

# An Algorithm for Dynamic Routing and Wavelength Assignment in WDM Network

Yun Wu, Songming Xu\*

Hainan College of Software Technology, China

114004519@qq.com,108743696@qq.com\*

**Keywords:** wavelength; transmitters' installment; degree constrained; improvement Dijkstra

**Abstract.** In WDM network, it is still a core issue to research routing. According to the characteristics of multi wavelength optical network and the setting of the wavelength converter, we change multi wavelength network to single wavelength dynamic network through the conversion function of wavelength converter, and put forward an equivalent model of degree-constrained shortest path calculation. Through the analysis of the model, we use the method of adding the critical point set and adding the virtual node, put forward an improved algorithm based on the the Dijkstra shortest path algorithm. The time complexity of the Dijkstra algorithm is  $O(n^5)$ , and that of the improved algorithm is  $O(n^3)$ . The time complexity reduces greatly. This improved algorithm can allocate network resource more reasonably, improve the utilization rate of load-balanced links, and save the time of data transmission and the resource of link.

## 1 Introduction

Only when different wavelengths of light-wave are propagated in the same optical fiber, The potential of optical fibers is really being tapped. With the network traffic keeping increasing exponentially and rapidly, people has a higher expect for network bandwidth. The WDM (Wavelength Division Multiplexing) network can transmit multi optical signals in a single optical fiber, and is considered to be the most potential optical network in solving bandwidth issue in next generation.

The RWA(Routing and Wavelength Assignment) is one of the core issues in network performance optimization. RWA is used to find a route between source node and destination node and assign wavelength when there is a request for establishment of a light path between the nodes in network. The RWA problem is a NPC(Nondeterministic Polynomial Complete) problem, for the convenience of research, so the RWA problem is usually split into routing issue and wavelength assignment issue to solve respectively. The RWA problem that optical path connection request is dynamic and random is referred to dynamic RWA problem. The performance index of the dynamic RWA problem is mainly the network blocking rate. Now the two issues can be considered together through wavelength converter.

In this paper, we research the RWA problem that the number of node wavelength converters is limited, and then put forward a WDM network model based on the number-limited wavelength conversion, and a dynamic RWA algorithm – an improved Dijkstra algorithm.

The all-optical network signal is only converted between electrical and optical when enters and leaves network, but exists in the form of light in transmitting and exchange process. Without electrical operation, optical signal has advantage of high bandwidth, good compatibility, scalable, reconfigurable, simple structure and high reliability. The PDH, SDH, ATM transmission can be used and improve the utilization of network resources. But due to the limitation of the technology,

the number of wavelengths that can be accommodated on a single optical fiber is limited, so utilization rate of wavelength is a very critical problem in the all-optical networks.

In order to improve wavelength utilization, wavelength converter is added in the switching nodes, so switching node can convert a wavelength to another through wavelength converter. The ability of wavelength converter is stronger, the wavelength utilization rate is higher. Research shows that when the wavelength of the switching node can be arbitrarily converted, the wavelength utilization rate is the highest. But the ability of conversion is stronger, technical requirement and cost are higher, so the current conversion range of wavelength converter is limited, they can only convert input wavelength to several specific output wavelength. In order to save cost, the number of wavelength converter is designed as few as possible for node. The limit on number of wavelength converter inevitably leads to increase of blocking probability, so we need to research the routing problem when the number of wavelength converter is limited, in order to reduce the impact of link bottleneck cause by lack of wavelength, and to improve the performance of network.

## **2 The limited number of wavelength conversion WDM network model and degree limitation network model**

Current WDM network based on wavelength conversion is divided into single hop and multi hop system, in single hop system, the wavelength of all links are the same, multi hop system due to the introduction of wavelength converters, the source to destination node of the optical channel can use multiple different wavelengths of optical path combination. But due to the wavelength converter technology is not very mature, and the price is expensive, so in actual use, we need to balance the number of wavelength converters in network settings.

The essence of wavelength converter is built to a wavelength path (WP) or virtual wavelength path (VWP) between the two routing nodes, the channel depend on wavelength converter to establish, due to the limited number of wavelength converters, so establishment of the channel is bound to be limited.

Based on the wavelength conversion interface restriction WDM network topology structure is equivalent to the degree restriction network topology structure:

Routing node is the note of degree-restricted network, the number of wavelength converters of routing nodes is the number of nodes in degree-restricted network, the established WP and VWP through a wavelength converter the actual existence of the edge in degree-restricted network, the WP and VWP can be established through a wavelength converter is a potential edge of restriction network. To find a route from source  $s$  to the end of  $t$  in WDM network is equivalent to the shortest path from the  $s'$  to  $t'$  in degree-restricted network. ( $s'$  is the equivalent node  $s$  in the degree-restricted network,  $t'$  is the equivalent node  $t$  in the degree-restricted network)

We first define the concept of the degree-restricted network: there are two classes of edges in network, one class is the actual edge, the other is a potential edge, each node in the network has the restricted degree, that is to say, the edge linked by each point cannot be connected to exceed this limit. We can add the potential edges to the network, but we have to ensure the constraints of node satisfaction degree.

Use mathematical symbols to define the network:

$E$  represents edge set;

$V$  represents the original node set;

$S$  represents the actual edge set;

$R$  represents the potentially additive edge set;

$D$  represents the maximum number of edges that can be connected to the nodes, namely the limitation of degree;

$\delta_i$  represents the maximum number of edges that can be added in  $S$  set.

To clear the meaning of:

a) If  $V_i$  has been connected to the edge of  $S$  concentration are equal to  $D-1$ , then  $\delta_i = 1$ ;

b) If  $V_i$  has been connected to the edge of  $S$  concentration are less than or equal to  $D-2$ , then  $\delta_i = 2$ ;

c) If  $V_i$  has been connected to the edge of  $S$  concentration are equal to  $D$ , then  $\delta_i = 0$ .

The job we have to do now is to find the shortest path from  $s'$  to  $t'$  in this degree restricted network.

### 3 Related Work

This problem has been studied by previous researchers, and on the basis of a large quantity of studies, we propose the minimum weight perfect matching to solve this problem.

#### Algorithm Description

First of all, building new graph  $G'$ , for  $\forall V_x \in V - \{V_s, V_t\}$ , construct  $V_{x_1}$  and  $V_{x_2}$ , add an edge of zero weight between  $V_{x_1}$  and  $V_{x_2}$ , construct two nodes for each edge of  $E_{xy} \in E$  (called side nodes), as well as add an edge of zero weight between them. We add an edge from  $V_{E_{xy}}$  to  $V_{x_1}$  and  $V_{x_2}$ , weight is  $W_{xy}/2$ , we also add an edge from  $V_{E_{xy}}$  to  $V_{y_1}$  and  $V_{y_2}$ , and the weight of the edge is also  $W_{xy}/2$ . When  $V_x = V_s$  or  $V_x = V_t$ , we only add an edge from  $V_{E_{xy}}$  to  $V_x$ . for all potential edge, if  $\delta_x = 1$ , we delete the edges from  $V_{E_{xy}}$  to  $V_{x_1}$  and also the edges from  $V_{E_{xy}}$  to  $V_{x_2}$ .

Finally, for the graph  $G'$ , we can get the shortest path from  $V_s$  to  $V_t$  by using the minimum weight perfect matching algorithm which is proposed by Edmonds in 1970s. The algorithm complexity is  $O(n^5)$ .

### 4 Dijkstra improved algorithm analysis

Due to the algorithm based on matching is relatively complicated, in terms of the characteristic of solving the problem for the shortest route, we proposed a modified version of the shortest path algorithm based on Dijkstra, and its time complexity is  $O(n^3)$ .

#### Algorithm analysis

In the Dijkstra algorithm, the shortest path is calculated by setting the label. The algorithm divides the nodes into two groups: calibration of permanent and temporary. Every time the algorithm selects the minimum distance value from temporary calibration group to add to the permanent calibration group, and update the node labeling in the temporary calibration group. Algorithm will not finish until all nodes are involved in the permanent calibration group.

(1)  $\delta_x = 2$ , the problem is to solve the general shortest path. The Dijkstra algorithm can be used to solve this problem;

(2)  $\delta_x = 0$ , the problem is the same as (1);

(3)  $\delta_x = 1$ , The problem is no longer as simple as (1) or (2). Now assume that the node is  $V_x$ , according to the edge of the refresh can be divided into two cases:

If update to  $V_x$  by the actual edge, we can certainly update the nodes connecting with  $V_x$  via  $V_x$ , because in the shortest path, it can connect with a potential edge through  $V_x$ , and the degree will not exceed the restriction of  $D$ ;

If update to  $V_x$  by the potential edge, we cannot update the nodes that connected with  $V_x$  via potential edge, otherwise it would be contrary to the degree restriction. An example is shown in Fig. 1.

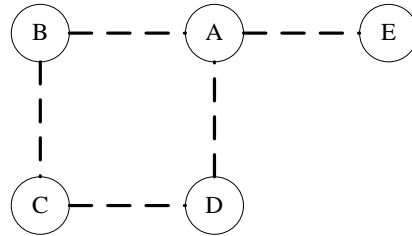


Figure 1. An example of algorithm

To calculate the shortest path from  $V_B$  to  $V_E$ , under the condition of  $D = 2$ , update  $V_A$  by  $E_{BA}$ . If update  $V_D$  via  $V_A$ , then it is unable to access  $V_E$  via  $V_B \rightarrow V_C \rightarrow V_D \rightarrow V_A \rightarrow V_E$ , because  $V_A$  has been permanently marked, so  $V_D$  should not be updated now.

Now we introduce virtual nodes  $V_{D'}$  and  $V_{A'}$  dynamically ( $V_{D'}$  inherits the original topology of  $V_D$ , and  $V_{A'}$  inherit the original topology of  $V_A$ ), add two edges  $E_{A'D}$  and  $E_{AD'}$ , which the weights of are the same as  $E_{AD}$ . After updating  $V_A$  via potential edge, don't update  $V_D$  via  $V_A$ , but only update the virtual node  $V_{D'}$ , so that we can avoid the situation which  $V_E$  is unreachable.

Through the above analysis, we can find, the Dijkstra algorithm is qualified when  $\delta_i = 0$  or  $\delta_i = 2$ , but not when  $\delta_i = 1$ . If  $\delta_i = 1$  and label  $V_i$  is exactly the smallest, update  $V_i$  via a potential edge, if  $E_{ia} \in R$ , when updating  $V_A$  with  $V_i$ ,  $V_i$  has been passed through twice, which is conflict with restriction of degree. Therefore, the single Dijkstra algorithm cannot solve this problem correctly.

### Algorithm Description

The algorithm is relatively complex, because it is designed through introducing the critical point set. The example shown in the following Tab.1 can help to understand the essence of the algorithm.

We assume that the graph is connected. *Step 1* is the initialization of the data structure that saves two values for each node, which is the difference from the Dijkstra algorithm; *Step 1.1* initializes the data structure of all the nodes; *Step 1.2* initializes the starting point; *Step 1.3* initializes permanent calibration set  $P$ , virtual set  $V'$ , critical node set  $C$ , as well as temporary calibration nodes set  $Q$ .

*Step 2* can be formed of many steps. First, select the node  $V_y$  with the smallest label value (*Step 2.1*); and then marked  $V_y$  permanently (*Step 2.2*); if  $V_y$  is just the node of original graph (*Step 2.3*), and the value of red label is smaller than the value of green label, and  $V_y$  belongs to  $Z$  (*Step 2.3.1*), it indicates that  $V_y$  have been already updated via a potential edge, and it is obvious that other nodes cannot be updated via the potential edge linking  $V_y$ , otherwise, degree restriction is exceeded because  $Deg(V_y) = D + 1$ . In order to avoid the degree restriction damaged, add two virtual nodes  $V_{y'}$  and  $V_{x'}$  ( $E_{xy} \in S$ ), and add  $V_{x'}$  into the adjacency list of  $V_y$  (*Step 2.3.1.1*), add  $V_{y'}$  into the adjacency list of  $V_x$  (*Step 2.3.1.2*), add  $V_x$  into  $C$  (*Step 2.3.1.3*), add  $V_{x'}$  and  $V_{y'}$  into  $V'$  (*Step 2.3.1.4*), add  $E_{x'y}$  and  $E_{xy'}$  into  $E$  (*Step 2.3.1.5*), through these operations, the degree of can be restricted in the prescribed range. If  $Deg(V_y) = D$  (*Step 2.4*), only add  $V_x$  linking by actual edge in  $Q$  into the adjacency list of  $V_y$  (*Step 2.4.1*), otherwise, add all the nodes that link with  $V_y$  and haven't been permanently marked into the adjacency list of  $V_y$  (*Step 2.4.2*).

The next step is update operation. First, according to the type of edge, decide to update potential edge or practical edge (*Step 2.5.1, Step 2.5.2*); if  $V_x$  can be updated via  $V_y$  (*Step 2.5.3*), then update

$V_x$  via  $V_y$ , and update the value of  $d[x]$  (Step 2.5.3.1, Step 2.5.3.2). At last, update the temporary label set  $Q$  (Step 2.6).

The time complexity of the algorithm is  $O(n^3)$ .

The algorithm is described with pseudocode as follows.

F: represents an active interface for 0 point sets

Z: represents the active interface has one and only one point set.

P: represents a set of nodes that have been permanently labeled

C: represents a critical point set

V': represents a virtual point set

Q: represents a temporary label set

$d[x][\text{green}]$ : represents the value through the updated actual edge

$d[x][\text{red}]$ : represents the value isn't through the updated actual edge

I.0 Procedure initialization

I.1 for each  $V_x \in V$

I.1.1  $d[x] = d[x][\text{green}] = d[x][\text{red}] = \infty$

I.1.2 end for

I.2  $d[s] = d[s][\text{green}] = d[s][\text{red}] = 0$

I.3  $P, V', C \leftarrow \emptyset$

I.4  $Q \leftarrow V$

I.0 end procedure

Degree-constrained shortest path algorithm is described with pseudocode as follows.

Input:  $G = (V, E)$  and Source  $V_s$  and  $E \leftarrow S \cap R$

Output:  $G' = (V \cup V', E')$  with shortest path information

1 Call initialization

2 for each  $V_x \in Q$

2.1  $V_y \leftarrow \arg V_x \in Q \min d[x]$  Exit if  $d[y] = \infty$

2.2  $P \leftarrow P \cup \{V_y\}$

2.3 if  $V_y \in V$  then

2.3.1 if  $d[V_y][\text{red}] < d[V_y][\text{green}] \wedge V_y \in Z$  then

2.3.1.1  $\text{adj}[y] \leftarrow \text{adj}[y] \cup \{V_{x'} \mid \forall V_{x'}, V_{x'} \in V' \wedge E_{xy} \in S\}$

2.3.1.2  $\text{adj}[x] \leftarrow \text{adj}[x] \cup \{V_{y'} \mid \forall V_x, V_x \in V \wedge E_{xy} \in S\}$

2.3.1.3  $C \leftarrow C \cup \{V_x\}$

2.3.1.4  $V' \leftarrow V' \cup \{V_{x'}, V_{y'}\}$

2.3.1.5  $E \leftarrow E \cup \{E_{xy}, E_{x'y}\}$

2.3.1 end if

2.4 else if  $V_y \in F$  then

2.4.1  $\text{adj}[y] \leftarrow V_x, \forall V_x, V_x \in Q \wedge E_{xy} \in S$

2.4.2 else  $\text{adj}[y] \leftarrow V_x, \forall V_x, V_x \in Q \wedge E_{xy} \in E$

2.3 end if

2.5 for each  $V_x$  in  $\text{adj}[y]$

2.5.1 if  $E_{xy} \in S$  then index = green

2.5.2 else if  $E_{xy} \in R$  then index = red

2.5.1 end if

2.5.3 if  $d[x][\text{index}] > d[y] + W_{xy}, V_x \notin P$  then

2.5.3.1  $d[x][\text{index}] \leftarrow d[y] + W_{xy}$

2.5.3.2  $d[x] \leftarrow \min(d[x][\text{green}], d[x][\text{red}])$

```

2.5.3         end if
2.5           end for
2.6           Q ← V ∪ V' - P
2           end for

```

An example for algorithm analysis is shown in Fig.2.

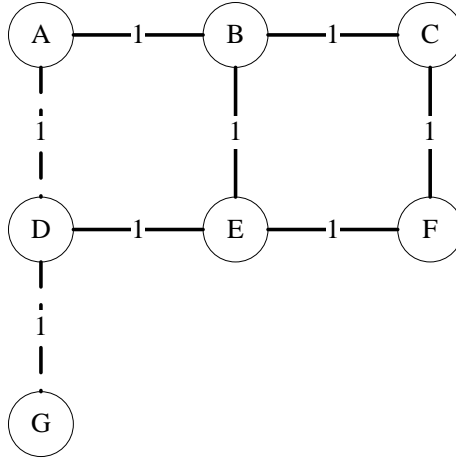


Figure 2. An example for algorithm analysis

The algorithm's calculation result for Fig.2 is shown in the Tab. 1.

Table 1. The algorithm's calculation result for Fig.2

Iteration	$V_A$	$V_B$	$V_C$	$V_D$	$V_E$	$V_F$	$V_G$	$V_{D'}$	$V_{E'}$
1( $V_y=V_A$ )	0/0	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	NULL	NULL
2( $V_y=V_B$ )	0/0	$\infty/1$	$\infty/\infty$	1/ $\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	NULL	NULL
3( $V_y=V_D$ )	0/0	$\infty/1$	$\infty/2$	1/ $\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	NULL	NULL
4( $V_y=V_C$ )	0/0	$\infty/1$	$\infty/2$	1/ $\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/\infty$	$\infty/2$
5( $V_y=V_{E'}$ )	0/0	$\infty/1$	$\infty/2$	1/ $\infty$	$\infty/\infty$	$\infty/3$	$\infty/\infty$	$\infty/\infty$	$\infty/2$
6( $V_y=V_F$ )	0/0	$\infty/1$	$\infty/2$	1/ $\infty$	$\infty/\infty$	$\infty/3$	$\infty/\infty$	$\infty/\infty$	$\infty/2$
7( $V_y=V_E$ )	0/0	$\infty/1$	$\infty/2$	1/ $\infty$	$\infty/4$	$\infty/3$	$\infty/\infty$	$\infty/\infty$	$\infty/2$
8( $V_y=V_{D'}$ )	0/0	$\infty/1$	$\infty/2$	1/ $\infty$	$\infty/4$	$\infty/3$	$\infty/\infty$	$\infty/5$	$\infty/2$
9( $V_y=V_G$ )	0/0	$\infty/1$	$\infty/2$	1/ $\infty$	$\infty/4$	$\infty/3$	$\infty/6$	$\infty/5$	$\infty/2$

## 5 Conclusion

This paper analyzes the wavelength conversion efficiency of the existing WDM network, and puts forward the problems of routing brought by the limitation on number of wavelength converters. In the WDM network with restricted number of wavelength conversion, the shortest path algorithm cannot meet the demand of routing, and the complexity of time based on the minimum weight perfect matching algorithm is relatively high. So we put forward the improved Dijkstra algorithm. Through the analysis of the key issue of problem, we introduce the critical point set for the nodes of  $\delta_i = 1$ , adjust the graph dynamically in the calculation process, restrict the refresh of nodes via virtual nodes, finally, resolve the problem of shortest path calculation, and the time complexity of algorithm is also significantly reduced. Aimed at the routing of the WDM network that wavelength converter is not used widely, this algorithm balances the optical path and interface constraints, improves the utilization rate of load-balanced links, and saves the time of data transmission and the resource of link.

## References

- [1] K. Lee, M. Shayman, Optical Network Design with Optical Constraints in Multi-hop WDM Mesh Networks[C], ICCCN'04, Oct. 2004.
- [2] S. Khuller, K. Lee and M. Shayman. On Degree Constrained Shortest Path[C]. ESA 2005.
- [3] S.Fekete, S.Huller, M.Klemmstein, B.Raghavachari and N. Young, "A Network-Flow technique for finding low-weight bounded-degree spanning trees"[J], Journal of Algorithms, 1997,24 : 310-324
- [4] E. Leonardi, M. Mellia, M. A. Marsan, Algorithms for the Logical Topology Design in WDM All-Optical Networks[J], Optical Networks Magazine, Jan. 2000, 35-46.
- [5] N.A.Riza , "Reconfigurable Optical Wireless"[C], LEOS '99, 1999,1 : 8-11
- [6]Xu Bao, Gao Suixiang, ran min, a computer simulation of the improved [J]. WDM network routing algorithm, 2006,23 (2): 121-124
- [7]Xiao Shiyuan, Liu Xiande, Jin Xin, a dynamic routing and wavelength assignment algorithm for wavelength conversion constrained WDM networks, Journal of electronics, 2005,33 (6): 1142 - 1140
- [8] P.Gruumohan, J. Hui "Topology Design for Free Space Optical Network"[C], ICCCN '2003, Oct.2003