Example:

Having just 4 optical channels per transmission system, each with a capacity of 100 Gbps, and the following set of demands.

ODU4 (100 Gbps)

0	0	0	0	0	0
0	**0**	0	9	0	0
0	0	**0**	0	0	0
0	9	0	**0**	0	0
0	0	0	0	**0**	0
0	0	0	0	0	**0**

The demands are first ordered considering two criteria's (type of ODU and length of the path) in a single matrix. Then using a fixed-alternate routing algorithm it is chosen the K-shortest paths which will possibly be used to route all demands.

In this example:

Scheduler

Matrix of demands processed (after the scheduler)		
Index	ODU type	Hops of the shortest path
1	4	1
2	4	1
3	4	1
4	4	1
5	4	1
6	4	1
7	4	1
8	4	1
9	4	1

Demand 1

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 1	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The first candidate path found in this case will be the shortest path for demand 1. It passes all the criteria once we have available capacity and grooming capacity in that path.

Grooming vector

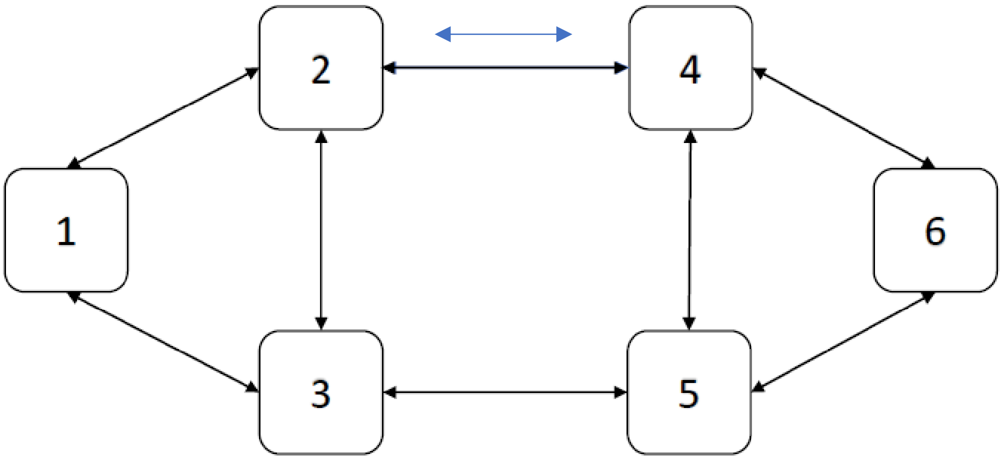
Initial vector:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4	-	-	-	-	4	-

After grooming the first demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4	-	-	-	-	3	1

The first demand is routed through link 2,4.



Demand 2

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 1	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The first candidate path found in this case will be the shortest path for demand 2. It passes all the criteria once we have available capacity and grooming capacity in that path.

Grooming vector

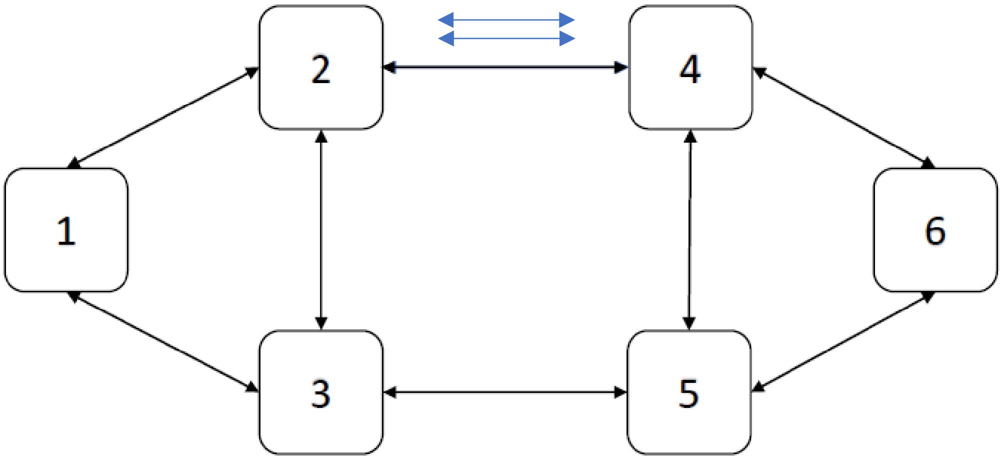
Initial vector:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4		-	-	-	3	1

After grooming the demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4		-	-	-	2	1,2

Demand routed through link 2,4.



Demand 3

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 3	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The first candidate path found in this case will be the shortest path for demand 3. It passes all the criteria once we have available capacity and grooming capacity in that path.

Grooming vector

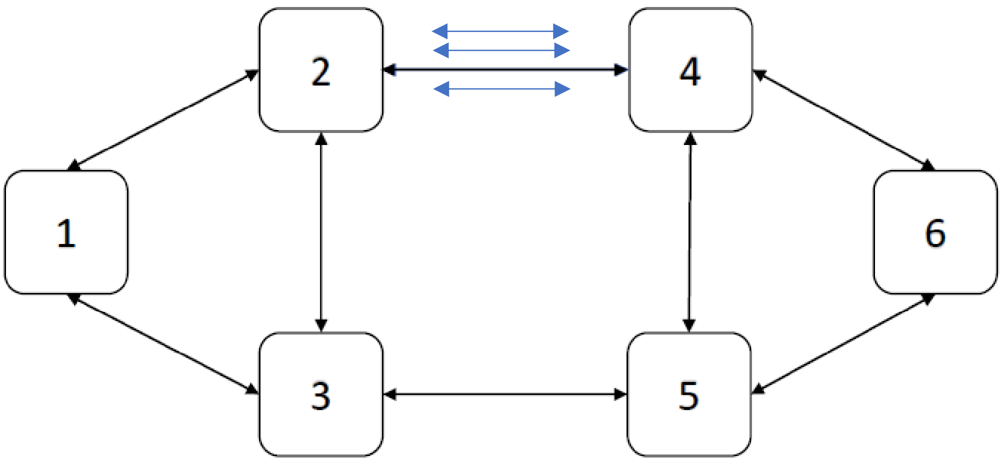
Initial vector:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4		-	-	-	2	1,2

After grooming the demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4		-	-	-	1	1,2,3

Demand routed through link 2,4.



Demand 4

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 4	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The first candidate path found in this case will be the shortest path for demand 4. It passes all the criteria once we have available capacity and grooming capacity in that path.

Grooming vector

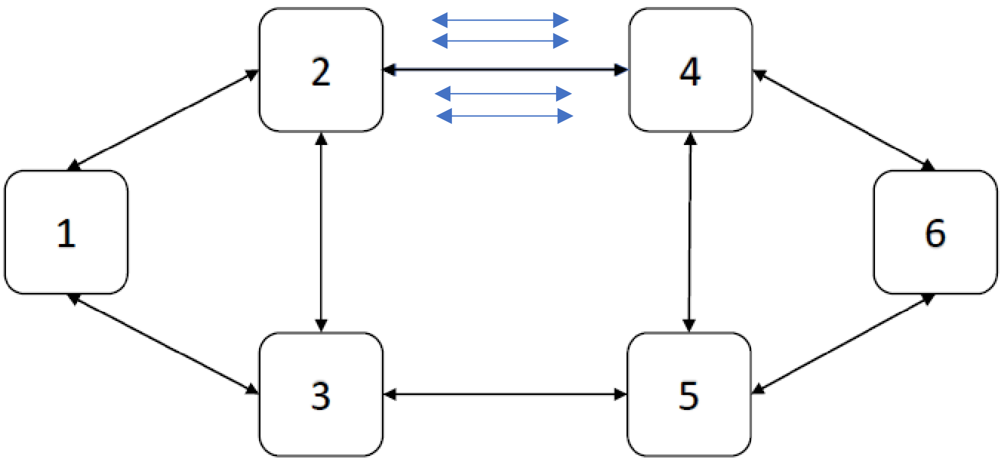
Initial vector:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4		-	-	-	1	1,2,3

After grooming the first demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4		-	-	-	0	1,2,3,4

Demand routed through link 2,4.



Demand 5

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 4	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The following demand (5) can’t use the first path because it contains a full link (see grooming vector for link (2,4) and as so it will be temporary eliminated for the purpose of routing demands of the same ODU type. So, we will try the next shortest path of the matrix for this demand.

Grooming vectors

Initial vectors:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,4	-	-	-	-	0	1,2,3,4

As we can see above the remaining capacity of this link for ODU4 demands is 0 so this link will be blocked for any other demands in this case.

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	4	-

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	4	-

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	4	-

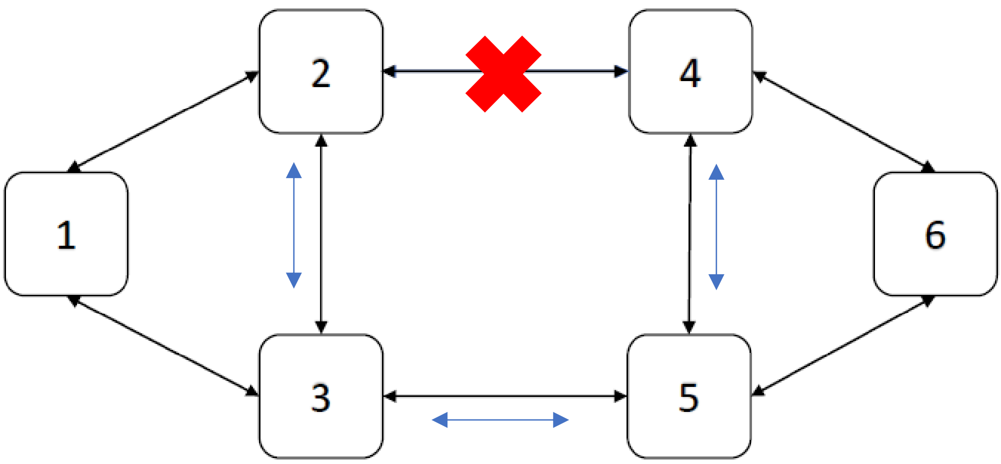
After grooming the demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	3	5

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	3	5

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	3	5

Demand routed through links (2,3) (3,5) and (5,4).



Demand 6

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 4	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The following demand (6) can't use the first path because it contains a full link (see grooming vector for link 2,4) and as so it was temporary eliminated for the purpose of routing demands of the same ODU type, in the previous iteration. So, we will try the next shortest path of the matrix for this demand which we can see below in green.

Grooming vectors

Initial vectors:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	3	5

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	3	5

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	3	5

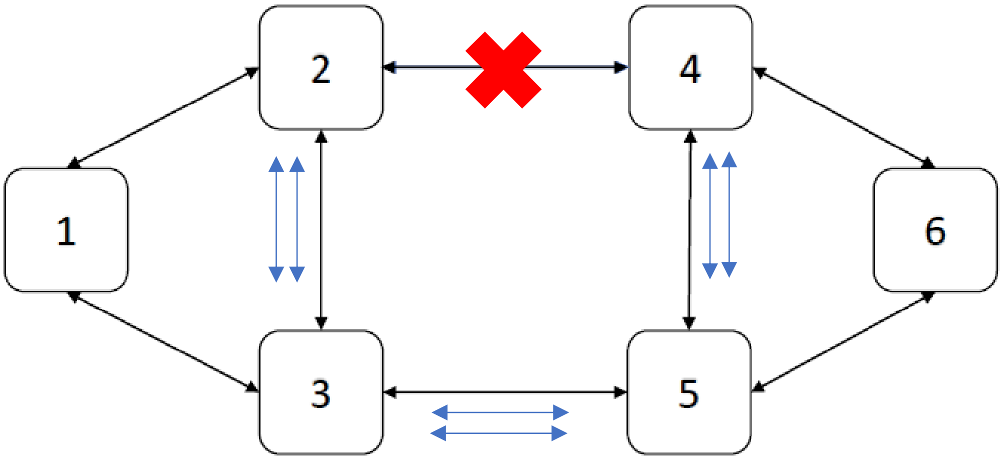
After grooming the demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	2	5,6

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	2	5,6

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	2	5,6

Demand routed through links (2,3) (3,5) and (5,4).



Demand 7

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 4	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The following demand (7) can't use the first path because it contains a full link (see grooming vector for link 2,4) and as so it was temporary eliminated for the purpose of routing demands of the same ODU type. So, we will try the next shortest path of the matrix for this demand which we can see below in green.

Grooming vectors

Initial vectors:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	2	5,6

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	2	5,6

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	2	5,6

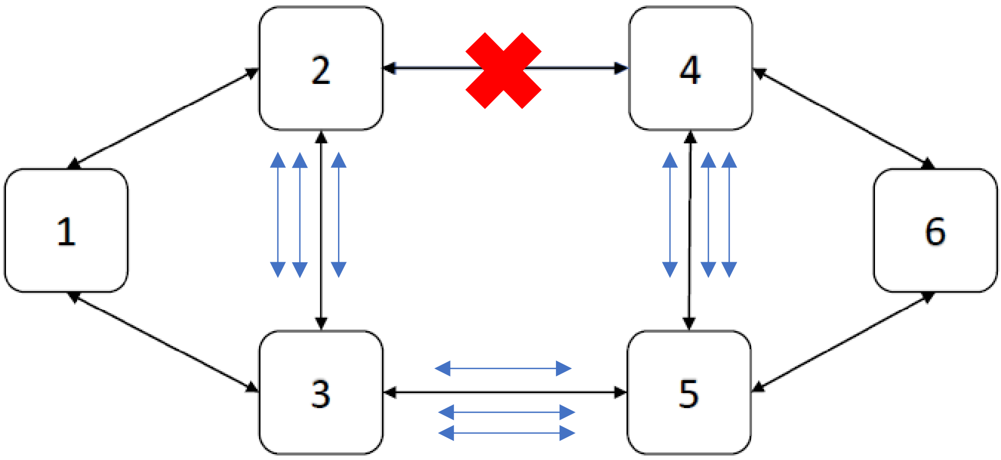
After grooming the demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	1	5,6,7

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	1	5,6,7

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	1	5,6,7

Demand routed through links (2,3) (3,5) and (5,4).



Demand 8

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 4	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

The following demand (8) can't use the first path because it contains a full link (see grooming vector for link 2,4) and as so it was temporary eliminated for the purpose of routing demands of the same ODU type. So, we will try the next shortest path of the matrix for this demand which we can see below in green.

Grooming vectors

Initial vectors:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	1	5,6,7

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	1	5,6,7

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	1	5,6,7

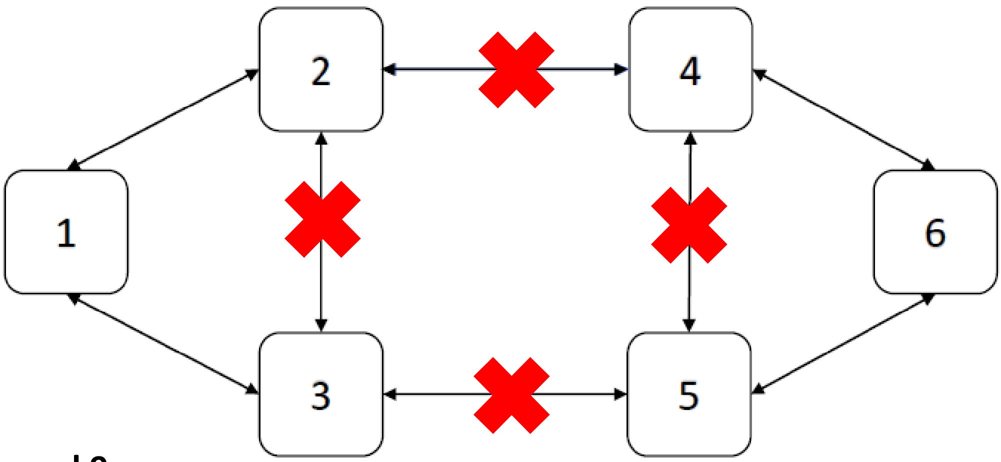
After grooming the demand:

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
2,3	-	-	-	-	0	5,6,7,8

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
3,5	-	-	-	-	0	5,6,7,8

Link	ODU0	ODU1	ODU2	ODU3	ODU4	Demands
5,4	-	-	-	-	0	5,6,7,8

Demand routed through links (2,3) (3,5) and (5,4). As we can see in these vectors the remaining capacity to aggregate ODU4 demands is zero which means that those links will be blocked in a future iteration where those links are needed. That's what is going to happen in demand 9.



Demand 9

Fixed-alternate routing algorithm

3-shortest paths matrix for demand 4	
Path (Links)	Number of hops
(2,4)	1
(2,3);(3,5);(5,4)	3
(2,1);(1,3);(3,5);(5,4)	4

Finally, we reach the last demand (9), and as all the paths are already completely full or at least one of the links of the path there won't be any candidate paths which means that blocking will occur, and this demand won't be routed.

Blocked demands matrix	
Index	9

Each time a demand is routed it is stored in an array which later will be used in the wavelength assignment process. After all of the demands have been processed this array which contains all of the routed demands must be sorted according to two criteria's, the type of ODU and the number of hops in its path. So below we have the final sorted matrix:

Index	ODU type	Number of hops
5	4	(2,3);(3,5);(5,4) = 3
6	4	(2,3);(3,5);(5,4) = 3
7	4	(2,3);(3,5);(5,4) = 3
8	4	(2,3);(3,5);(5,4) = 3
1	4	(2,4) = 1
2	4	(2,4) = 1
3	4	(2,4) = 1
4	4	(2,4) = 1

Wavelength assignment

In this case once we 4 wavelengths available per link and also 4 demands routed in each of those links there isn´t going to appear any difficulties.

Wavelength assignment	
Index of demand	Wavelength
5	1
6	2
7	3
8	4
1	5
2	6
3	7
4	8

Once all the demands are routed and a wavelength is attributed then we finish our algorithm presenting all the information in a Cost Report.