Bounds for two static optimization problems on routing and spectrum allocation of anycasting

Yasutaka Miyagawa[a], Yosuke Watanabe[a], Maiko Shigeno[a,], Kiyo Ishii[b], Atsuko Takefusa[b,c], Akiko Yoshise[a]

[a] *University of Tsukuba Tsukuba, Ibaraki, 305-8573, Japan*
[b] *National Institute of Advanced Industrial Science and Technology (AIST) Tsukuba, Ibaraki, 305-8568, Japan*
[c] *National Institute of Informatics, Chiyoda-ku, Tokyo 101-8430, Japan*

Abstract

Elastic optical networks with optical-orthogonal frequency division multiplexing have been addressed enthusiastically for communication networks in the last decade because they result in high bandwidth efficiency. Routing and spectrum allocation (RSA) problems need to be solved when we transmit demands in an elastic optical network. This research deals with static RSA models for anycast transmission, which is one-to-one-of-many transmission in inter-datacenter networks. Two static RSA optimization models are considered. One minimizes the maximum number of spectrum slots needed to allocate given demands. The other maximizes the traffic volume of demands served under a given spectrum slot number. For both models, lower and upper bounds are developed in order to obtain exact optimal solutions. One-side bounds of the problems are evaluated by relaxing spectrum continuity constraints. For the other side bounds, several greedy algorithms are investigated. We conducted computational experiments to confirm whether relaxation problems can give tight bounds and to determine greedy algorithmic behaviors by using each of route selection criterion and each of demand ordering policies. The results show that the solutions obtained by relaxing spectrum continuity constraints are almost optimal. They also indicate that exact optimal solutions are obtained efficiently by using these bounds.

Keywords: elastic optical network, anycast, greedy algorithm, integer linear programming

1. Introduction

The rapid growth of internet traffic and new network applications has led to the development of high-capacity and cost-effective optical fiber transmission technologies. Optical networks based on wavelength division multiplexing (WDM) are not considered to be sufficient for the exponential growth of communication bandwidth demand. In the last decade, elastic optical networks with optical-orthogonal frequency division multiplexing (OFDM), which results

in high bandwidth efficiency, have been addressed enthusiastically for communication networks [1, 2, 3, 4, 5, 6].

Routing and spectrum allocation (RSA) problems need to be solved when we transmit demands in an elastic optical network. RSA problems are roughly divided into two models: static and dynamic. In static models, all demands are known in advance. The purposes of static models are designing networks including finding the minimum number of spectrum slots admitting all demands, calculating evaluation indices such as an active ratio, reallocating all demands when impairment occurs or when a centralized reallocation is possible like software definition networks, and so on [7, 8, 9, 10, 11, 12, 13, 14]. For some cases, exact optimal solutions are desired. However, in [7, 8], the NP-hardness of the static RSA problem was proved. Thus, we use integer linear programming (ILP) only when we have a lot of time to solve RSA. Heuristic algorithms are also developed for when we need to solve RSA quickly. As well as greedy approaches, several metaheuristic approaches have been investigated for RSA [11, 15, 16]. In dynamic models, once a demand arrives at the network, we immediately find a route for the demand with enough contiguous spectrum slots. Dynamic models intend to find good RSA rules so as to improve blocking ratios for practical use and applications. In most literature, greedy heuristic algorithms for dynamic models are evaluated [17, 18, 19, 20, 21, 22, 23].

Recently, not only unicast but also anycast transmission in elastic optical networks have been researched [11, 24, 25, 26, 27, 28, 29]. Anycasting is one-to-one-of-many transmission in inter-datacenter networks. It reduces latency and increases redundancy, and it can manage a huge peak throughput and high traffic burstiness. Thus, anycasting has attracted much attention in both academia and real-life implementations.

In this research, we deal with RSA problems for anycast transmission, where we employ the problem setting discussed in [11]. We consider two static optimization models: the minimax slot number model and maximum traffic volume model. The former minimizes the maximum number of spectrum slots needed to allocate given demands. The latter maximizes the traffic volume of demands served under a given spectrum slot number. For both models, lower and upper bounds are developed in order to obtain exact optimal solutions. We estimate the one-side bounds of problems by relaxing the constraint for allocation of the same spectrum over the links. For the other side bounds, we investigate the accuracy of greedy algorithms. There are a lot of variations of greedy algorithms, which play important roles not only in the dynamic model but also for heuristics in the static model. We reorganize these algorithms from three points of view: a criterion for selecting routes, spectrum allocation strategy, and policy for demand ordering processed one by one. To construct effective algorithms, it is important to see which is superior in terms of the route selection criterion, spectrum allocation strategy, and demand ordering policy and to verify the effect of combinations of these. In this paper, we focus on greedy algorithms for static RSA and confirm algorithmic behaviors by using a criterion for selecting routes and policy for demand ordering. In Abkenar–Rahbar [13], some greedy algorithms are also discussed and compared. A distinctive trait of our research

is establishing greedy algorithms in a structured manner.

The rest of the paper is organized as follows. In Section 2, we present our model with assumptions and notations and give integer linear programming (ILP) formulations. In Section 3, we describe relaxation problems for RSA models. We also formulate problems which find feasible RSA solutions by using solutions obtained by the relaxation problems. In Section 4, we describe greedy algorithms for RSA from the point of view of routing criterion and demand ordering policy. Numerical experiments are shown in Section 5. We evaluate the quality of solutions obtained by the greedy algorithms and characterize the route selection criteria and demand ordering policies by performing numerical experiments. We next find exact optimal solutions by using bounds derived from the greedy algorithms and the relaxation problems. Finally, Section 6 concludes our results.

2. Anycast RSA Model on Elastic Optical Networks

We represent an optical network by a bi-directed graph $G = (V, E)$, where elements in V stand for nodes of the network, that is, data centers, communication endpoints, switches, and so on, and elements in E are associated with fibers of the network. If $(u, v) \in E$, the opposite directional edge (v, u) also belongs to E. A whole spectrum of each fiber (edge) is sliced into a lot of small spectrum slots. Assume that the sliced spectrum slots are indexed by consecutive numbers in accordance with frequencies. Let $S = \{1, 2, \ldots, s, \ldots\}$ be a set of indices of the sliced spectrum slots.

We consider anycasting, which is one of the transmission types for network communication, and assume that the optical network is an intra-datacenter network, where a set of data centers is denoted by $C(\subseteq V)$. A client node sends a query to any one of possible data centers. The data center that receives the query sends back its response to the client node. A set of queries is denoted by D. For query $d(\in D)$, $\tau(d)$ represents its response. Both a query and response are referred to as demands. Let $\tilde{D} = D \cup \{\tau(d) \mid d \in D\}$ be a set of demands. For each demand, we need to determine a route to transmit data with a possible data center according to the routing table. Routing is concerned with choosing a directed path between its client node and a data center in the network. For each query $d \in D$, let P_d be a set of candidate directed paths from the client node of d to a data center. For each response $d \in \{\tau(d) \mid d \in D\}$, P_d implies a set of candidate directed paths from a data center to its client node. Although the closest data center under a certain distance metric is usually chosen, it is important how a client selects the best data center and the best route between them. Our model allows for different routings for a query and for its response so long as a query-response pair accesses the same data center.

In elastic optical networks, we need to determine not only routing but also spectrum slots used to transmit each demand. The required number of spectrum slots for transmission depends on the volume of transmission data and on the length of a path chosen as a route. The larger the volume of data is and the longer the path length is, the more slots for transmission we need. Each demand

$d \in \tilde{D}$ has a predetermined volume of transmission data, which is represented by its bit-rate h_d. Usually, the volume of downstream demand is much larger than the volume of upstream demand, that is, $h_d \ll h_{\tau(d)}$ for $d \in D$. The number of spectrum slots needed to transmit demand d along path $p(\in P_d)$ is denoted by n_{dp}. According to [11], it is given by

$$n_{dp} = 2\lceil \frac{h_d}{4m_p\Delta_s} \rceil, \tag{1}$$

where a parameter m_p equals 1 when the length of path p is above 1500 km, $m_p = 2$ when it is above 750 km, $m_p = 3$ when it is above 375 km, $m_p = 4$ when it is below 375 km, and Δ_s is the width of frequency slice for which we use 6.25GHz. It is assumed that transponders operate with polarization division multiplexing for all m_p. If we consider spectrum guard bands, the number of slots for guard bands is added to (1). In spectrum slot allocation, all spectrum slots allocated for a demand must be adjacent, that is, a demand has to be allocated to slots having consecutive indices. In addition, the same spectrum slots must be allocated for a demand in each fiber on the path to transmit the demand. The former constraint is called "spectrum contiguity," and the latter is called "spectrum continuity." Moreover, each spectrum slot is allocated to at most one demand. Note that our model assumes that each data center has enough capacity and, hence, there are no constraints for data center capacities.

The above RSA problem for anycasting is formulated as an ILP formulation. Although discussion on performance with different ILP formulations can be found in [10, 29], our formulation corresponds to [11]. We employ a Boolean variable x_{dps} that equals 1 if demand d is carried along path $p \in P_d$ using spectrum slots from s.

$$\sum_{p \in P_d} \sum_{s \in S} x_{dps} = 1, \ \forall d \in \tilde{D} \tag{2}$$

$$\sum_{d \in \tilde{D}} \sum_{\substack{p \in P_d \\ e \in p}} \sum_{\substack{s' \in S \\ s - n_{dp} + 1 \le s' \le s}} x_{dps'} \le 1, \ \forall e \in E, \forall s \in S \tag{3}$$

$$\sum_{p \in P_d} \sum_{s \in S} head(p) x_{dps} = \sum_{p \in P_{\tau(d)}} \sum_{s \in S} tail(p) x_{\tau(d)ps}, \ \forall d \in D \tag{4}$$

$$\sum_{s \in S} (s + n_{dp} - 1) x_{dps} \le w, \ \forall d \in \tilde{D}, \ \forall p \in P_d \tag{5}$$

Constraint (2) assigns one path and one starting spectrum slot to each demand. Constraint (3) guarantees that each spectrum slot for each edge is allocated at most one demand. The left-hand-side of (3) represents the number of demands which use the sth slot in edge e. Constraint (4) ensures that the data center that receives a query sends its response. In (4), $head(p)$ and $tail(p)$ stand for the head (end node) and tail (start node) of path p. Constraint (5) represents that the index of allocated spectrum slots is no more than w, that is, the number of

slots prepared on each edge. Minimizing w under constraints (2)–(5), we can find the minimum number of spectrum slots on each edge needed to allocate all demands. We call this model the *minimax slot number model*. When we maximize the volumes of demands served under the given number of spectrum slots w, the problem, called the *maximum traffic volume model*, is maximizing

$$t := \sum_{d \in \tilde{D}} h_d z_d$$

subject to

$$\sum_{p \in P_d} \sum_{s \in S} x_{dps} = z_d, \ \forall d \in \tilde{D} \tag{6}$$

$$z_d = z_{\tau(d)}, \ \forall d \in D \tag{7}$$

and (3)–(5), where z_d is a Boolean variable that equals 1 if demand d is served. Note that w is a decision variable in the minimax slot number model, while it is a constant number in the maximum traffic volume model.

3. Relaxation Problem

Relaxation problems for optimization problems help in not only obtaining exact optimal solutions but also in evaluating solutions with heuristic algorithms. In this section, we consider relaxation problems for RSA models. By relaxing the condition of "spectrum continuity," we obtain a lower bound for the minimax slot number model. To do this, we solve the following problem.

$$(\mathrm{LB}_w) \quad \begin{array}{lll} \text{minimize} & w & \\ \text{subject to} & \displaystyle\sum_{p \in P_d} x_{dp} = 1, & \forall d \in \tilde{D} \quad (8) \\ & \displaystyle\sum_{p \in P_d} head(p) x_{dp} = \sum_{p \in P_{\tau(d)}} tail(p) x_{\tau(d)p}, & \forall d \in D \quad (9) \\ & \displaystyle\sum_{d \in \tilde{D}} \sum_{\substack{p \in P_d \\ e \in p}} n_{dp} x_{dp} \leq w, & \forall e \in E, \quad (10) \end{array}$$

where x_{dp} is a Boolean variable that equals 1 if demand d is carried along path p. This relaxation is also discussed in [8, 10]. Constraints (8) and (9) correspond to (2) and (4). From (3) and (5), we obtain

$$\sum_{d \in \tilde{D}} \sum_{\substack{p \in P_d \\ e \in p}} n_{dp} \sum_{s \in S} x_{dps} \leq w, \quad \forall e \in E, \tag{11}$$

which corresponds to (10). Since we relax the spectrum continuity constraint, we can allocate spectrum slots to each demand contiguously for each edge one

by one. This implies that the formulation does not represent spectrum slot allocation explicitly. For a feasible solution x and w for the ILP formulation of the minimax slot number model, by setting $x'_{dp} = \sum_{s \in S} x_{dps}$, a solution to x' and w is feasible in (LB_w). Thus, the optimal value of (LB_w) is a lower bound of the optimal slot number w^*.

We can derive a feasible solution for the original problem from a solution of this relaxed problem (LB_w). Let $\tilde{p}_d$ be the path selected for demand d in the solution of (LB_w). For these paths $\{\tilde{p}_d \mid d \in \tilde{D}\}$, we assign consecutive spectrum slots with the following problem.

$$
(\mathrm{UB}_w) \left| \begin{array}{lll}
\text{minimize} & w & \\
\text{subject to} & \displaystyle\sum_{s \in S} x_{ds} = 1, & \forall d \in \tilde{D} \qquad (12) \\
& \displaystyle\sum_{\substack{d \in \tilde{D} \\ e \in \tilde{p}_d}} \sum_{\substack{s' \in S \\ s - n_{d\tilde{p}_d} + 1 \le s' \le s}} x_{ds'} \le 1, & \forall e \in E,\ \forall s \in S \qquad (13) \\
& \displaystyle\sum_{s \in S} (s + n_{d\tilde{p}_d} - 1) x_{ds} \le w, & \forall d \in \tilde{D}, \qquad (14)
\end{array} \right.
$$

where x_{ds} is a Boolean variable that equals 1 if demand d is assigned to consecutive spectrum slots from s. Since (UB_w) gives a feasible solution, the objective value of (UB_w) is an upper bound of w^*.

A similar relaxation problem can be applied to the maximum traffic volume model. For a given number of spectrum slots w, the relaxation problem

$$
(\mathrm{UB}_t) \left| \begin{array}{ll}
\text{maximize} & \displaystyle\sum_{d \in \tilde{D}} h_d z_d \\
\text{subject to} & \displaystyle\sum_{p \in P_d} x_{dp} = z_d, \quad \forall d \in \tilde{D} \\
& (9),\ (10)
\end{array} \right.
$$

gives an upper bound of the maximum traffic volume t^*, and

$$
(\mathrm{LB}_t) \left| \begin{array}{ll}
\text{maximize} & \displaystyle\sum_{d \in \tilde{D}} h_d z_d \\
\text{subject to} & \displaystyle\sum_{s \in S} x_{ds} = z_d, \quad \forall d \in \tilde{D} \\
& (13),\ (14)
\end{array} \right.
$$

gives a lower bound of t^*.

4. Greedy Algorithms

Several greedy algorithms for RSA problems may proceed to find an available path and allocate spectrum slots for each demand one by one. A lot of the

algorithms consist of two-steps: the routing step and slot allocation step. The routing step selects an available path from a candidate path set by using some criterion, for example, the shortest distance or number of free slots (see, for example, [9, 19, 20, 23, 27]). If an available path is found for allocation, the routing step is stopped, and the spectrum allocation step is triggered for this path. The allocation step decides contiguous spectrum slots along the path by using some strategy, for example, first-fit or best-fit (see, for example, [19, 20, 22, 23]). When no available path is found, both the query and its response are rejected.

In addition to the variations of the criterion for selecting routes and for the strategy of spectrum slot allocation, greedy algorithms for static RSA problems depend on the ordering policy of demands to proceed one by one. Our research focuses on the criterion of selecting routes and on the policy of ordering demands. Some criteria for selecting routes use the results of spectrum slot allocation along a path. Such criteria assume that spectrum slots are allocated by following the classical first-fit rule, which allocates a demand to available consecutive slots having the minimum indices. We focus on the processing order of demands instead of the allocation strategy because static models know all demands a priori and attempt to serve all demands at the same time. We can see several variations in the processing order of demands in [8, 9, 22].

We take up five routing criteria and seven demand ordering policies. Let $\hat{P}_d$ be the set of available paths in P_d for demand d under the condition of several spectrum slots being occupied. The length of path p and the number of hops of path p are denoted by $len(p)$ and $hop(p)$, respectively. The index set of used spectrum slots in edge e is given by $\hat{S}(e)$, and the index set of spectrum slots for allocating the target demand along path p by following the first-fit rule is given by $S_{new}(p)$. For demand d, we select a path among available candidate paths $\hat{P}_d$ with the following criteria.

1. shortest path first (SPF)
$$\min_{p \in \hat{P}_d} len(p)$$
2. path minimizing the highest slot index to allocate the target demand first (MHIF)
$$\min_{p \in \hat{P}_d} \max_{s \in S_{new}(p)} s$$
3. path minimizing the highest used slot index after the target demand is allocated first (MUIF)
$$\min_{p \in \hat{P}_d} \max_{s \in S_{new}(p) \cup (\cup_{e \in p} S(e))} s$$
4. path having largest the number of common slots first (LCSF). That is, the selected path has as many slots as possible so that they are available to allocate the target demand and simultaneously used in other edges.
$$\max_{p \in \hat{P}_d} |S_{new}(p) \cap \cup_{e \in E} \hat{S}(e)|$$

5. path having most free slots over hops first (FSoHF)

$$\min_{p \in \hat{P}_d} \sum_{e \in p} (|S_{new}(p)| + |\hat{S}(e)|)/hop(p)$$

The ordering policy of D is due to the way priority is given to demands that need many resources. Let p_d be a shortest path among P_d, that is, $len(p_d) = \min_{p \in P_d} len(p)$. We consider the seven ordering policies below.

1. descending order of response's bit-rates (br)

$$h_{\tau(d)}$$

2. descending order of the minimum number of required slots (s_min)

$$\min_{p \in P_d} n_{\tau(d)p}$$

3. descending order of the maximum number of required slots (s_max)

$$\max_{p \in P_d} n_{\tau(d)p}$$

4. descending order of the shortest path lengths (len)

$$len(p_d)$$

5. descending order of the average path lengths (lenoh)

$$\min_{p \in P_d} len(p)/hop(p)$$

6. descending order of the number of hops of a shortest path (hop)

$$hop(p_d)$$

7. random order (rnd)

In policies 2–6, if several demands have the same evaluation value, they are ordered by their responses' bit-rates.

The general description of the greedy algorithm for the maximum traffic volume model is given as the following.

step 1 Sort queries $d_1, d_2, \ldots, d_{|D|}$ according to the ordering policy. Set $i = 1$.

step 2 For query d_i,

- **2-1** If there is no available path, i.e., $\tilde{P}_{d_i} = \emptyset$, go to step 4.
- **2-2** Find a path p in $\tilde{P}_{d_i}$ in accordance with the route selection criterion.
- **2-3** Assign slots along the path p by following the first-fit rule.

step 3 For response $\tau(d_i)$,

3-1 If there is no available path, i.e., $\tilde{P}_{\tau(d_i)} = \emptyset$, remove the accommodated query d_i in step 2 and go to step 4.

3-2 Find a path p in $\tilde{P}_{\tau(d_i)}$ in accordance with the route selection criterion.

3-3 Assign slots along the path p by following the first-fit rule.

step 4 If all demands are searched for, stop. Otherwise, increment i by 1 and go to step 2.

As a variation of this algorithm, we may perform step 3 before step 2. Such an algorithm is called "type II," while the original one described above is called "type I." For the minimax slot number model, all paths in P_d are always available, that is, $P_d = \tilde{P}_d$ for all d. Thus, neither 2-1 nor 3-1 occur.

Note that the selected path is not influenced by the previous procedure when we adopt SPF as the route selection criterion in the minimax slot number model. In general, the numbers of slots for serving all demands are different according to the order of serving demands, as shown in Fig. 1, even if the accommodating route is defined. Assume that there are three demands: d_1 transmits from a to f, d_2 from a to e through c, and d_3 from a to e through b. The numbers of spectrum slots needed for the demands, n_{dp}, are s, s, and $s+1$ respectively. If we accommodate demands in this order, d_1 uses spectrum slots from 1, demand d_2 uses spectrum slots from $s+1$, and demand d_3 uses spectrum slots from $2s+1$. Then, the number of slots we need is $3s+1$. However, if we accommodate demand d_3 first, we can serve these demands within $2s+1$ slots.

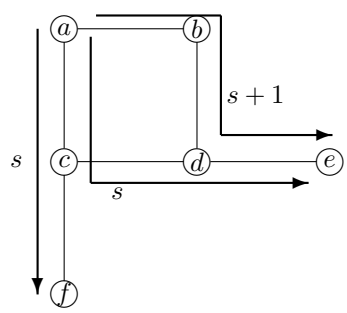


Figure 1: Example of spectrum slot allocation when paths for transmission are fixed

Meanwhile, the number of slots for serving all demands does not depend on the order of demands if the route for transmission is defined by the shortest path, which is uniquely defined in anycasting.

Theorem 1. *Assume that the graph G has a shortest path uniquely from each node to a data center. For the minimax slot number model, the number of slots obtained by the greedy algorithm employing SPF does not depend on the order of demands, that is, it is unaffected by step 1 of the algorithm.*

Proof. A collection of shortest paths from a node to one of the data centers forms rooted trees whose roots are data centers. We use in-trees for queries

and out-trees for responses. With respect to a feasible solution, if slot s is not occupied in an edge in one of the trees, slot s is not also occupied in the descendant subtree of the edge. Thus, spectrum slot allocation for edges incident to data centers decides feasible allocation, where the number of slots we need is given by the maximum number of total slots among edges incident to data centers. Hence, the number of slots does not depend on the order of demands.

5. Numerical Experiments

The upper/lower bounds were assessed in order to obtain exact optimal solutions for RSA problems of anycasting. For each model of RSA, we first evaluated the quality of solutions obtained by the greedy algorithms discussed in the previous section and characterized the route selection criteria and the demand ordering policies by performing numerical experiments. We next found exact optimal solutions by using bounds derived from the greedy algorithms and from the relaxation of spectrum continuity as discussed in Section 3.

5.1. Experimental Setup

We employed two network topologies, EURO16, which is the core part of the pan-European fiber-optic network [30], and the Japan Photonic Network (JPN) 25 [31], which are shown in Figures 2 and 3, respectively. In both figures, link distances are marked in kilometers, although several distances are omitted in the area having dense links in Figure 3. The data centers are shown as the black nodes in each figure. Other details are presented in Table 1. In our model, undirected links in these networks were replaced by pairs of bidirectional edges. The sets of demands were generated randomly. The client node of a query was selected uniformly at random from nodes except for data centers, $V \setminus C$. For each $d \in D$, the volume of response $h_{\tau(d)}$ was assigned a uniform random number from 40 to 400 Gb/s, whereas the volume of query h_d was set to 10 Gb/s. Candidate paths P_d were provided by the k-th shortest paths from the client node of d to one of the data centers. We prepared five shortest paths. These setups are similar to [11].

Table 1: Experimental networks

topology	# nodes	# links (undirected)	link length	node degree	
				average	maximum
EURO16	16	23	218 – 783 (km)	2.88	4
JPN25	25	43	29 – 931 (km)	3.44	5

We implemented algorithms with Java 1.8 and performed them on a MacBook Pro with OS X Sierra (version 10.12.4), a 2 GHz Intel Core i5 processor, and 8GB of memory. The ILP formulations were solved by using CPLEX 12.7 solver [32].

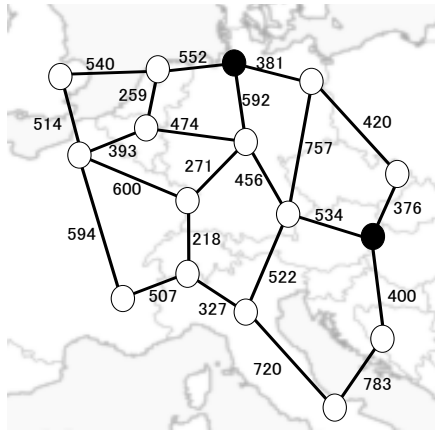


Figure 2: EURO16 [30] with marked the link distances in kilometers

5.2. Minimax Slot Number Model

First, we dealt with the problem of finding the minimum number of spectrum slots to accommodate all demands. We computed the optimal slot number w^* for 20 pairs of query and response ($|D| = 20$) by using the ILP formulated in Section 2 without any technical devices within moderately reasonable computational times. Figures 4 and 5 display the characterization of instances we used. Each of them shows the relationship between the optimal slot number w^* and $\underline{Sn} := \sum_{d \in \tilde{D}} n_{dp_d} \cdot hop(p_d)$, which is given by the sum of the numbers of slots that need to be transmitted along the shortest path for every demand (recall that p_d implies the shortest path between the client node of d and one of the data centers). Dividing $\underline{Sn}$ by the number of edges derives a lower bound of the optimal slot number, that is, $\underline{Sn}/|E| \leq w^*$. Marks in these figures are distinguished by computational times (seconds) for which details are discussed later. In both networks, the larger the optimal slot number w^* was, the more time solving problems tended to take. Moreover, the correlation between w^* and $\underline{Sn}$ was stronger in JPN25 rather than in EURO16, which means that more deft routing and spectrum assignment may be needed to reduce the slot number accommodating all demands in EURO16.

To obtain good upper bounds, we compared the performance of greedy algorithms for these 50 instances. Tables 2 and 3 show the relative tolerances of the gap between the value obtained by algorithm A, denoted by w_A, and the optimal value w^*, i.e.,

$$\frac{w_A - w^*}{w^*}.$$

Clearly, the route selection criterion had a stronger influence than the demand

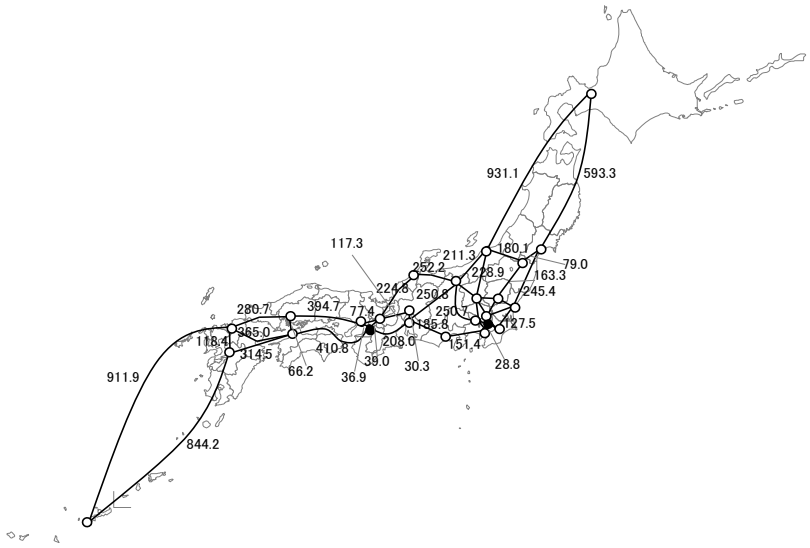


Figure 3: JPN25 [31] with marked the link distances in kilometers

ordering policy. In addition, type II tended to bring better solutions than type I. The best average of relative tolerances was achieved by the type II algorithm employing hop and MHIF or MUIF in both networks. In EURO16, however, algorithms using s_min instead of hop brought about slightly smaller standard deviations. With respect to the route selection criterion, MHIF and MUIF were superior to SPF, LCSF, and FSoHF. In particular, LCSF and FSoHF tended to select longer paths, which was a waste of network resources. Indeed, the total numbers of used slots $\hat{Sn} := \sum_{d \in \tilde{D}} n_{d\hat{p}_d} \cdot hop(\hat{p}_d)$, where $\hat{p}_d$ is the selected path for demand d, was large for LCSK and FSoHF. Table 4 shows the averages of $\hat{Sn}$ for the route selection criterion among all demand ordering policies and types I and II for all 50 instances. Algorithms employing FSoHF seemed to select a path making a detour.

Table 5 indicates the number of instances for which each algorithm obtained the optimal slot number among the 50 instances. For JPN25, algorithms employing MHIF and MUIF obtained the optimal slot numbers w^* in about 20–

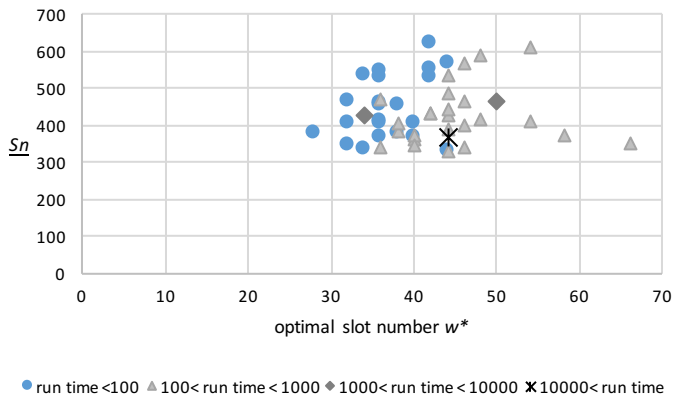


Figure 4: Relationship between optimal slot number w^* and $\underline{Sn}$ for 50 instances in EURO16

Table 2: Average relative tolerances and standard deviations, which are in parentheses, for 50 instances in EURO16

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	0.73 (0.30)	0.30 (0.12)	0.35 (0.16)	0.86 (0.26)	1.94 (0.72)
	s_min		0.27 (0.13)	0.29 (0.14)	0.73 (0.23)	2.31 (0.81)
	s_max		0.30 (0.12)	0.35 (0.12)	0.86 (0.26)	1.94 (0.72)
	len		0.31 (0.13)	0.40 (0.13)	0.74 (0.29)	2.38 (0.87)
	lenoh		0.60 (0.22)	0.66 (0.26)	1.29 (0.41)	1.33 (0.73)
	hop		0.25 (0.10)	0.33 (0.13)	0.66 (0.24)	2.70 (0.87)
	rnd		0.37 (0.17)	0.37 (0.18)	0.89 (0.27)	2.20 (0.78)
II	br		0.18 (0.09)	0.18 (0.09)	0.86 (0.26)	1.40 (0.41)
	s_min		0.18 (0.07)	0.18 (0.07)	0.72 (0.22)	1.42 (0.42)
	s_max		0.18 (0.09)	0.18 (0.09)	0.86 (0.26)	1.40 (0.41)
	len		0.27 (0.11)	0.26 (0.10)	0.73 (0.27)	1.44 (0.41)
	lenoh		0.40 (0.13)	0.40 (0.13)	1.15 (0.36)	1.10 (0.42)
	hop		0.18 (0.08)	0.18 (0.08)	0.64 (0.20)	1.12 (0.41)
	rnd		0.25 (0.10)	0.24 (0.11)	0.95 (0.33)	1.04 (0.43)

30% instances, although the algorithms of LCSF and FSoHF obtained optimal solutions in few instances. For EURO16, the greedy algorithms found optimal solutions in very few instances. This result implies that problems in JPN25 can be solved more easily than in EURO15, from which we know that the difficulty for finding optimal solutions is influenced rather by the configuration than the size of networks.

These properties we discussed above were also found in other instances. We display box-plot diagrams of the obtained slot numbers for 1000 instances in Figures 6 and 7. We also show the results for the 1000 instances with 100 pairs of query and response ($|D| = 100$) in Figures 8 and 9. In these figures, the labels in the horizontal axis stand for algorithms. For example, "br_I" means type I algorithm using demand ordering policy "br." The route selection criterion employing each algorithm is shown at the bottom of the labels. The

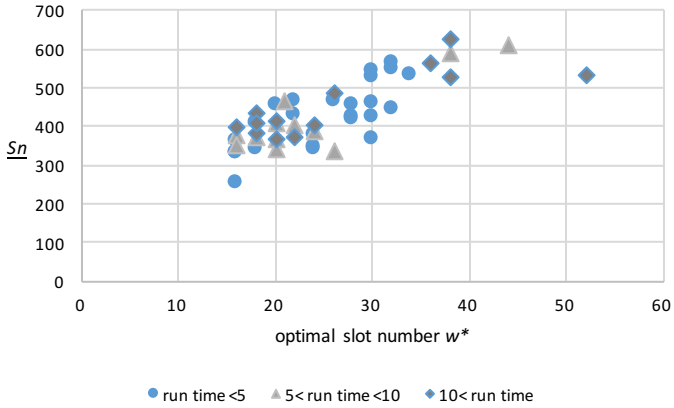


Figure 5: Relationship between optimal slot number w^* and $\underline{Sn}$ for 50 instances in JPN25

statistic information , that is, averages and standard deviations, of these results are shown in tables at Appendix. The difference in the algorithms' performance was more remarkable for the instances with 100 demands than with 20 demands. These results also assured that type II tended to bring better solution than type I, and that MHIF and MUIF were superior stably to the other route selection criterion. Table 6 indicates the percentage of instances for which each algorithm achieved the best solution among our greedy algorithms for 1000 instances with 100 demands. Since neither of the algorithms with SPF, LCSF, and FSoHF achieved the best solutions in these instances, the table indicates only those for MHIF and MUIF. From these results, we conclude that the route selection criterion is important and that we have to select a shorter path under slot conditions. In addition, we have to employ type II with demand ordering policy s_min, s_max, or hop to get better solutions.

We now turn to finding the optimal slot numbers. It may be necessary to take a lot time to solve the ILP formulated in Section 2 to find an exact optimal solution. Indeed, there was an instance that was computed in 8 hours in our computational environment. One of the reasons for taking a long time is the symmetric structure of feasible solutions. Usually, there are many optimal solutions in our problem. For example, exchanging the slots for demands that are assigned to the same path is also feasible. To reduce the number of optimal solutions, we replace the objective function w by

$$|\tilde{D}| \cdot |S| \cdot w + \sum_{d \in \tilde{D}} \sum_{p \in P_d} \sum_{s \in S} d \cdot s \cdot x_{dps} \tag{15}$$

without changing the optimal value of w. The computational time for 50 instances was reduced by about half by introducing (15) in EURO16. Moreover, we used the lower and upper bounds obtained by solving (LB_w) and (UB_w). In

Table 3: Average relative tolerances and standard deviations, which are in parentheses, for 50 instances in JPN25

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	1.36 (0.45)	0.15 (0.14)	0.15 (0.15)	0.76 (0.43)	2.03 (0.62)
	s_min		0.12 (0.13)	0.13 (0.13)	0.85 (0.38)	2.11 (0.65)
	s_max		0.13 (0.15)	0.13 (0.15)	0.91 (0.39)	2.08 (0.60)
	len		0.14 (0.14)	0.13 (0.11)	0.69 (0.30)	2.28 (0.66)
	lenoh		0.14 (0.13)	0.13 (0.11)	0.75 (0.33)	2.16 (0.58)
	hop		0.12 (0.12)	0.11 (0.11)	0.70 (0.34)	2.32 (0.73)
	rnd		0.21 (0.17)	0.21 (0.17)	0.76 (0.43)	2.07 (0.71)
II	br		0.14 (0.15)	0.15 (0.15)	0.78 (0.42)	1.96 (0.57)
	s_min		0.12 (0.13)	0.12 (0.13)	0.82 (0.44)	2.02 (0.54)
	s_max		0.12 (0.15)	0.12 (0.15)	0.92 (0.42)	2.01 (0.54)
	len		0.12 (0.11)	0.12 (0.11)	0.74 (0.34)	2.07 (0.57)
	lenoh		0.13 (0.11)	0.13 (0.11)	0.81 (0.31)	2.03 (0.53)
	hop		0.10 (0.10)	0.10 (0.10)	0.71 (0.36)	2.09 (0.59)
	rnd		0.19 (0.16)	0.20 (0.16)	0.70 (0.44)	1.92 (0.59)

Table 4: Average of total numbers of used slots $\hat{Sn}$

	SPF	MHIF	MUIF	LSCF	FSoHF
EURO16	454.76	623.21	618.19	755.35	670.28
JPN25	430.48	461.99	456.09	498.26	532.57

Table 5: Number of instances for which each algorithm obtained optimal value among 50 instances

		EURO16					JPN25				
		SPF	MHIF	MUIF	LCSF	FSoHF	SPF	MHIF	MUIF	LCSF	FSoHF
I	br	0	0	0	0	0	0	15	15	1	0
	s_min		0	0	0	0		15	14	0	0
	s_max		0	0	0	0		13	12	0	0
	len		0	0	0	0		13	13	0	0
	lenoh		0	0	0	0		13	12	0	0
	hop		0	0	0	0		15	17	1	0
	rnd		0	0	0	0		9	11	1	0
II	br		1	1	0	0		16	16	1	0
	s_min		0	0	0	0		16	16	0	0
	s_max		1	1	0	0		14	14	0	0
	len		0	0	0	0		13	13	0	0
	lenoh		0	0	0	0		12	12	0	0
	hop		2	2	0	0		16	17	1	0
	rnd		0	0	0	0		10	9	2	0

our experiments, the lower bound obtained from (LB_w) was equivalent to the upper bound obtained from (UB_w) for more than 90% of our instances. That is to say, we obtained optimal solutions for these instances where lower and upper bounds coincided with each other. The computational time for solving (LB_w) and (UB_w) was shorter than for solving the original ILP formulation of the min-imax slot number model. We compare the average slot numbers and average computational time in Figures 10 and 11, where the value obtained by the greedy

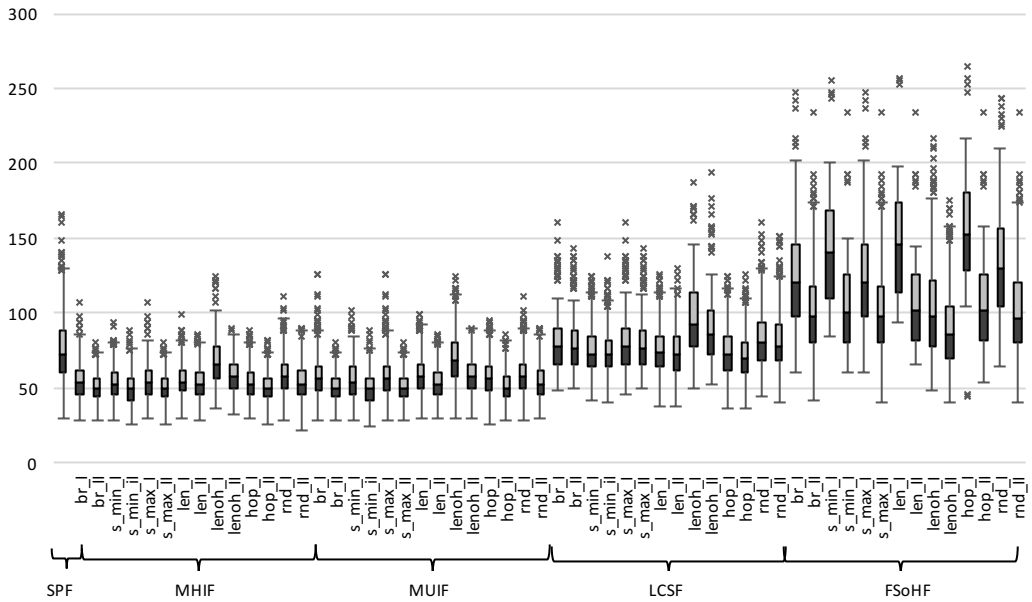


Figure 6: Box-plot diagram of obtained slot numbers for EURO16 with 20 demands

algorithm is also compared. The greedy algorithm returned the best solution among the type II algorithms of MHIF and MUIF with s_min, s_max, and hop. The results are averages among the 50 instances for each $|D| = 20, 30, \ldots, 100$. Note that the vertical axis in the graph for computational time is given by the logarithmic scale. From the result, the upper bound when using the greedy algorithm was not much better than that when using (UB_w). However, the computational time for solving (UB_w) grew drastically according to $|D|$. Thus, there was a trade-off between the accuracy and computational time for evaluating upper bounds. When the upper bound was used for finding exact optimal solutions, the value of (UB_w) may not have been a very good upper bound even if it took a lot of time to compute and may have additionally taken a lot of time to obtain exact optimal solutions. Thus, we employed the upper bound obtained by the greedy algorithm as input to compute the minimax slot number model. In the meantime, we can solve (LB_w) as fast as the greedy algorithms. The computational time for (LB_w) did not grow that rapidly even when $|D|$ was increased, although it used the IP solver. Therefore, before solving the ILP formulation defined in Section 2, we obtained a lower bound by solving (LB_w) and an upper bound from the best solution of six greedy algorithms. We then solve the ILP formulation by adding the constraint that the optimal w^* is bounded by these lower and upper bounds. The upper bound of w^* defines the size of the decision variable x_{dps} in the ILP of the minimax slot number model. The original formulation adapts it as the slot number w obtained by the algorithm

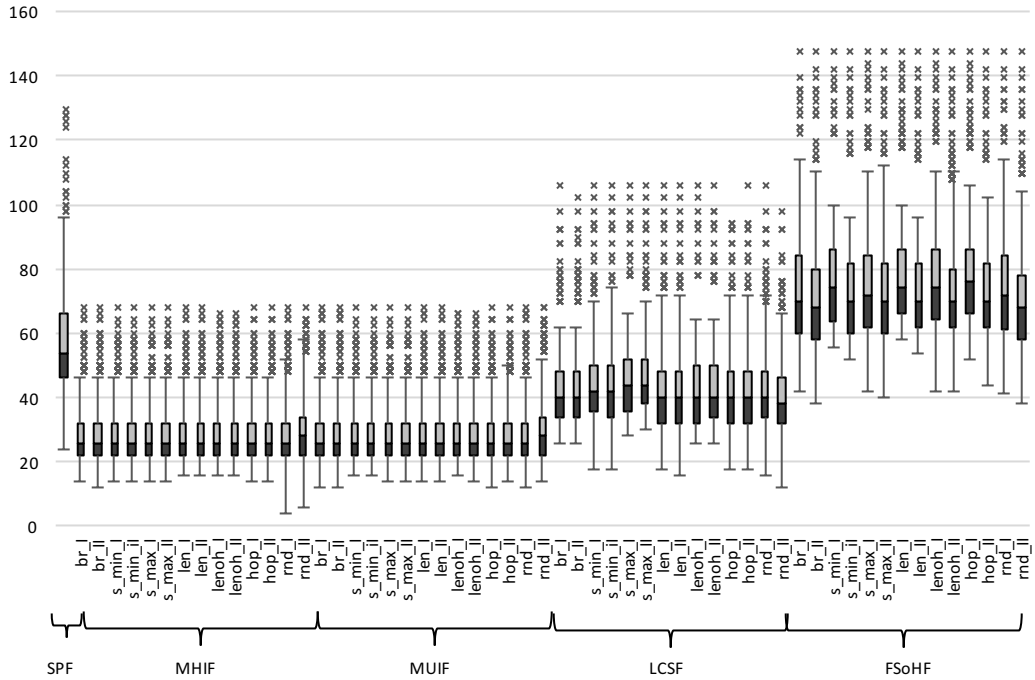


Figure 7: Box-plot diagram ofobtained slot numbers for JPN25 with 20 demands

with SPF. As shown in Table 7, the result shows that the good upper and lower bounds were effective at reducing the computational time. Since the ILP formulation using lower and upper bounds can obtain optimal solutions efficiently, we apply it to larger instances whose size of demands increased from 30 to 100. Table 8 shows the computational time for each $|D|$. In JPN25, we could obtain optimal solutions within a reasonable time for $|D| = 100$, while in EURO16, the computational time was very long when $|D| = 50$. In some instances, our method can obtain an optimal solution faster than solving (UB_w). Surprisingly, the optimal values of (LB_w) coincided with the optimal slot numbers w^* in all instances, although the paths selected by (LB_w) were not always optimal because the optimal value of (UB_w) was not equivalent to w^*.

We finally consider the appropriate number of candidate paths. If we prepare many candidate paths, the objective values may be improved. We evaluated the improvement ratios of objective values in EURO16 when the number of candidate paths increased by a factor of two. That is to say, we computed the optimal slot numbers when there were 10 candidate paths for each client node. The average of the improvement ratio was 7.38% for the 50 instances

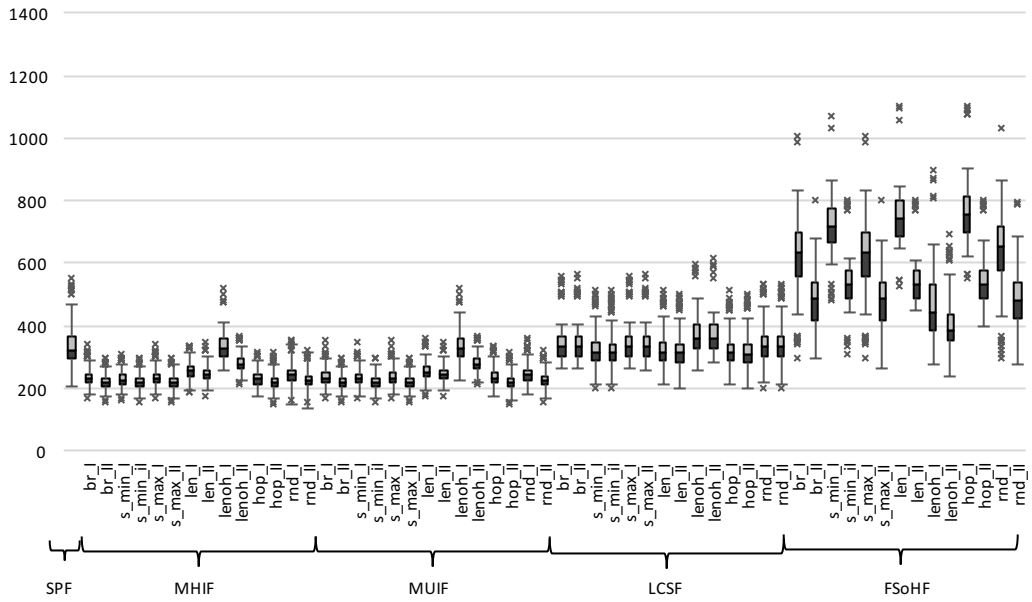


Figure 8: Box-plot diagram of obtained slot numbers for EURO16 with 100 demands

with $|D| = 20$. Here, the percentage of the improvement ratio was defined by

$$\frac{\text{the value for 5 candidate paths} - \text{the value for 10 candidate paths}}{\text{the value for 5 candidate paths}} \times 100.$$

However, in 26% of instances, the objective values were not changed. Instead of improvement, the several instances needed to take large amount of time for computation. In our case, the computational time increased 4.27 times on average. Figure 12 displays the relationship between the improvement ratio of the objective values and the computational time. The horizontal axis means the percentage of the improvement ratio, and the vertical axis means the computational time ratio, i.e., (the time for 10 candidate paths)/(the time for 5 candidate paths). We can see there was a trade-off between the optimal slot number and computational time. Meanwhile, there were instances for which the objective value was not improved even if the computational time increased.

We finally summarize the results in our experiments for the minimax slot number model.

- With respect to the performance of the greedy algorithms,
  - the route selection criterion had a stronger influence than the demand ordering policy.
  - Type II tended to bring better solutions than type I.

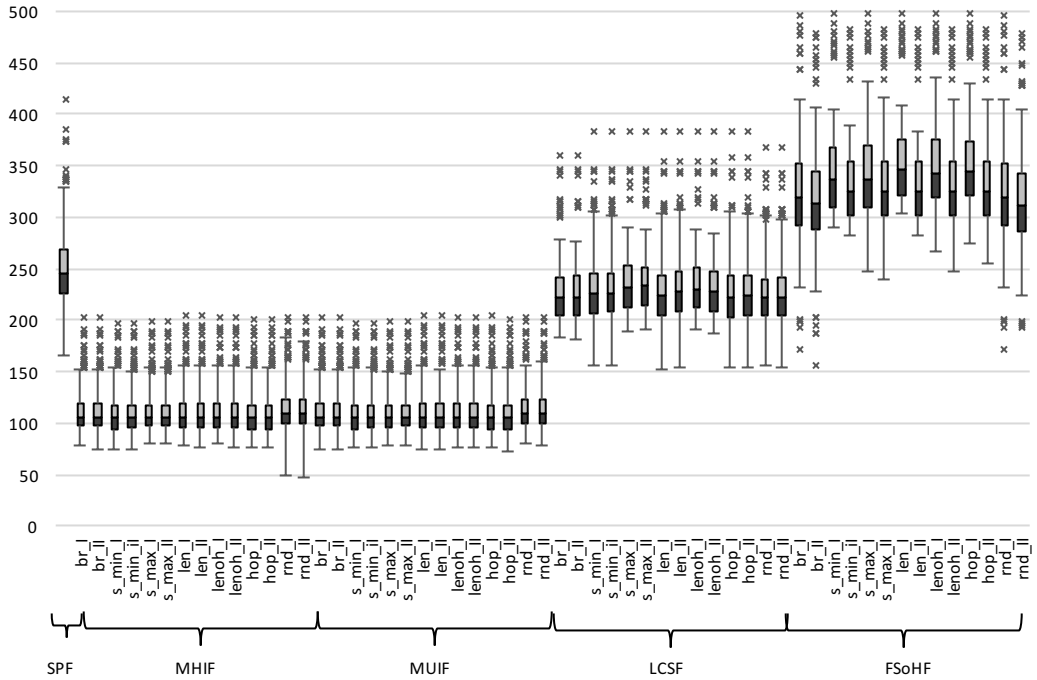


Figure 9: Box-plot diagram of obtained slot numbers for JPN25 with 100 demands

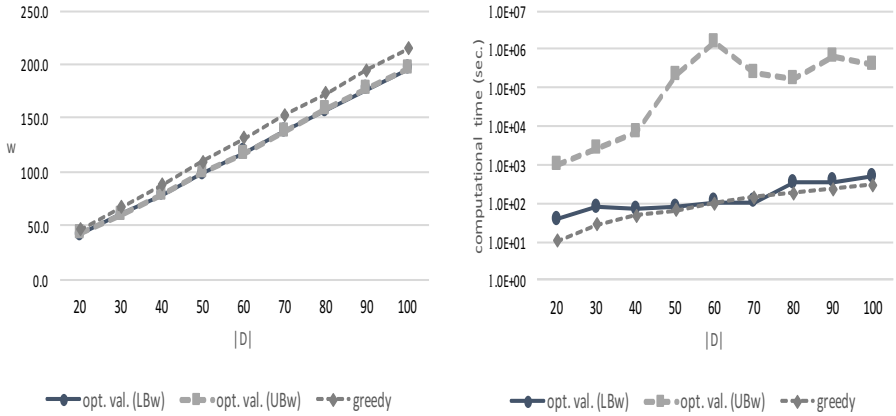
  - MHIF and MUIF is superior among our route selection criterion. Namely, it is better to select a shorter path under slot condition.
  - Problems in JPN25 could be solved more easily than in EURO15

- With respect to the exact algorithm,
  - the lower bound obtained from (LB_w) was equivalent to the upper bound obtained form (UB_w) for more than 90 % of our instances.
  - There was a trade-off between the accuracy and computational time for evaluating upper bounds by greedy algorithms and by solving (UB_w).
  - The ILP formulation using lower and upper bounds could obtain optimal solutions efficiently.
  - the optimal values of (LB_w) coincided with the optimal slot numbers w^* in all our instances

- There was a trade-off between the optimal slot number and computational time by increasing the number of candidate paths. We could say that the five candidate paths were appropriate in our instances.

Table 6: Percentage of instances for which each algorithm achieved best solution among our greedy algorithms for 1000 instances with $|D| = 100$

	EURO16				JPN25			
	type I		type II		type I		type II	
	MHIF	MUIF	MHIF	MUIF	MHIF	MUIF	MHIF	MUIF
br	1.6	0.8	33.5	33.1	21.4	21.5	21.9	21.8
s_min	2.1	1.6	35.0	34.5	40.4	41.0	40.4	40.4
s_max	1.6	0.8	33.5	33.1	30.6	32.1	30.7	30.4
len	0.0	0.0	0.0	0.0	16.3	13.3	17.8	17.6
lenoh	0.0	0.0	0.0	0.0	12.3	15.1	12.6	12.8
hop	2.9	1.9	27.1	25.4	38.1	38.4	41.5	41.5
rnd	0.6	0.4	8.1	7.9	8.6	7.2	8.8	8.1

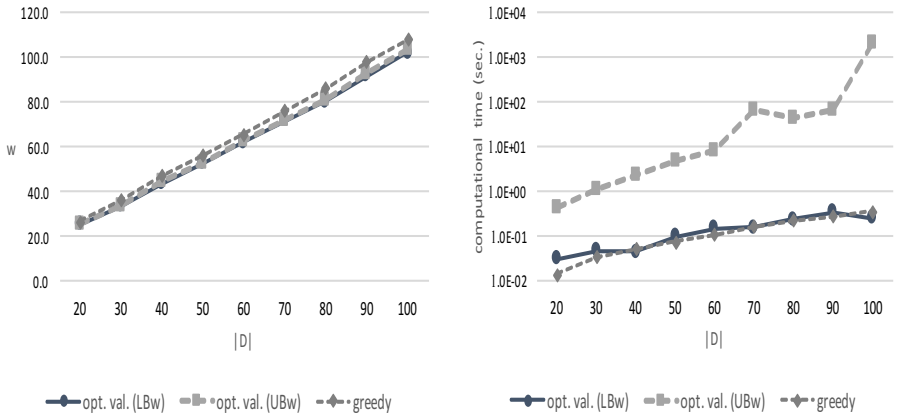


(a) upper and lower bounds of w (b) computational time

Figure 10: Comparing averages of upper and lower bounds of w and computational times in accordance with size of demand $|D|$ in EURO16

5.3. Maximum Traffic Volume Model

We now deal with the problem of maximizing the traffic volume of demands served under a given spectrum slot number. First, we compared the performance of the greedy algorithms. By setting the slot number of each instance to its optimal slot number w^* that was calculated by the previous subsection, we could transmit 100% of demands in the instances if we executed transmission in accordance with the optimal RSA solution. Tables 9 – 12 show the averages of traffic volume rates, that is, the sum of served demand volumes over the total volume of all demands, and standard deviations of them. Tables 9 and 10 represent the results for EURO16 with $|D| = 20$ and 50, respectively, and Tables 11 and 12 are the results for JPN25 with $|D| = 20$ and 100, respectively. Note that the type I and type II algorithms employing SPF had the same performance because the route used for each demand was fixed without affecting the usage of spectrum slots and no (directed) edge was shared between a query



(a) upper and lower bounds of w (b) computational time

Figure 11: Comparing averages of upper and lower bounds of w and computational times in accordance with size of demand $|D|$ in JPN25

Table 7: Comparing computational time (sec.) for solving minimax slot number of 50 instances in each network with $|D| = 20$

	EURO16			JPN25		
	original	obj.func. (15)	upper/lower bounds	original	obj.func. (15)	upper/lower bounds
average	889.43	393.40	4.86	6.83	5.17	0.58
min	7.85	10.41	0.74	1.01	0.89	0.26
max	30008.76	7717.16	29.31	21.09	27.25	1.14

and its response. Thus, the cells for the type II algorithms with SPF are empty in these tables. In our results, a larger traffic volume rate was achieved for instances in JPN25 than in EURO16, which implies that JPN25 has a tractable structure for not only the minimax slot number model but also the maximum traffic volume model. Depending on the size of $|D|$, the average of the traffic volume rate was not affected too much, but the standard deviation was smaller when $|D|$ was large. The reason is that nearly optimal combinations of served demands may increase when $|D|$ is large. For almost all of the cases, the type II algorithms were superior to the type I algorithms, especially for algorithms with FSoHF. Among the type II algorithms, MHIF and MUIF performed better with respect to the route selection criterion, and hop performed better with respect to the demand ordering policy. In EURO16, the algorithm with SPF and len performed well when $|D| = 20$. Interestingly, algorithms with rnd were better when $|D|$ became large. We can confirm the tendency described above in Table 13, which indicates the number of instances for which each algorithm can serve 100% of demands for each network with $|D| = 20$. Obviously, the

Table 8: Computational time (sec.) for solving minimax slot number of 50 instances

	$\|D\|$	30	40	50	60	70	80	90	100
EURO16	average	146.68	805.69	4855.46					
	min	4.14	7.31	28.90					
	max	3603.51	14853.63	155186.05					
JPN25	average	1.54	4.21	7.32	13.84	31.36	72.39	192.29	758.24
	min	0.70	1.44	3.37	6.04	10.68	16.51	23.54	31.72
	max	4.05	17.77	20.46	105.13	271.13	914.13	3740.22	16993.45

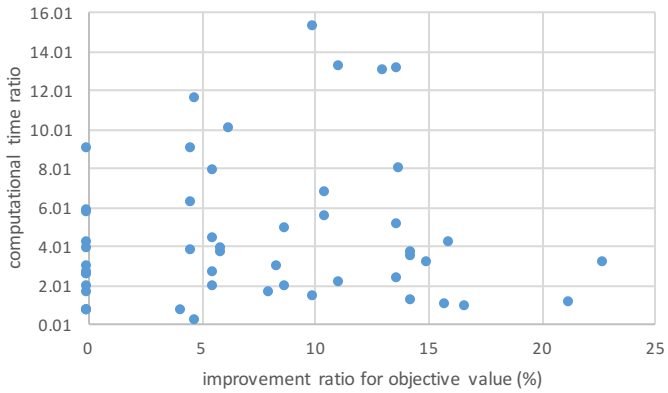


Figure 12: Relationship between improvement ratio of objective value and computational time

instances for which the algorithms obtained the optimal slot number in the minimax slot number model served all of the demands with the algorithm in the maximum traffic volume model. By comparing Tables 5 and 13, algorithms employing MHIF and MUIF accomplished all of the demands for only such instances, while algorithms with LCSF and FSoHF could serve 100% of demands for other instances. In EURO16 with $|D| = 50$, there existed no other instance that achieved the 100% traffic volume rate with any of the greedy algorithms. Among our 50 instances for JPN25 with $|D| = 100$, only algorithms employing s_min and MHIF or MUIF achieved the 100% traffic volume rate for only one instance for which only the algorithms obtained the optimal slot number in the minimax slot number model. The larger the size of the demands was, the more difficult it was to achieve the 100% traffic volume demand.

We next compared the performance of the greedy algorithms in accordance with the size of demands, where the spectrum slot number was fixed. We set the spectrum slot number to the maximum value of the optimal slot number w^* in the minimax slot number model with $|D| = 20$ among the 50 instances, for which we obtained optimal solutions. For EURO16, the slot number w was fixed to 66, and for JPN25, $w = 52$. Hence, when $|D| = 20$, all of the demands

Table 9: Average traffic volume rates and standard deviations, which are in parentheses, for 50 instances in EURO16 with $|D| = 20$

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	86.59 (6.24)	83.94 (7.13)	83.84 (6.95)	69.10 (9.55)	49.44 (11.18)
	s_min	85.07 (8.23)	82.01 (8.22)	83.68 (7.03)	70.85 (9.72)	55.40 (11.76)
	s_max	86.59 (6.24)	83.94 (7.13)	83.84 (6.95)	69.10 (9.55)	58.23 (11.96)
	len	89.77 (5.77)	84.19 (7.93)	81.44 (7.88)	75.71 (9.14)	61.00 (13.44)
	lenoh	88.58 (6.44)	81.84 (7.18)	80.93 (7.57)	72.30 (8.40)	74.93 (11.16)
	hop	88.85 (6.52)	85.10 (7.61)	83.30 (9.32)	76.70 (9.16)	55.35 (9.22)
	rnd	87.29 (4.86)	83.87 (5.01)	84.66 (4.29)	69.60 (8.93)	62.92 (10.41)
II	br		89.41 (6.21)	89.73 (5.88)	73.51 (10.35)	85.84 (7.22)
	s_min		87.54 (7.37)	87.51 (6.95)	73.82 (10.57)	83.85 (8.56)
	s_max		89.41 (6.21)	89.73 (5.88)	73.51 (10.35)	85.84 (7.22)
	len		86.42 (7.17)	86.30 (6.99)	78.85 (8.41)	88.55 (5.71)
	lenoh		85.36 (5.00)	85.36 (5.00)	77.19 (8.54)	89.12 (5.59)
	hop		89.41 (5.02)	89.20 (5.20)	81.45 (8.96)	87.13 (6.94)
	rnd		86.68 (5.28)	86.63 (5.22)	74.63 (5.81)	86.27 (4.92)

Table 10: Average traffic volume rates and standard deviations, which are in parentheses, for 50 instances in EURO16 with $|D| = 50$

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	84.51 (3.55)	82.34 (5,96)	82.96 (5.51)	74.31 (4.87)	44.92 (6.48)
	s_min	83.64 (4.08)	81.49 (6.00)	81.79 (5.24)	74.25 (6.54)	47.54 (8.67)
	s_max	84.51 (3.55)	82.34 (5.96)	82.96 (5.51)	74.31 (4.87)	57.63 (10.44)
	len	90.32 (4.16)	82.88 (5.87)	83.51 (5.19)	78.21 (7.82)	49.72 (7.06)
	lenoh	88.40 (4.04)	76.20 (6.29)	76.00 (6.21)	78.82 (5.00)	76.43 (9.84)
	hop	86.57 (4.35)	86.68 (6.31)	85.06 (6.05)	79.87 (6.04)	44.51 (5.88)
	rnd	89.54 (2.36)	88.24 (2.85)	87.87 (2.84)	80.45 (4.21)	62.66 (9.55)
II	br		87.26 (4.88)	87.45 (4.98)	76.49 (4.91)	83.28 (3.99)
	s_min		86.61 (4.34)	86.66 (4.57)	78.56 (6.19)	81.85 (5.81)
	s_max		87.26 (4.88)	87.45 (4.98)	76.49 (4.91)	83.28 (3.99)
	len		85.68 (4.38)	85.26 (4.54)	82.82 (5.94)	89.23 (3.97)
	lenoh		84.53 (3.69)	84.53 (3.69)	79.83 (4.18)	89.69 (3.38)
	hop		90.69 (4.13)	90.72 (4.14)	83.20 (4.68)	84.80 (4.68)
	rnd		90.56 (3.17)	90.67 (3.14)	81.68 (4.52)	88.23 (2.70)

could be served in almost all of the instances. The larger the size of the demands $|D|$ was, the lower the traffic volume rate became. Figures 13 and 14 show the frequency at which each algorithm obtained the best traffic volume rate among our greedy algorithms for the 1000 instances for $|D| = 20, \ldots, 100$. This result shows that the best method depended on the network and the traffic congestion level. The demand ordering policy rnd became efficient when the size of demand $|D|$ was large in both networks. In EURO16, lenhop was a superior policy. For the route selection criterion, SPF and FSoHF performed well in EURO16, and MHIF and MUIF performed well in JPN25. This result implies that it is hard to select a dominant criterion of route selection and a dominant policy of demand ordering. In other words, it is worth testing the quality of their solutions for every route selection criterion and every demand ordering policy.

We now attempt to obtain an exact optimal solution for the maximum traffic volume model. Similar to the case of minimax slot number models, we used upper and lower bounds. We compare the average of the bounds of traffic

Table 11: Average traffic volume rates and standard deviations, which are in parentheses, for 50 instances in JPN25 with $|D| = 20$

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	92.96 (5.57)	94.30 (5.57)	94.40 (5.39)	91.77 (6.73)	78.15 (12.11)
	s_min	91.85 (6.46)	93.51 (5.88)	93.47 (5.69)	90.17 (7.86)	86.93 (9.47)
	s_max	93.07 (5.39)	93.95 (5.44)	93.66 (5.41)	91.00 (8.47)	87.55 (7.84)
	len	93.10 (5.16)	93.97 (6.29)	94.50 (5.28)	93.30 (6.24)	89.66 (8.10)
	lenoh	93.55 (5.30)	94.51 (5.10)	94.54 (4.87)	93.54 (6.72)	90.51 (7.22)
	hop	92.32 (5.94)	94.66 (5.54)	95.23 (4.91)	93.19 (6.09)	87.95 (7.61)
	rnd	92.53 (5.95)	91.48 (6.99)	91.64 (7.20)	89.48 (6.63)	85.34 (11.04)
II	br		94.42 (5.45)	94.59 (5.29)	90.98 (7.31)	91.31 (6.94)
	s_min		93.94 (5.73)	94.16 (5.57)	91.05 (7.28)	90.07 (7.69)
	s_max		94.12 (5.45)	94.12 (5.45)	89.97 (8.75)	89.89 (7.72)
	len		94.39 (5.33)	94.43 (5.32)	92.62 (6.18)	91.35 (6.59)
	lenoh		94.90 (4.53)	94.80 (4.51)	92.38 (6.54)	92.26 (6.08)
	hop		95.29 (4.88)	95.38 (4.88)	92.74 (6.10)	91.10 (6.14)
	rnd		92.01 (6.23)	91.71 (6.63)	89.17 (7.14)	89.79 (7.53)

Table 12: Average traffic volume rates and standard deviations, which are in parentheses, for 50 instances in JPN25 with $|D| = 100$

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	91.61 (5.03)	94.36 (3.96)	94.36 (3.66)	91.34 (5.50)	77.83 (7.82)
	s_min	90.62 (5.58)	94.62 (4.27)	94.70 (4.32)	90.73 (5.28)	85.10 (7.03)
	s_max	90.57 (5.82)	93.49 (5.11)	93.76 (4.95)	90.99 (5.78)	86.65 (6.27)
	len	92.77 (3.42)	95.68 (2.82)	95.71 (2.91)	93.88 (3.44)	89.56 (4.80)
	lenoh	93.27 (3.72)	95.72 (2.73)	95.87 (2.63)	93.31 (4.22)	90.42 (4.58)
	hop	92.49 (4.16)	95.95 (3.14)	95.87 (3.22)	93.68 (3.66)	88.64 (4.88)
	rnd	93.53 (4.06)	96.40 (2.27)	96.44 (2.39)	93.67 (4.03)	88.01 (6.14)
II	br		94.52 (3.95)	94.52 (3.95)	90.99 (5.95)	90.09 (5.88)
	s_min		94.84 (4.34)	94.84 (4.34)	90.56 (5.88)	88.96 (6.02)
	s_max		93.83 (5.03)	93.85 (5.02)	90.58 (6.55)	88.98 (6.09)
	len		96.12 (2.17)	96.08 (2.21)	94.28 (3.23)	90.08 (4.32)
	lenoh		95.93 (2.56)	95.93 (2.56)	93.95 (4.09)	90.87 (4.24)
	hop		96.12 (3.05)	96.12 (3.05)	93.94 (3.29)	89.70 (4.60)
	rnd		96.64 (2.17)	96.45 (2.43)	93.68 (4.04)	91.50 (4.81)

volume rates and the average of the computational time in Figures 15 and 16, where upper bounds were obtained from (UB_t) and lower bounds were obtained from (LB_t) and when using the greedy algorithms. As discussed above, since there were no dominant greedy algorithms, the lower bound was given by the best solution among all 63 patterns of our greedy algorithms. The results are averages among the 50 instances for each $|D| = 20, \ldots, 100$. Although there were about 60% of instances in which the values of (UB_t) and (LB_t) coincided with each other, the averages of the traffic volume rate were nearly the same. In addition, the computational time for solving (LB_t) did not grow drastically for the sizes of our instances. However, the greedy algorithms did not obtain very good solutions even though much time was spent on computing the 63 patterns. In particular, the relative tolerances of gaps between the upper bounds and solutions obtained by the greedy algorithms were large when $|D|$ became large. Thus, we employed lower bounds obtained by (LB_t) to compute the exact optimal traffic volume rate. That is to say, our algorithm for obtaining an exact

Table 13: Number of instances for which each algorithm could transmit all demands among 50 instances with $|D| = 20$

		EURO16					JPN25				
		SPF	MHIF	MUIF	LCSF	FSoHF	SPF	MHIF	MUIF	LCSF	FSoHF
I	br	0	0	0	0	0	11	15	15	12	1
	s_min	0	0	0	0	0	7	15	14	10	4
	s_max	0	0	0	0	0	10	13	12	12	4
	len	0	0	0	0	0	8	13	13	11	4
	lenoh	0	0	0	0	0	11	13	12	14	7
	hop	1	0	0	0	0	8	15	17	13	3
	rnd	0	0	0	0	0	12	9	11	6	8
II	br		1	1	0	0		16	16	12	10
	s_min		0	0	0	1		16	16	12	7
	s_max		1	1	0	0		14	14	10	6
	len		0	0	0	0		13	13	9	6
	lenoh		0	0	0	1		12	12	10	11
	hop		2	2	0	0		16	17	12	6
	rnd		0	0	0	0		10	9	6	9

optimal solution first solves (UB_t) and (LB_t). If the upper and lower bounds obtained from (UB_t) and (LB_t) coincide with each other, the solution of (LB_t) is optimal, which implies that our algorithm stops. Otherwise, we solve the ILP formulation in Section 2 by adding the constraint that the optimal value is bounded by these lower and upper bounds. In this case, some instances needed to take a lot of time to solve the problems as cutting off of the search of IP solver was not successful. Therefore, we added the additional constraint (11). Table 14 indicates the precision of the bounds and computational time for finding the exact optimal traffic volume. The 3rd and 4th rows, denoted by "UB=LB" and "UB=opt.val.," show the numbers of instances in which the objective values of (UB_t) and (LB_t) were equivalent and in which the objective value of (UB_t) was equivalent to the exact optimal value, respectively, among our 50 instances, where w was also fixed to 66 for EURO16 and 52 for JPN25. The precision of bounds worsened when the size of demands $|D|$ was large. Similar to the minimax slot number model, the relaxation of spectrum continuity derived good bounds for the optimization problem. In the maximum traffic volume model, the upper bounds did not achieve the exact value for some instances. However, the upper bound gave a good approximation, as shown in Figure 17, where the exact value and its upper and lower bounds obtained by (UB_t) and (LB_t) are plotted for instances whose upper and lower bounds did not coincide with each other. The triangle marker shows the exact traffic volume rate, and the vertical line shows the gaps between lower and upper bounds. The horizontal axis shows the instances rearranged in the ascending order of the computational time to find optimal solutions. As we can see in this graph, the upper bounds were almost optimal even if they were not equivalent to the optimal solution. This graph also shows that the computational time did not depend simply on the exact traffic volume rate or the gaps between the upper and lower bounds. A lot of computational time was spent certifying the optimality by reducing this slightly small gap between the exact optimal solution and upper bound.

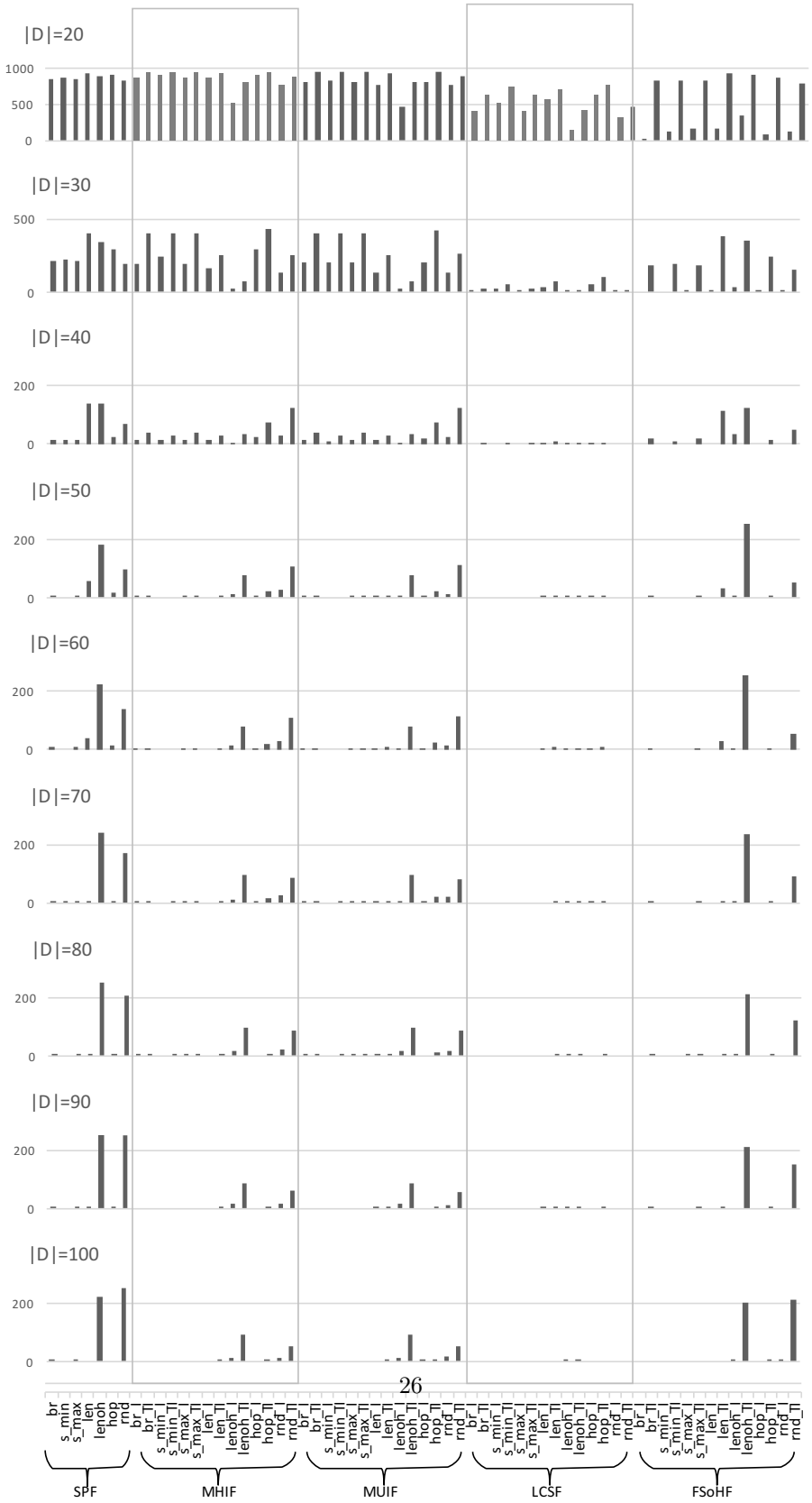


Figure 13: Frequency of achieving best traffic volume rates among greedy algorithms for each $|D|$ of 1000 instances in EURO16

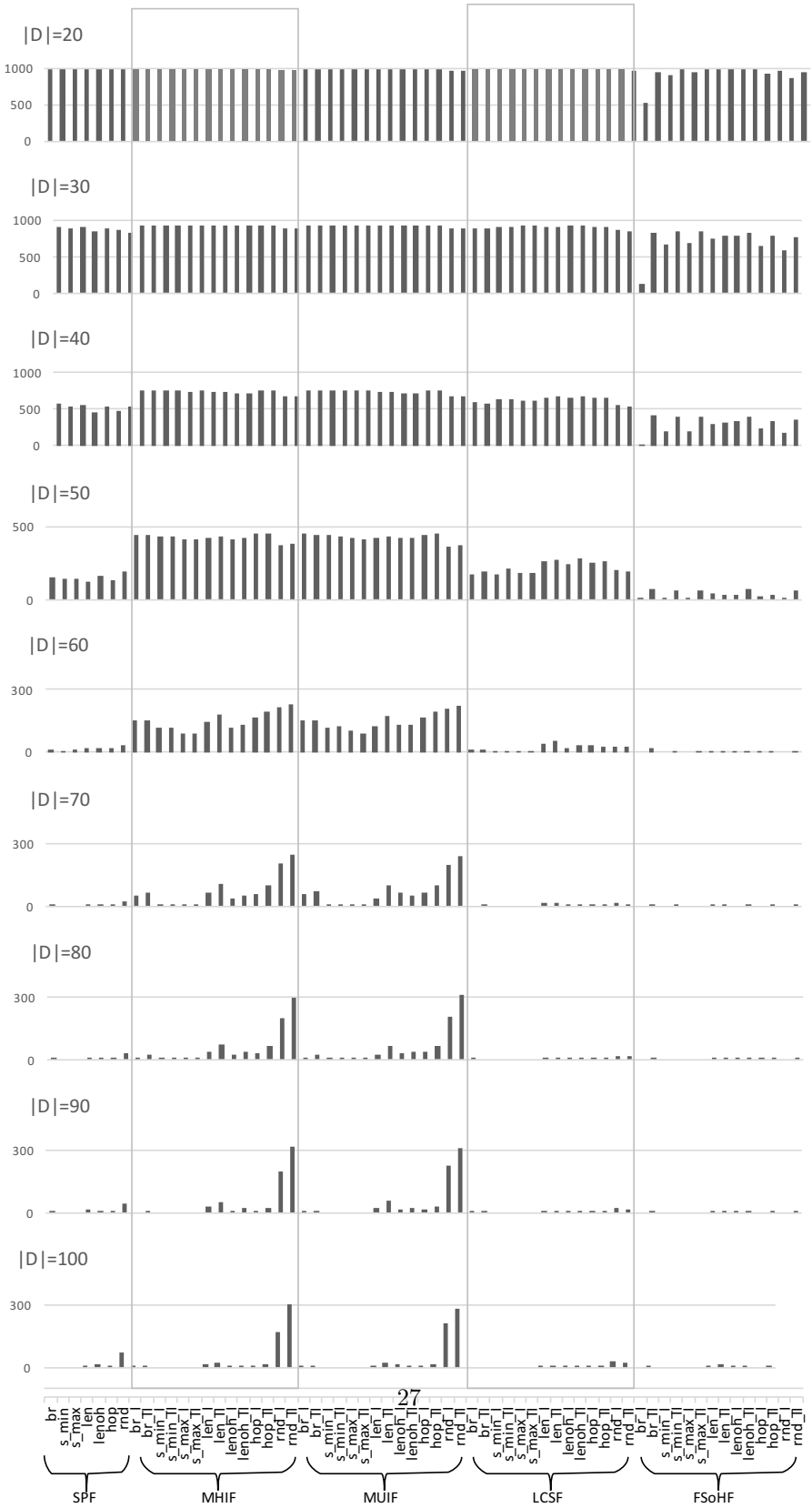
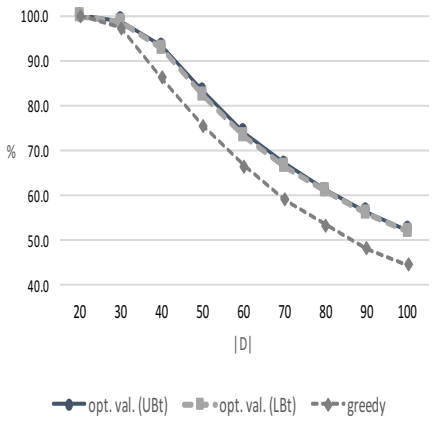
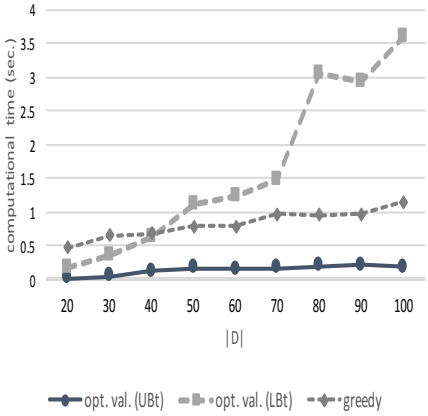


Figure 14: Frequency of achieving best traffic volume rates among greedy algorithms for each $|D|$ of 1000 instances in JPN25



(a) lower and upper bounds of traffic volume rate



(b) computational time

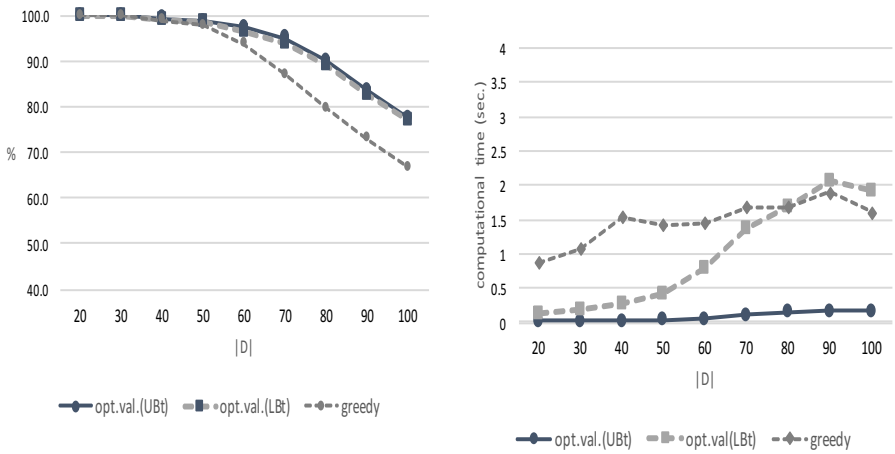
Figure 15: Comparing lower and upper bounds of traffic volume rate and computational times for obtaining them in accordance with demand size $|D|$ in EURO16

Table 14: Precision of bounds and computational time to for finding exact solutions

		EURO16			JPN25						
	$\|D\|$	20	30	40	20	30	40	50	60	70	80
# instances	UB=LB	49	46	40	50	50	46	38	35	35	32
	UB = opt.val.	49	50	47	–	–	48	49	46	42	41
time (sec.)	average	0.29	6.78	38.96	0.15	0.22	1.19	2.07	34.43	9536.13	1878.73
	max	5.73	245.76	636.59	0.24	0.36	36.17	9.02	659.86	448096.44	47491.66
	min	0.12	0.14	0.25	0.08	0.13	0.16	0.22	0.29	0.58	0.72

We summarize the results in our experiments for the maximum traffic volume model.

- With respect to the performance of the greedy algorithms,
  - JPN25 had a more tractable structure than EURO15.
  - Type II were superior to Type I
  - the best method depended on the network and the traffic congestion level.
  - The demand ordering policy rnd become efficient when the size of demand was large.
- With respect to the exact algorithm,
  - the upper bound obtained from (UB_t) was equivalent to the lower bound obtained from (LB_t) for about 60% of our instances.
  - the relaxation of spectrum continuity derived good upper bounds.



(a) lower and upper bounds of traffic volume rate

(b) computational time

Figure 16: Comparing lower and upper bounds of traffic volume rate and computational times for obtaining them in accordance with demand size $|D|$ in JPN25

6. Conclusion

In this paper, we investigated upper/lower bounds and computed exact optimal solutions for RSA problems of anycasting. Our computational experiments showed that the solutions obtained by relaxing the spectrum continuity constraint were almost optimal. This result might be unique to anycast transmission. For the minimax slot number model, the lower bound given by relaxing the spectrum continuity constraint performed well in terms of both computational time and accuracy in our experiments. Surprisingly, our lower bound coincided with the optimal slot numbers in all of our instances. Further work is needed to verify the optimality of this lower bound. For the maximum traffic volume model, the relaxation of the spectrum continuity also gave good bounds. However, when we found the exact solutions, small gaps between bounds caused the IP solver to take a lot of time to certify the optimality. Thus, we need other techniques in order to obtain an exact solution efficiently with an IP solver for larger size instances.

For the greedy algorithms, the route selection criterion was important for the minimax slot number model. We conclude that it is necessary to select a shorter path under some slot conditions. In the maximum traffic volume model, there was no dominant greedy algorithm, and the best route selection criterion and best demand ordering policy depended on the network and traffic congestion level. However, when the number of demands increased, the algorithm employing random ordering of demands performed well. This fact implies that we may expect that greedy algorithms with a route selection criterion that is appropriate for a given network will perform in dynamic cases as well as in static cases

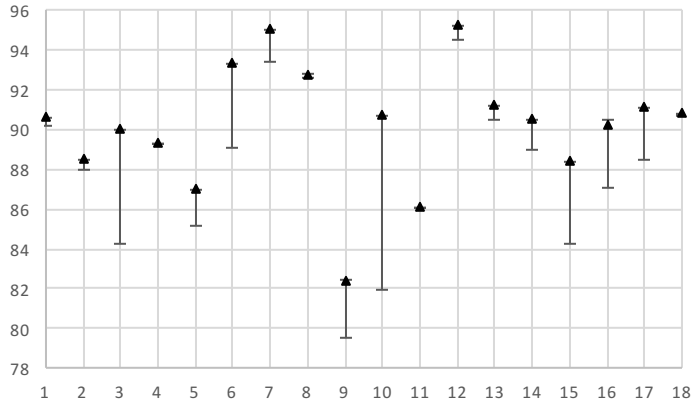


Figure 17: Comparing exact values and upper and lower bounds obtained from (UB_t) and (LB_t) for instances with $|D| = 80$ in JPN25

when the traffic congestion level is high in our instances. However, the relative tolerances of the gaps between objective values obtained by the greedy algorithms and the optimal values were large when the number of demands became large. Therefore, we might need a distinct method for the greedy algorithm for the maximum traffic volume model.

Acknowledgements

We would like to thank Tomohiro Kudoh for the valuable comments. This research is supported in part by MEXT Grant-in-Aid for Scientific Research (B) No. 16H03118.

References

[1] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies," IEEE Commun. Mag., vol. 47, pp. 66–73, Nov. 2009.

[2] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, "Elastic optical networking: A new dawn for the optical layer?," IEEE Commun. Mag., vol. 50, pp. s12–s20, Feb. 2012.

[3] G. Zhang, M. De Leenheer, A. Morea, and B. Mukherjee, "A survey on OFDM-based elastic core optical networking," IEEE Commun. Surv. Tut., vol. 15, pp. 65–87, Feb. 2013.

[4] I. Tomkos, S. Azodolmolky, J. Solé-Pareta, D. Careglio and E. Palkopoulou, "A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges," Proc. IEEE, vol. 102, pp. 1317–1337, Sept. 2014.

[5] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Salama, and G.N. Rouskas, "Spectrum management techniques for elastic optical networks: A survey," Opt. Switch. Netw., vol. 13, pp. 34–48, July 2014.

[6] B.C. Chatterjee, N. Sarma, and E. Oki,"Routing and spectrum allocation in elastic optical networks: A tutorial," IEEE Commun. Surv. Tut., vol. 17, pp. 1776–1800, May, 2015.

[7] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," INFOCOM, 2011, Proc. IEEE pp. 1503-1511, Apr. 2011.

[8] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," J. Lightwave Technol., vol. 29, pp. 1354–1366, Mar. 2011.

[9] M. Klinkowski and K. Walkowiak, "Routing and spectrum assignment in spectrum sliced elastic optical path network," IEEE Comm. Lett., vol. 15, pp. 884–886, June 2011.

[10] L. Velasco, M. Klinkowski, M. Ruiz, and J. Comellas, "Modeling the routing and spectrum allocation problem for flexgrid optical networks," Photonic Netw. Commun., vol. 24, pp. 177–186, Dec. 2012.

[11] M. Kucharzak, K. Walkowiak, P. Kopec, D. Bulira, and A. Kasprzak, "Integer programming, constraint programming, and metaheuristic approaches for static optimization of anycast flows in elastic optical networks, " Networks, vol. 66, pp. 253–266, Sept. 2015.

[12] R.W. Alaskar, I. Ahmad, and A. Alyatama, "Offline routing and spectrum allocation algorithms for elastic optical networks," Opt. Switch. Netw., vol. 21, pp. 79–92, July 2016.

[13] F.S. Abkenar and A.G. Rahbar, "Study and analysis of routing and spectrum allocation (RSA) and routing, modulation and spectrum allocation (RMSA) algorithms in elastic optical networks (EONs)," Opt. Switch. Netw., vol.23, pp.5–39, Jan. 2017.

[14] H. Wu, F. Zhou, Z. Zhu, and Y. Chen,"An experimental comparison of routing and spectrum assignment algorithms in elastic optical networks," In: S. Lasaulce, T. Jimenez, and E. Solan (eds), Network Games, Control, and Optimization, Avignon, France, pp. 211–221, 2017.

[15] R. Goścień, K. Walkowiak, and M. Klinkowski, "Tabu search algorithm for routing, modulation and spectrum allocation in elastic optical network with anycast and unicast traffic," Comput. Netw., vol. 79, pp. 148–165, Mar. 2015.

[16] P. Nagły and K. Walkowiak "Simulated annealing algorithm for minimization of bandwidth fragmentation in elastic optical networks with multicast and unicast flows," In: K. Jackowski, R. Burduk, K. Walkowiak, M. Wozniak, H. Yin (eds) , Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science, vol 9375. pp. 318–327, Oct. 2015.

[17] X. Wan, N. Hua, H. Zhang, and X. Zheng,"Study on dynamic routing and spectrum assignment in bitrate flexible optical networks," Photonic Netw. Commun., vol. 24. pp. 219–227, May 2012.

[18] J. Morell and G. Sahin, "Distance-adaptive routing and spectrum assignment of deadline-driven requests in reconfigurable elastic optical networks," in Proc. of ICSNC 2012, pp. 175–179, Nov. 2012.

[19] L. Zhang, W. Lu, X. Zhou, and Z. Zhu, "Dynamic RMSA in spectrum-sliced elastic optical networks for high-throughput service provisioning," in Proc. Int. Conf. Comput. Netw. Commun., pp. 380–384, Jan. 2013.

[20] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," J. Lightwave Technol., vol. 31, pp. 15–22, Jan. 2013.

[21] I. Olszewski, "Dynamic RSA problem for time-varying traffic in spectrum sliced elastic optical path network," Int. J. Electron. Telecommun., vol. 61, pp. 179–184, June 2015.

[22] B.C. Chatterjee, W. Fadini, and E. Oki "A spectrum allocation scheme based on first-last-exact fit policy for elastic optical networks," J. Netw. Comput. Appl., vol. 68, pp. 164–172, June 2016.

[23] E. Biernacka, J. Domżał, and R. Wójcik, "Investigation of dynamic routing and spectrum allocation methods in elastic optical networks," Int. J. Electron. Telecommun., vol. 63, pp. 85–92, Mar. 2017.

[24] K. Walkowiak and M. Klinkowski, "Joint anycast and unicast routing for elastic optical networks: Modeling and optimization," in Proc. IEEE ICC 2013, pp. 9–13, June 2013.

[25] K. Walkowiak, R. Goścień, and M. Klinkowski, "On minimization of the spectrum usage in elastic optical networks with joint unicast and anycast traffic," in Proc. Asia Communications and Photonics Conference, Sept. 2013.

[26] L. Zhang and Z. Zhu,"Dynamic anycast in inter-datacenter networks over elastic optical infrastructure." in Proc. IEEE ICNC, pp. 491–495, Feb. 2014.

[27] L. Zhang and Z. Zhu, "Spectrum-efficient anycast in elastic optical inter-datacenter networks," Opt. Switch. Netw., vol. 14, pp.250-259, Aug. 2014.

[28] M. Aibin and K. Walkowiak. "Simulated annealing algorithm for optimization of elastic optical networks with unicast and anycast traffic," in Proc. of IEEE ICTON, Th.B3.6, July 2014.

[29] L. Peng, K. Park, and C-H. Youn, "Investigation on static routing and resource assignment of elastic all-optical switched intra-datacenter networks," Sci. China Inform. Sci., vol. 59, 102304, Sept. 2016.

[30] S. D. Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz, "Pan-European optical transport networks: An availability-based comparison," Photonic Netw. Commun., vol. 5, pp. 203–225, May 2003.

[31] Japan Photonic Network Model. IEICE Technical Committee on Photonic Network. (http://www.ieice.org/˜pn/jpn/jpnm.html, https://www.ieice.org/cs/pn/jpn/JPNM/JPNM_v20161013sn(E).zip)

[32] IBM ILOG CPLEX Optimization Studio. (http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/)

Appendix

Table 15: The average numbers of needed slots and its standard deviations, which are in parentheses, for 1000 instances of 20 demands in EURO16.

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	75.17 (21.85)	54.65 (11.19)	56.94 (13.58)	79.72 (17.82)	197.79 (36.46)
	s_min		53.48 (10.93)	55.64 (12.22)	74.78 (15.46)	193.24 (36.34)
	s_max		54.65 (11.19)	56.94 (13.58)	79.72 (17.82)	197.79 (36.46)
	len		55.67 (10.87)	58.93 (12.22)	74.85 (15.49)	189.17 (16.24)
	lenoh		67.89 (15.91)	69.81 (15.94)	96.86 (25.59)	199.67 (36.74)
	hop		53.44 (10.80)	56.80 (12.24)	73.01 (15.62)	191.59 (36.27)
	rnd		58.81 (12.65)	59.28 (13.03)	82.29 (18.94)	201.30 (36.35)
II	br		50.36 (9.86)	50.39 (9.87)	78.93 (17.52)	161.46 (31.67)
	s_min		49.97 (10.07)	50.00 (10.05)	73.61 (14.87)	142.83 (28.80)
	s_max		50.36 (9.86)	50.39 (9.87)	78.93 (17.52)	161.46 (31.67)
	len		52.93 (9.98)	52.93 (9.92)	73.69 (15.11)	156.26 (31.63)
	lenoh		58.19 (11.17)	58.20 (11.17)	88.63 (22.57)	159.57 (32.07)
	hop		50.57 (9.93)	50.58 (9.95)	71.35 (14.73)	158.34 (31.51)
	rnd		54.00 (10.80)	54.03 (10.79)	81.49 (18.32)	162.23 (31.93)

Table 16: The average numbers of needed slots and its standard deviations, which are in parentheses, for 1000 instances of 20 demands in JPN25.

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	56.68 (15.83)	28.44 (8.51)	28.50 (8.66)	41.76 (11.80)	63.49 (14.88)
	s_min		27.85 (8.42)	27.89 (8.40)	43.87 (12.43)	63.42 (14.65)
	s_max		27.81 (8.37)	27.80 (8.39)	45.01 (12.55)	63.38 (14.44)
	len		28.40 (8.43)	28.48 (8.44)	41.83 (13.14)	65.16 (14.19)
	lenoh		28.52 (8.57)	28.56 (8.56)	42.57 (13.31)	65.36 (14.39)
	hop		28.25 (8.48)	28.38 (8.49)	41.28 (12.67)	65.11 (14.47)
	rnd		28.44 (8.51)	28.50 (8.66)	41.76 (11.18)	65.48 (15.34)
II	br		28.41 (8.54)	28.44 (8.69)	49.23 (14.93)	65.53 (14.87)
	s_min		27.77 (8.47)	27.76 (8.49)	43.55 (12.26)	61.85 (14.34)
	s_max		27.76 (8.42)	27.74 (8.44)	45.58 (12.52)	63.40 (14.43)
	len		28.21 (8.58)	28.26 (8.55)	41.96 (12.80)	65.18 (14.16)
	lenoh		28.37 (8.67)	28.39 (8.65)	43.07 (12.81)	65.39 (14.37)
	hop		28.15 (8.55)	28.16 (8.55)	41.38 (12.49)	65.11 (14.46)
	rnd		29.86 (9.49)	29.91 (9.59)	40.12 (11.25)	65.54 (15.46)

Table 17: The average numbers of needed slots and its standard deviations, which are in parentheses, for 1000 instances of 100 demands in EURO network.

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	331.58 (52.07)	235.62 (22.20)	235.85 (22.74)	338.58 (50.29)	868.01 (87.26)
	s_min		229.14 (20.43)	232.39 (22.02)	319.33 (41.88)	840.28 (90.59)
	s_max		235.62 (22.20)	235.85 (22.74)	338.58 (50.29)	868.01 (87.26)
	len		255.76 (22.74)	252.61 (23.44)	321.82 (42.45)	829.73 (91.34)
	lenop		333.11 (40.33)	333.29 (40.35)	370.25 (56.38)	871.36 (86.93)
	hop		231.81 (22.72)	234.62 (23.72)	317.20 (41.36)	842.26 (92.37)
	rnd		243.44 (24.08)	244.13 (24.92)	337.95 (49.52)	871.36 (86.93)
II	br		220.26 (19.28)	220.26 (19.30)	337.38 (49.04)	710.72 (70.21)
	s_min		219.40 (20.45)	219.44 (20.46)	316.37 (42.22)	627.68 (74.46)
	s_max		220.26 (19.28)	220.26 (19.30)	337.38 (49.04)	710.72 (70.21)
	len		246.66 (21.27)	246.68 (21.21)	316.40 (43.73)	668.85 (69.85)
	lenoh		278.75 (22.99)	278.75 (22.99)	368.04 (56.91)	709.87 (70.99)
	hop		220.69 (21.33)	220.88 (21.32)	312.77 (42.65)	699.63 (72.56)
	rnd		226.94 (21.02)	226.96 (20.95)	337.08 (49.83)	709.99 (71.39)

Table 18: The average numbers of needed slots and its standard deviations, which are in parentheses, for 1000 instances of 100 demands in JPN25 network.

		SPF	MHIF	MUIF	LCSF	FSoHF
I	br	248.85 (31.77)	110.25 (17.63)	110.16 (17.70)	224.34 (29.77)	248.34 (31.56)
	s_min		108.17 (17.81)	108.12 (17.84)	227.95 (29.78)	252.46 (30.08)
	s_max		109.30 (17.00)	108.93 (17.23)	234.51 (29.56)	250.30 (31.21)
	len		109.88 (18.60)	110.10 (18.43)	225.44 (30.46)	261.55 (32.07)
	lenoh		110.11 (18.27)	110.05 (18.26)	232.59 (30.03)	232.59 (30.03)
	hop		108.33 (18.07)	108.37 (18.17)	224.90 (30.35)	264.69 (29.81)
	rnd		112.84 (18.35)	112.92 (18.38)	223.15 (29.67)	254.70 (32.43)
II	br		110.12 (17.70)	110.13 (17.70)	225.43 (29.73)	249.04 (31.38)
	s_min		108.14 (17.83)	108.14 (17.83)	228.18 (29.67)	245.18 (29.73)
	s_max		109.31 (16.98)	109.32 (17.00)	234.79 (29.16)	250.78 (30.93)
	len		109.70 (18.76)	109.70 (18.76)	229.05 (30.36)	261.47 (32.14)
	lenoh		109.91 (18.44)	109.90 (18.45)	229.87 (30.15)	263.64 (32.12)
	hop		108.18 (18.24)	108.18 (18.25)	225.79 (30.42)	264.66 (29.87)
	rnd		112.73 (18.41)	112.70 (18.40)	224.12 (29.60)	255.77 (32.07)