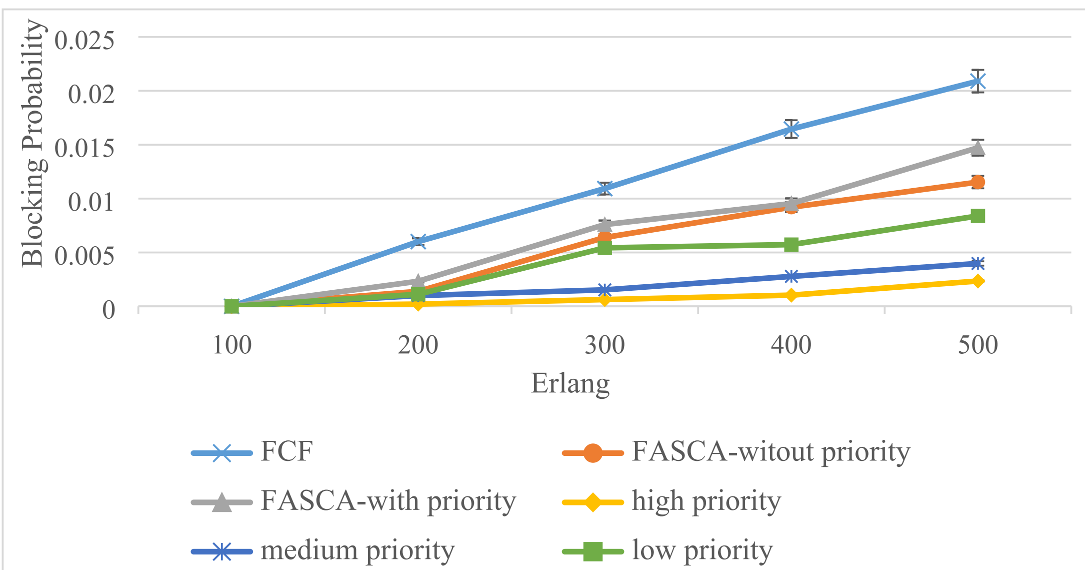

Fig.13. Blocking Probability of First Core Fit (FCF), Fragmentation Aware Spectrum and Core Assignment (FASCA-without priority) and Fragmentation Aware Spectrum and Core Assignment (FASCA-with priority) under the JPN-12 topology.

14. Allocate the lowest cost block found from the *Block* on Core *c*
15. **Break** //go to line 18
16. **End if**
17. **End For**
18. **If** (*Block* $\neq$ 0) //The input connection can be assigned
19. **Break** //go to line 22
20. **End If**
21. **End For**
22. **If** (*Block* $\neq$ 0) //The input connection can be assigned
23. Setup the connection
24. **Else**
25. Block the connection
26. **End If**

Algorithm 4: FASCA-with priority algorithm

According to Line 12 of the FASCA-with priority algorithm, the index of the first block that has the lowest cost is stored in *Block* and then in Line 13, it is checked that whether *Block* is not equal to 0, and then according to Line 14, the candidate block can be assigned to the connection. But if *Block* = 0, it means that no empty available block is found in that core and the next core must be checked. According to Line 22, the connection can be successfully set up if *Block* is not equal to 0. Otherwise, the connection will be blocked. The time complexity of FASCA with priority algorithm is O($n \times D \times C \times k$).

According to the flowchart of FASCA-with priority algorithm, shown in Fig. 7, firstly the priority of the request is specified in the candidate path and then the first assignable core is determined for that request based on its priority. After that, the empty blocks are searched according to the request and the cost of each one is calculated and the block with the lowest cost is selected for allocation. If no empty block is found, the next core is searched. Finally, if the request could not be allocated in the

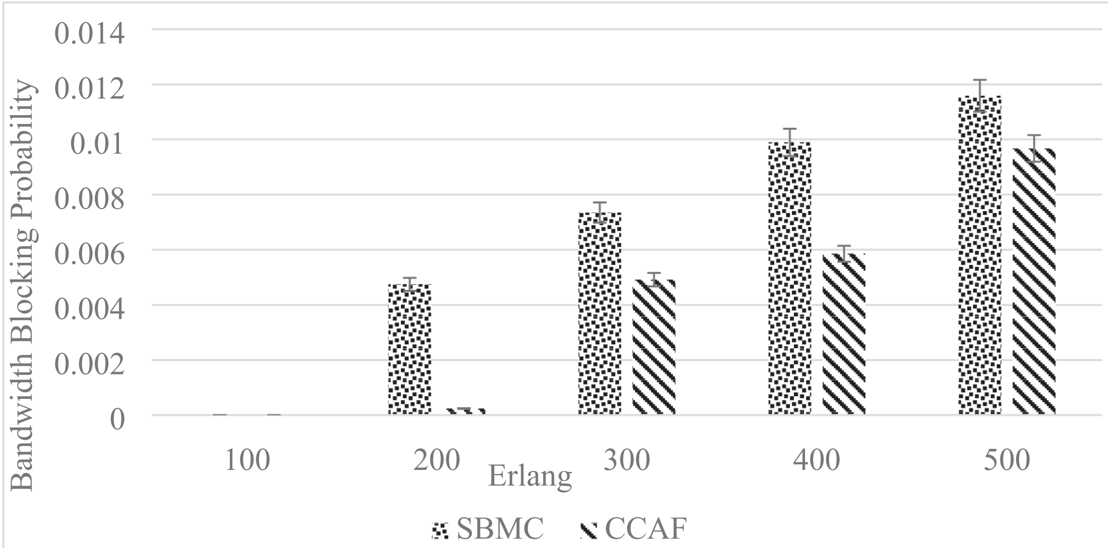

Fig.14. Bandwidth blocking probability on CCAF and SBMC under the NSFNET topology.

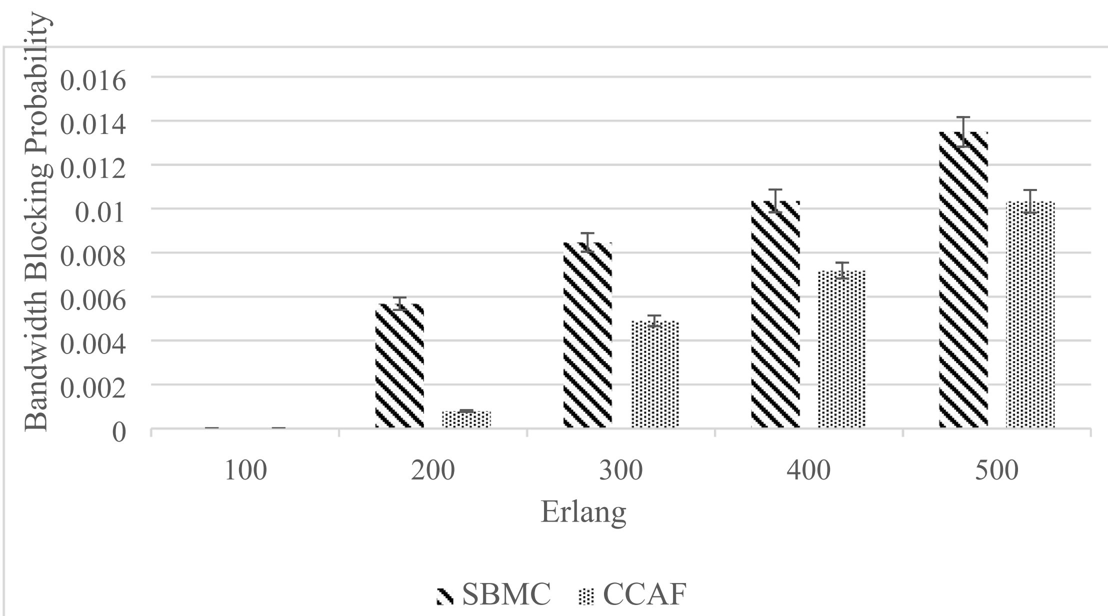

Fig.15. Bandwidth blocking probability on CCAF and SBMC under the JPN-12 topology.

C cores, the next candidate path would be searched, and if we could not allocate the request in the *k-th* path, the request would be blocked.

Example 5:. *Assume an arriving new connection needs* 50 *FSs and* $C = 7$. *Firstly, the priority type of the new request is checked. If the connection is a high-priority, first Core* 1 *is searched for allocation. If there is no available block in Core* 1 *for* 50 *FSs, Core* 2 *is searched and the same process continues until at least one available block is found until Core* 7. *For example, if there are two available blocks in Core* 3, *based on the LowestCostBlock* (*Path, Required_FS, Core*) *function, the block that causes the lowest cost is selected for assignment.*

If the connection priority is medium, first Core $\lceil log_2^C \rceil$ (i.e., Core 3 here) is searched for allocation. If there is no available block in Core 3 for 50 FSs, Core 4 is searched and the same process continues until at least one available block is found until Core 7. If at least one empty block is not suitable to the request in any of these five cores, the connection will be blocked.

If the connection priority is low, first Core $\lceil log_2^C \rceil + 1$ (i.e., Core 4) is searched for allocation. If there is no available block in Core 4 for 50 FSs, Core 5 is searched and the same process continues until at least one available block is found until Core 7. If at least one empty block is not suitable for the request in any of these four cores, the connection will be blocked. Otherwise, the connection can be successfully established.

4. Performance evaluation

In this section, the simulation results of the CCAF, FASCA-without priority and FASCA-with priority algorithms in the NSFNET and JPN-12 topologies are evaluated. The NSFNET topology with 14 nodes and 21 links and the JPN-12 topology with 12 nodes and 16 links are displayed in Fig. 8 and Fig. 9 (with distances in km), respectively. In these topologies, we assume $C = 7$ and we have $n = 300$ FSs on each core. The

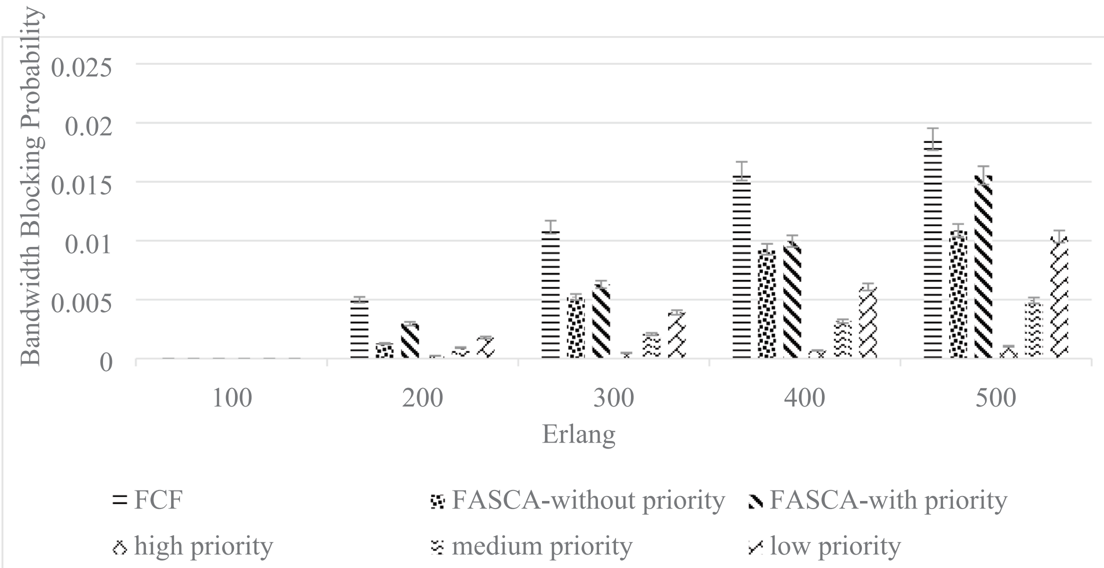

Fig.16. Bandwidth Blocking Probability of FCF, FASCA-with priority and FASCA-without priority under the NSFNET topology.

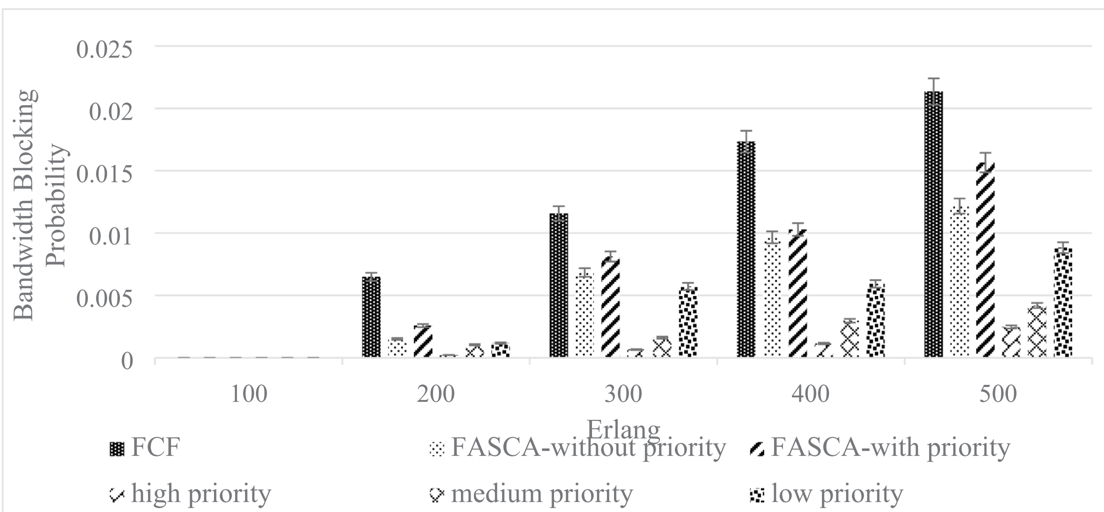

Fig.17. Bandwidth Blocking Probability of FCF, FASCA-with priority and FASCA-without priority under the JPN-12 topology.

bandwidth of each FS is considered as 12.5 GHz. The requested capacity of bandwidth is distributed from 12.5 Gbps to 237.5 Gbps randomly. Therefore, one fiber link can be provided the capacity of 3.75 THz.

The required slots for each connection are considered to be 80 to 100 FSs using the Uniform distribution. Network load in Erlang is obtained from the $\mu \times h$ which μ is the arrival rate and h is the mean holding time. It is assumed that the network load is 100 to 500 Erlangs. Note that for traffic loads smaller than 100 Erlangs, the blocking probability is almost zero. The results of the evaluated algorithms in Erlang 100 to 500 can be clearly compared and the best and worst algorithms can be identified. Connection holding time is also obtained using the Exponential distribution with mean 100 s. The arrival rate of connections is also based on the Exponential distribution. The k-shortest path algorithm is used to determine the shortest path between any source and destination pair nodes. Each point in the following diagrams is simulated 10 times and is averaged for 100,000 requested connections. In the diagrams, in order to evaluate the CCAF, FASCA-without priority and FASCA-with priority algorithms, the SBMC algorithm [9] and the FCF [18] algorithm have been used as our benchmark algorithms. For the FASCA-with priority algorithm, 20% of requests are generated with high-priority and 35% of requests are generated with medium-priority and 45% of requests are generated with low-priority. The performance evaluation results are depicted with 95% confidence intervals. The performance metrics are specified as follows:

- Blocking Probability (*BP*) is the ratio of the number of blocked connections to the total number arriving connections during the simulation time. The blocking probability is defined as Eq.(6).

$$BP = \frac{number\ of\ blocked\ connections}{total\ connections} \tag{6}$$

- Bandwidth Blocking Probability (*BBP*) is the ratio of the blocked FS requests to total number of requested FSs. Parameters *B* and *AR* are

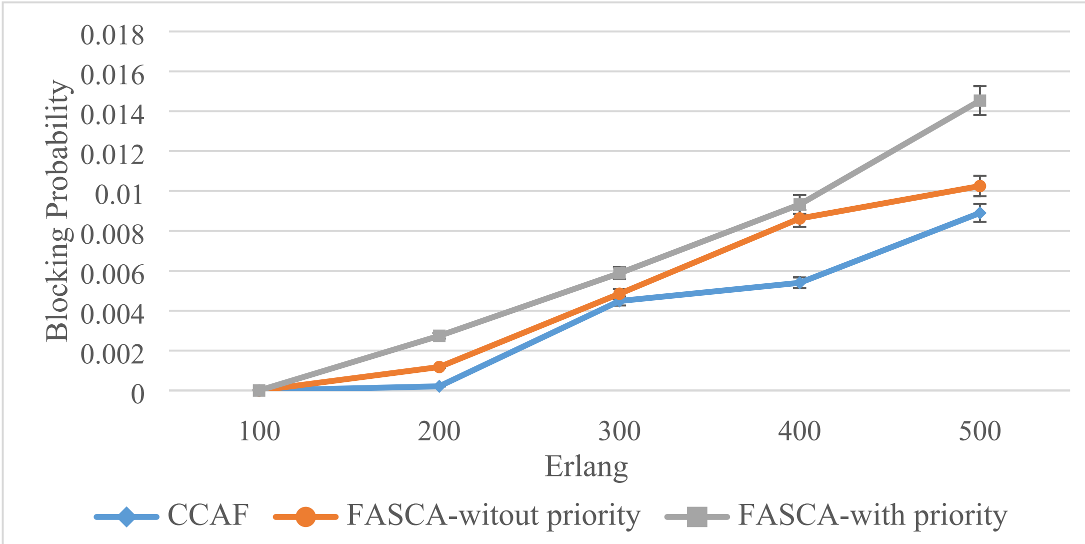

Fig.18. Blocking probability on algorithms CCAF, FASCA-with priority and FASCA-without priority under the NSFNET topology.

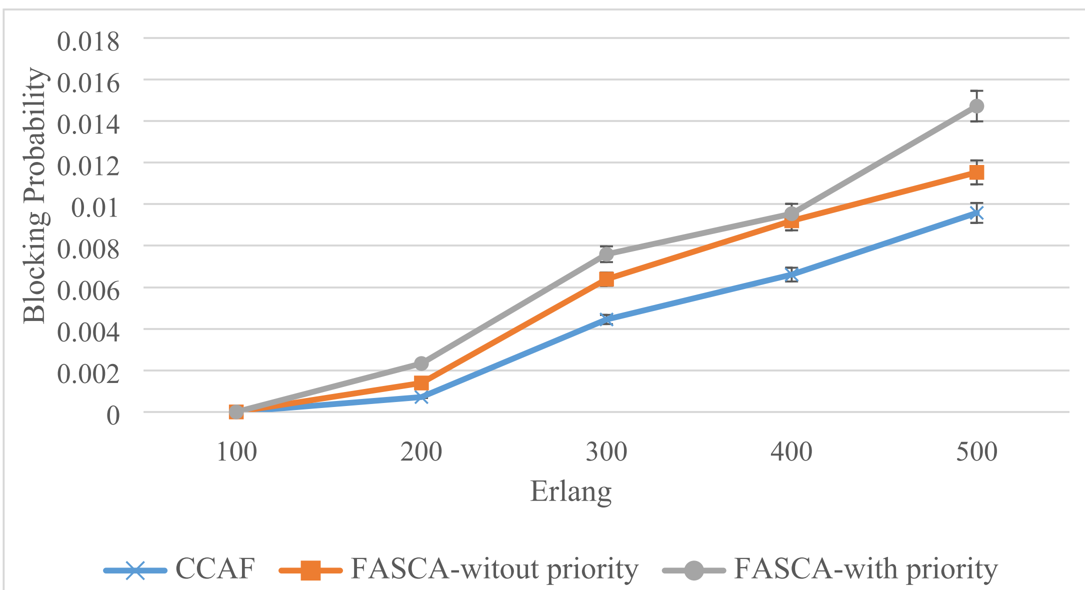

Fig.19. Blocking probability on algorithms CCAF, FASCA-with priority and FASCA-without priority under the JPN-12 topology.

set of blocked connections and arriving connections. Parameter b_i is the amount of bandwidth for connection *i*. The bandwidth blocking probability is defined as Eq.(7).

$$BBP = \frac{\sum_{i\in B} b_i}{\sum_{i\in AR} b_i} \tag{7}$$

- Spectrum Utilization (*SU*) is used to specify spectrum performance. Let *ATR* be the set of accepted connections and HT_i be holding time of connection *i*. The spectrum utilization is expressed by Eq.(8).

$$SU = \frac{\sum_{i\in ATR} b_i \times HT_i}{Total\ spectrum \times Simulation\ time} \tag{8}$$

Figs. 10 and 11 present the simulation results of the SBMC and our proposed CCAF algorithm under the NSFNET and JPN-12 topologies, respectively. According to the results, blocking probability for CCAF is smaller than SBMC in both topologies. Improving the blocking probability is due to the control of spectral fragmentation in the cores, which is possible to reduce the fragmentation of the spectrum by considering the block cost function to determine the appropriate spectral block for every connection request.

As it can be observed from the results, the blocking probability of both algorithms in JPN-12 is higher than in NSFNET because the average shortest path hop in JPN-12 is higher than in NSFNET, and therefore, majority of requests have to travel through more hops in JPN-12 than in NSFNET. Note that average shortest path hops in JPN-12 and NSFNET are 2.45 and 2.16, respectively.

Figs. 12 and 13 display the simulation results of FASCA-with priority and FASCA-without priority compared to the FCF algorithm under NSFNET and JPN-12, respectively. As it can be seen, the proposed FASCA-with priority and FASCA-without priority algorithms have the smallest blocking probability since it improves the blocking probability by controlling the fragmentation in the cores in both topologies. According to the results, the blocking probability for FASCA-without priority is smaller than for FASCA-with priority in both topologies. This reduction in blocking probability is due to the fact that in the FASCA-

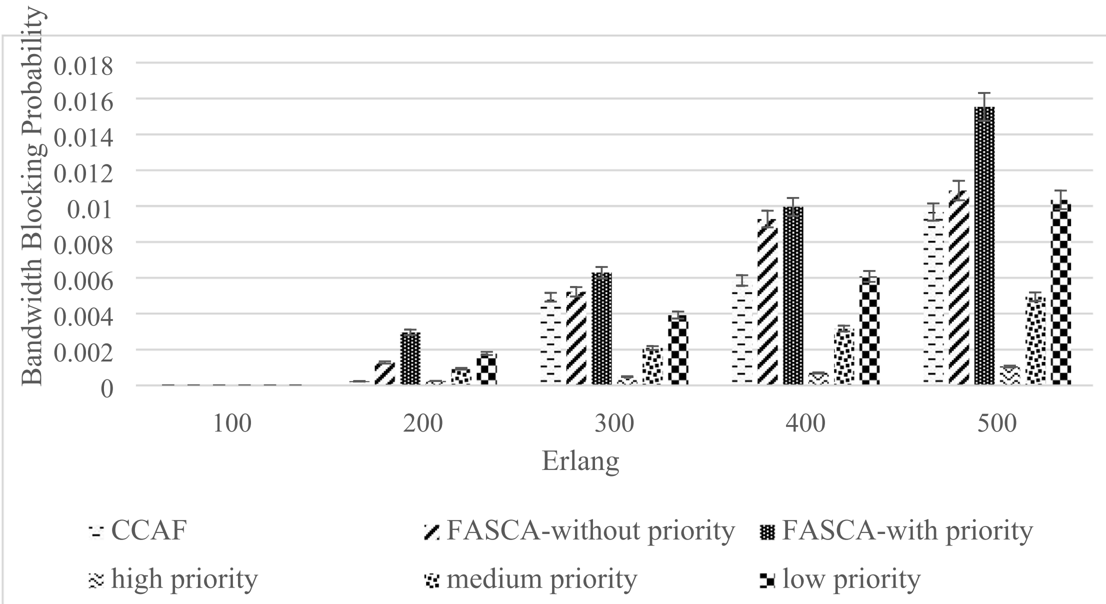

Fig.20. Bandwidth blocking probability of CCAF, FASCA-with priority and FASCA-without priority under the NSFNET topology.

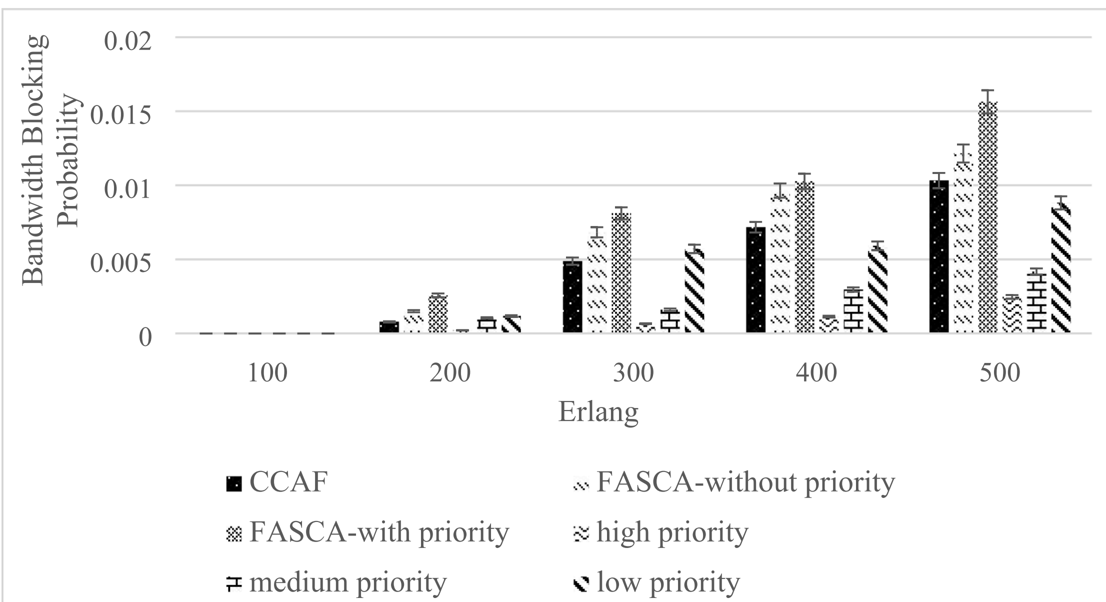

Fig.21. Bandwidth blocking probability of CCAF, FASCA-with priority and FASCA-without priority under the JPN-12 topology.

with priority, high-priority incoming requests are assigned first and then lower-priority requests. In this case, the blocking probability of low-priority requests increases. According to the results, the blocking probability for low-priority requests is higher than both medium-priority requests and high-priority requests, and the blocking probability for high-priority requests is smaller than medium-priority requests. According to results, the blocking probability of three proposed algorithms in JPN-12 topology is higher than in the NSFNET topology.

Figs. 14 and 15 illustrate the bandwidth blocking probability between CCAF and SBMC under NSFNET and JPN-12, respectively. Based on the results, the proposed CCAF algorithm has lower bandwidth blocking probability than SBMC in both topologies. By reducing the bandwidth blocking probability, this algorithm is more efficient in controlling fragmentation compared to SBMC. Improving the bandwidth blocking probability is due to the control of spectrum fragmentation in cores, which is possible to reduce the spectrum fragmentation by considering the block cost function to determine appropriate spectral block for connection request. In other words, in CCAF, the block that has the lowest cost is selected for allocation from the available blocks. However, in the SBMC algorithm, the first available block is selected for allocation to the input request, regardless of the spectrum fragmentation. According to the simulation results, the bandwidth blocking probability of two algorithms in JPN-12 is higher than in NSFNET.

Figs. 16 and 17 show a comparison of the bandwidth blocking probability of FASCA-with priority, FASCA-without priority and FCF algorithms under the NSFNET and JPN-12 topologies. As it can be observed, FASCA-with priority and FASCA-without priority have the smallest bandwidth blocking probability compared to SBMC in both topologies. This reduction in the bandwidth blocking probability is due to the consideration of the cost function of available blocks, where the

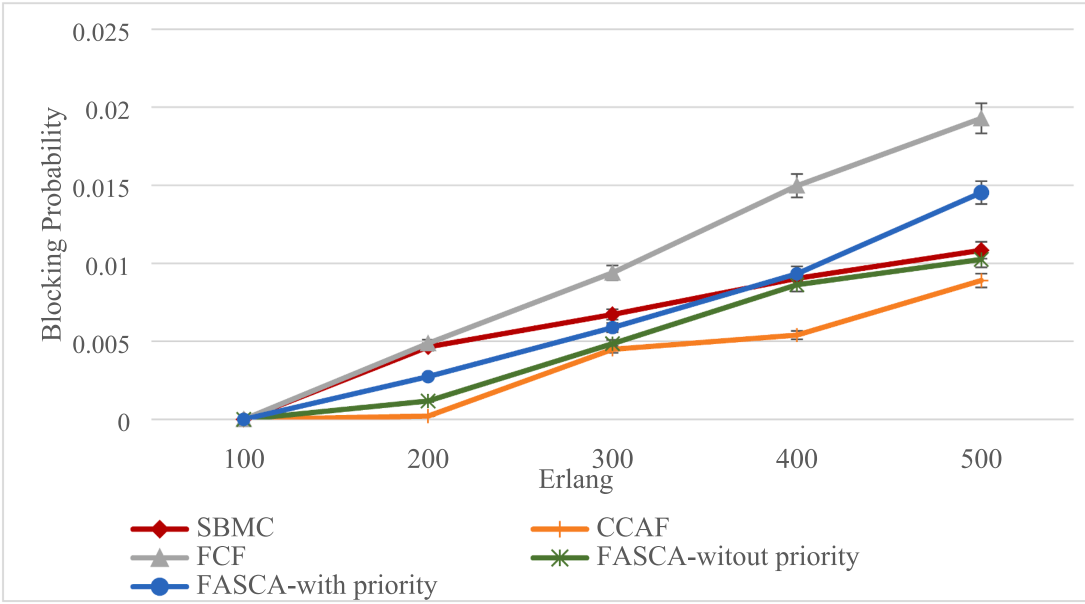

Fig.22. Blocking probability of CCAF, FASCA-with priority, FASCA-without priority, SBMC and FCF under the NSFNET topology.

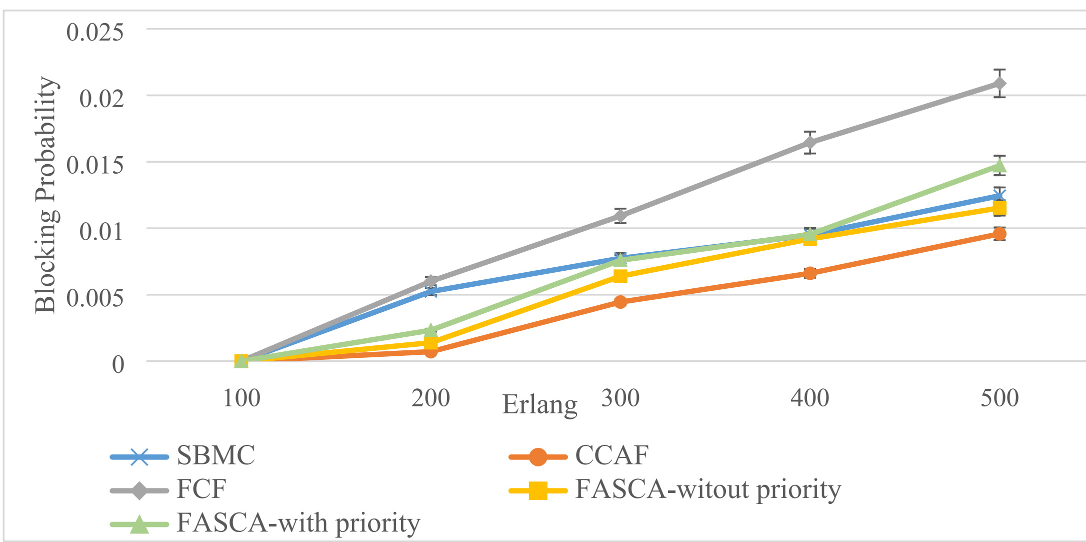

Fig.23. Blocking probability of CCAF, FASCA-with priority, FASCA-without priority, SBMC and FCF under the JPN-12 topology.

block with the lowest cost is selected for allocation. According to the results, bandwidth blocking probability in FASCA-without priority is smaller than in FASCA-with priority. This reduction in bandwidth blocking probability is due to the fact that in the FASCA-with priority algorithm, high-priority incoming requests are assigned first and then lower-priority requests. In this case, the bandwidth blocking probability of low-priority requests increases. According to the results, the bandwidth blocking probability for low-priority requests is higher than both medium-priority requests and high-priority requests, and the bandwidth blocking probability for high-priority requests is smaller than medium-priority requests. Based on the results, the bandwidth blocking probability of three algorithms in JPN-12 is higher than in NSFNET.

Fig. 18 and 19 compare the three proposed CCAF, FASCA-without priority and FASCA-with priority algorithms in this paper under NSFNET and JPN-12 topologies. As it can be seen from the results, the CCAF algorithm has lower blocking probability than the FASCA-without priority algorithm and the FASCA-with priority algorithm in both topologies. This reduction in the blocking probability is due to the consideration of core classification and the cost function of the available empty blocks. In other words, the block that causes the least fragmentation in the spectrum is preferred compared to other blocks. According to the results, blocking probability under FASCA-without priority is smaller than FASCA-with priority. This is because of the fact that in the FASCA-with priority algorithm, high-priority incoming requests are assigned first and then lower-priority requests. In this case, the blocking probability of low-priority requests increases. According to the results, the blocking probability of low-priority requests is higher than both medium-priority requests and high-priority requests, and the blocking probability of high-priority requests is smaller than medium-priority requests. Based on the results, the blocking probability of three proposed algorithms in the JPN-12 topology is higher than in the NSFNET topology.

Figs. 20 and 21 compare all three proposed CCAF, FASCA-without priority and FASCA-with priority algorithms in this paper in terms of bandwidth blocking probability under NSFNET and JPN-12 topologies, respectively. The CCAF algorithm has lower bandwidth blocking

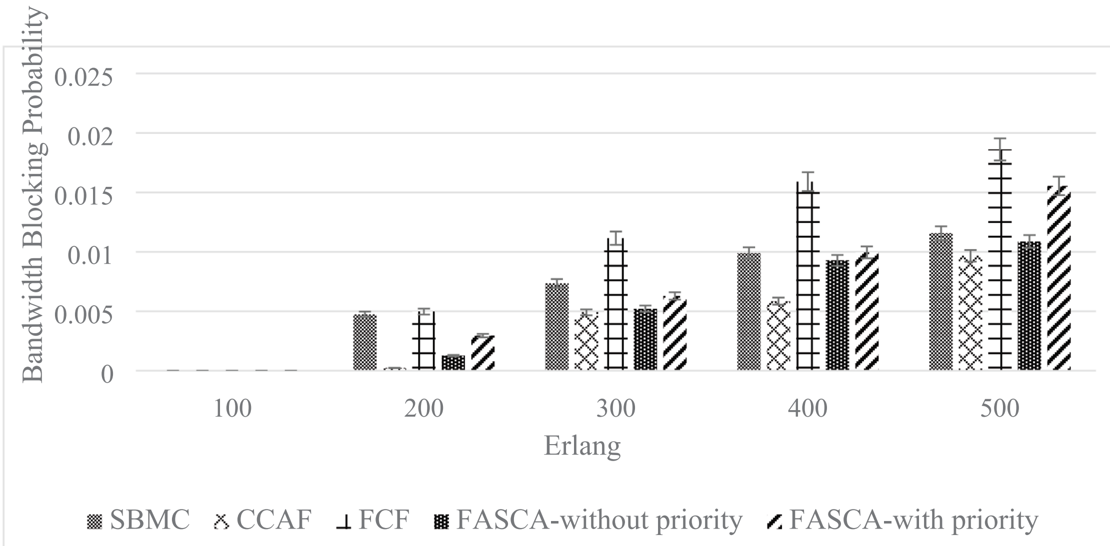

Fig.24. Bandwidth blocking probability of CCAF, FASCA-with priority, FASCA-without priority, SBMC and FCF under the NSFNET topology.

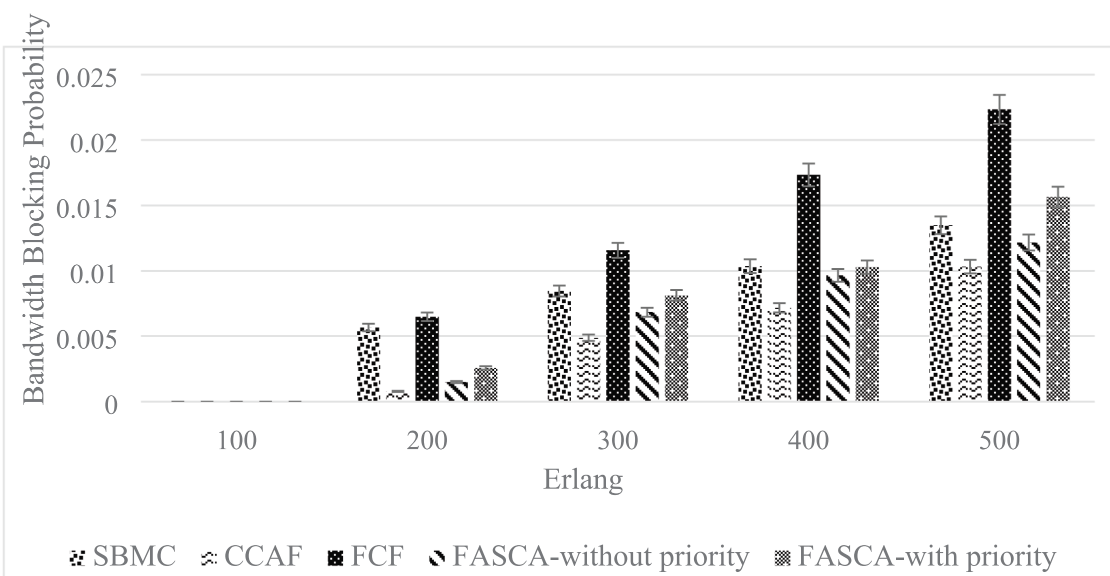

Fig.25. Bandwidth blocking probability of CCAF, FASCA-with priority, FASCA-without priority, SBMC and FCF under the JPN-12 topology.

probability than both FASCA-without priority and FASCA-with priority in both topologies. This reduction in bandwidth blocking probability is due to the consideration of the core classification. In all of the proposed algorithms, the block with the lowest cost is selected for allocation and the FASCA-without priority algorithm has smaller bandwidth blocking probability than FASCA-with priority. According to the simulation results, the bandwidth blocking probability of the three proposed algorithms in JPN-12 is higher than in NSFNET.

Figs. 22 and 23 show the blocking probability between the five algorithms under the NSFNET and JPN-12 topologies. Here, the FCF algorithm has the highest blocking probability and the proposed CCAF algorithm has the lowest blocking probability among the algorithms. The blocking probability of the benchmark SBMC algorithm is less than the FCF algorithm in both topologies. The FASCA-without priority algorithm has smaller blocking probability than FASCA-with priority. According to the results, the FASCA-with priority algorithm has a high blocking probability compared to the SBMC algorithm in Erlangs 400 and 500. On the other hand, in other traffic loads, the blocking probability of FASCA-with priority is lower than SBMC. The blocking probability of the algorithms in the JPN-12 topology is higher than in the NSFNET topology.

Figs. 24 and 25 compare all three proposed CCAF, FASCA-without priority and FASCA-with priority algorithms and two benchmark algorithms in terms of bandwidth blocking probability under NSFNET and JPN-12 topologies. The CCAF algorithm has lower bandwidth blocking probability than both FASCA-without priority and FASCA-with priority and the benchmark algorithms. This reduction in bandwidth blocking probability is due to the consideration of the core classification and block cost. Both CCAF and SBMC algorithms consider core classification, but the proposed CCAF algorithm has lower bandwidth blocking probability than SBMC. Improving the bandwidth blocking probability is due to the control of spectrum fragmentation in cores, which is possible to reduce the spectrum fragmentation by considering the block cost function to determine appropriate a spectral block for a given connection request. The FASCA-with priority algorithm has a high bandwidth blocking probability compared to the SBMC algorithm in traffic loads of

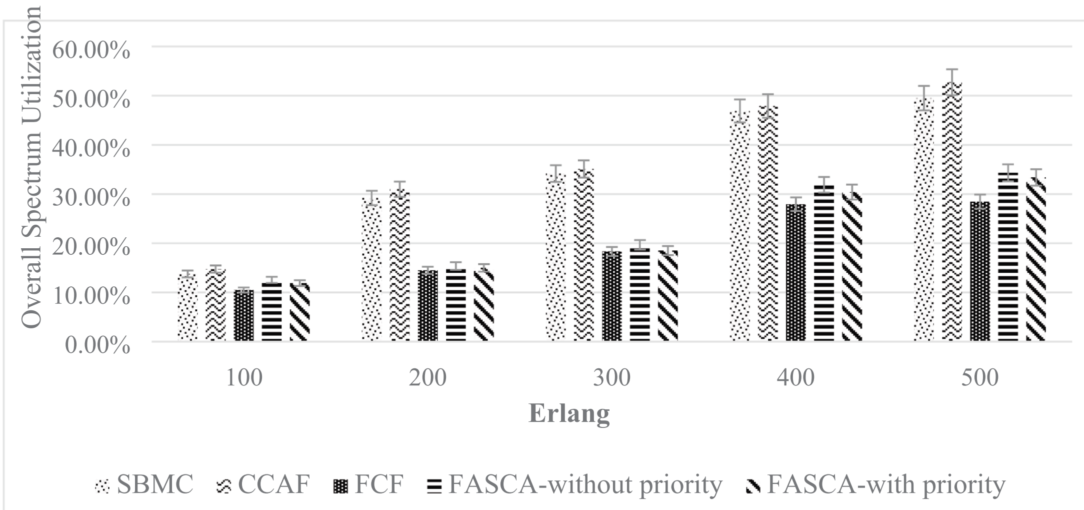

Fig.26. Spectrum utilization of CCAF, FASCA-with priority, FASCA-without priority, SBMC and FCF under the NSFNET topology.

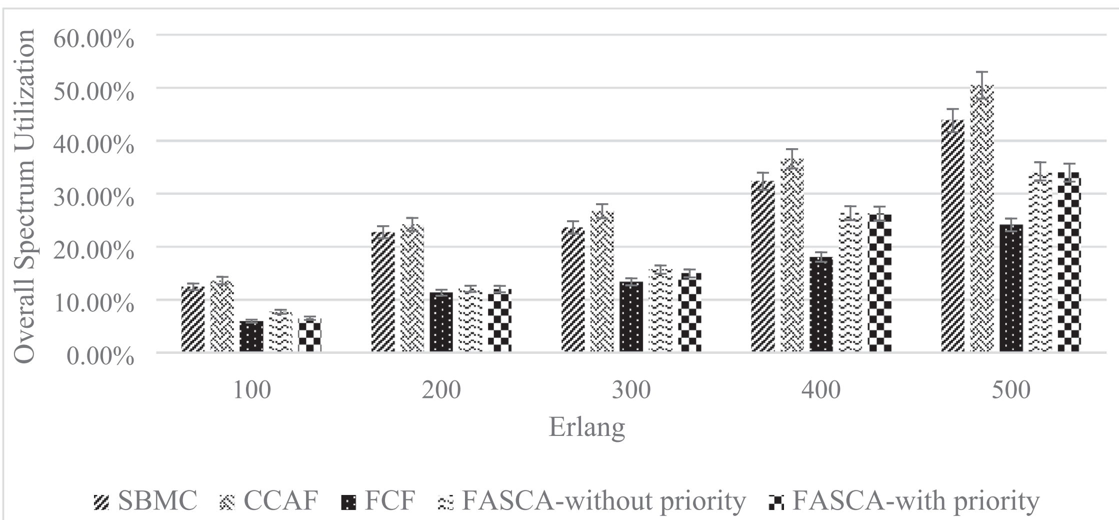

Fig.27. Spectrum utilization of CCAF, FASCA-with priority, FASCA-without priority, SBMC and FCF under the JPN-12 topology.

400 and 500 Erlangs, but the bandwidth blocking probability of FASCA-with priority algorithm is lower than SBMC algorithm in other traffic loads. Comparing the two topologies, it can be seen that NSFNET has a lower blocking probability than JPN-12.

Figs. 26 and 27 depict the overall spectrum utilization between the five algorithms under NSFNET and JPN-12. Here, spectrum utilization in CCAF is better than other algorithms, so it can be stated that this algorithm is the best algorithm among the three proposed CCAF, FASCA-without priority and FASCA-with priority algorithms. This is because the core classification method is considered in the CCAF algorithm. Comparing FASCA-with and the FASCA-without priority with each other, the FASCA-with priority has better performance under most of traffic loads than the other one.

5. Conclusion

One of the most important problems in SDM-EONs is spectrum fragmentation, which is caused by the presence of non-aligned slots or non-adjacent slots. Therefore, the fragmentation problem increases the blocking probability due to the scattering of the frequency slots. In this paper, we have tried to solve this problem by introducing CCAF, FASCA-with priority and FASCA-without priority. The CCAF algorithm reduces the blocking probability by considering the cost function to determine the appropriate block in the spectrum and core classification. The FASCA-with priority and FASCA-without priority algorithms reduce the blocking probability by considering the cost function. Finally according to the simulation results, CCAF is better than other FASCA-without priority and FASCA-with priority algorithms in terms of blocking probability and spectrum utilization. Comparing the FASCA-with and the FASCA-without priority with each other, the FASCA-with priority has better performance under most of traffic loads than the other one.

6. Future works

The algorithms presented in this paper can be expanded in some ways. Some of the development strategies are as follows:

- Modulation level can be considered to optimize the number of requested FS.
- Load Balancing can be considered in the SDM-EONs.
- Since another important issue in optical networks is energy consumption, the proposed algorithms in this paper can be examined in terms of energy consumption.

- Survivability and link failure issues can be considered for the proposed algorithms.

CRediT authorship contribution statement

Elham Pourkarimi: Conceptualization, Methodology, Software, Writing – original draft. **Akbar Ghaffarpour Rahbar:** Writing - review & editing.

Declaration of Competing Interest

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

References

[1] E.d. Harstead, R. Sharpe, Forecasting of access network bandwidth demands for aggregated subscribers using Monte Carlo methods, IEEE Communications Magazine 53 (3) (2015) 199–207.
[2] B.C. Chatterjee, S. Ba, E. Oki, Fragmentation problems and management approaches in elastic optical networks: a survey, IEEE Communications Surveys & Tutorials 20 (1) (2018) 183–210.
[3] E. OKI, N. WADA, S. OKAMOTO, N. YAMANAKA, K.-I. SATO, Optical networking paradigm: Past, recent trends and future directions, IEICE Transactions on Communications E100.B (9) (2017) 1564–1580.
[4] B.C. Chatterjee, et al., Routing and Wavelength Assignment for WDM-based Optical Networks:Quality-of-Service and Fault Resilience, Springer 410 (2016).
[5] Y. Liu, R. He, S. Wang, C. Yu, Temporal and Spectral 2D Fragmentation-Aware RMSA Algorithm for Advance Reservation Requests in EONs, IEEE Access 9 (2021) 32845–32856.
[6] B.C. Chatterjee, N. Sarma, E. Oki, Routing and spectrum allocation in elastic optical networks: A tutorial, IEEE Communications Surveys & Tutorials 17 (3) (2015) 1776–1800.
[7] L. Bonani, M. Forghani-elahabad, M. Abbade, Network Fragmentation Measure in Elastic Optical Networks, IEEE, Sao Paulo, Brazil, 2019.
[8] F. Shirin Abkenar, A. Ghaffarpour Rahbar, Study and analysis of routing and spectrum allocation (RSA) and routing, modulation and spectrum allocation (RMSA) algorithms in elastic optical networks (EONs), Optical Switching and Networking 23 (2017) 5–39.
[9] F. Yousefi, A.G. Rahbar, Novel fragmentation-aware algorithms for multipath routing and spectrum assignment in elastic optical networks-space division multiplexing (EON-SDM), Optical Fiber Technology 46 (2018) 287–296.
[10] F. Yousefi, A.G. Rahbar, M. Yaghubi-Namaad, Fragmentation-aware algorithms for multipath routing and spectrum assignment in elastic optical networks, Optical Fiber Technology 53 (2019), 102019.
[11] P.M. Moura, N.L. da Fonseca, R.A. Scaraficci, Fragmentation aware routing and spectrum assignment algorithm, IEEE, Sydney, NSW, Australia, 2014.
[12] M. Klinkowski, P. Lechowicz, K. Walkowiak, Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking, Optical Switching and Networking 27 (2018) 58–78.
[13] Y. Awaji, J. Sakaguchi, B.J. Puttnam, R.S. Luís, J.M.D. Mendinueta, W. Klaus, N. Wada, High-capacity transmission over multi-core fibers, Optical Fiber Technology 35 (2017) 100–107.
[14] S. Hanczewski, M. Stasiak, J. Weissenberg, Non-full-available queueing model of an EON node, Optical Switching and Networking 33 (2019) 131–142.
[15] H. Xuan, et al., Routing, spectrum and core assignment for multi-domain elastic optical networks with multi-core fibers, Optical Fiber Technology 59 (2020), 102040.
[16] M. Jafari-Beyrami, A.G. Rahbar, S. Hosseini, On-demand fragmentation-aware spectrum allocation in space division multiplexed elastic optical networks with minimized crosstalk and multipath routing, Computer Networks 181 (2020), 107531.
[17] S. Paira, J. Halder, M. Chatterjee, U. Bhattacharya, On Energy Efficient Survivable Multipath Based Approaches in Space Division Multiplexing Elastic Optical Network: Crosstalk-Aware and Fragmentation-Aware, IEEE Access 8 (2020) 47344–47356.
[18] F. Yousefi, A.G. Rahbar, A. Ghadesi, Fragmentation and Time Aware Algorithms in Spectrum and Spatial Assignment for Space Division Multiplexed Elastic Optical Networks (SDM-EON), Computer Networks 174 (2020), 107232.
[19] F. Yousefi, A.G. Rahbar, Novel crosstalk, fragmentation-aware algorithms in space division multiplexed-Elastic Optical Networks (SDM-EON) with considering physical layer security, Optical Switching and Networking 37 (2020), 100566.
[20] A. Athari Beyragh, A. Ghaffarpour Rahbar, S.-M. Hosseini Ghazvini, M. Nickray, IF-RSCA: intelligent fragmentation-aware method for routing, spectrum and core assignment in space division multiplexing elastic optical networks (SDM-EON), Optical Fiber Technology 50 (2019) 284–301.
[21] R. Zhu, et al., Protected Resource Allocation in Space Division Multiplexing-Elastic Optical Networks with Fluctuating Traffic, Journal of Network and Computer Applications 174 (2021), 102887.
[22] Y. Zhao, L. Hu, R. Zhu, X. Yu, Y. Li, W. Wang, J. Zhang, Crosstalk-aware spectrum defragmentation by re-provisioning advance reservation requests in space division multiplexing enabled elastic optical networks with multi-core fiber, Optics express 27 (4) (2019) 5014, https://doi.org/10.1364/OE.27.005014.
[23] R. Zhu, A. Samuel, P. Wang, S. Li, L. Li, P. Lv, M. Xu, Survival Multipath Energy-aware Resource Allocation in SDM-EONs during Fluctuating Traffic, Journal of Lightwave Technology 39 (7) (2021) 1900–1912.
[24] J. Yuan, R. Zhu, Y. Zhao, Q. Zhang, X. Li, D.i. Zhang, A. Samuel, A spectrum assignment algorithm in elastic optical network with minimum sum of weighted resource reductions in all associated paths, Journal of Lightwave Technology 37 (21) (2019) 5583–5592.
[25] M.A. Cavalcante, H.A. Pereira, D.A.R. Chaves, R.C. Almeida, Optimizing the cost function of power series routing algorithm for transparent elastic optical networks, Optical Switching and Networking 29 (2018) 57–64.
[26] A. Muhammad, G. Zervas, R. Forchheimer, Resource allocation for space-division multiplexing: Optical white box versus optical black box networking, Journal of Lightwave Technology 33 (23) (2015) 4928–4941.
[27] Y. Hirota, et al., Dynamic spectrum allocation based on connection alignment for elastic optical networks, IEEE, Colombo, Sri Lanka, 2015.
[28] Xu, Y. and Y.-C. Kim. Dynamic routing and spectrum allocation to minimize fragmentation in elastic optical networks. in 2017 20th Conference of Open Innovations Association (FRUCT). 2017. St. Petersburg, Russia: IEEE.
[29] Y. Zhao, L. Hu, C. Wang, R. Zhu, X. Yu, J. Zhang, Multi-core virtual concatenation scheme considering inter-core crosstalk in spatial division multiplexing enabled elastic optical networks, China Communications 14 (10) (2017) 108–117.