数理解析研究所講究録 1373 巻 2004 年 175-183

An Optimal Employment Problem with Multiple-Choice and Partial Recall

玉置光司 (Mitsushi Tamaki)
Department of Business Administration, Aichi University

1 Introduction

We consider an employment problem where a company wants to employ m workers through the coming n periods and a sufficiently large number of persons apply for this employment. These applicants are assumed to be rankable in the order of desirability (1 being the best, 2 the second best and so forth) and appear in random order over these periods. At the end of each period, the company is allowed a partial recall, that is, it chooses any applicant that have arrived in that period. When the company employs an applicant, a loss is incurred depending on the rank of the applicant (we consider loss instead of profit only for ease of description). The applicants not chosen are lost immediately and unavailable later. The problem of the company is to determine, at the end of each period, how many topmost applicants to choose from among those that have arrived in that period based on the full memory of the relative ranks of the applicants that have arrived by that time, in order to minimize the expected total loss.

To make this problem more precise and avoid unnecessary complication, we consider this in the framework of the infinite formulation, i.e., infinite secretary problem as defined and originally studied by Gianini and Samuels(1976) (see also Gianini(1977) and Sec.5 of Samuels(1991)). Let $U_i, i=1,2,\ldots$, denote the arrival time of the i-th best of an infinite countable sequence of rankable applicants. The basic assumption of the infinite formulation is that $U_1, U_2, \cdots$ are independent and uniformly distributed on the unit interval $I=(0,1]$. We introduce a discretization that allows a partial recall by dividing I into n equal subintervals

$$I_k \equiv \left(\frac{k-1}{n}, \frac{k}{n}\right], \qquad k=1,2,\cdots,n.$$

This implies that, at the end of each subinterval, we can choose any applicant that have arrived in that subinterval. We also introduce a loss function $q(i), i=1,2,\ldots$, which denotes a loss for choosing the i-th best. $q(i)$ is naturally assumed to be non-decreasing in i. Since we have to employ m workers, if the number of applicants employed in the first $n-1$ subintervals amounts to k, then we choose exactly the top $m-k$ applicants in the last subinterval.

The loss functions of special interest are as follows:

Example 1: For $\beta>1$,

$$q(i)=\begin{cases} 1, & \text{if } i=1 \\ \dfrac{\beta(\beta+1)\cdots(\beta+i-2)}{(i-1)!}, & \text{if } i\geq 2. \end{cases}$$

It should be noted that $q(i)\equiv i$ for $\beta=2$, so that in this case the loss is just the rank of the applicant, and so the objective of the problem can be interpreted as minimizing the expected total ranks of the applicants chosen. Since $q(i)$ is concave, linear or convex depending on $\beta < 2$, $\beta = 2$ or $\beta > 2$, β can be considered as a parameter that reflects our attitude toward the risk.

Example 2: For some positive integer N,

$$q(i) = \begin{cases} 0, & \text{if } 1 \le i \le N \\ 1, & \text{if } i \ge N+1. \end{cases}$$

The objective for this loss function can be interpreted as minimizing the expected number of the chosen applicants whose ranks exceed N. N also reflects our attitude toward the risk. That is, we are easy-going if N is large, but severe if N is small.

In Section 2, we derive the optimality equation of the problem and obtain the structure of the optimal policy. In Section 3, we derive an important formula which makes it easy to calculate the optimal values and the related decision numbers recursively.

When m is a multiple of n, say $m = cn$, there exists an easily practicable employment policy called a deterministic rule, which chooses exactly the top c applicants in each subinterval(independent of the previous applicants that appeared in the preceeding subintervals). Numerical results show that this policy works well when m is large and n is small.

2 Formulation and optimal policy

Suppose that we have to choose k more applicants in the remaining r subintervals (in other words, we have already chosen $m-k$ applicants in the first $n-r$ subintervals). Then the next decision epoch takes place at the end of subinterval I_{n-r+1} after having observed the infinite ordered sequence $(i_1, i_2, \cdots)$, $i_1 < i_2 < \cdots$, where $i_j, j \ge 1$, represents the relative rank of j-th best in I_{n-r+1} among all the applicants that have arrived by time k/n. At most top k can be candidates for choice, so that the finite sequence $(i_1, \cdots, i_k)$ is a sufficient information for our decision. We thus denote the state of this decision epoch by $(r, k; i_1, \cdots, i_k)$. Let $v_k^r(i_1, \cdots, i_k)$ denote the minimum expected loss starting from state $(r, k; i_1, \cdots, i_k)$. We first introduce the joint probability mass function and the loss function.

$p^r(i_1, \cdots, i_k)$: The joint probability mass function that the ranks of best, 2nd best, $\cdots$, k-th best in I_{n-r+1} among all the applicants that have arrived by time k/n are $i_1, i_2, \cdots, i_k$ respectively, where $i_1 < i_2 < \cdots < i_k$.

$R_j(t)$: The expected loss incurred by choosing an applicant at time t whose rank relative to all its predecessors is j $(1 \le j,\ 0 < t \le 1)$.

These quantities are given as follows.

Lemma 1.

For $1 \le r < n$ and $i_1 < i_2 < \cdots < i_k$,

$$p^r(i_1, \cdots, i_k) = \left(\frac{1}{n-r+1}\right)^k \left(1 - \frac{1}{n-r+1}\right)^{i_k - k}. \tag{1}$$

Lemma 2.

For $1 \le j$ and $0 < t \le 1$,

$$R_j(t) = \sum_{i=j}^{\infty} q(i) \binom{i-1}{j-1} t^j (1-t)^{i-j}. \tag{2}$$

If $q(i)$ is increasing in i, then

(i) $R_j(t)$ is decreasing in t.

(ii) $R_{j+1}(t) > R_j(t)$.

(iii) $\lim_{j\to\infty} R_j(t) = \infty$ for all $t \in (0,1]$.

Proof. See Mucci(1973).

Remark: In some cases, closed form of $R_j(t)$ can be obtained;

Example 1:

$$R_j(t) = \frac{q(j)}{t^{\beta-1}}, \qquad j \geq 1.$$

Example 2:

$$R_j(t) = \begin{cases} 1 - \sum_{i=0}^{N-j} \binom{i+j-1}{i} t^j (1-t)^i, & \text{if } 1 \leq j \leq N \\ 1, & \text{if } j \geq N+1, \end{cases}$$

We now have the following optimality equations

$$v_k^r(i_1, \cdots, i_k) = \min_{0\leq j\leq k} \left\{ \sum_{t=1}^{j} R_{i_t}\left(\frac{n-r+1}{n}\right) + v_{k-j}^{r-1} \right\}, \tag{3}$$

where

$$v_k^r = \sum \sum_{i_1<i_2<\cdots<i_k} \cdots \sum v_k^r(i_1, i_2, \cdots, i_k) p^r(i_i, i_2, \cdots, i_k), \quad 1 \leq k \leq m,\ 2 \leq r < n \tag{4}$$

with the boundary condition

$$v_k^1 = \sum_{j=1}^{k} R_j\left(\frac{1}{n}\right). \tag{5}$$

Therefore the expected total loss is given by

$$V_m^n = \min_{0\leq j\leq m} \left\{ \sum_{i=1}^{j} R_i\left(\frac{1}{n}\right) + v_{m-j}^{n-1} \right\}. \tag{6}$$

v_k^1 is increasing convex in k, because, from Lemma 2,

$$\begin{aligned} v_{k+1}^1 - v_k^1 &= R_{k+1}\left(\tfrac{1}{n}\right) > 0 \\ v_{k+1}^1 + v_{k-1}^1 - 2v_k^1 &= R_{k+1}\left(\tfrac{1}{n}\right) - R_k\left(\tfrac{1}{n}\right) > 0. \end{aligned} \tag{6'}$$

As the following lemma shows, v_k^r in effect inherits this property for all r.

Lemma 3

v_k^r is increasing convex in k for each r.

Proof. See the Appendix.

Now define, for $0 \le j \le k$,

$$A_j^r(i_1, \cdots, i_j) = \sum_{t=1}^{j} R_{i_t}\left(\frac{n-r+1}{n}\right) + v_{k-j}^{r-1}. \tag{7}$$

Then we have

Lemma 4

For fixed r and k, and a given sequence $(i_1, \cdots, i_k)$, $-A_j^r(i_1, \cdots, i_j)$ is unimodal with respect to j.

Proof. Let, for $1 \le j \le k$

$$B_j^r(i_1, \cdots, i_j) \equiv A_j^r(i_1, \cdots, i_j) - A_{j-1}^r(i_1, \cdots, i_{j-1}).$$

Then, it suffices to show that $B_j^r(i_1, \cdots, i_j)$ is increasing in j. From (7), we can write

$$B_j^r(i_1, \cdots, i_j) = R_{i_j}\left(\frac{n-r+1}{n}\right) - (v_{k+1-j}^{r-1} - v_{k-j}^{r-1}). \tag{8}$$

Since $R_{i_j}\left(\frac{n-r+1}{n}\right)$ is increasing in j from Lemma 2 (ii), whereas $v_{k+1-j}^{r-1} - v_{k-j}^{r-1}$ is non-increasing in j from Lemma 3, the result follows.

Let

$$c_k^r(i_1, \cdots, i_k) = \max\{1 \le j \le k : B_j^r(i_1, \cdots, i_j) \le 0\}, \tag{9}$$

with $\max\{\phi\} = 0$(tjis convention is used throughout this paper). Then, from Lemma 4, we immediately have

Lemma 5

In state $(r, k; i_1, \cdots, i_k)$, it is optimal to choose exactly the top $c_k^r(i_1, \cdots, i_k)$ applicants at the end of I_{n-r+1}.

Remark: When $B_j^r(i_1, \cdots, i_j) = 0$ for some j, i.e., $A_j^r(i_1, \cdots, i_j)$ attains its minimum at $j-1$ and j, we assume to choose exactly the top j applicants.

Define

$$\phi_j^r(k) \equiv v_{k+1-j}^{r-1} - v_{k-j}^{r-1}, \quad 1 \le j \le k. \tag{10}$$

It is easy to see from (8) and (9) that, in state $(r, k; i_1, \cdots, i_k)$, we never choose j-th best applicant if $R_j\left(\frac{n-r+1}{n}\right) > \phi_j^r(k)$, while we possibly choose j-th best if $R_j\left(\frac{n-r+1}{n}\right) \le \phi_j^r(k)$ and the value of i_j is sufficiently small. More specifically if we define, for $1 \le j \le k$,

$$i_j^r(k) = \max\left\{j \le i : R_i\left(\frac{n-r+1}{n}\right) \le \phi_j^r(k)\right\}. \tag{11}$$

We can give, from Lemma 5, another way of describing the optimal choice as follows.

Lemma 6

There exists a sequence of decision numbers $\{i_j^r(k)\}_{j=1}^k$, such that the optimal decision in state $(r,k;i_1,\cdots,i_k)$ chooses j-th best applicant provided $i_j \le i_j^r(k)$, irrespective of the values of $i_1,\cdots,i_{j-1},i_{j+1},\cdots,i_k, 1\le j\le k$.

The sequence $\{i_j^r(k)\}_{j=1}^k$ satisfies the following monotonicity properties.

Lemma 7

(i) $i_j^r(k) \ge i_{j+1}^r(k)$.
(ii) $i_j^r(k) \le i_j^r(k+1)$.
(iii) $i_j^r(k) \ge i_{j+1}^r(k+1)$. In particular, if $i_{j+1}^r(k+1) \ne 0$, then $i_j^r(k) = i_{j+1}^r(k+1)$.

Proof.

(i) and (ii) are immediate since $R_{i_j}(\cdot)$ is increasing in j, whereas $\phi_j^{r-1}(k)$ is non-increasing in j.
(iii) is immediate since $\phi_k^{r-1}(j)$ depends on k and j only through $k-j$.

It should be noted that, if we define

$$K = K(r,k) = \max\{j : i_j^r(k) \ne 0\}, \tag{12}$$

K denotes the maximum possible number of applicants that can be chosen in subinterval I_{n-r+1} when k more applicants must be chosen.

We now have

Lemma 8

If it is optimal to choose exactly the top c applicants in state $(r,k;i_1,\cdots,i_k)$, then it is optimal to choose exactly the top c or top $(c+1)$ applicants in state $(r,k+1;i_1,\cdots,i_k,i_{k+1})$ for any i_{k+1}. Or equivalently

$$c_{k+1}^r(i_1,\cdots,i_{k+1}) = c_k^r(i_1,\cdots,i_k) \quad \text{or} \quad c_k^r(i_1,\cdots,i_k)+1.$$

Proof.

Write c instead of $c_k^r(i_1,\cdots,i_k)$ for simplicity. First assume $c<k$. Then, from the definition of c, we have

$$i_{c+1} > i_{c+1}^r(k). \tag{13}$$

Since $i_{c+1} < i_{c+2}$, we have from (13) and Lemma 7 (iii)

$$i_{c+2}^r(k+1) \le i_{c+1}^r(k) < i_{c+1} < i_{c+2},$$

which implies that it is not optimal to choose $(c+2)$-nd best in state $(r,k+1;i_1,\cdots,i_{k+1})$. On the other hand, from the definition of c and Lemma 7 (ii)

$$i_c \le i_c^r(k) \le i_c^r(k+1), \tag{14}$$

which implies that the optimal policy chooses c-th best applicant in state $(r,k+1;i_1,\cdots,i_{k+1})$. when $c=k$, the result is trivial since (14) still holds.

Appendix

Proof of Lemma 3

(a) v_k^r **is increasing in** k.

Though increasing nature of v_k^r with respect to k is intuitively clear, we formally can explain this as follows: Let f_k and f_{k+1} denote the optimal policies that can be used when D-M starts from states (r, k) and $(r, k+1)$ respectively. Then, if we denote by u_k^r the expected loss incurred when D-M uses f_{k+1}, starting from (r, k), (he stops choosing as soon as the number of applicants chosen reaches k), then obviously $u_k^r < v_{k+1}^r$. On the other hand, $v_k^r \le u_k^r$ by definition. Thus the proof is complete.

(b) **convexity of** v_k^r

It suffices to show that, for any realization $(i_1, \cdots, i_k, i_{k+1})$,

$$v_{k+1}^r(i_1, \cdots, i_{k+1}) + v_{k-1}^r(i_1, \cdots, i_{k-1}) - 2v_k^r(i_1, \cdots, i_k) \ge 0. \tag{A.1}$$

We show (A.1) by induction on r. For $r = 1$, (A.1) obviously holds from (6').

Assume that (A.1) holds for $r-1$. Then Lemmas 4-8 in fact hold because these lemmas are based on this induction hypothesis. Write now simply c, c' and c'' instead of $c_k^r(i_1, \cdots, i_k), c_{k-1}^r(i_1, \cdots, i_{k-1})$ and $c_{k+1}^r(i_1, \cdots, i_{k+1})$ respectively. Then we have from (3)

$$\begin{aligned} & v_{k+1}^r(i_1, \cdots, i_{k+1}) + v_{k-1}^r(i_1, \cdots, i_{k-1}) - 2v_k^r(i_1, \cdots, i_k) \\ &= \sum_{t=1}^{c'} R_{i_t}\left(\frac{n-r+1}{n}\right) + \sum_{t=1}^{c''} R_{i_t}\left(\frac{n-r+1}{n}\right) - 2\sum_{t=1}^{c} R_{i_t}\left(\frac{n-r+1}{n}\right) \\ &+ \left(v_{k-1-c'}^{r-1} + v_{k+1-c''}^{r-1} - 2v_{k-c}^{r-1}\right). \end{aligned} \tag{A.2}$$

Let A represent the right hand side of (A.2). When $1 \le c \le k-1$, we can show $A \ge 0$ by distinguishing four possible cases depending on the values of c' and c'' from Lemma 8.

Case 1 ($c' = c - 1$ and $c'' = c + 1$):

We have, from Lemma 2 (ii),

$$A = R_{i_{c+1}}\left(\frac{n-r+1}{n}\right) - R_{i_c}\left(\frac{n-r+1}{n}\right) + 0 > 0.$$

Case 2 ($c' = c$ and $c'' = c$):

We have from the induction hypothesis,

$$A = 0 + (v_{k+1-c}^{r-1} + v_{k-1-c}^{r-1} - 2v_{k-c}^{r-1}) \ge 0.$$

Case 3 ($c' = c - 1$ and $c'' = c$):

We have $i_c \le i_c^r(k)$ from the definition of c and $R_{i_c^r(k)}\left(\frac{n-r+1}{n}\right) \le \phi_c^r(k)$ from the definition of $i_c^r(k)$. Thus

$$R_{i_c}\left(\frac{n-r+1}{n}\right) \le R_{i_c^r(k)}\left(\frac{n-r+1}{n}\right) \le \phi_c^r(k),$$

which from (10) implies

$$A \;=\; -R_{i_c}\left(\frac{n-r+1}{n}\right) + (v_{k+1-c}^{r-1} - v_{k-c}^{r-1}) \ge 0.$$

Case 4 ($c' = c$ and $c'' = c+1$):
We have $R_{c+1}\left(\frac{n-r+1}{n}\right) > \phi^r_{c+1}(k)$ from the definition of c, and $R_{i_{c+1}}\left(\frac{n-r+1}{n}\right) \geq R_{c+1}\left(\frac{n-r+1}{n}\right)$ due to $i_{c+1} \geq c+1$. Therefore, from (10)

$$A = R_{i_{c+1}}\left(\frac{n-r+1}{n}\right) - (v^{r-1}_{k-c} - v^{r-1}_{k-c-1}) > 0.$$

For $c = 0$ ($c = k$), two cases can be distinguished; (i) $c' = 0, c'' = 1$ ($c' = k-1, c'' = k+1$) and (ii) $c' = 0, c'' = 0$ ($c' = k-1, c'' = k$). The proof is omitted since we can prove similarly.

3 Algorithm for calculating $\{i^r_j(k)\}$ and v^r_j

Let $p^r_j(i)$ be the probability that the rank of j-th best in I_{n-r+1} relative to all its predecessors is i.
Then

$$p^r_j(i) = \binom{i-1}{j-1}\left(\frac{1}{n-r+1}\right)^j\left(1-\frac{1}{n-r+1}\right)^{i-j}, \qquad i \geq j.$$

Algorithm

(i) Initialize $\quad v^1_i = \sum_{j=1}^{i} R_j\left(\frac{1}{n}\right), \quad 1 \leq i \leq m.$

(ii) Assume $\{v^{r-1}_i\}_{i=1}^m$ are given. Also assume that we are in state (r, k).

First calculate

$$\varphi^r_i \equiv v^{r-1}_i - v^{r-1}_{i-1}, \qquad 1 \leq i \leq m,$$

and define, for fixed r and k,

$$\phi^r_j(k) = \varphi^r_{k+1-j}, \qquad 1 \leq j \leq k,$$

and

$$K = K(k,r) = \max\left\{1 \leq j \leq k : R_j\left(\frac{n-r+1}{n}\right) \leq \phi^r_j(k)\right\},$$

with $\max\{\phi\} = 0$.

Then the decision number is calculated as

$$i^r_j(k) = \max\left\{i \geq j : R_i\left(\frac{n-r+1}{n}\right) \leq \phi^r_j(k)\right\}, \quad 1 \leq j \leq K,$$

and the probability that exactly the top j applicants are chosen is expressed as

$$q_j = \begin{cases} 1 - Q_1, & \text{if } j = 0 \\ Q_j - Q_{j+1}, & \text{if } 1 \leq j \leq K-1 \\ Q_K, & \text{if } j = K \end{cases}$$

where

$$Q_j = \sum_{i=j}^{i_j^r(k)} p_j^r(i), \qquad 1 \le j \le K.$$

Finally we obtain, for $K \ge 1$,

$$v_k^r = \sum_{j=1}^{K} \left(\sum_{i=j}^{i_j^r(k)} R_i \left(\frac{n-r+1}{n} \right) p_j^r(i) \right) + \sum_{j=0}^{K} v_{k-j}^{r-1} q_j.$$

When $K = 0, v_k^r = v_k^{r-1}$.

We present some numerical results of Example 1(Example 2) in Table 1(Table 2).

Table 1

$m\backslash n$		5	10	20
20	β=1.5	71.999 (1.0191)	73.805 (1.0712)	75.478 (1.1850)
	β=2	238.35 (1.0489)	253.54 (1.1833)	267.99 (1.4926)
	β=3	2195.4 (1.1387)	2595.0 (1.5414)	2979.7 (2.6849)
	β=4	17269 (1.2667)	23049 (2.1693)	28768 (5.5617)
	β=5	1.2222×10^5 (1.4319)	1.8453×10^5 (3.2516)	2.5234×10^5 (12.6815)
60	β=1.5	357.88 (1.0071)	361.14 (1.0272)	364.33 (1.0741)
	β=2	1914.5 (1.0186)	1960.1 (1.0714)	2005.2 (1.1969)
	β=3	43151 (1.0544)	46174 (1.2128)	49141 (1.6280)
	β=4	7.7062×10^5 (1.1071)	8.7444×10^5 (1.4409)	9.7975×10^5 (2.4496)
	β=5	1.1602×10^7 (1.1765)	1.4128×10^7 (1.7837)	1.6894×10^7 (3.9779)
100	β=1.5	762.95 (1.0043)	767.19 (1.0168)	771.38 (1.0463)
	β=2	5190.6 (1.0114)	5266.6 (1.0443)	5342.3 (1.1231)
	β=3	1.8620×10^5 (1.0339)	1.9427×10^5 (1.1324)	2.0226×10^5 (1.3844)
	β=4	5.1868×10^6 (1.0670)	5.6256×10^6 (1.2710)	6.0699×10^6 (1.8452)
	β=5	1.1956×10^8 (1.1110)	1.3611×10^8 (1.4709)	1.5380×10^8 (2.6217)

The values of V_m^n for given triplet (m, n, β). The value in the parenthsis is the ratio of $\bar{V}_m^n / V_m^n$, where $\bar{V}_m^n$ is the value by the corresponding deterministic rule.

Table 2

$m\backslash n$		5	10	20
20	N=10	10.0672 (1.0132)	10.1670 (1.0670)	10.2706 (1.1660)
	N=20	2.7641 (1.2630)	3.2272 (1.5906)	3.5282 (2.0321)
	N=30	0.4335 (2.0604)	0.6888 (3.2822)	0.8921 (4.8122)
60	N=30	30.0016 (1.0007)	30.0127 (1.0116)	30.0340 (1.0524)
	N=50	11.1748 (1.0796)	11.7461 (1.2330)	12.1752 (1.4733)
	N=60	4.8185 (1.2733)	5.6392 (1.6211)	6.1886 (2.1170)
	N=70	1.6007 (1.6817)	2.2454 (2.4386)	2.7095 (3.4732)
	N=80	0.4065 (2.5423)	0.7420 (4.2219)	1.0061 (6.6233)
	N=90	0.0845 (4.1788)	0.2089 (8.2448)	0.3348 (13.9215)
100	N=50	50.0000 (1.0000)	50.0011 (1.0028)	50.0046 (1.0214)
	N=90	12.2822 (1.1117)	13.1478 (1.3024)	13.7705 (1.5955)
	N=100	6.2286 (1.2754)	7.2923 (1.6275)	8.0092 (2.1354)
	N=110	2.6594 (1.5730)	3.5736 (2.2166)	4.2170 (3.1138)
	N=120	0.9442 (2.1183)	1.5405 (3.3124)	1.9894 (5.0040)
	N=130	0.2862 (2.7960)	0.5879 (5.4103)	0.8559 (8.7161)
	N=140	0.0732 (4.7934)	0.2020 (9.5177)	0.3305 (16.7344)

The values of V_m^n for given triplet (m, n, N). The value in the parenthsis is the ratio of $\bar{V}_m^n/V_m^n$, where $\bar{V}_m^n$ is the value by the corresponding deterministic rule.

参考文献

[1] Gianini, J., "The infinite secretary problem as the limit of the finite problem", *Ann. Probab.***5** , p636-644. (1977)

[2] Gianini, J., and Samuels, S. M. "The infinite secretary problem", *Ann. Probab.* **4**, p418-432. (1976)

[3] Mucci, A. G., "On a class of secretary problems", *Ann Probab.* **1**, p417-427. (1973 b)

[4] Samuels, S. M., "Secretary Problem", in : Ghosh. B. K., Sen. P. K., eds, Handbook of sequential analisis, Marcel Decker, Yew York, p381-405. (1991)