An Efficient Algorithm for the Minimum-Range Ideal Problem

02401130 *根本 俊男（筑波大学 社会工学研究科）

1. Introduction

Suppose we are given a poset (partially ordered set) $\mathcal{P} = (E, \preceq)$, a real weight $w(e)$ associated with each element $e \in E$ and a positive integer k. A subset of E is called an *ideal* I of $\mathcal{P}$ if $e \preceq e' \in I$ always implies $e \in I$. We consider the problem as follows:

$$\mathrm{P_{range}}: \quad \text{Minimize} \quad \max_{e\in I} w(e) - \min_{e\in I} w(e)$$
$$\text{subject to} \quad I \in \mathcal{I}(\mathcal{P}), |I| = k \qquad (1.1)$$

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. We call this problem the minimum-range ideal problem($\mathrm{P_{range}}$). We hope that this problem serves as a subproblem for leveling resources for large scale scheduling planning.

The optimization problem on the ideal is valuable because many applications in real-life are formalized as the ideal problem. Therefore, various types of this class of problems have been well researched. It is interesting to note that the cardinality-restricted ideal problem is $\mathcal{NP}$-hard if the objective function is linear, but it has a strongly polynomial algorithm if the objective is to minimize the range. To the author's knowledge, no one has ever considered $\mathrm{P_{range}}$.

The min-range problem is also an interesting combinatorial optimization problem. Several researchers have studied a number of min-range problems, including the assignment problem [3], the spanning tree problem [1], and the cut problem [2]. For these problems, a general algorithm has been proposed in [3]. Simply applying the general algorithm in [3], $\mathrm{P_{range}}$ can be solved in $O(mn)$ time, where $n = |E|$ and m is the smallest number of arcs to represent $\mathcal{P}$.

However, we present an $O(n \log n + m)$ algorithm for this problem. This algorithm is different from the algorithms proposed for the above min-range combinatorial problems. 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.

2. The Minimax Ideal Problem

We consider the *minimax ideal problem* and the *maximin ideal problem* which play an important role to solve $\mathrm{P_{range}}$. The former is defined as follows:

$$\mathrm{P_{minimax}}: \min\{\max_{e\in I} w(e) | (1.1)\}.$$

The following algorithm solves $\mathrm{P_{minimax}}$.

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

Step 1: Put $J := \emptyset$.

Step 2: Repeat the following (a) and (b) k times.

(a) $C := \{e | e$ is a minimal element of $\mathcal{P}(E - J)\}$.

(b) If $C = \emptyset$, then stop (there is no ideal of size k of $\mathcal{P}$). Otherwise find a min-weight element $\hat{e}$ in C, and $J := J \cup \{\hat{e}\}$.

Theorem 2.1: MINIMAX($\mathcal{P}, k$) *computes a minimax ideal of* $\mathcal{P}$ *in* $O(n \log n + m)$. □

Similarly, we define the latter($\mathrm{P_{maximin}}$) and construct an algorithm, which is called MAXIMIN($\mathcal{P}$,k).

3. A Naive Algorithm

For real values α and β with $\alpha < \beta$, let $L(\alpha) = \{e | e \in E, w(e) \le \alpha\}$ and $H(\beta) = \{e | e \in E, w(e) > \beta\}$, and for an element $e \in E$, let $F(e) = \{\hat{e} | e \preceq \hat{e}, \hat{e} \in E\}$. Then, a set $E - \cup_{e\in L(\alpha)} F(e) - \cup_{e\in H(\beta)} F(e)$ is an ideal of $\mathcal{P}$. For simplicity, we use $\mathcal{P}(\alpha, \beta]$ as an abbreviation for the subposet induced by the above ideal. Similarly, we abbreviate $\mathcal{P}(E - \cup_{e\in L(\alpha)} F(e))$ to $\mathcal{P}(\alpha, \infty)$, and $\mathcal{P}(E - \cup_{e\in H(\beta)} F(e))$ to $\mathcal{P}(-\infty, \beta]$.

First, we show that there exists the subposet of $\mathcal{P}$ whose min-range ideal is found easily. Given a real value α_i, compute the optimal value of $\mathrm{P_{minimax}}$ on $\mathcal{P}(\alpha_i, \infty)$, denoted by β_{i+1}, if $\mathcal{P}(\alpha_i, \infty)$ has an ideal of size k. Then, identify $\mathcal{P}(\alpha_i, \beta_{i+1}]$.

Lemma 3.1: *For all ideal I of size k of* $\mathcal{P}(\alpha_i, \beta_{i+1}]$, *we have* $\max\{w(e) | e \in I\} = \beta_{i+1}$. □

Combining this lemma and the fact that $\mathrm{P_{range}}$ becomes equivalent to $\mathrm{P_{maximin}}$ if every ideal has the same maximum weight, we have the lemma below.

Lemma 3.2: *Let I_{i+1} be a maximin ideal of* $\mathcal{P}(\alpha_i, \beta_{i+1}]$. *Then I_{i+1} is a min-range ideal of* $\mathcal{P}(\alpha_i, \beta_{i+1}]$. □

Next, we shall expand the discussion above into a naive algorithm. Let α_0 be a sufficiently small value. Then we have $\mathcal{P}(\alpha_0, \infty) = \mathcal{P}$ and we can find the first appropriate value β_1 by applying MINIMAX($\mathcal{P}$,k). Furthermore, by applying MAXIMIN($\mathcal{P}(\alpha_0, \beta_1]$,$k$) a min-range ideal I_1 of $\mathcal{P}(\alpha_0, \beta_1]$ is obtained. Define $\alpha_1 = \min\{w(e) | e \in I_1\}$. After that, repeat the above processes for $\alpha_i (i \ge 1)$ obtained in the previous iteration until there is no ideal of size k of $\mathcal{P}(\alpha_i, \infty)$. We

*Toshio NEMOTO: A Doctoral Candidate studying at Doctoral program in Socio-Economic Planning, University of Tsukuba, Tsukuba, 305 JAPAN.
e-mail: nemoto@shako.sk.tsukuba.ac.jp

prove the fact that there is a min-range ideal of $\mathcal{P}$ in $\{I_i | i = 1, \ldots, q\}$ obtained above. Therefore, it takes $O(n(n \log n + m))$ time to find a min-range ideal. This time is not faster than the general algorithm [3].

4. Improved Implementation

To improve the time of the naive algorithm to $O(n \log n + m)$, we introduce three new ideas.

Firstly, we examine a method to identify $\mathcal{P}(-\infty, \beta)$. Let J' be the set J obtained by repeating Step 2 of MINIMAX$(\mathcal{P}, k)$ while $\min\{w(e) | e \in C\} \leq \beta$ holds after MINIMAX$(\mathcal{P}, k)$ is done, where β is the optimal value of P_{minimax}. Then, we have that $J' = E - \cup_{e \in H(\beta)} F(e)$, i.e.,$\mathcal{P}(J') = \mathcal{P}(-\infty, \beta)$.

Secondly, we propose how to identify $\mathcal{P}(\alpha, \infty)$. In general, no method except for the basic search-method is found. However, by using the information of $\mathcal{P}(-\infty, \beta)$ obtained in the previous step, $E - \cup_{e \in L(\alpha)} F(e)$ can be identified by the following three operations.

$D^1(e)$: Deletion of $F(e)$ for $e \in (E - \cup_{e \in H(\beta)} F(e)) - \tilde{I}$.
$D^2(e)$: Deletion of $F(e)$ for each $e \in \tilde{I}(\alpha)$.
$D^3(e)$: Deletion of $F(e)$ for each remaining $e \in L(\alpha)$.
Here, $\tilde{I}$ is a maximin ideal of $\mathcal{P}(-\infty, \beta]$, $\alpha = \min\{w(e) | e \in \tilde{I}\}$ and $\tilde{I}(\alpha) = \{e | e \in \tilde{I}, w(e) = \alpha\}$. In fact, it is not necessary to identify $\mathcal{P}(\alpha, \infty)$ completely to find a minimax ideal of it. The above operations are carried out as the need arises.

Thirdly, 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, define the *maximin-order of* $\mathcal{P}$ by MAXIMIN$(\mathcal{P}, n)$.

Theorem 4.1: *For an ideal J of $\mathcal{P}$ and an integer k with $0 \leq k \leq |J|$, the subset $I \subseteq J$ consisting of k elements in minimax-order (maximin-order) of $\mathcal{P}$ is a minimax ideal (maximin ideal) of $\mathcal{P}(J)$.* □

Hence, given the minimax-order and the maximin-order of $\mathcal{P}$ as a preprocessor, it is not necessary to arrange the set of all the minimal elements of the underlying subposet to solve P_{minimax} and P_{maximin}.

Now, let us describe an efficient implementation of the naive algorithm.

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

Step 1: Define the minimax-order and the maximin-order of $\mathcal{P}$, and give each element in E the index in the minimax-order. Put $J := \emptyset$, $S := E$, *range* $:= \infty$, $\alpha := -\infty$, $\beta := -\infty$ and $i := 1$.

Step 2: While $S \neq \emptyset$, do the following, (2-1) to (2-5).

(2-1): Repeat the following until $|J| = k$.

(a) If $e_i \in S$ and $w(e_i) \leq \alpha$, then $D^3(e_i)$.
If $e_i \in S$ and $\alpha < w(e_i) \leq \beta$, then $S := S - \{e_i\}$ and $J := J \cup \{e_i\}$.
If $e_i \in S$ and $\beta < w(e_i)$, then $\beta := w(e_i)$, $S := S - \{e_i\}$ and $J := J \cup \{e_i\}$.

(b) $i := i + 1$. If $i > n$ and $|J| < k$, then go to Step 3. If $i > n$ and $|J| = k$, then go to (b$'$).

(2-2): While $w(e_i) \leq \beta$, do the following.

(a$'$) If $e_i \in S$ and $w(e_i) \leq \alpha$, then $D^3(e_i)$.
If $e_i \in S$ and $\alpha < w(e_i)$, then $S := S - \{e_i\}$, $J := J \cup \{e_i\}$, find the biggest $\hat{e} \in J$ in the maximin-order and $D^1(\hat{e})$.

(b$'$) $i := i + 1$. If $i > n$, then call (2-3) and (2-4), and go to Step 3.

(2-3): Find the min-weight element $\tilde{e} \in J$ and $\alpha := w(\tilde{e})$. If *range* $> \beta - \alpha$, then *range* $:= \beta - \alpha$ and $\beta^* := \beta$.

(2-4): While $w(\tilde{e}) = \alpha$, do $D^2(\tilde{e})$ and find the min-weight element $\tilde{e} \in J$.

Step 3: Retrieve a min-range ideal by using β^*.

Theorem 4.2: RANGE$^+(\mathcal{P}, k)$ *correctly computes a min-range ideal of* $\mathcal{P}$ *in* $O(n \log n + m)$ *time.* □

5. Lower Bound

We shall consider a lower bound for P_{range}.

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

Lemma 5.1: Cl(k) *has an* $\Omega(n \log n)$ *lower bound.* □

We prove the fact that Cl(k) is n-transformable to P_{range}. Therefore, we have the following lemma.

Lemma 5.2: P_{range} *has an* $\Omega(n \log n + m)$ *lower bound.* □

Theorem 5.3: *The algorithm* RANGE$^+(\mathcal{P}, k)$ *requires* $\theta(n \log n + m)$ *time, which is optimal.* □

References

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

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

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