Routing and wavelength/sub-wavelength path assignment to maximizing accommodated traffic demands on optical networks

Yosuke WATANABE*, Kiyo ISHII**, Toshiki SATO*, Atsuko TAKEFUSA**, Tomohiro KUDOH**,***, Maiko SHIGENO*, Akiko YOSHISE*

*Graduate School of Systems and Information Engineering, University of Tsukuba
Tsukuba, Ibaraki, 305-8573, Japan
E-mail:s1520564@sk.tsukuba.ac.jp
**National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Ibaraki, 305-8568, Japan
***Information Technology Center, University of Tokyo
Bunkyo, Tokyo, 113-8658, Japan

Received 15 November 2015

Abstract

Wavelength division multiplexing (WDM) technology transmits multiple optical communication channels in an optical fiber. Routing and wavelength assignment (RWA) problems on WDM network have widely attracted interest of many researchers. Recently, so called sub-wavelength paths which are smaller granular paths than the wavelength-paths are discussed. In this paper, we deal with the RWA problem with considering sub-wavelength assignment on optical networks. One of the purpose of our study is to investigate active rates for optical networks. We formulate a sub-wavelength path assignment problem maximizing accommodated traffic demands by integer programming and solve it by a cutting plane algorithm. Since, in actuality, RWA is done individually for each demand on the time when the demand occurs, we consider a greedy type on-line algorithm. Numerical experiments show the efficiency of our algorithms and give some observation for active rates. Moreover, we verify the efficiency of our greedy-type algorithm on realistic situations which follow 10/40/100 Gbps system used for the current communication on optical networks. Our experimental results conclude that the active rates are depending on the configurations of the underlying graphs, and that our greedy-type algorithm is efficient for several kinds of instances.

keywords : Optical network, Routing and wavelength assignment, Sub-wavelength assignment, Integer programming, Cutting plane method, Greedy algorithm

1 Introduction

Wavelength division multiplexing (WDM) technology transmits multiple optical communication channels in an optical fiber. This technology provides the capability of transferring huge amounts of data in an optical network. The data are carried through the optical channels called wavelength-paths in the optical network. Routing and wavelength assignment (RWA) problems on WDM networks have widely attracted interest of many researchers (Ramaswami et al., 2010). We can find a lot of considerable interest models in RWA problems. One of the such models is a static RWA model to find the minimum number of wavelengths to satisfy all of the given demands. Another model analyzes the blocking performance of wavelength routed networks under dynamic demands.

In a practical manner, smaller granular paths, called sub-wavelength such as optical-channel data unit (ODU), than the wavelength paths are introduced, since the data capacity of each wavelength path is huge such as 10 to 100 Gbps. The data capacity of each sub-wavelength path is such as 1 to 100 Gbps by considering ODU paths (Ishii et al., 2014). Consequently, sub-wavelength path assignment is required as well as wavelength path assignment (Hu and Leida, 2004; Liu and Rouskas, 2013; Yetginer and Rouskas, 2009).

In this paper, we deal with the RWA problem with considering sub-wavelength assignment on optical networks. Our study is to investigate active rates for optical networks. Although the active rate which stands for the ratio of used resource on an optical network is discussed experimentally, its theoretical value might be not known. To obtain the upper value of the active rate, we solve a sub-wavelength path assignment problem maximizing accommodated traffic demands exactly. This problem is related to integer multicommodity flow problems and bin packing problems, both of which are known as NP-hard. Therefore, it is difficult to solve our problem directly by formulating an integer optimization problem. In this study, we propose a cutting plane algorithm for the problem and verify the efficiency of the algorithm by numerical experiments. We also observe the active rates of networks.

For a large size instance, our cutting plane algorithm does not obtain an optimal solution in practical computational time. Thus, we modify the optimization problem for which the cutting plane algorithm obtains a near optimal solution. In addition, we propose a simple heuristic algorithm and verify that the algorithm runs faster and obtains suitable solutions.

The rest of the paper is organized as follows. In Section 2, we formulate our problems as integer programming problems. In Section 3, we propose a cutting plane algorithm for the problems, and describe methods to obtain upper and lower bounds. In addition, a simple greedy-type algorithm is presented. This algorithm can be applied for cases given dynamic demands. In Section 4, we show the experimental results for several instances. Finally in Section 5, we summarize our conclusions.

2 Problem formulation

We represent an (physical) optical network by graph $(N, \tilde{F})$, where N is the set of nodes such as switches and communication endpoints and so on, and $\tilde{F}$ is the set of fibers. Each fiber in $\tilde{F}$ is replaced by a pair of unidirectional links in opposite directions. The set of such links is given by F. Let W be a set of wavelengths available at each fiber and g the capacity of a single wavelength.

We denote a set of traffic demands by D. Each traffic demand $d \in D$ has source-destination pair (s_d, t_d) and bandwidth c_d. We assume a symmetric traffic communication, that is, an up-line requires the same transmission capacity as that of an down-line and the up-line and the down-line use the same route with the opposite direction. Thus, we only consider the up-lines in our model. Namely, we assume that each traffic demand d is unidirectional, i.e., the routing between a given source-destination pair is a directed path from the source to the destination in (N, F). Our model assumes that traffic grooming is allowed, i.e., multiplexing of many lower-rate traffic demands onto one wavelength, and demultiplexing them. That is to say, several traffic demands with low bandwidths can be combined in order to be served through one wavelength on a link. We also assume that wavelength conversions are available in every node. Thus, it is not required that the wavelengths on all links of a route are same. Therefore, our network model is one of the virtual wavelength path (VWP) network models (Sato et al. 1994). We need to determine a sub-wavelength path from s_d to t_d and a wavelength for each link in the path for each traffic demand d.

Our objective is to find how many traffic demands are served and to analyze the activity rate in our model.

At first, we consider a static case, that is, all traffic demands D are given beforehand. We introduce a 1-0 variable z_d for whether traffic demand d is served or not. Since we need to determine routing for each traffic demand and assignment of wavelength of each link in the route, we introduce a 1-0 variable x_{fwd} for whether traffic demand d is used link $f \in F$ with wavelength $w \in W$ or not. The routing is restricted by a single path between the source-destination pair, and the wavelength assignment is restricted by its capacity.

When we consider the static model, a solution maximizing the number of served traffic demands tends to accommodate traffic demands with lower bandwidth and shorter route. However, it lacks fairness and equality of network users. Therefore, in order to exclude arbitrariness from accommodations of traffic demands, we restrict accommodations in order of precedence of traffic demands. For this precedence constraint, assume that D is given by consecutive integers $\{1, 2, \ldots, |D|\}$ and that traffic demand d is not served unless all of traffic demands $1, \ldots, d-1$ are served.

Our problem is formulated as below, where $\delta^+ i$ and $\delta^- i$ are the sets of links from node i and into node i, respectively, and f^{r} means the reverse direction link of f.

$$
(\mathrm{P}_1) \left|
\begin{array}{lr}
\text{maximize} \sum_{d \in D} z_d & (1) \\
\text{subject to} \sum_{f \in \delta^+ i} \sum_{w \in W} x_{fwd} - \sum_{f \in \delta^- i} \sum_{w \in W} x_{fwd} = \begin{cases} z_d & (i = s_d) \\ -z_d & (i = t_d) \\ 0 & (\text{otherwise}) \end{cases} \quad (i \in N,\ d \in D) & (2) \\
\sum_{d \in D} c_d (x_{fwd} + x_{f^{\mathrm{r}}wd}) \leq g \quad (f \in F,\ w \in W) & (3) \\
z_d \geq z_{d+1} \quad (d \in D) & (4) \\
x_{fwd} \in \{0, 1\} \quad (f \in F, w \in W, d \in D) & (5) \\
z_d \in \{0, 1\} \quad (d \in D) & (6)
\end{array}
\right.
$$

Equation (1) is the objective function maximizing the total number of served traffic demands. The first constraint, Eq. (2), guarantees that a traffic demand is routed by a single path. Equation (3) stands for the capacity constraint of wavelengths. Equation (4) is the precedence constraint. Equations (5) and (6) represent that each variable is given by binary.

The problem without the precedence constraint is referred to as (P_2).

When using an appropriate solver, i.e., a software to solve integer optimization problems, adding some valid inequalities might make to find an optimal solution faster (see Nemhauser and Wolsey, 1998, for example). We introduce two types of valid inequalities for capacity constraints: For any integer $k (\geq 2)$,

$$\sum_{\substack{d \in D \\ c_d > g/k}} (x_{fwd} + x_{f^{\mathrm{r}}wd}) \leq k - 1 \quad (f \in F, w \in W), \tag{7}$$

and, for any $p (\geq 1/2)$,

$$\sum_{\substack{d \in D \\ c_d > pg}} x_{fwd} + \sum_{\substack{d \in D \\ g/2 \geq c_d > \max\{g/3, (1-p)g\}}} \frac{(x_{fwd} + x_{f^{\mathrm{r}}wd})}{2} \leq 1 \quad (f \in F, w \in W). \tag{8}$$

Equation (7) means that at most $k - 1$ traffic demands having bandwidth greater than g/k can be served in a wavelength of each link. When a traffic demand having bandwidth greater than pg is served in wavelength w of the link f, equation (8) is to limit served traffic demands in the residual capacity.

3 Algorithms

3.1 Exact algorithm for static cases

We first consider the algorithm to obtain exact optimal solutions of (P_1) and (P_2). The exact optimal solutions are needed to estimate the active rate of networks. Unfortunately, the problems (P_1) and (P_2) have too many variables to solve them by using solvers. Thus, we employ the cutting plane method by relaxing the capacity constraints. This relaxed problem ignores wavelength assignment. We introduce a variable x_{fd}, which is equivalent to $\sum_{w\in W} x_{fwd}$. Thus, the first constraint Eq. (2) of (P_1) (resp. (P_2)) can be written by

$$\sum_{f\in\delta^+ i} x_{fd} - \sum_{f\in\delta^- i} x_{fd} = \begin{cases} z_d & (i = s_d) \\ -z_d & (i = t_d) \\ 0 & \text{(otherwise)} \end{cases} \qquad (i \in N,\ d \in D),$$

and the capacity constraint Eq. (4) is replaced by the constraint for capacities for fibers:

$$\sum_{d\in D} c_d(x_{fd} + x_{f^{\mathrm{r}}d}) \le g|W| \ \ (f \in F). \tag{9}$$

Moreover, the valid inequalities Eqs. (7) and (8) are written by

$$\sum_{\substack{d\in D \\ c_d > g/k}} x_{fd} + x_{f^{\mathrm{r}}d} \le (k-1)|W| \ \ (f \in F), \tag{10}$$

and

$$\sum_{\substack{d\in D \\ c_d > pg}} x_{fd} + \sum_{\substack{d\in D \\ g/2 \ge c_d > \max\{g/3,(1-p)g\}}} \frac{(x_{fd} + x_{f^{\mathrm{r}}d})}{2} \le |W| \ \ (f \in F). \tag{11}$$

We denote these relaxed problems of (P_1) and (P_2) with the above valid inequalities by ($\overline{\mathrm{P}_1}$) and ($\overline{\mathrm{P}_2}$), respectively.

For an optimal solution x^* of ($\overline{\mathrm{P}_1}$) (resp. ($\overline{\mathrm{P}_2}$)) and link $f \in F$, let $D(f) = \{d \in D \mid x^*_{fd} + x^*_{f^{\mathrm{r}}d} = 1\}$. If the traffic demands in $D(f)$ can be assigned to wavelengths in W with satisfying the capacity constraint for each link $f \in F$, that is, x^* is feasible for (P_1) (resp. (P_2)), the solution is optimal for (P_1) (resp. (P_2)). We can check feasibility of x^* for (P_1) (resp. (P_2)) by solving the following problem:

$$(\mathrm{BP}(f)) \left| \begin{array}{ll} \text{maximize} & \displaystyle\sum_{d\in D(f)} \sum_{w\in W} \lambda^{(f)}_{wd} := \Lambda(f) \\ \text{subject to} & \displaystyle\sum_{d\in D(f)} c_d \lambda^{(f)}_{wd} \le g \ \ (w \in W) \\ & \displaystyle\sum_{w\in W} \lambda^{(f)}_{wd} \le 1 \ \ (d \in D(f)) \\ & \lambda^{(f)}_{wd} \in \{0,1\} \ \ (w \in W, d \in D(f)) \end{array} \right.$$

If $|D(f)| = \Lambda(f)$ holds for all $f \in F$, then we can obtain a feasible wavelength assignment. If there exists $h \in F$ with $|D(h)| > \Lambda(h)$, we can add to ($\overline{\mathrm{P}_1}$) (resp. ($\overline{\mathrm{P}_2}$)) the cutting plane

$$\sum_{d\in D(h)} (x_{fd} + x_{f^{\mathrm{r}}d}) \le \Lambda(h) \ \ (f \in F). \tag{12}$$

Our algorithm is described as below:

Step 0 Let $(\overline{\mathrm{P}_1})^+$ (resp. $(\overline{\mathrm{P}_2})^+$) be the problem ($\overline{\mathrm{P}_1}$) (resp. ($\overline{\mathrm{P}_2}$)).

Step 1 Obtain an optimal solution x^* by solving $(\overline{\mathrm{P}_1})^+$ (resp. $(\overline{\mathrm{P}_2})^+$) and find $D(f)$ with respect to x^* for each $f \in F$.

Step 2 For each $f \in F$, solve (BP(f)). Let $H = \{f \in F \mid |D(f)| > \Lambda(f)\}$. If $H = \emptyset$, then Stop. We can obtain an optimal solution from x and λ. Otherwise, add the constraint

$$\sum_{d\in D(h)} (x_{fd} + x_{f^{\mathrm{r}}d}) \le \Lambda(h) \ \ (f \in F, h \in H) \tag{13}$$

to $(\overline{\mathrm{P}_1})^+$ (resp. $(\overline{\mathrm{P}_2})^+$) and go to Step 1. □

This algorithm can find exact optimal solutions for instances having not so many traffic demands and several wavelengths in each fiber. However, the larger the numbers of traffic demands or wavelengths in a fiber become, the more iterations of the algorithm might be needed, since the patterns of the set $D(f)$ may increase exponentially. In this case, we need to stop the algorithm before obtaining an optimal solution. Then, we get an upper bound of the optimal value from the objective value of $(\overline{\mathrm{P}_1})^+$ (resp. $(\overline{\mathrm{P}_2})^+$). On the other hand, the problem is referred to as $(\overline{\mathrm{P}_1(\alpha)})$ (resp. $(\overline{\mathrm{P}_2(\alpha)})$ if the capacity constraint Eq. (9) of $(\overline{\mathrm{P}_1})$ (resp. $(\overline{\mathrm{P}_2})$) is replaced by

$$\sum_{d \in D} c_d(x_{fd} + x_{f^{\mathrm{r}}d}) \leq \alpha g|W| \ \ (f \in F)$$

for $0 < \alpha < 1$. By replacing **Step 0** of our algorithm by

Step 0' Let $(\overline{\mathrm{P}_1})^+$ (resp. $(\overline{\mathrm{P}_2})^+$) be the problems $(\overline{\mathrm{P}_1(\alpha)})$ (resp. $(\overline{\mathrm{P}_2(\alpha)})$),

we can find a feasible solution, which gives a lower bound. This algorithm with a small value α finds a feasible solution rapidly, and it does a nearly optimal solution by setting α to nearly 1. By setting an appropriate value to α, we can employ a feasible solution instead of an optimal solution, when the gap between upper and lower bounds is sufficiently small.

3.2 Greedy-type algorithm

To obtain a suitable feasible solution efficiently, we propose a simple heuristic algorithm. The algorithm iterates to find a path constructed from links including a wavelength with the enough rest capacity for a demand. Recall that D is given by consecutive integers $\{1, 2, \dots, |D|\}$. For a feasible solution $(x_{fwd})_{f \in F, w \in W, d \in D}$ and a demand $\tilde{d}$, let $F_{\tilde{d}}(x)$ be the set of links $f \in F$ such that there exists wavelength $w \in W$ with $g - \sum_{d<\tilde{d}}(x_{fwd} + x_{f^{\mathrm{r}}wd}) \geq c_{\tilde{d}}$. Our algorithm is greedy-type, because we decide whether a demand can be served or not sequentially one by one. This greedy-type algorithm is described as below:

Step 0 Set $x_{fwd} \leftarrow 0$ for all $f \in F, w \in W, d \in D$ and $\tilde{d} = 1$.

Step 1 Find a shortest path P from $s_{\tilde{d}}$ to $t_{\tilde{d}}$ in $(N, F_{\tilde{d}}(x))$. If $t_{\tilde{d}}$ is not reachable from $s_{\tilde{d}}$ in $(N, F_{\tilde{d}}(x))$, then go to Step 3.

Step 2 For each link $f \in P$, find $w* \in \arg\min\{g - \sum_{d<\tilde{d}}(x_{fwd} + x_{f^{\mathrm{r}}wd}) \mid w \in W, g - \sum_{d<\tilde{d}}(x_{fwd} + x_{f^{\mathrm{r}}wd}) \geq c_{\tilde{d}}\}$ and update $x_{fw^*\tilde{d}} \leftarrow 1$.

Step 3 $d \leftarrow d + 1$. If $d \leq |D|$, go to Step 1. $\Box$

We perform **Step 1** by executing the breadth first search. **Step 2** is regarded as a step of the best fit algorithm for the bin packing problem for each link in P. Each iteration runs in $\mathrm{O}(|F| + |W|)$ time. If we implement by using a suitable data structure like a priority queue, it runs in $\mathrm{O}(|F| + \log |W|)$ time.

Note that this greedy-type algorithm can be applied when each demand has to be decided to be served as soon as it occurs.

4 Experimental results

In this section, we evaluate active rates for several instances by using the exact algorithm, and accuracy of the greedy-type algorithm. The active rate is defined by

$$\frac{\sum_{f \in F} \sum_{w \in W} \sum_{d \in D} c_d x_{fwd}}{|\tilde{F}||W|} \tag{14}$$

for a solution $(x_{fwd})_{f \in F, w \in W, d \in D}$, which means the rates of used resource on the network. We generate two types of instances. In instances of Type 1, bandwidth c_d of traffic demand d is randomly chosen from $\{0.1, 0.2, \dots, 0.9, 1.0\}$. In Type 2, we adopt only 0.1, 0.4, and 1.0 bandwidths of traffic demands. Type 2 follows 10/40/100 Gbps system used for the current communication on optical networks. In order to measure activity rates, we use Type 1 instances. On the other hand, we employ Type 2 instances to establish the accuracy of the greedy-type algorithm in realistic situations.

We adopt two graphs. One is a 5×5 grid graph that is configured as Fig. 3. The other is Japan Photonic Network Model (JPN) 12. JPN has been developed based on the publicly available information of nationwide transportation network by Technical Committee on Photonic Network and is published at `http://www.ieice.org/~pn/jpn/jpnm.html`. We use JPN12 which is constructed from 12 city nodes.

In all of our instances, the single wavelength capacity g is fixed to 1. The exact algorithm is executed by FICO Xpress 7.6. The valid inequalities of Eq. (10) are adapted for $k = 2, 3, 4$ and of Eq. (11) for $p = 0.6$ and 0.7. In our experiments, we perturbe the objective function as

$$(|F||D| + 1) \sum_{d \in D} z_d - \sum_{f \in F} \sum_{d \in D} x_{fd},$$

which means that a solution with shorter routes tends to be chosen when there are a lot of optimal solutions. The greedy-type algorithm is implemented by Java. Our numerical experiments are done on Intel Core$^{\text{TM}}$i7 (3.20 GHz) processor with 12.0 GB memory.

4.1 Active rates

By using instances of Type 1, we evaluate active rates given by Eq. (14).

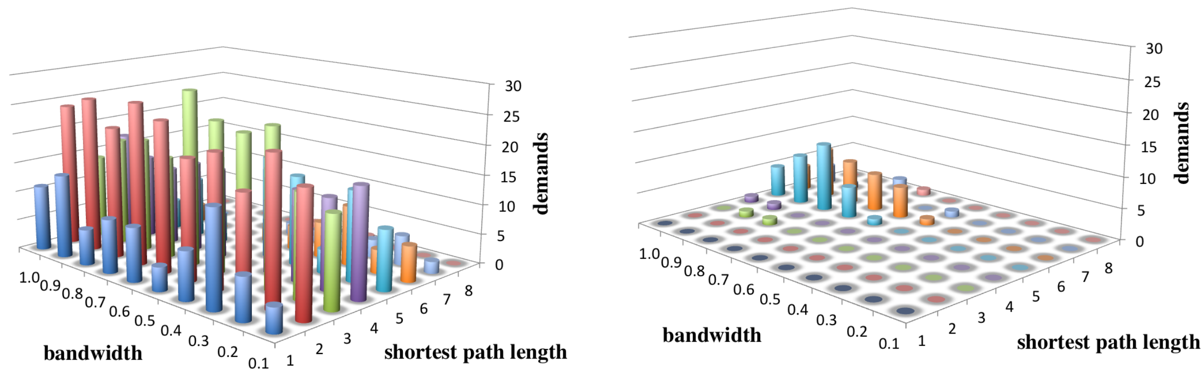
At first, we generated 10 instances (A – J) having 5 wavelengths per a fiber and 100 traffic demands on the 5×5 grid graph, where each pair of source s_d and destination t_d for traffic demand d was given by randomly. The computational results obtained by our exact algorithm are shown in Table 1, where "opt.value", "iterations", and "time" show the number of accommodated traffic demands, the number of iterations and the computational time of our algorithm, respectively. For instance C for (P_2), our algorithm stops before obtaining an optimal solution because the upper bound of iterations is set to 99.

Table 1: Computational results for Type 1 instances on the 5×5 grid graph. Each instance has 5 wavelengths per a fiber and 100 traffic demands.

	P_1 (with the precedence constraint)				P_2 (without the precedence constraint)			
instance	opt. value	iterations	time (sec)	active rate	opt. value	iterations	time (sec)	active rate
A	70	33	4248	0.63	89	2	7288	0.71
B	66	0	39	0.69	86	15	1862	0.71
C	86	0	33	0.68	–	99	250268	–
D	74	5	475	0.69	92	0	7	0.79
E	69	8	736	0.68	93	10	1021	0.76
F	65	8	736	0.63	89	12	29986	0.69
G	66	8	771	0.66	92	26	19034	0.72
H	68	54	56151	0.68	94	23	93541	0.72
I	80	17	1970	0.72	94	8	1008	0.75
J	69	2	235	0.68	93	10	1017	0.76

The precedence constraint makes about 20% decrease of the number of accommodated traffic demands. Indeed, exclusion of the precedence constraint allows to choose traffic demands having shorter shortest path and fewer bandwidth. Figure 1 shows the numbers of served traffic demands and unserved traffic demands in (P_2) according to its bandwidth and the length of a shortest path between its source-destination pair, among 900 traffic demands except in instance C. Figure 2 shows the ratios of them. Although the served traffic demands by (P_1) may be longer path and bigger bandwidth than ones by (P_2), the active rates are worse at

(P_1) than (P_2), which may be effected by the number of served traffic demands. Figure 3 shows the average of active rates among all instances for each fiber. We can observe that the active rate tends to be higher at the center of the network.



served traffic demands **unserved traffic demands**

Figure 1: The numbers of served and unserved traffic demands in P_2 without the precedence constraint.

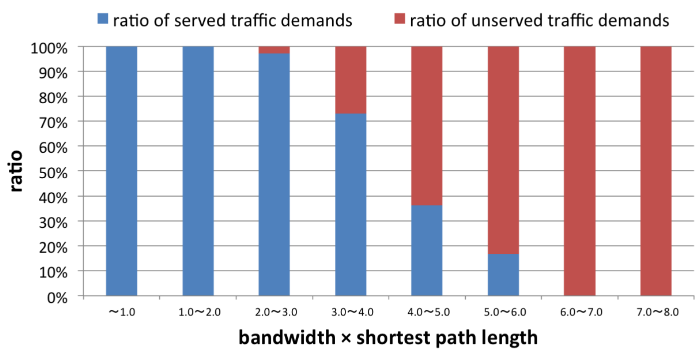


Figure 2: The ratio of served and unserved traffic demands in P_2 without the precedence constraint.

The computational time of our exact algorithm is not so fast, especially when solving (P_2). When the given traffic demands are increased to 200 by adding new 100 traffic demands, we inspected that the algorithm solving (P_2) needs more computational time. (Note that the solutions do not change for (P_1) even if additional traffic demands are given, because of the precedence constraint.) In order to obtain appropriate feasible solutions quickly, we execute heuristic algorithm which uses $(\overline{P_2(\alpha)})$. We set $\alpha = 0.95$ and restrict computational time as at most 86400 sec (= 24 hours). If the algorithm does not stop within 86400 sec, we reduce α by 0.05. In order to evaluate these lower bounds, we obtain upper bounds by solving the exact algorithm with time restriction of 3600 sec. Table 2 shows the result. In our experiment, we obtained feasible solutions whose approximation ratios were at least 0.9. By relaxing the capacity constraint, we can get appropriate feasible solutions although how to determine a suitable value α remains our problem. We observe that the active rate is improved since there are several traffic demand candidates to be accommodated. The difference of active rates between with the precedence constraint and without it becomes striking by increasing traffic demands.

We next establish the instances (K – T) of Type 1 with 10 wavelengths and 100 traffic demands on JPN12.

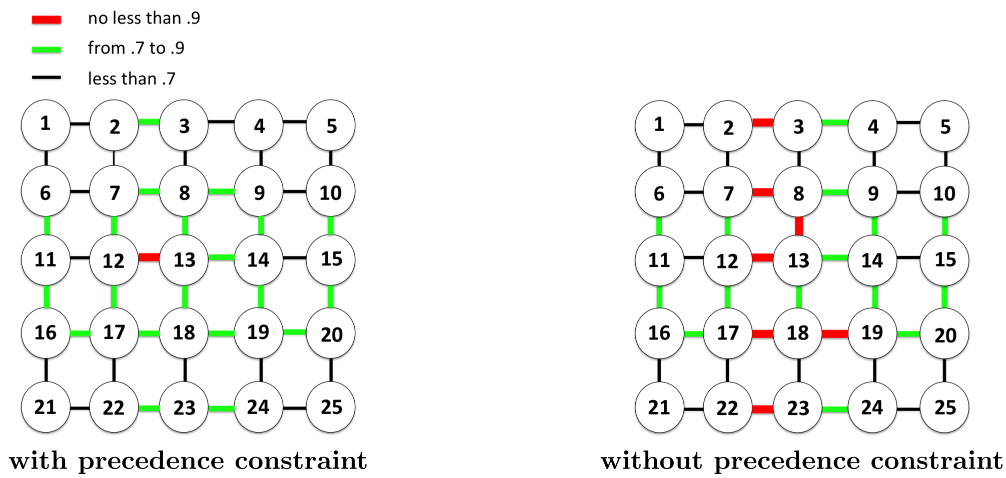


Figure 3: Average active rate of each fiber on the 5×5 grid graph.

Table 2: Lower and upper bounds for (P_2) on the 5×5 grid graph with 5 wavelengths with 200 traffic demands

instance	lower bound				upper bound	
	value	time (sec)	α	active rate	value	time (sec)
A	133	3658	0.95	0.83	137	3600
B	129	6802	0.80	0.70	141	3600
C	148	1534	0.85	0.74	158	3600
D	142	3650	0.95	0.85	146	2654
E	130	27197	0.85	0.75	141	3600
F	137	60984	0.90	0.79	143	3600
G	130	877	0.85	0.74	140	3600
H	134	5860	0.80	0.70	147	3600
I	135	3648	0.80	0.73	146	2651
J	142	33973	0.95	0.83	146	3600

Each pair of source and destination for a traffic demand is given by randomly according to the population of the corresponding city. Table 3 shows the result for evaluating 10 instances. For the most of instances, our exact algorithm finds an optimal solution without adding any cutting planes Eq. (12). Active rates are lower than that of the 5×5 grid graph. This might be caused by the configurations of graphs, and the deviation of a distribution of traffic demands. We also confirm on JPN12 that precisely picking out demands affects active rates. The average active rate of each fiber is drown in Fig. 4. We can see the bottlenecks are between Tokyo and Nagoya and between Tokyo (Hachioji) and Nagano.

Finally, we compare active rates by types of data. As we see, the active rates for Type 1 instances are around 0.65–0.70 for P_1 and 0.70–0.75 for P_2 on the 5×5 grid graphs. When we adopt the demand capacity of Type 2, active rates tend to be not so different from ones of Type 1, although they are slightly better than ones of Type 1. This tendency is the same to instances on JPN12. From these computational results, we see that the active rates are affected by underlying graphs and locations of source-destination pairs of demands.

Table 3: Computational results for Type 1 instances on JPN12. Each instance has 10 wavelengths per a fiber and 100 traffic demands.

	P_1 (with the precedence constraint)				P_2 (without the precedence constraint)			
instance	opt. value	iterations	time (sec)	active rate	opt. value	iterations	time (sec)	active rate
K	64	0	10	0.41	93	0	6	0.52
L	55	0	10	0.43	85	0	6	0.49
M	75	1	32	0.54	95	0	6	0.60
N	50	0	10	0.38	82	0	6	0.49
O	52	3	68	0.44	80	0	6	0.49
P	54	0	11	0.37	83	0	6	0.43
Q	55	0	11	0.45	90	0	6	0.55
R	69	0	11	0.46	89	0	6	0.51
S	69	2	53	0.42	90	1	18	0.53
T	51	0	10	0.44	79	0	6	0.48

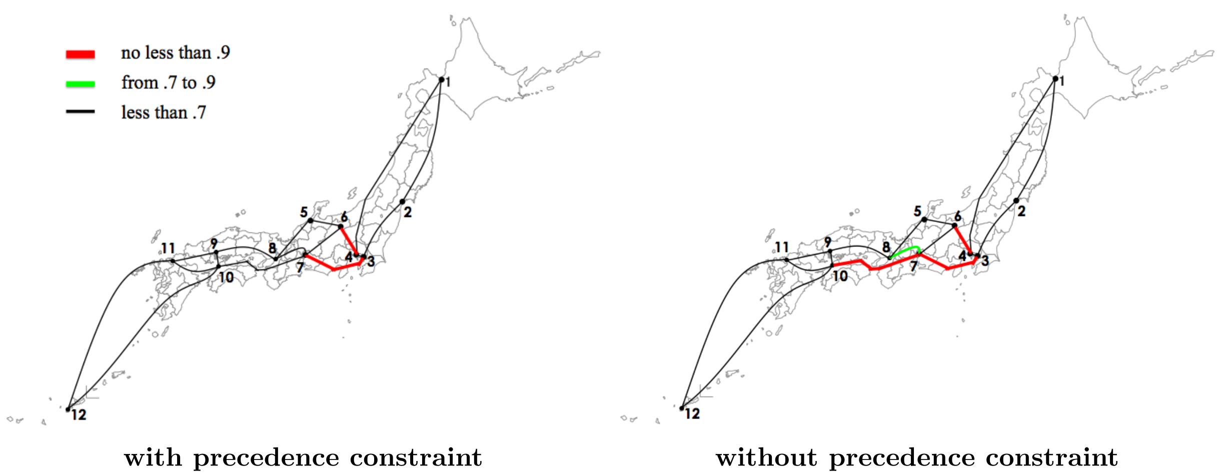


Figure 4: Average active rate of each fiber on JPN12

4.2 Efficiency of greedy-type algorithm

We compare solutions obtained by the exact algorithm and by the greedy-type algorithm for realistic instances, Type 2, in order to evaluate the greedy-type algorithm. Recall that an instance of Type 2 has bandwidths randomly chosen from only $\{0.1, 0.4, 1.0\}$. Note that, when the bandwidth is limited to $\{0.1, 0.4, 1.0\}$, we can find an exact optimal solution without adding any cutting planes Eq. (12) in the exact algorithm. For each link f, the valid inequality Eq. (10) for $k = 3$ gives the upper bound of the number of traffic demands with bandwidth 0.4 as twice the number of wavelengths except for assigned to traffic demands having bandwidth 1.0. Therefore, all traffic demands having bandwidth 0.4 or 1.0 can be assigned to wavelengths in accordance with the solution of relaxing problem ($\overline{\mathrm{P}_1}$) (resp. ($\overline{\mathrm{P}_2}$)). In addition, traffic demands having bandwidth 0.1 are able to be accommodated in the residual wavelength capacities. So, we expect to obtain an optimal solution of Type 2 faster than one of Type 1.

On the 5×5 grid graph, we generated 5 instances (a – e), where each fiber has 5 wavelengths and 100 traffic demands have randomly chosen pairs of a source and a destination. The computational results obtained by our exact algorithm and greedy-type algorithm are shown in Table 4. In the table, the columns of "opt. value" and "value" show the numbers of accommodated traffic demands, and the columns of "time" mean computational times. The column of "first loss" gives the demand number which is not served first by the greedy-type algorithm. That is to say, if we considered the precedence constraint, then the greedy-type algorithm returned the solution which accommodates all traffic demands in front of the first loss demand. The first loss demand could be served if a route is decided accurately when it is less than the optimal value of P_1. The difference between the demand number of the first loss and the optimal value of P_1 might describe the dissatisfaction of unserved demands. Although the demand number of the first loss is smaller than the optimal value of P_1, the total number of accommodated demands is greater than the optimal value of P_1 and is near the optimal value of P_2. In addition, the greedy-type algorithm runs by far faster than the exact algorithm. Thus we can conclude that the greedy-type algorithm is practical for these instances.

Table 4: Comparing the exact algorithm and greedy-type algorithm for Type 2 instances on the 5×5 grid graph

	exact algorithm				greedy-type algorithm		
	P_1		P_2				
instance	opt.value	time (sec)	opt. value	time (sec)	value	time (sec)	first loss
a	84	4	91	3600	84	5×10^{-4}	65
b	80	9	93	3600	82	5×10^{-4}	53
c	80	6	94	2	86	4×10^{-4}	69
d	82	5	96	2	89	5×10^{-4}	77
e	74	4	95	3	91	5×10^{-4}	59

It is obvious that the solution obtained by the greedy-type algorithm depends on the order of demands. The precedence constraint is also affected by the order of demands. To investigate the effect of the order of demands, we resolve each instance of (a–e) 1000 times by reordering demands randomly. Table 5 shows the statistical data of the results. In the table, the columns of "max" and "min" show the largest numbers and the smallest numbers, respectively, of accommodated traffic demands. The columns of "ave", "med" and "s.d." show the averages, medians and standard deviations, respectively, of the numbers of accommodated traffic demands.

Table 5: The statistical data of the result for Type 2 instances on the 5×5 grid graph

	exact algorithm					greedy-type algorithm									
	P_1 (with the precedence constraint)					solution					until the first loss				
instance	max	min	ave	med	s.d.	max	min	ave	med	s.d.	max	min	ave	med	s.d.
a	87	53	74.31	75	5.20	90	74	83.77	84	2.54	87	37	64.13	65	7.11
b	91	60	78.19	79	4.85	89	72	83.20	83	3.02	80	35	58.96	59	8.06
c	93	59	81.92	82	4.93	94	79	87.78	88	2.29	88	29	67.98	69	7.42
d	96	69	85.93	86	4.37	94	75	88.72	89	2.42	90	32	67.58	69	9.31
e	95	66	82.83	83	4.91	91	75	85.76	86	2.55	81	27	61.63	63	8.62

The average numbers of accommodated demands by the greedy-type algorithm tend to be larger than that by P_1. There exist, however, 0.3 % instances where the greedy-type algorithm serves more traffic demands than the exact algorithm for P_1. On the other hand, the standard deviations of the greedy-type algorithm's solutions are the smallest. Although the difference of the largest number and the smallest number of accommodated traffic demands by P_1 is approximately 35%, the difference between them is less than 20% by the greedy-type algorithm. So, the effect of the order of traffic demands is few for the greedy-type algorithm. From this point of view, we also conclude that solution found by greedy-type algorithm is passable accurate on the 5×5 grid graph. The difference of active rates is not large among the algorithms. Since there are many detours on the 5×5 grid graph, a lot of traffic demands tend to be accommodated by long paths.

We next examine instances of Type 2 on JPN12. We generate 5 instances (f–j) where each fiber has 10 wavelengths and each of 100 traffic demands has a source–destination pair given by randomly according to the population of the corresponding city. The computational results obtained by our algorithms are shown in Table 6.

Table 6: Comparing the exact algorithm and greedy-type algorithm for Type 2 instances on JPN12

	exact algorithm				greedy-type algorithm		
	P_1		P_2				
instance	opt. value	time (sec)	opt. value	time (sec)	value	time (sec)	first loss
f	92	3	97	1	95	2×10^{-4}	86
g	63	1	88	1	78	3×10^{-4}	63
h	53	1	86	1	82	3×10^{-4}	53
i	75	2	90	1	88	5×10^{-4}	75
j	61	2	92	1	81	4×10^{-4}	61

Like the 5×5 grid graph, the number of accommodated traffic demands by the greedy-type algorithm is greater than the optimal value of P_1. Table 7 shows the statistics data for resolving 1000 times for each instance of f–j by reordering the traffic demands.

Table 7: The statistical data of the result for Type 2 instances on JPN12

	exact algorithm					greedy-type algorithm									
	P_1 (with the precedence constraint)					solution					until the fist loss				
instance	max	min	ave	med	s.d.	max	min	ave	med	s.d.	max	min	ave	med	s.d.
f	97	65	86.87	87	5.01	97	92	95.39	95	0.92	97	65	86.19	87	5.00
g	82	43	63.23	63.5	5.96	87	68	80.64	81	3.30	82	43	62.91	63	6.04
h	76	39	58.76	59	5.89	85	70	80.79	81	2.51	75	39	58.32	58	5.88
i	86	46	67.36	68	6.33	90	73	85.08	85	2.61	83	46	66.58	67	6.22
j	89	47	71.69	72	6.44	92	77	86.465	87	2.87	86	47	70.32	71	6.29

As the case of the 5×5 grid graph, the average number of accommodated traffic demands by the greedy-type algorithm also tends to be larger than that of P_1. On JPN12, the number of accommodated demands by P_2 and that by the greedy-type algorithm accord in about 0.5%. Especially, in 10% execution for instance f, the greedy-type algorithm succeeds to accommodate as many as traffic demands accommodated by P_2. The standard deviations by the greedy-type algorithm are small rather than ones by P_1. So, we conclude that the greedy-type algorithm also finds solutions with good accuracy on JPN12.

In JPN12, we see that the average numbers of accommodated traffic demands by the greedy-type algorithm are closer to ones by P_1 rather than by P_2. It is caused by asymmetricity of JPN12 unlike the 5×5 grid graph. There are fewer edge-disjoint paths between source-destination pairs in JPN12 than in the 5×5 grid graph. So, the number of detours is small, which derives that few difference in numbers of accommodated traffic demands.

5 Conclusion

In this paper, we studied a RWA problem with considering sub-wavelength assignment on optical networks. We formulated static models of this problem as integer programming problems and proposed an algorithm to obtain an exact optimal solution by cutting plane method.

By using our algorithm, we evaluated active rates for instances with a few wavelength and 100-200 traffic demands on the 5×5 grid graph and JPN12. For instances with 5 wavelength on the 5×5 grid graph, the active rates are approximately 60–75%. However, for instances with 10 wavelengths on JPN12, the active rates are as low as 43%. Our experiments show that the active rates are affected by underlying graphs. Indeed, we see that the bottleneck fibers on JPN12 are between Tokyo and Nagoya and between Tokyo (Hachioji) and Nagano. This fact might be caused by not only configurations of graphs but also traffic distribution according to the population of corresponding cities. A further experiment establishing an additional fiber between Tokyo and Nagoya showed that the active rate was improved the average to 62%. In this way, for the effective utilization of the network resources, our exact algorithm can contribute for a network design including the decision of extension of fibers.

Moreover, we compared the numbers of accommodated traffic demands on realistic situations which follows 10/40/100 Gbps system used for the current communication on optical networks. Our numerical experiments verified that the greedy-type algorithm was useful in almost cases. The greedy-type algorithm also works for on-line cases, where each traffic demand is served when it occurs.

Our future work is to experiment by increasing the number of wavelength so that the instances can be said to be practical. Furthermore, additional experiments for dynamic traffic demands are needed in order to verify the effect of the greedy-type algorithm for on-line cases.

References

[1] Hu, J. Q. and Leida B., Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks, Proceedings on IEEE INFOCOM 2004 (2004).

[2] Ishii, K., Kurumida J. and Namiki S., Towards large-capacity, energy-efficient, and sustainable communication networks. Synthesiology, Vol. 7, No. 1 (2014), pp. 30–43.

[3] Liu, Z. and Rouskas G. N., Scalable optimal traffic grooming in WDM rings incorporating fast RWA formulation, Proceedings of IEEE International Conference (2013), pp. 3958–3962.

[4] Nemhauser, G. L. and Wolsey L. A., Integer and Combinatorial Optimization (1988), Wiley, Interscience.

[5] Ramaswami, R., Sivarajan K. N. and Sasaki G. H., Optical Networks: A Practical Perspective (2010), Morgan Kaufmann, Burlington.

[6] Sato, K., Okamoto S. and Hadama H., Network performance and integrity enhancement with optical path layer technologies, IEEE Journal on Selected Areas in Communications, Vol. 12, No. 1 (1994), pp.159–170.

[7] Yetginer, E. and Rouskas G. N., Power efficient traffic grooming in optical WDM networks, Proceedings of Global Telecommunications Conference (2009), pp.1–6.