How simple routing algorithms are good for solving RWA problem in Survival Optical Networks?

Dieu Linh Truong *
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
Email: linhtd@soict.hust.edu.vn

Quang Huy Duong
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
Email: huydq.vp@gmail.com

ABSTRACT

Since Routing and Wavelength Assignment (RWA) in optical networks is NP-complete problem. Many researches have proposed from simple to complex heuristic algorithms for solving this RWA problem. The shortest path of Dijkstra and sometimes the shortest pair of disjoint paths of Suurballe are used for solving this problem. Such simple heuristic algorithms are usually prejudged as inefficient leading to the conception of complex algorithms. In this paper we analyse quantitatively to see how the simple algorithms of Dijikstra and of Suurballe are good in solving RWA problem in survivable optical networks. The comparisons are made against the optimal solution, when it is available, and against a lower bound, otherwise. The results show that simple algorithms once give solutions, they give very good solutions.

Keywords

Routing and Wavelength Assignment, survival optical networks, Dijkstra algorithm, Suurballe algorithm.

1. INTRODUCTION

Thanks to the ability to provide large bandwidth, the optical networks have become more and more popular in backbone, metro core and access networks. The modern optical networks are moving from exploiting a single wavelength per fiber in the first generation optical networks to exploiting multiple wavelengths by using Wavelength Division Multiplexing technique in the second generation optical networks. Moreover, in the second generation optical networks, the switching task as well as intelligent functions are moved to the optical layer eliminating the need to convert optical signal to electrical form by expensive devices. Consequently, each network connection needs to use single wavelength along all its links. The routing problem in optical networks is usually divided to two sub-problems:

- Routing, i.e., identifying a path for each connection.
- Wavelength assignment, i.e, choosing a wavelength for using by each connection.

The entire problem is so called RWA (Routing and Wavelength Assignment) although the two steps can be performed in any order. Once a connection path has been assigned a wavelength is called a lightpath.

There are different possible objective functions for RWA problem, such as minimising the number of used wavelengths per link, or minimising the total number of used wavelength in the entire network. The RWA problem has been proven NP-complete in [2]. Some researches have proposed Integer Linear Programs (ILP) for modelling the optimal solutions for this problem [3, 6, 9]. The common weakness of these ILPs is that they can only be solved in small size networks. Therefore, many researches have proposed heuristic algorithms for this RWA problem with the expectation to obtain solutions in larger size network in acceptable delay while approaching the optimal solutions. Those heuristic algorithms include both simple ones, as listed in [15], and complex ones as listed in [1, 5, 8, 10, 14].

Intuitively, we think that the simple algorithms are not as efficient as the complex algorithms. That is why researchers usually forget simple algorithm but spend times and energy to design complex algorithms. However complex routing algorithms consume a lot of computation resource and are difficult to implement in the practice. We suspected that simple routing algorithms are not so bad, thus it might be not necessary to investigate computation resource on complex algorithms. The objective of this paper is to analyse simple algorithms for solving RWA problems to see how good those algorithms are. The answer of this question would help researchers to decide to orient their researches to complex or simple algorithms while dealing with RWA related problems.

The two simple algorithms usually used in the Routing task for RWA problem in general, and RWA problem in survivable network in particular, are the shortest path algorithm of Dijkstra [4] and the shortest pair of disjoint path algorithm of Suurballe [11, 12]. In this research, we will estimate the efficiency of these simple routing algorithms in solving RWA problem in survival optical networks. The algorithms will be evaluated against the optimal solutions when this later is available and against a lower bound otherwise.

*Corresponding author

SoICT '14 December 04 - 05 2014, Hanoi, Viet Nam

ACM 978-1-4503-2930-9/14/12 ...$15.00.
http://dx.doi.org/10.1145/2676585.2676625.

The rest of the paper is organised as follows: Section 2 will present some background concepts on the survivability issue and RWA problem in survivable optical networks. In this section, we will also list the simple routing algorithms as well as the well-known wavelength assignment strategies usually used in solving RWA problem. In Section 3, the optimal solution for RWA problem for survivable networks is modelled mathematically. Section 4 comes up with 2 complete RWA heuristics that use two simple routing algorithms in evaluation. Section 5 presents a way to estimate a lower bound of the optimal solution. Then, Section 6 summaries the numerical experiments and analysis. Finally, Section 7 concludes the paper.

2. BACKGROUND ON SURVIVABLE ROUTING IN OPTICAL NETWORKS

Optical networks function mainly on circuit switching style due to very fast large bandwidth of fiber while network nodes have limited processing function. This circuit switching style makes data transmission processes follow strictly stable routes assigned to them during all the transmission time. This characteristic implies that optical networks are very sensible to fiber cuts or optical node failures. The optical nodes are maybe Optical Line Terminal, Optical Network Units, Optical Cross-connects etc.. Those optical nodes can be recovered easily if we reserve another backup optical node for each of them. The backup node will replace the failed nodes when failures happen. However, it is more complex to recover the network from fiber cuts. The fiber cuts happen principally because of mistaken cuts during construction events. According to [13] there are in average 4.39 cuts/year/1000 miles of fiber. Even the fiber cut rate is not high for a single fiber, it increases quickly in networks with high density. Fiber repair takes many hours or even several days which cause unacceptable interruption of data transmission service. The common solution to avoid this long service interruption is to employ a path protection scheme to make the network survivable even while fiber cuts.

In path protection scheme, each transmission connection, called *working connection*, must have a *backup connection* that shares the same source and destination nodes with the *working connection*. The *backup connection* needs to be reserved before hand and will replace the *working connection* when the latter fails due to any error. Consequently, the *backup connection* must not be affected when the *working connection* fails. The transmission connection is thus survivable. If all transmission connections are survivable then the network is said survivable.

For the sake of simplification, it is usually assumed that there are at most a single failure in the network at a time. That means a failure must be recovered before another failure may occur. Under this single failure circumstance, a *backup connection* needs only to be disjoint with the *working connection* in order to be unaffected when the *working connection* fails.

The RWA problem for survivable network is stated as follows:

- *Given* a physical topology $\mathbb{G}$ containing all nodes and links of the networks. The number of maximum available wavelengths in a link is $\mathbb{W}$ for each direction.
- *Given* a set of connection requests $\mathbb{M} = \{(s, d, bw)\}$ where (s, d, bw) is a connection from s to d and requesting for bandwidth equivalent of bw wavelengths, $bw \geq 1$.
- The *goal* is to find route and to assign wavelengths to each connection request in $\mathbb{M}$ *such that* each connection request has a *working path* and a link-disjoint *backup path*. The solution must also minimise the total number of used wavelengths in all links of the whole network.

This RWA problem is more complex than that in non survivable networks due to principally more complex routing task. In order to solve it, the problem is also broken to two sub-problem: Routing sub-problem and Wavelength Assignment sub-problem. Following subsections list some simple algorithms for the Routing sub-problem and the Wavelength Assignment sub-problem.

2.1 Simple survivable routing algorithms

In non-survivable networks, the shortest path algorithm of Dijkstra [4] can be used for seeking the path for each connection as the shortest path between the source node and the destination node of the connection. Then, a wavelength can be assigned to this path by using a wavelength assignment strategy amongst those are listed in 2.2. However, in survivable networks, for each connection request, we need to find a *working path* and a *backup path* disjoint with it. These two paths can be found by using one of two algorithms described below.

2.1.1 Two shortest paths

For each connection request $r = (s, d, bw)$ in the network $\mathbb{G}$, this algorithm works as follows:

- The *working path* is searched first as the shortest path between the source node s and the destination node d in the network $\mathbb{G}$.
- All the links used by the *working path* are then temporary excluded from $\mathbb{G}$.
- The *backup path* is searched as the shortest path between the source node s and the destination node d in the residual network $\mathbb{G}$. In so doing, the *backup path* is surely link-disjoint with the *working path*.

The complexity of the algorithm is roughly two times the complexity of Dijkstra algorithm, so it is in the order of $O(|\mathbb{E}| + |\mathbb{V}| \log |\mathbb{V}|)$ where $|\mathbb{E}|$ is the set of network links and $\mathbb{V}$ is the set of network nodes.

The algorithm is very simple, the working and backup paths satisfy the disjoint condition. Since the working and the backup paths are two shortest paths, they will use practically small amount of resources.

2.1.2 Shortest pair of disjoint paths

sec:Suurballe Suurballe have been proposed in [11, 12] a simple algorithm for finding a pair of disjoint paths between a source and a destination such that the total length of the two path are minimal. If we use this algorithm in the Routing step of the RWA problem for survivable networks, we are sure to find for each connection request the two disjoint paths that go through the least of links and thus use the least of resources. The Suurballe algorithm requires two iterations of Dijkstra algorithm so its complexity is also $O(|\mathbb{E}| + |\mathbb{V}| \log |\mathbb{V}|)$. The details on the Suurballe algorithm is not explained in this paper but can be found in [11, 12].

In comparison with using simply two times Dijkstra algorithms as in the "two shortest paths" approach, Suurballe algorithm provides the optimal pair of *working path* and *backup path* in terms of resource usage.

2.2 Simple wavelength assignment strategies

This section list the wavelength assignment strategies that are usually used in simple RWA solutions for non-survivable as well as survivable optical networks. The objective of Wavelength assignment task is to chose wavelengths for using by each connection whose path has been identified by the Routing step. From the original network topology $\mathbb{G}$, we build different graphs $G_1, G_2, G_3, \dots G_{\mathbb{W}}$ so that graph G_j contains only edges where the wavelength λ_j is available. Then, routings are performed on these graphs separately. The process may result in up to $\mathbb{W}$ lightpaths, one for each graph. A lightpath amongst them is selected as the final RWA solution for the request by using one of the following methods:

- **First Fit**: select the first shortest lightpath found when performing routing successively from G_1 to $G_{\mathbb{W}}$.
- **Best Fit**: select the shortest lightpath from all lightpaths found in all graphs.
- **Densest Fit**: select the lightpath found in the the graph with the most edges. If no lightpath is found in that graph, then try on the second densest graph and so on.
- **Random Fit**: select randomly a lightpath in from those found in all graphs.

The work in [7] have evaluated the efficiency of 8 non-survivable RWA simple algorithms that use the shortest path algorithm for the Routing step and First Fit, Best Fit, Densest Fit or Random Fit for the Wavelength assignment step. In this paper, we evaluate only simple routing algorithm in the whole RWA solution in survivable networks. Although the wavelength assignment task may be similar to those listed above, the Routing step in our case is definitively different to that in non-survivable networks.

3. MATHEMATICAL MODEL FOR OPTIMAL SOLUTIONS

This section presents a mathematical model for the optimal solution of the RWA problem in survival optical networks.

3.1 Variables

The network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is seen as a directed graph where $\mathbb{V}$ is the set of nodes and $\mathbb{E}$ is the set of arcs. Each link between two nodes $i, j \in \mathbb{V}$ has as two arcs $(i,j) \in \mathbb{E}$ and $(j,i) \in \mathbb{E}$ in opposite direction, each arc has $\mathbb{W}$ wavelengths. Let $\Gamma^+(i)$ be the set of arcs going out from node i and $\Gamma^-(i)$ be the set of arcs coming into node i.

Following decision valuables define the paths and the wavelength for each connection request. s, d, i, j are node in $\mathbb{V}$:

$$\alpha_{ij}^{sdw} = \begin{cases} 1 & \text{if the request for connection } (s,d,bw) \in \mathbb{M} \text{ uses wavelength } w \text{ on arc } (i,j) \text{ for its working path} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{ij}^{sdw} = \begin{cases} 1 & \text{if the request for connection } (s,d,bw) \in \mathbb{M} \text{ uses wavelength } w \text{ on arc } (i,j) \text{ for its backup path} \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{ij}^{sd} = \begin{cases} 1 & \text{if the request for connection } (s,d,bw) \in \mathbb{M} \text{ uses any wavelength on arc } (i,j) \text{ for its working path} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{ij}^{sd} = \begin{cases} 1 & \text{if the request for connection } (s,d,bw) \in \mathbb{M} \text{ uses any wavelength on arc } (i,j) \text{ for its backup path} \\ 0 & \text{otherwise} \end{cases}$$

$$l_{sd}^{w} = \begin{cases} 1 & \text{if the request for connection } (s,d,bw) \in \mathbb{M} \text{ uses wavelength } w \text{ in the working path} \\ 0 & \text{otherwise} \end{cases}$$

$$l'^{w}_{sd} = \begin{cases} 1 & \text{if the request for connection } (s,d,bw) \in \mathbb{M} \text{ uses wavelength } w \text{ in the backup path} \\ 0 & \text{otherwise} \end{cases}$$

3.2 Objective function

The objective of the RWA problem is to minimise the total wavelength used in all links by both working and backup paths of connection requests in the whole network.

$$\min \sum_{(i,j)\in\mathbb{E}} \sum_{(s,d,bw)\in\mathbb{M}} \sum_{w\in\mathbb{W}} \left(\alpha_{ij}^{sdw} + \beta_{ij}^{sdw}\right) \tag{1}$$

3.3 Constraints

The variables must satisfy the following constraints:

- For each request $(s,d,bw) \in \mathbb{M}$, following constraints form its working connection:

$$\sum_{i\in\Gamma^-(l)} \alpha_{il}^{sdw} - \sum_{j\in\Gamma^+(l)} \alpha_{lj}^{sdw} = \begin{cases} l_{sd}^{w} & \text{if } l = d \\ -l_{sd}^{w} & \text{if } l = s \\ 0 & \text{else} \end{cases} \tag{2}$$

$$\forall l \in \mathbb{V}, \forall w \in \mathbb{W}$$

$$\sum_{w\in\mathbb{W}} \alpha_{ij}^{sdw} = \alpha_{ij}^{sd} \times bw^{sd}, \tag{3}$$

$$\forall (i,j) \in \mathbb{E}$$

$$\sum_{w\in\mathbb{W}} l_{sd}^{w} = bw^{sd} \tag{4}$$

Constraints (2), (3) and (4) guarantee that all bw requested wavelengths of a connection are routed on identical path.

- For each request $(s,d,bw) \in \mathbb{M}$, following constraints form its backup connection:

$$\sum_{i\in\Gamma^-(l)} \beta_{il}^{sdw} - \sum_{j\in\Gamma^+(l)} \beta_{lj}^{sdw} = \begin{cases} l_{sd}^{w} & \text{if } l = d \\ -l_{sd}^{w} & \text{if } l = s \\ 0 & \text{else} \end{cases} \quad (5)$$

$$\forall l \in \mathbb{V}, \forall w \in \mathbb{W}$$

$$\sum_{w\in\mathbb{W}} \beta_{ij}^{sdw} = \beta_{ij}^{sd} \times bw^{sd}, \quad (6)$$

$$\forall (i,j) \in \mathbb{E}$$

$$\sum_{w\in\mathbb{W}} l'^{w}_{sd} = bw^{sd} \quad (7)$$

Constraint (5), (6) and (7) guarantee that all bw wavelengths of a connection are routed on identical path.

- Working path and backup path of a connection do not share any common link:

$$\alpha_{ij}^{sd} + \alpha_{ji}^{sd} + \beta_{ij}^{sd} + \beta_{ji}^{sd} \leq 1 \quad (8)$$

$$\forall (i,j) \in \mathbb{E}, (s,d,bw) \in \mathbb{M}$$

- A wavelength in a link can be used by at most one connection, regardless it is working or backup connection.

$$\sum_{s,d} \alpha_{ij}^{sdw} + \sum_{s,d} \beta_{ij}^{sdw} \leq 1 \quad (9)$$

$$\forall (i,j) \in \mathbb{E}, (s,d,bw) \in \mathbb{M}$$

4. SIMPLE RWA ALGORITHMS FOR EVALUATION

In order to evaluate the efficiency of Dijkstra based algorithm "Two shortest path" and "Shortest pair of disjoint paths" of Suurballe in solving RWA problem for survivable networks, these two algorithms will be combined with the Best Fit strategy for creating a complete RWA algorithm as follows. For each connection request $r = (s, d, bw) \in \mathbb{M}$ to be allocated in the network $\mathbb{G}$ in a survivable style, a set of bw wavelengths required by the connection is sought first. Then, we try to find a path between the source s and the destination d of the request in the network composing of only those wavelengths. If we cannot obtain a path, we another wavelength set will be tried.

Obviously, with a connection request $r = (s, d, bw)$, there are $C_{\mathbb{W}}^{bw}$ possible wavelength sets that r can use. However, trying with such a large number of sets does not allow the RWA algorithm running in acceptable time. Consequently, we propose a method for selecting a significantly smaller number of trial wavelength sets, but still maintain a reasonable number of choices. Routing algorithm is performed iteratively with all selected trial wavelength sets resulting in different paths. After all, the shortest path amongst all obtained paths is picked as the final result and the corresponding trial wavelength set is assigned to the path.

4.1 Trial wavelength sets

For each connection, a trial wavelength set is selected as follows. We rely on the total number arcs of the graph $\mathbb{G}$ that still have a particular wavelength available. Let us indexing wavelengths by $w_0, w_1, w_2, ..., w_{\mathbb{W}-1}$. Let EoW be an array to store the total number available arcs for each wavelength. Then, $EoW[w_i]$ gives the number of arcs of $\mathbb{G}$ where wavelength w_i is still unused.

- `Sort(`EoW`)` sorts array EoW in the increasing order and removing zero elements so that

  $$0 < EoW[w_{i_0}] \leq EoW[w_{i_1}] \leq ... \leq EoW[w_{i_{\mathbb{W}-1}}]$$

  Trial wavelength sets are groups of bw wavelengths present in bw consecutive elements of the EoW array. The first set is $\{EoW[w_{i_0}], EoW[w_{i_1}], ..., EoW[w_{i_{bw-1}}]\}$. The second set is $\{EoW[w_{i_1}], EoW[w_{i_2}], ..., EoW[w_{i_{bw}}]\}$, and so on.
  In so doing, we obtain $W - bw + 1$ sets of wavelengths sorted in increasing order of their popularity in network edges.

- `getFirstEoW()` picks the bw first wavelengths of EoW. This is the first trial wavelength set. These wavelengths are available in the least number of arcs.

- `getNextEoW()` picks the next set of bw trial wavelengths. These wavelengths are available in more number of arcs.

4.2 Shortest Path-Based RWA

This section presents how the above selected trial wavelength sets are injected in the routing algorithm "Two shortest paths" in order to form the whole RWA solution. The whole RWA solution in this case is called Shortest Path-based RWA (SPH). For each connection request (s, d, bw), the working path is sought first, then the backup path. The main idea of SPH is as follows:

- For each trial wavelength set, we create $\mathbb{G}_{trial}$ as a subgraph of $\mathbb{G}$ which contains only the edges where all wavelengths in the trial set are available. Then Dijkstra algorithm is run on $\mathbb{G}_{trial}$ for finding the shortest path between the source and the destination of the request.

- The final working path is selected as the shortest one amongst all found after trying all trial wavelength sets. The working path uses the wavelengths of the $\mathbb{G}_{trial}$ it comes from. This wavelength assignment follows, in some extend, the Best-Fit strategy.

- Then all edges of the working path are temporarily excluded from $\mathbb{G}$ and the backup path for it is sought in similar way as what happens for the working path.

The algorithm is illustrated in pseudo-code in Alg. 1

4.3 Suurballe-Based RWA

Suurballe-based RWA solution (SBH) uses Suurballe for the routing step with the selected trial wavelength sets. For each connection request and for each trial wavelength set, the working and backup paths are jointly routed by using Suurballe algorithm. As a result, the working and backup paths use the same wavelength set. The best path pair amongst all the pairs found is selected as the final working and backup path. The algorithm is illustrated in Alg. 2.

Algorithm 1 SPH

```
sort(EoW);
trialW ←getFirstEoW() {get the first trial wavelength set}
while trialW ≠ ∅ do
    p ← dijkstra(s,d);
    if p is shorter than workPath then
        workingPath ← p;
        workingWavelength ← trialW;
    end if
    trialW ←getNextEoW();
end while
assign workingWavelength to workingPath;
remove edges of workingPath from 𝔾
sort(EoW);
trialW ←getFirstEoW();
while trialW ≠ ∅ do
    p ←dijkstra(s,d);
    if p is shorter than backupPath then
        backupPath ← p;
        backupWavelength ← trialW;
    end if
    trialW ←getNextEoW();
end while
assign backupWavelength to backupPath;
```

Algorithm 2 SBH

```
sort(EoW);
trialW ←getFirstEoW();
while trialW ≠ ∅ do
    pathPair ←Suurballe(s,d);
    if pathPair is shorter than bestPair then
        bestPair ← pathPair;
        bestWavelength ← trialW;
    end if
    trialW ←getNextEoW();
end while
assign bestWavelength to bestPair;
```

5. LOWER BOUND OF THE OPTIMAL SOLUTION

Since it is not easy to get the optimal RWA solution for large size networks, in this section, we estimate a lower bound for each RWA optimal solution. The lower bound will be used instead of the optimal solution for benchmarking heuristics in large size networks.

Let SP^{sd} be the total length of the shortest disjoint path pair between node s and node d while ignoring the wavelength constraint. The total length of the working and backup paths of a connection request (s, d, bw) is never smaller than SP^{sd}. Thus a lower bound of the optimal solution is derived as:

$$LB_e = \sum_{\forall(s,d,bw)\in\mathbb{M}} SP^{sd} \times bw \qquad (10)$$

The value SP^{sd} can be easily identified by using Suurballe algorithm for the pair of nodes s and d.

The lower bound is in fact the optimal solution of the original RWA problem when the number of wavelength per link $\mathbb{W}$ is set to ilimited so that the wavelength assignment task can be eliminated. The lower bound is always smaller or equal to the optimal value, and the optimal value is always smaller or equal value of the objective function of the heuristic solution. Although the lower bound may be far from the optimal value, the closer to the lower bound the heuristic is, the closer to the optimal it is. Figure 1 illustrates the relationship between the lower bound, the optimal value and the heuristic solution value. When the heuristic value is equal to lower bound value, they must be all the optimal values.

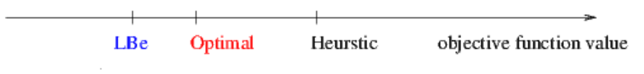

Figure 1: Relationship between lower bound, optimal value and heuristic solution value.

6. NUMERICAL EVALUATION

6.1 Data Description

For evaluate the efficiency of the Shortest path algorithm of Dijkstra and the Shortest pair of disjoint paths of Suurballe in solving RWA problem for survivable networks, we evaluate the performance of SPH and SBH algorithms. The two algorithms are tested on networks of different sizes and densities. The network size varies from $|\mathbb{V}| = 15, 20, 25,$ to 50. The network densities vary by changing average nodal degree of network nodes. Connection request set $\mathbb{M}$ are generated so that there are connection requests between every pair of nodes. Each connection request asks for a bandwidth of $bw = 1 \ldots 3$ wavelengths.

6.2 Performance evaluation against optimal solution in small size networks

We define the network capacity as the total number of wavelengths available in all network links. The network capacity is competed as: $|\mathbb{E}| \times |\mathbb{W}|$, where $|\mathbb{E}|$ is the set of arcs of the network. The network capacity can also be roughly calculated in function of the number of network nodes and the average node degree d by the expression $|\mathbb{V}| \times d \times |W|$.

Table 1: Performance of SPH and SBH in small networks

Networks	LB_e	Opt.	SPH	SBH	SPH vs. Opt. (%)	SBH vs. Opt. (%)	Network capacity	SPH vs. Net. cap. (%)	SBH vs. Net. cap. (%)
$\|\mathbb{V}\|$=5, d=4, $\mathbb{W}$=8	75	75	87	75	16	0	160	54	47
$\|\mathbb{V}\|$=7, d=4, $\mathbb{W}$=8	116	116	127	116	9	0	224	57	52
$\|\mathbb{V}\|$=7, d=5, $\mathbb{W}$=8	133	133	152	136	14	2	272	56	50
$\|\mathbb{V}\|$=9, d=4, $\mathbb{W}$=12	280	280	-	-	-	-	432	-	-
$\|\mathbb{V}\|$=9, d=4, $\mathbb{W}$=14	280	280	296	-	6	-	504	59	-
$\|\mathbb{V}\|$=9, d=5, $\mathbb{W}$=8	220	220	-	-	-	-	352	-	-
$\|\mathbb{V}\|$=9, d=5, $\mathbb{W}$=10	220	220	257	227	17	3	440	58	52
$\|\mathbb{V}\|$=9, d=5, $\mathbb{W}$=12	220	220	252	220	15	0	528	48	42

Table 1 shows the total number of wavelengths used by all connections in the networks in lower bound, in optimal solution, in SPH, in SBH and the percentages of wavelengths that SPH and SBH use more than optimal solution in small size networks, the network capacity, the percentage of wavelengths that SPH, SBH use in comparison with the network capacity. The differences between SPH and the optimal solutions remain between 6 and 16%, meanwhile these differences are much smaller in the case of SBH, i.e., from 0 to 3%. These number demonstrates that, when it can find a solution Suurballe based algorithm finds nearly or even optimal solutions in most of cases. This is understandable since Suurballe algorithm finds the shortest pairs of working and back up path (in term of number of links in the case of this paper) so these two paths uses the least number of wavelengths along their links. That characteristic leads SBH approaches to the optimal solutions.

Differently, SPH uses the Shortest path algorithm for identifying the working path, then the backup path is the shortest one obtained after excluding links of the working path. The working path can be thus very short but the backup path may be very long if the working path has used some critical links in the networks. That explains why SPH does not perform as well as SBH. Even though, the largest gap of 16% over the optimal solutions is still very good for the simple algorithm SPH.

However, we remark that in few cases when the problem is feasible (cases marked with hyphen in numerical tables), SPH cannot find a solution, SBH even cannot find a solution more frequently. This is explained by the fact that the Routing and wavelength assignment of these simple heuristics are solved separately so that with the selected trial wavelength sets, the routing step could end unsuccessfully.

When we tried to reduce network capacities by reducing the number of wavelengths $|\mathbb{W}|$, no optimal solution can be found, that means those networks become infeasible, or they cannot be survivable for the given request set $\mathbb{M}$. We remark also that while increasing network capacity by increasing $|\mathbb{W}|$ for each network topology, as soon as the there exist the optimal solution, the network capacity get to two times greater than the optimal value. That means at least the network capacity must be twice the amount of resources required by the optimal solutions for making survivable RWA problem feasible.

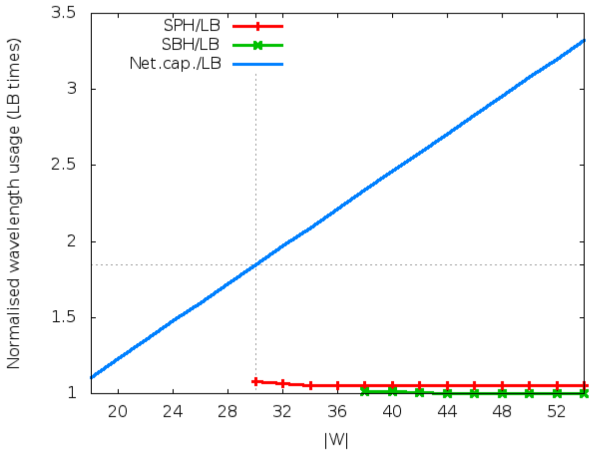

Figure 2: Result in networks with 15 nodes, 4 average degree

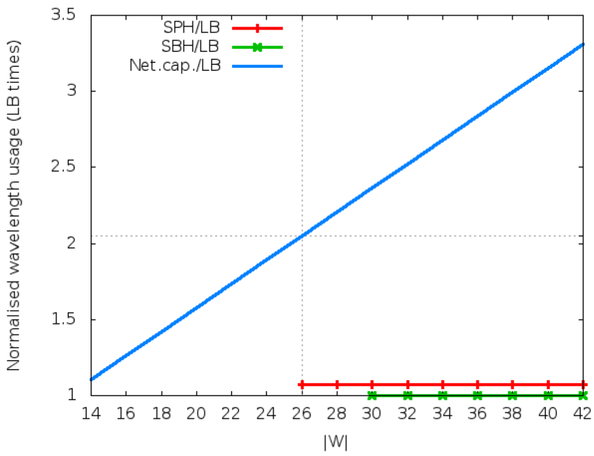

Figure 3: Result in networks with 15 nodes, 5 average degree

6.3 Performance evaluation against lower bounds in large size networks

Since ILP model of the optimal solution could not be run on these big networks, therefore in those networks we com-

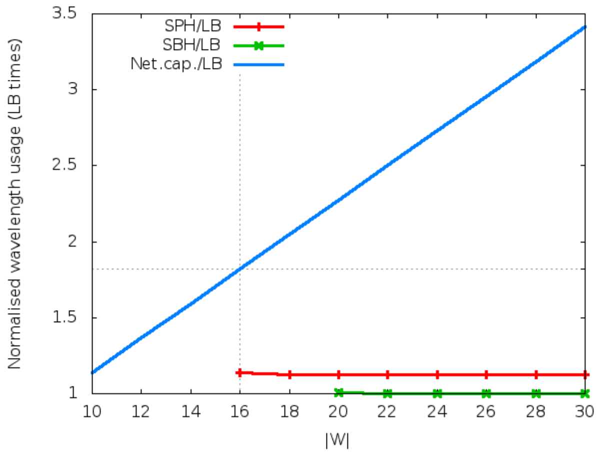

Figure 4: Result in networks with 15 nodes, 6 average degree

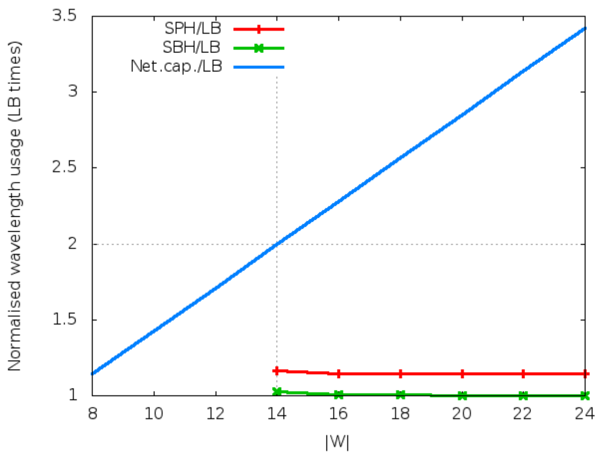

Figure 5: Result in networks with 15 nodes, 7 average degree

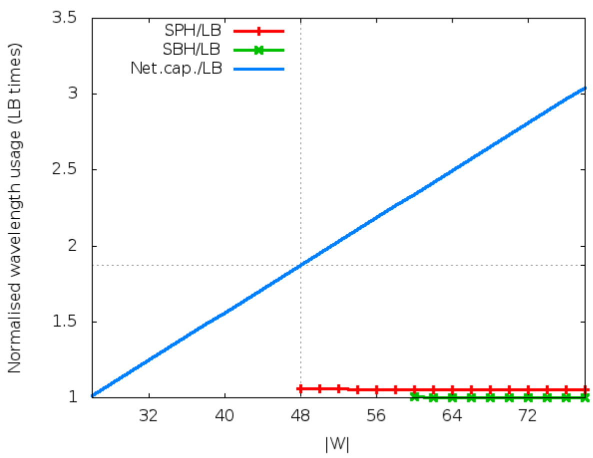

Figure 6: Result in networks with 20 nodes, 4 average degree

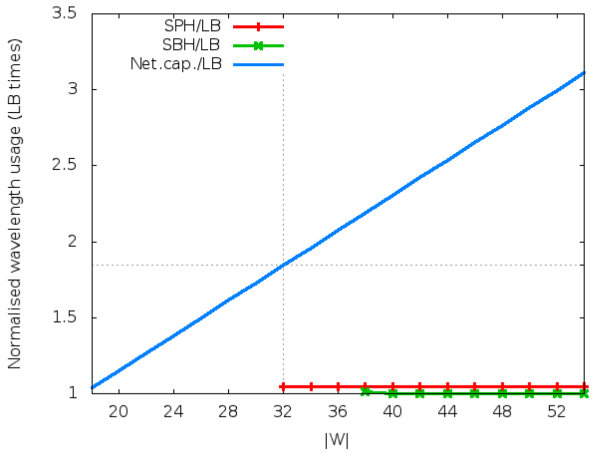

Figure 7: Result in networks with 20 nodes, 5 average degree

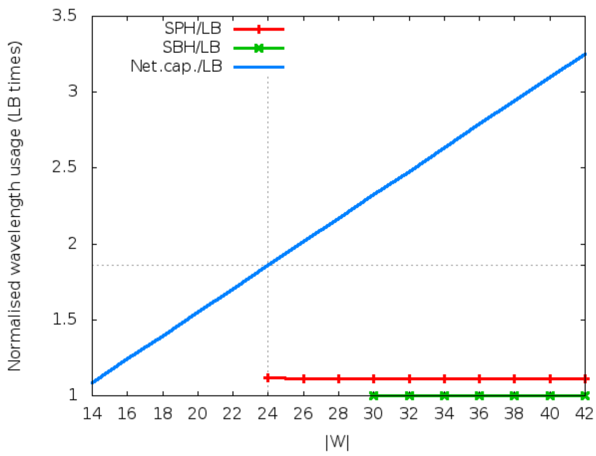

Figure 8: Result in networks with 20 nodes, 6 average degree

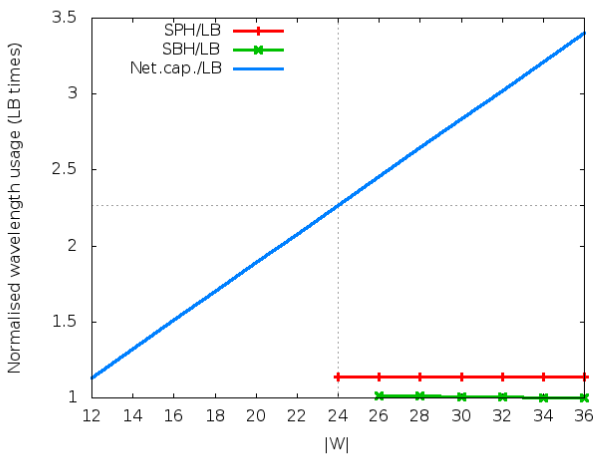

Figure 9: Result in networks with 20 nodes, 7 average degree

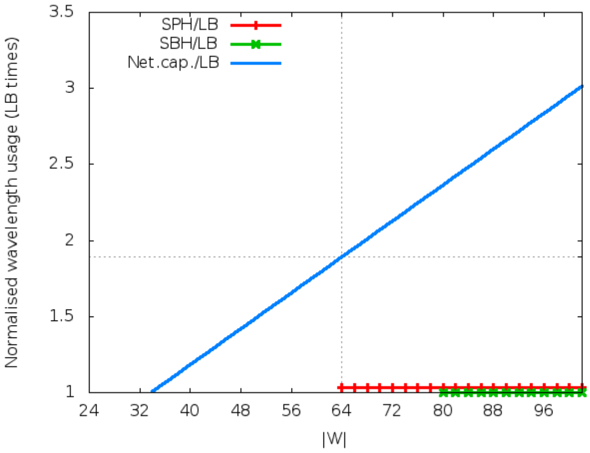

Figure 10: Result in networks with 25 nodes, 4 average degree

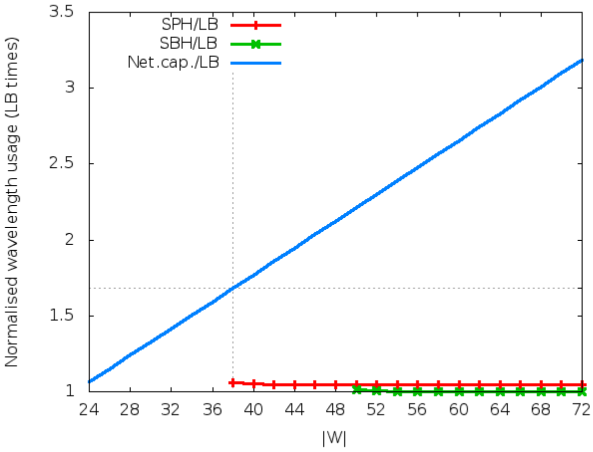

Figure 11: Result in networks with 25 nodes, 5 average degree

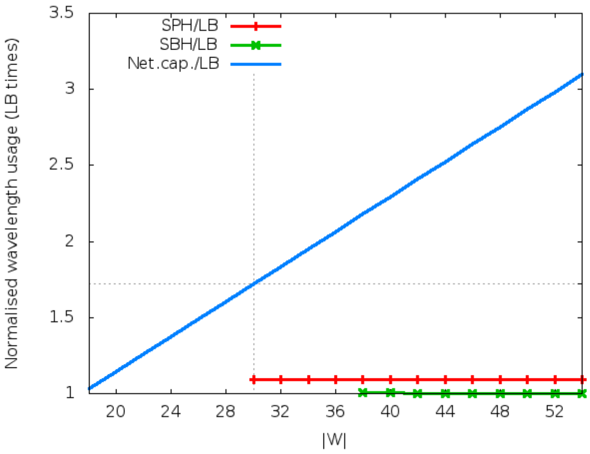

Figure 12: Result in networks with 25 nodes, 6 average degree

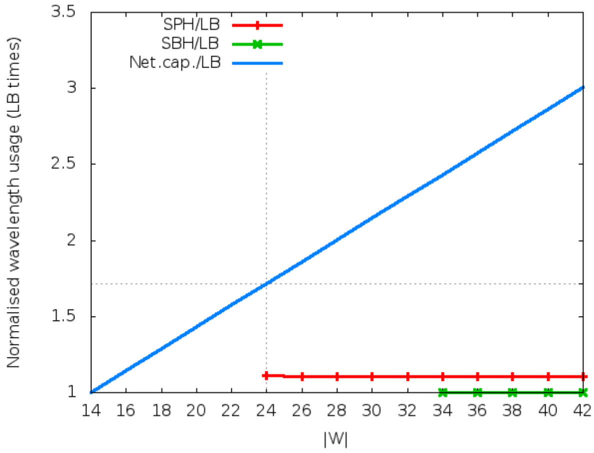

Figure 13: Result in networks with 25 nodes, 7 average degree

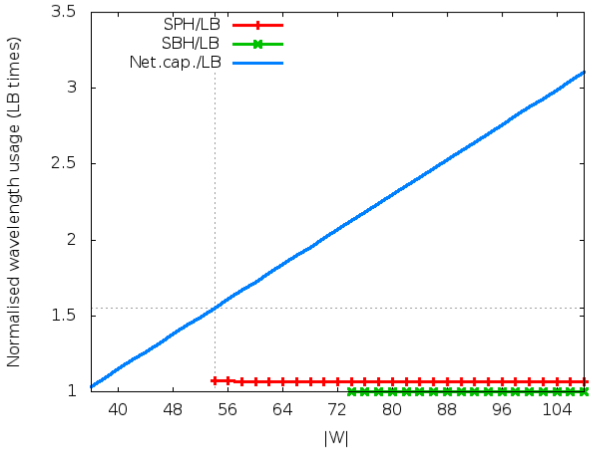

Figure 14: Result in networks with 50 nodes, 7 average degree

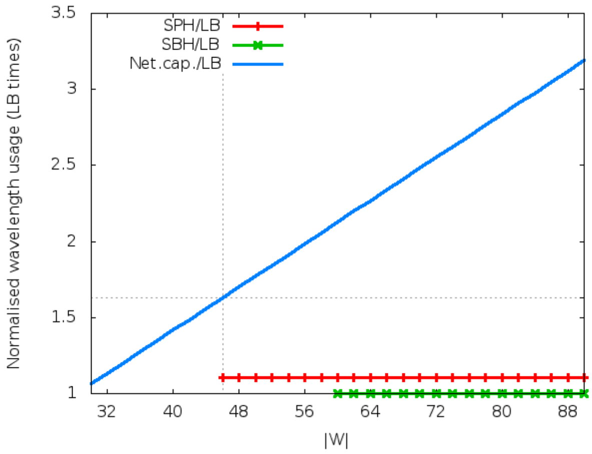

Figure 15: Result in networks with 50 nodes, 8 average degree

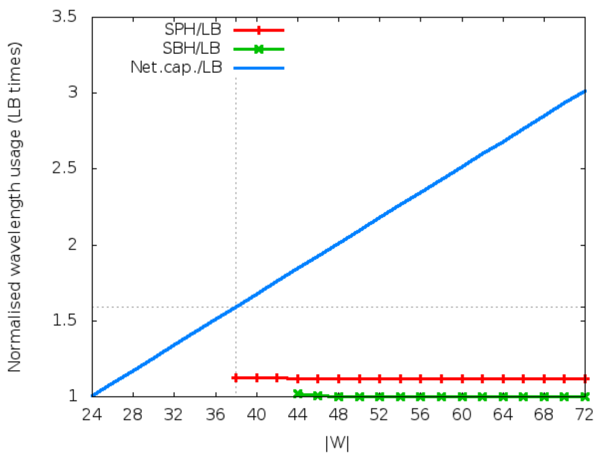

Figure 16: Result in networks with 50 nodes, 9 average degree

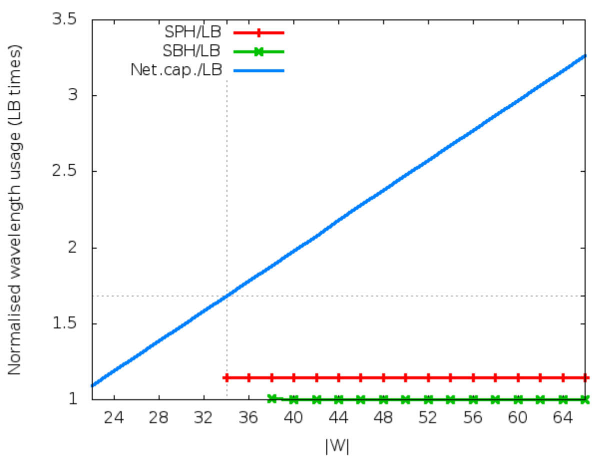

Figure 17: Result in networks with 50 nodes, 10 average degree

pare SPH and SBH against the lower bounds. Figures from 2 to 17 show normalised wavelength usages of SPH, SBH. The normalised wavelength usage is defined as ratio between a solution value and the lower bound. Concretely, the normalised wavelength usage of SPH (or SBH) is the ratio between the total number of wavelength used by SPH (or SBH) in the network and LB_e. For comparison, we show also in these figures the normalised wavelength usage of network capacity as the ratio between network capacity and the LB_e.

According to (10), for each network topology, the lower bound LB_e is independent of the network capacities, i.e. the number of wavelength $\mathbb{W}$. Therefore in Figures from 2 to 17, we can see that the normalised network capacity increases linearly with $\mathbb{W}$. The normalised network cost lines for SPH and SBH are nearly horizontal at most of figure with a light decline at the beginning. This phenomena can be explain by the fact that the increment of network capacities can help SPH or SBH find better until the network capacity become sufficient large to give SPH and SBH any route choices. Beyond that point, bigger network capacity does not provide additional alternative route choices and thus does not lead to better solutions.

The figures show that SBH provides always better solutions than SPH. This confirms again that Suurballe algorithm performs better than using Two shortest paths using Dijsktra algorithm in the RWA problem for survivable networks. In most of case the normalised values for SBH are 1, i.e., SBH gives the optimal solutions (see Figure 1 for illustration). SPH provides solutions that are just slightly above lower bound so they are also very close to the optimal solutions.

For each network topology, when increasing the number of wavelength $|\mathbb{W}|$, the network capacity increases. We remark that, SPH begins to find solution when the network capacity is around 2 times the lower bound (see point lines in the figures) which is even smaller or equal to 2 times the optimal values. Remember that in the previous section, we have found that for making the RWA feasible, at least the network capacity must be 2 times the optimal values in small size networks. Therefore, we suspect that even in large size networks, SPH has a very good capability in finding solution.

SBH starts finding solution when the network capacity is much larger than in the case of SPH. Hence, the capacity to find a solution of Suurballe based algorithm in low capacity network is much weaker than that based on Dijkstra algorithm.

With the above analysis, we suggest to use Suurballe algorithm for routing when we would like to find nearly optimal solution in robust capacity networks. On the other hand, we should you Dijkstra algorithm for routing step in order to find very good solution in any networks.

7. CONCLUSION

In this paper, we have conducted several tests for analysing the efficiency of two simple but well-known algorithms: the shortest path algorithm of Dijkstra and the shortest pair of disjoint path algorithm of Suurballe in solving the RWA problem in survivable networks. We have also proposed a way to calculate the lower bound of the RWA problem for survivable networks since this problem is NP-complete. The results show that counterintuitive, simple algorithms such as Suurballe based RWA solution demonstrates a high ability to solve this problem nearly optimally. Meanwhile, Dijkstra based RWA solution find very good solution in any networks. However Dijkstra based RWA solution are clearly more capable to find solution in low capacity network than Suurballe based RWA solution.

Acknowledgements

The research in this paper is supported by the grant number T2014-124 titled "Evaluation of simple algorithms in Routing problem for Optical networks" funded by Hanoi University of Science and Technology in 2014.

8. REFERENCES

[1] N. Banerjee and S. Sharan. A evolutionary algorithm for solving the single objective static routing and wavelength assignment problem in WDM networks. In *Proceedings of International Conference on Intelligent Sensing and Information Processing, 2004.* , pages 13–18, 2004.

[2] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high bandwidth optical WAN's. *IEEE Transactions on Communications*, 40(7):1171–1182, Jul 1992.

[3] K. Christodoulopoulos, K. Manousakis, and E. Varvarigos. Comparison of routing and wavelength assignment algorithms in wdm networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–6, Nov 2008.

[4] E. W. Dijkstra. Numerische Mathematik. *A note on two problems in connexion with graphs*, (1):269 – 271, 1959.

[5] B. Jaumard, C. Meyer, and B. Thiongane. On column generation formulations for the RWA problem. *Discrete Applied Mathematics*, 157(6):1291 – 1308, 2009. Reformulation Techniques and Mathematical Programming.

[6] B. Jaumard, C. Meyer, B. Thiongane, and X. Yu. Ilp formulations and optimal solutions for the rwa problem. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, volume 3, pages 1918–1924 Vol.3, Nov 2004.

[7] K. Li. Heuristic algorithms for routing and wavelength assignment in WDM optical networks. In *IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008)*, pages 1–8, April 2008.

[8] A. E. Ozdaglar and D. P. Bertsekas. Routing and Wavelength Assignment in Optical Networks. *IEEE/ACM Transaction on Networking*, 11(2):259–272, Apr. 2003.

[9] R. Ramaswami and K. Sivarajan. Routing and wavelength assignment in all-optical networks. *Networking, IEEE/ACM Transactions on*, 3(5):489–500, Oct 1995.

[10] N. Skorin-Kapov. Heuristic algorithms for the routing and wavelength assignment of scheduled lightpath demands in optical networks. *IEEE Journal on Selected Areas in Communications*, 24(8):2–15, Aug 2006.

[11] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4(3):125–145, 1974.

[12] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.

[13] M. To and P. Neusy. Unavailability analysis of long-haul networks. *IEEE Journal on Selected Areas in Communications*, 12(1):100–109, Jan. 1994.

[14] Y. Wang, T. H. Cheng, and M. H. Lim. A Tabu search algorithm for static routing and wavelength assignment problem. *IEEE Communications Letters*, 9(9):841–843, Sep 2005.

[15] H. Zang, J. P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 1:47–60, 2000.