Journal of the Operations Research
Society of Japan
Vol. 42, No. 1, March 1999

AN EFFICIENT ALGORITHM FOR THE MINIMUM-RANGE IDEAL PROBLEM

Toshio Nemoto
Bunkyo University

(Received January 12, 1998)

Abstract Suppose we are given a partially ordered set, a real-valued weight associated with each element and a positive integer k. We consider the problem which asks to find an ideal of size k of the partially ordered set such that the range of the weights is minimum. We call this problem the minimum-range ideal problem. This paper shows a new and fast $O(n \log n + m)$ algorithm for this problem, where n is the number of elements and m is the smallest number of arcs to represent the partially ordered set. It is also proved that this problem has an $\Omega(n \log n + m)$ lower bound. This means that the algorithm presented in this paper is optimal.

1. Introduction

In loading k containers into a ship out of a pile of containers, a problem of considerable importance is how to select the k containers. Each of the containers has a different weight. To keep the balance of the ship, each of the k containers should be as equal in weight as possible. However, the need to save time and ensure safety impose some restrictions. For example, we can never carry a container into the ship until we have carried every container located above it; there might be slope constraints that prevent the walls of a pile of containers from being too steep. In short, there are precedence restrictions to the selection of k containers.

To formalize our mathematical model, we consider an arbitrary finite set E and *precedence constraints* between two elements $e, e' \in E$: the element e *precedes* the element e', denoted by $e \preceq e'$, means that we must select e if we decide to select e'. This precedence constraint $\preceq$ is a *partial order* on E since it satisfies reflexivity, antisymmetry and transitivity. We call the pair $\mathcal{P} = (E, \preceq)$ a *partially ordered set* or *a poset* for short. For a poset $\mathcal{P} = (E, \preceq)$ a subset I of E is called an *ideal* if $e \preceq e' \in I$ always implies $e \in I$. Furthermore, assume that there is a real-valued weight $w(e)$ associated with each element $e \in E$. For a nonempty ideal I of $\mathcal{P}$, the *range of* I is defined to be the maximum difference among weights of elements in I, i.e., $\max_{e \in I} w(e) - \min_{e \in I} w(e)$. Here, assume the range of the empty ideal to be 0. Namely, the objective of our mathematical model is to find an ideal of size k in which the range of the ideal is as small as possible. We call such an ideal a *minimum-range ideal* (*of size* k) and this problem the *minimum-range ideal problem.*

The optimization problem on the ideal is valuable because many applications in real-life are formalized as the ideal problem, including the so-called *closure* problem (see [8]). Therefore, various types of problems have been well researched. Recently for the cardinality-restricted ideal problem including our problem some results were studied by [2]. To the author's knowledge, no one has ever considered the minimum-range ideal problem. We hope that an efficient algorithm for this problem serves as a subroutine of algorithms for another combinatorial optimization problems such as stable marrige problems, scheduling plannings and so on.

The minimum-range problem is also an interesting combinatorial optimization problem. Several researchers have studied minimum-range problems, including the minimum-range assignment problem [6], the minimum-range spanning tree problem [1] and [4], and the minimum-range cut problem [5]. For these problems, a general algorithm has been proposed in [6]. Essentially, the above three problems use general algorithm approach with their particular property. Simply applying the general algorithm in [6], the minimum-range ideal problem can be solved in $O((m+n)n)$ time, where n is the number of elements of a given poset $\mathcal{P}$ and m is the smallest number of arcs to represent $\mathcal{P}$.

However, we propose a faster $O(n \log n + m)$ time algorithm by a new approach. There are two major differences. One is that the general algorithm approach essentially makes use of a sorted list of the weights, whereas, instead of it, our algorithm makes use of two new ordered lists introduced here as a preprocessor scheme. The other is that the general algorithm approach requires the feasibility-check procedure, whereas, our algorithm does not use it. Instead, it has two procedures, each of which solves the *minimax ideal problem* and the *maximin ideal problem*, respectively. Furthermore, the point we wish to emphasize is that our algorithm solves the minimum-range problem as a sequence of a pair of the minimax ideal problem and the maximin ideal problem in such a way that the total running time is comparable to the time to solve a pair of them once. This approach leads us to an $O(n \log n + m)$ time algorithm. This is optimal because we prove that the minimum-range ideal problem has an $\Omega(n \log n + m)$ lower bound.

This paper is organized as follows. Section 2 presents descriptions of the minimum-range ideal problem and some definitions. Section 3 considers the minimax ideal problem and the maximin ideal problem, which play an important role to solve the minimum-range ideal problem. Section 4 introduces new two orders, minimax-order and maximin-order, defined on a poset. Section 5 gives an $O(n \log n + m)$ algorithm. Finally, Section 6 proves that the minimum-range ideal problem has an $\Omega(n \log n + m)$ lower bound and our improved algorithm is optimal.

2. Preliminaries

We consider the *minimum-range ideal problem* for a poset $\mathcal{P} = (E, \preceq)$ and a positive integer k as follows:

$$\begin{aligned} \mathrm{P_{range}}: \quad & \text{Minimize} \quad \max_{e \in I} w(e) - \min_{e \in I} w(e) && (2.1)\\ & \text{subject to} \quad I \in \mathcal{I}(\mathcal{P}), && (2.2)\\ & \qquad\qquad\quad |I| = k, && (2.3)\end{aligned}$$

where $\mathcal{I}(\mathcal{P})$ is the set of all the ideals of $\mathcal{P}$, and $|X|$ is the cardinality of a finite set X. For convenience, let $\mathcal{I}_k(\mathcal{P})$ denote the set of all the feasible ideals of $\mathrm{P_{range}}$. Without loss of generality, we assume throughout this paper that $\mathcal{I}_k(\mathcal{P}) \neq \emptyset$ and for each element $e \in E$ there exists an ideal including e in $\mathcal{I}_k(\mathcal{P})$. We claim that the cardinality constraint, $|I| = k$, is essential to this problem since the range is nondecreasing with respect to k. Hence, an ideal of size 0 or 1 becomes optimal when the cardinality constraint is dropped or replaced with $|I| \leq k$. Furthermore, we can easily understand from above that the solution for $|I| = k$ also gives the one for $|I| \geq k$.

The important observation for the minimum-range ideal problem is that we can transform it to an enumeration problem. For a closed interval $[\alpha, \beta]$ with $\alpha < \beta$, suppose that there is an ideal I such that $w(e) \in [\alpha, \beta]$ for all element $e \in I$. If there is no closed interval $[\alpha', \beta']$ properly included in $[\alpha, \beta]$, then the closed interval $[\alpha, \beta]$ is called *critical*. We call such an ideal a *critical ideal*. Thus, we can reduce the minimum-range ideal problem to the problem of finding a critical interval $[\alpha, \beta]$ such that $\beta - \alpha$ is minimum. Therefore, if we can enumerate all critical intervals, then we can identify a minimum-range ideal in the critical ideals.

The general approach to enumerate all critical intervals is described as follows. First, we sort all distinct values in weights. Then, for all distinct intervals, we apply a feasibility check—which there exists an ideal in the interval or not. To find all critical intervals it is necessary to do the feasibility check $O(n)$ times by the systematic approach [6]. For the ideal problem one feasibility check requires $(m+n)$ times. Therefore, we can solve the problem in $O((m+n)n)$ times by general approach.

On the other hand, We improve this time bound to $O(n \log n + m)$ by a new approach. Before discussing the new algorithm, we introduce some technical terms. For an element $e \in E$ an element e' is an *lower neighbor* of e if $e' \preceq e$ and there exists no element $\hat{e}$ such that $e \preceq \hat{e} \preceq e'$. If an element $e \in E$ has no lower neighbor, then we say that e is *minimal*. For a subset $H \subseteq E$ we call $\mathcal{P}(H) = (H, \preceq_H)$ a *subposet of* $\mathcal{P}$ *induced by* H if $\preceq_H$ is the set of all the partial orders between e and e' such that $\{e, e'\} \subseteq H$.

3. The Minimax(Maximin) Ideal Problem

In this section we consider the *minimax ideal problem* and the *maximin ideal problem* which play an important role to solve $\mathrm{P}_{\mathrm{range}}$. For a poset $\mathcal{P} = (E, \preceq)$ and a positive integer k, each problem is defined respectively as follows:

$$\mathrm{P}_{\mathrm{minimax}}: \quad \text{Minimize} \quad \max_{e \in I} w(e) \tag{3.1}$$

$$\text{subject to} \quad I \in \mathcal{I}_k(\mathcal{P}), \tag{3.2}$$

$$\mathrm{P}_{\mathrm{maximin}}: \quad \text{Maximize} \quad \min_{e \in I} w(e) \tag{3.3}$$

$$\text{subject to} \quad I \in \mathcal{I}_k(\mathcal{P}). \tag{3.4}$$

An optimal solution of $\mathrm{P}_{\mathrm{minimax}}$ (resp., $\mathrm{P}_{\mathrm{maximin}}$) is called a *minimax ideal* (resp., *maximin ideal*) of $\mathcal{P}$. A minimax ideal is found by applying the following algorithm based on a "greedy" principle — that is, it makes the cheapest choice at each step.

Algorithm MINIMAX($\mathcal{P}, k$)

Step 1: Put $J := \emptyset$ and $C := \{e | e \text{ is a minimal element of } \mathcal{P}\}$.

Step 2: Repeat the following ($\sharp$) k times.

($\sharp$) If $C \neq \emptyset$, then find a minimum-weight element $\hat{e}$ in C, and update $J := J \cup \{\hat{e}\}$ and $C := \{e | e \text{ is a minimal element of } \mathcal{P}(E - J)\}$; else stop (there is no feasible ideal of $\mathcal{P}$).

Step 3: Stop. The current J is a minimax ideal.

The validity of this algorithm is shown below.

Theorem 3.1: MINIMAX($\mathcal{P}, k$) *computes a minimax ideal of* $\mathcal{P}$ *in* $O(n \log n + m)$.

(Proof) For an ideal $\hat{I}$ found by MINIMAX($\mathcal{P}, k$), let $\hat{e}$ be a maximum-weight element in $\hat{I}$, and let C' and J' be the set C and J, respectively, in MINIMAX($\mathcal{P}, k$) just before $\hat{e}$ is chosen. Similarly, for a minimax ideal I^*, let e^* be a maximum-weight element in I^*.

Suppose $w(\hat{e}) > w(e^*)$. If $e^* \notin J'$, then $I^* \cap C' \neq \emptyset$, i.e., I^* has an element $\tilde{e} \in I^* \cap C'$ such that $w(e^*) < (w(\hat{e}) \leq) w(\tilde{e})$, contradicting the fact that e^* is the maximum-weight element in I^*. If $e^* \in J'$, then there exists an element $\tilde{e} \in C' \cap I^*$ such that $w(\tilde{e}) \leq w(e^*)(< w(\hat{e}))$ because $|I^*| > |J'|$ and e^* has maximum-weight in I^*. This contradicts the choice of $\hat{e}$. Consequently, we have $w(\hat{e}) = w(e^*)$.

We now turn to the time complexity analysis. In Step 2, it is not difficult to renew the set C if we have the list of lower neighbors of each element in $\mathcal{P}(E-J)$. That is, let C_i be C at ith iteration, we obtain $C_{i+1} = (C_i - \{\hat{e}\}) \cup \{e|$ the list of lower neighbors of e have just become empty by removing $\hat{e}$ from $\mathcal{P}(E-J)\}$. Notice that finding C_1 and identifying the element whose list have just become empty require $O(m+n)$ time in the whole of the algorithm. By having a heap data structure for C, finding a minimum-weight element $\hat{e}$ in C, inserting new members to C and deleting $\hat{e}$ from C are carried out in $O(\log n)$, respectively. Since each operation is done for an element at most once, it takes $O(n \log n)$ time. In total, this algorithm requires $O(n \log n + m)$ time. □

From $\max\min_{e\in I}(-w(e)) = -\min\max_{e\in I} w(e)$, we obtain an algorithm to solve P_{maximin} correctly by replacing "find a minimum-weight" in Step 2 of MINIMAX($\mathcal{P}$,k) with "find a maximum-weight". We call it MAXIMIN($\mathcal{P}$,k).

Remark: Based on an algorithm proposed by Gabow and Tarjan[3]for the bottleneck spanning tree problem, we can construct another $O(m \log^* n)$ algorithm for the minimax ideal problem. Here, $\log^*$ is the *iterated logarithm*, defined by $\log^{(0)} x = x, \log^{(i+1)} x = \log\log^{(i)} x$ and $\log^* x = \min\{i | \log^{(i)} x \leq 1\}$. Notice that $\log^* x$ is very slowly growing function. This is an improvement if m is sufficiently small, i.e., $m << (\frac{n}{2})^2$. However, we do not refer to this approach in this paper since it is not suited for constructing an efficient algorithm for the minimum-range ideal problem.

4. New Two Orders on a Poset

We introduce a new preprocessor scheme, which uses two new orders on E defined below. In carrying out MINIMAX($\mathcal{P}, n$), every element in E belongs to J in turn. The *minimax-order of* $\mathcal{P}$ is defined as the order in which MINIMAX($\mathcal{P}, n$) considers the element. In a similar fashion, let the *maximin-order of* $\mathcal{P}$ be the order in which MAXIMIN($\mathcal{P}, n$) considers the element. Notice that the minimax-order is defined uniquely by specifying the rule for selecting a minimum-weight element in C. We show a fundamental theorem on these orders, which is a base for a validation of a new algorithm for P_{range} proposed in next section.

Theorem 4.1: *For an ideal I of $\mathcal{P}$, an order defined by restricting the minimax-order (resp., the maximin-order) of $\mathcal{P}$ on I is a minimax-order (resp., a maximin-order) of $\mathcal{P}(I)$.*

(Proof) Let us give each element in I an index, $1, \ldots, |I|$, in a minimax-order of $\mathcal{P}$. Suppose e_i is also the ith element in I in the sense of a minimax-order of $\mathcal{P}(I)$. We show that e_{i+1} is also the $(i+1)$th element in the same sense by induction. Let C' and J' be the set C and J, respectively, just after e_i has belonged to J in MINIMAX($\mathcal{P}, n$). If $e_{i+1} \notin C' \cap I$, then there exists $\hat{e} \in C' \cap I$ such that $\hat{e} \preceq e_{i+1}$ since I is an ideal. This contradicts the fact that e_{i+1} is the first element in I which belongs to J after e_i has belonged to J. Therefore, $e_{i+1} \in C' \cap I$. Since e_{i+1} is the smallest in the minimax-order of $\mathcal{P}$ in $C' \cap I$, e_{i+1} is a minimum-weight element in $C' \cap I$. It follows that e_{i+1} is also the $(i+1)$th element in the sense of the minimax-order of $\mathcal{P}(I)$. Similarly, we have the fact that e_1 has minimum-weight in the set of all the minimal elements of $\mathcal{P}(I)$. Hence, e_1 is also the first element in I in the minimax-order of $\mathcal{P}(I)$. Consequently, we prove this theorem by induction. □

From Theorem 3.1 and Theorem 4.1 we have the following corollary.

Corollary 4.2: *For an ideal J of $\mathcal{P}$ and a positive integer k with $k \leq |J|$, a subset $I \subseteq J$ consisting of k elements in minimax-order (resp., maximin-order) of $\mathcal{P}$ is a minimax ideal (resp., maximin ideal) of $\mathcal{P}(J)$.* □

Example: Consider the minimax-order and the maximin-order on a poset $\mathcal{P}$ represented by the Hasse diagram shown in Figure 1. The weight of each element is attached at the lower left of the element. We have the minimax-order and the maximin-order as (a) and (b) in Figure 2, respectively.

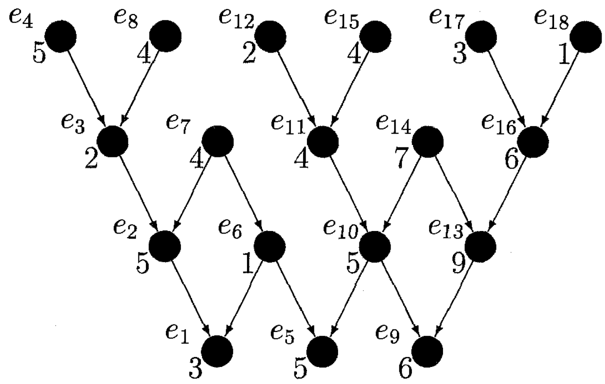
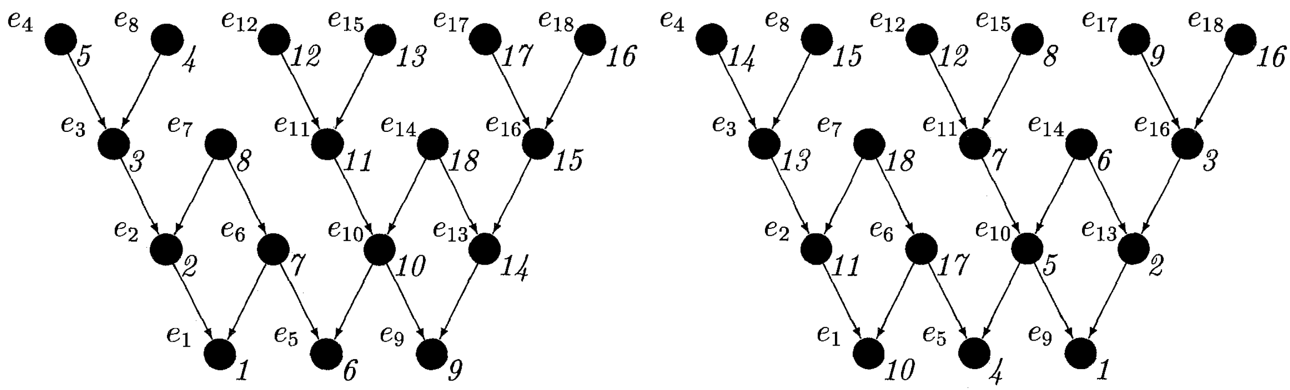


Figure 1: A poset $\mathcal{P}$ with weights.



(a) The minimax-order. **(b)** The maximin-order.

Figure 2: The minimax-order and the maximin-order of the poset shown in Figure 1.

5. Algorithm

In this section we propose an algorithm RANGE$(\mathcal{P}, k)$ for finding a minimum-range ideal of size k as follows.

Algorithm RANGE$(\mathcal{P}, k)$

Step 1: Define a minimax-order and a maximin-order of $\mathcal{P}$.
Put $(\alpha^*, \beta^*) := (-\infty, \infty)$, $S := E$ and $J = \emptyset$.

Step 2:

(2-1) Repeat the following (♯1)

(♯1) If $S \neq \emptyset$, then find the smallest element $\hat{e} \in S$ in minimax-order, and update $J := J \cup \{\hat{e}\}$ and $S := S - \{\hat{e}\}$; else go to Step 3.

until $|J| = k$. Put $\beta := \max\{w(e)|e \in J\}$.

(2-2) Repeat the following (♯2)

(♯2) If $S \neq \emptyset$, then find the smallest element $\hat{e} \in S$ in minimax-order, and if $w(\hat{e}) \leq \beta$, then update $J := J \cup \{\hat{e}\}$ and $S := S - \{\hat{e}\}$.

until $(w(\hat{e}) > \beta$ or $S = \emptyset)$.

(2-3) If $|J| > k$, then repeat the following (♭1)

(♭1) Find the largest element $\check{e} \in J \cup S$ in maximin-order, and update $J := J - \{\check{e}\}$ and $S := S - \{\check{e}\}$.

until $|J| = k$. Put $\alpha := \min\{w(e)|e \in J\}$ and $D := \{e|e \in J, w(e) = \alpha\}$. If $\beta^* - \alpha^* > \beta - \alpha$, then update $(\alpha^*, \beta^*) := (\alpha, \beta)$.

(2-4) Repeat the following (♭2)

(♭2) Find the largest element $\check{e} \in J \cup S$ in maximin-order, and update $J := J - \{\check{e}\}$, $S := S - \{\check{e}\}$ and $D := D - \{\check{e}\}$.

until $D = \emptyset$. Go to (2-1).

Step 3: Put $\beta := \beta^*$, $S := E$ and $J := \emptyset$. Then do (2-2),(2-3) and stop. The current J is a minimum-range ideal.

Let α_i and β_i $(i = 1, \cdots, l)$ be α and β computed at ith iteration of Step2, where l is the number of iteration in Step 2. And let $S_i^{(2-*)}$, $J_i^{(2-*)}$ and D_i denote the set S, J and D obtained just after Step (2-$*$) at ith iteration of Step 2. Here, we assume $S_0^{(2-4)} = E$, $J_0^{(2-4)} = \emptyset$, $\alpha_0 = -\infty$ and $\beta_0 = \infty$.

Lemma 5.1: *For a minimal element e of $\mathcal{P}(E - (J_i^{(2-2)} \cup S_i^{(2-2)}))$ we have either of the following statements:* (a) $w(e) \leq \alpha_{i-1}$; *or* (b) *there is an element $e' \in S_i^{(2-2)}$ such that $e' \preceq e$ on $\mathcal{P}$.*

(Proof) Let $\bar{e}$ be the smallest element in D in the sense of maximin-order. If there is a minimal element $\tilde{e}$ of $\mathcal{P}(E - (J_i^{(2-2)} \cup S_i^{(2-2)}))$ such that $w(\tilde{e}) > \alpha_{i-1}$, then $\tilde{e}$ is greater than $\bar{e}$ in maximin-order. Because all element deleted from $J_{i-1}^{(2-3)} \cup S_{i-1}^{(2-3)}$ after $\bar{e}$ is deleted at Step (2-4) in $(i-1)$th iteration of Step 2 have the same weight α_{i-1}. Thus, from the definition of maximin-order, $\tilde{e}$ satisfies one of the following two cases: (1) $\bar{e} \preceq \tilde{e}$ on $\mathcal{P}$; or (2) there is an element e' such that $w(e') < \alpha_{i-1}$ and $e' \preceq \tilde{e}$ on $\mathcal{P}$. Case (1) contradicts the assumption that $\tilde{e}$ is minimal element of $\mathcal{P}(E - (J_i^{(2-2)} \cup S_i^{(2-2)}))$. In the case (2), we have $e' \notin J_i^{(2-2)}$ from the definition of the set D. Therefore, we have $e' \in S_i^{(2-2)}$. □

Lemma 5.2: *For an ideal $\tilde{I}$ of $\mathcal{P}$ such that $\alpha_{i-1} < w(e) \leq \beta_i (e \in \tilde{I})$, $\tilde{I} \subseteq J_i^{(2-2)}$.*

(Proof) If there exists an element $\tilde{e} \in \tilde{I} - J_i^{(2-2)}$, then $\tilde{e}$ satisfies one of the following two cases: (1) $\tilde{e} \in E - (J_i^{(2-2)} \cup S_i^{(2-2)})$; or (2) $\tilde{e} \in S_i^{(2-2)}$. In case of (1), there are two cases from Lemma 5.1. In case of (a) in Lemma 5.1 we have $w(e) \leq \alpha_{i-1}$. In case of (2) and (b) in Lemma 5.1, $\tilde{I}$ includes an element e'' with $w(e'') > \beta_i$ since for any minimal element e of $\mathcal{P}(S_i^{(2-2)})$ we have $w(e) > \beta_i$. Both cases contradict the condition $\alpha_{i-1} < w(e) \leq \beta_i (e \in \tilde{I})$. □

Suppose that the set $J_{i-1}^{(2-4)} \cup S_{i-1}^{(2-4)}$ is an ideal of $\mathcal{P}$ from here till Lemma 5.8. Then, the following two lemmas immediate follows from Corollaly 4.2.

Lemma 5.3: *The set $J_i^{(2-1)}$ is a minimax ideal of size k of $\mathcal{P}(J_{i-1}^{(2-4)} \cup S_{i-1}^{(2-4)})$.*

Lemma 5.4: *The set $J_i^{(2-2)}$ is an ideal of $\mathcal{P}(J_{i-1}^{(2-4)} \cup S_{i-1}^{(2-4)})$.*

From Lemma 5.4 and Corollary 4.2 we have the following lemma.
Lemma 5.5: *The set $J_i^{(2-3)}$ is a maximin ideal of size k of $\mathcal{P}(J_i^{(2-2)})$.*

Lemma 5.6: *For all ideal I of size k of $\mathcal{P}(J_i^{(2-2)})$, $\max\{w(e)|e \in I\} = \beta_i$.*

(Proof) Suppose that there exists an ideal I' of $\mathcal{P}(J_i^{(2-2)})$ such that $|I'| = k$ and $\max\{w(e)|e \in I'\} < \beta_i$. Then, I' is also an ideal of $\mathcal{P}(J_{i-1}^{(2-4)} \cup S_{i-1}^{(2-4)})$ from Lemma 5.4. This contradicts that β_i is the optimal value of $\mathrm{P_{minimax}}$ on $\mathcal{P}(J_{i-1}^{(2-4)} \cup S_{i-1}^{(2-4)})$. □

Combining Lemma 5.5 and Lemma 5.6, we have the following lemma.
Lemma 5.7: *The set $J_i^{(2-3)}$ is a minimum-range ideal of size k of $\mathcal{P}(J_i^{(2-2)})$. Thus, $J_i^{(2-3)}$ is a critical ideal of $\mathcal{P}$.*

Lemma 5.8: *The set $J_i^{(2-4)} \cup S_i^{(2-4)}$ is an ideal of $\mathcal{P}$.*

(Proof) Immediate from Corollary 4.2. □

Notice that the set $E(= J_0^{(2-4)} \cup S_0^{(2-4)})$ is an ideal of $\mathcal{P}$. Therefore, by induction, we have
Theorem 5.9: *Each set $J_i^{(2-3)}(i = 1, \cdots, l)$ is a critical ideal of $\mathcal{P}$.*

Now, let us define for α_i

$$\mathcal{I}_k(\mathcal{P}, \alpha_i) = \{I \,|I \in \mathcal{I}_k(\mathcal{P}), \min\{w(e)|e \in I\} \le \alpha_i\}. \tag{5.1}$$

Then we have the following lemma.
Lemma 5.10: *Each set $J_i^{(2-3)}$ $(i = 1, \cdots, l)$ of $\mathcal{P}$ is a minimum-range ideal of size k in $\mathcal{I}_k(\mathcal{P}, \alpha_i) - \mathcal{I}_k(\mathcal{P}, \alpha_{i-1})$.*

(Proof) For a set $J_i^{(2-3)}$, $\alpha_{i-1} < w(e)(\le \beta_i)$ $(e \in J_i^{(2-3)})$ and there is an element $e \in J_i^{(2-3)}$ with $w(e) = \alpha_i$. Therefore, $J_i^{(2-3)} \in \mathcal{I}_k(\mathcal{P}, \alpha_i) - \mathcal{I}_k(\mathcal{P}, \alpha_{i-1})$. Suppose I^* is a minimum-range ideal on $\mathcal{I}_k(\mathcal{P}, \alpha_i) - \mathcal{I}_k(\mathcal{P}, \alpha_{i-1})$ such that $\max\{w(e)|e \in I^*\} - \max\{w(e)|e \in I^*\} < \beta_i - \alpha_i$. Notice that $\min\{w(e)|e \in I^*\} \le \alpha_i$ since $I^* \in \mathcal{I}_k(\mathcal{P}, \alpha_i)$. Therefore

$$\beta_i - \alpha_i \;>\; \max\{w(e)|e \in I^*\} - \min\{w(e)|e \in I^*\}, \tag{5.2}$$
$$\ge\; \max\{w(e)|e \in I^*\} - \alpha_i. \tag{5.3}$$

Hence we have $\beta_i > \max\{w(e)|e \in I^*\}$. In follows that $\alpha_{i-1} < w(e) < \beta_i(e \in I^*)$, and $I^* \subseteq J_i^{(2-2)}$ from Lemma 5.2. This contradicts Lemma 5.6 since I^* is also an ideal of $\mathcal{P}(J_i^{(2-2)})$. □

Combining the above lemma with the fact that

$$\cup\{\mathcal{I}_k(\mathcal{P}, \alpha_i) - \mathcal{I}_k(\mathcal{P}, \alpha_{i-1})|i = 1, \cdots, l\} = \mathcal{I}_k(\mathcal{P}, \alpha_l) - \mathcal{I}_k(\mathcal{P}, \alpha_0) \tag{5.4}$$
$$= \mathcal{I}_k(\mathcal{P}) - \emptyset \tag{5.5}$$
$$= \mathcal{I}_k(\mathcal{P}), \tag{5.6}$$

we have the following conclusion.
Theorem 5.11: *The set of all critical ideals of $\mathcal{P}$ is $\{J_i^{(2-3)}|i = 1, \cdots, l\}$. Thus, the algorithm* RANGE$(\mathcal{P}, k)$ *enumerates all critical ideals.*

Theorem 5.12: *The algorithm* RANGE$(\mathcal{P}, k)$ *correctly computes a minimum-range ideal of size k of $\mathcal{P}$ in $O(n \log n + m)$ time.*

(Proof) From Theorem 5.11 Step 2 finds a critical interval $[\alpha^*, \beta^*]$ such that $\beta^* - \alpha^*$ is minimum in the all critical interval. And Step 3 identifies a minimum-range ideal. We now analyze the time complexity. In Step 1, defining the minimax-order and the maximin-order requires $O(n \log n + m)$ time from Theorem 3.1. In Step 2, one iteration of (♯1),

(♯2), (♭1) or (♭2) is done in O(1) time. And finding a set D in Step (2-4), i.e., finding the smallest element in maximin-order in the set of all the minimum-weight elements in J can be done in $O(\log |J|)$ time by using an appropriate data structures such as a heap date structure with respect to the maximin-order and weights, respectively. Notice that we build two data structures for the same set J by two criterions. The number of iteration is $O(n)$ and $\max\{|J_i| | i = 1, \cdots, l\} \leq n$. Therefore Step 2 requires $O(n \log n)$. Step 3 require also $O(n \log n)$ since it is similar to carrying out one iteration of Step 2. Therefore, it takes $O(n \log n + m)$ time in total. □

Example: Consider the minimum-range ideal problrm (of size 5) on the poset $\mathcal{P}$ shown in Figure 1. RANGE($\mathcal{P}$, 5) execute as indicated in Table 1. At termination, we have a minimum-range ideal $\{e_5, e_9, e_{10}, e_{13}, e_{16}\}$.

Table 1: Illustrating RANGE($\mathcal{P}$, 5).

Step	S	J	α	β	D
1	E	$\emptyset$			
(2-1)	$\{e_5,e_6,e_7,e_9,e_{10},e_{11},e_{12}, e_{13},e_{14},e_{15},e_{16},e_{17},e_{18}\}$	$\{e_1,e_2,e_3,e_4,e_8\}$		5	
(2-2)	$\{e_9,e_{10},e_{11},e_{12},e_{13},e_{14}, e_{15},e_{16},e_{17},e_{18}\}$	$\{e_1,e_2,e_3,e_4,e_8, e_5,e_6, e_7\}$		5	
(2-3)	$\{e_9,e_{10},e_{11},e_{12},e_{13},e_{14}, e_{15},e_{16},e_{17}\}$	$\{e_1,e_2,e_3,e_4,e_5\}$	**2**	**5**	e_3
(2-4)	$\{e_9,e_{10},e_{11}, e_{13},e_{14},e_{15}, e_{16},e_{17}\}$	$\{e_1,e_2,e_5\}$	2	5	
(2-1)	$\{e_{11}, e_{13},e_{14},e_{15},e_{16},e_{17}\}$	$\{e_1,e_2,e_5,e_9,e_{10}\}$	2	6	
(2-2)	$\{e_{13},e_{14},e_{16},e_{17}\}$	$\{e_1,e_2,e_5,e_9,e_{10}, e_{11},e_{15}\}$	2	6	
(2-3)	$\{e_{13},e_{14},e_{16},e_{17}\}$	$\{e_5,e_9,e_{10},e_{11},e_{15}\}$	**4**	**6**	e_{11},e_{15}
(2-4)	$\{e_{13},e_{14},e_{16}\}$	$\{e_5,e_9,e_{10}\}$	4	6	
(2-1)	$\{e_{14}\}$	$\{e_5,e_9,e_{10},e_{13},e_{16}\}$	4	9	
(2-2)	$\emptyset$	$\{e_5,e_9,e_{10},e_{13},e_{16},e_{14}\}$	4	9	
(2-3)	$\emptyset$	$\{e_5,e_9,e_{10},e_{13},e_{16}\}$	**5**	**9**	e_5,e_{10}
(2-4)	$\emptyset$	$\{e_9,e_{13},e_{16}\}$	5	9	
3	$\{e_{13},e_{14},e_{16},e_{17}\}$	$\{e_5,e_9,e_{10},e_{11},e_{15}\}$	4	6	e_{11},e_{15}

6. A Lower Bound

We shall consider a lower bound for P_{range} and show that RANGE($\mathcal{P}$, k) is optimal. First, we introduce the following problem.

The closest k numbers problem : Given n real numbers and a positive integer k, find k numbers whose range is smallest.

Lemma 6.1: *The closest k numbers problem has an $\Omega(n \log n)$ lower bound.*

(Proof) For N real numbers, which are treated as a multiset and denoted by U, finding a minimum difference among them has an $\Omega(N \log N)$ lower bound (see Chapter 5 of [7]). This problem is called *the closest pair problem.* Let $U_{\lceil \frac{k}{2} \rceil}$ be a multiset which has $\lceil \frac{k}{2} \rceil |U|$ real numbers such that each real number in U contributes to $U_{\lceil \frac{k}{2} \rceil}$ $\lceil \frac{k}{2} \rceil$ times, where $\lceil \frac{k}{2} \rceil$ is the smallest integer larger than or equal to $\frac{k}{2}$. Now, we consider the closest k numbers problem on $U_{\lceil \frac{k}{2} \rceil}$. Then the minimum-range of the closest k numbers problem on $U_{\lceil \frac{k}{2} \rceil}$ is equal to that of the closest pair problem on U since there are $\lceil \frac{k}{2} \rceil$ same numbers in $U_{\lceil \frac{k}{2} \rceil}$. Hence, an arbitrary algorithm for the closest k numbers problem requires at least $N \log N - N\lceil \frac{k}{2} \rceil$ time. Noting that $n = N\lceil \frac{k}{2} \rceil (N = \frac{n}{\lceil \frac{k}{2} \rceil})$ and k is a fixed integer, we get the above statement. □

Lemma 6.2: *The minimum-range ideal problem (of size k) has an $\Omega(n \log n + m)$ lower bound.*

(Proof) For any positive integers n and m satisfying $m \leq (\frac{n}{2})^2$, let us define a bipartite graph $G = (E_1 \cup E_2, A)$ with disjoint element sets E_1 and E_2 such that $|E_1| + |E_2| = n$, $|E_1| \geq |E_2|$, and $|E_1|(|E_2| - 1) < m \leq |E_1||E_2|$, and an arc set $A = \{(e, e')|e' \in E_1, e \in E_2 - \{\hat{e}\}\} \cup \{(\hat{e}, e')|e' \in E_1'\}$ for a subset $E_1' \subset E_1$ with $|E_1'| = m - |E_1|(|E_2| - 1)$ and a special element $\hat{e} \in E_2$ (see Figure 3). Then, consider the minimum-range ideal problem (of size k) on the poset $\mathcal{P}$ represented by $G = (E_1 \cup E_2, A)$ defined above. Notice that E_1 is the set of all the minimal elements of $\mathcal{P}$. If $k \leq |E_1|$, every element in $E_2 - \{\hat{e}\}$ is not able to belong to an ideal of size k. Therefore, this problem is transformable to the closest k numbers problem on E_1 and hence has an $\Omega(|E_1| \log |E_1|)$ lower bound due to Lemma 6.1. From the facts that $|E_1| \geq \frac{n}{2}$ and that it has a trivial $\Omega(m + n)$ lower bound, $\Omega(n \log n + m)$ lower bound follows. We get the same condition if $k > |E_1|$, by reversing all arcs of G. □

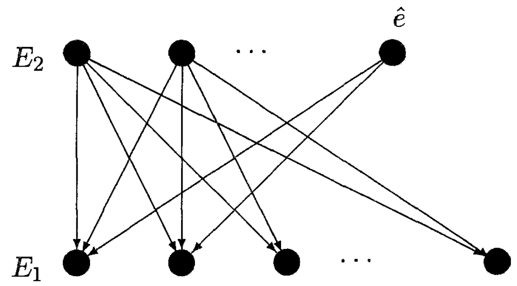


Figure 3: The bipartite graph $G = (E_1 \cup E_2, A)$ in Lemma 6.2.

Combining this lemma with Theorem 5.12, we have the theorem below.

Theorem 6.3: *The algorithm* RANGE($\mathcal{P}, k$) *requires* $\Theta(n \log n + m)$ *time, which is optimal.* □

References

[1] P. M. Camerini, F. Maffioli, S. Martello and P. Toth: Most and least uniform spanning trees. *Discrete Applied Mathematics*, **15** (1986) 181-197.

[2] U. Faigle and W. Kern: Computational complexity of some maximum average weight problems with precedence constraints. *Operations Research*, **42** (1994) 688-693.

[3] H. N. Gabow and R. T. Tarjan: Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, **9** (1988) 411-417.

[4] Z. Galil and B. Schieber: On finding most uniform spanning tree. *Discrete Applied Mathematics*, **20** (1988) 173-175.

[5] N. Katoh and K. Iwano: Efficient algorithms for minimum range cut problems. *Networks*, **24** (1994) 395-407.

[6] S. Martello, W.R. Pulleyblank, P. Toth and D. de Werra: Balanced optimization problems. *Operations Research Letters*, **3** (1984) 275-278.

[7] F. P. Preparata and M. I. Shamos: *Computational Geometry: An Introduction* (Springer-Verlag, New York, 1985).

[8] J. C. Picard and M. Queyranne: Selected applications on minimum cuts in networks. *INFOR*, **20** (1982) 394-422.

Toshio Nemoto
Department of Business and Information
Faculty of Information and Communication
Bunkyo University
1100 Namegaya, Chigasaki 253-8550,JAPAN
E-mail:nemoto@shonan.bunkyo.ac.jp