满意化秘书问题：当闭式解遇上模拟退火

Roberto Brera[a]，Feng Fu[a,b,*]

[a]达特茅斯学院数学系，美国新罕布什尔州汉诺威，邮编 *03755*[b]达特茅斯盖塞尔医学院生物医学数据科学系，美国新罕布什尔州黎巴嫩，邮编 *03756*

arXiv:2302.03220v1 [physics.soc-ph] 7 Feb 2023

摘要

秘书问题及其各种扩展一直受到广泛研究，为最优停止理论提供了有益见解。原始解法是设定一个停止阈值，从规模为n 的候选人池中立即拒绝r 名候选人，然后接受此后面试的第一位优于先前被拒绝的r 名候选人的候选人。现实中，人们常常会在求职者中划定一条界线，以区分界线之上和之下的候选人。在此，我们探讨这种满意化秘书问题：即使选择次优候选人（找到排名前d 的任意一位候选人），也视为招聘成功。我们采用多重停止准则$(r_1, r2, \cdots, r_d)$，当先前的选择标准未能选出候选人时，依次降低期望。我们计算了获得排名前d 候选人的概率，其闭式解与计算机模拟结果高度吻合。寻找最优停止策略的穷举搜索需要处理庞大的参数空间，因此计算时间很快就会变得极其漫长。鉴于此，我们采用模拟退火这一有效的计算方法来寻找最优r_i 值。尽管所需计算时间显著少于穷举搜索，该方法的表现仍相当不错。本研究揭示了如何结合自适应搜索策略与计算方法，最大限度地提高获得满意（次优）结果的可能性。

关键词： 最优停止、优化、自适应决策、应用概率

1. 引言

秘书问题提出了一个概念上简单却在数学上十分优雅的模型，用于研究最优决策策略，从而在一系列现实场景中最大限度地提高找到最佳选择（或匹配对象）的概率，例如找到最理想的职位候选人或最佳伴侣。自这一如今广为人知的问题首次发表于 1960 年 2 月的《科学美国人》以来，关于其求解方法及各种变体的研究已取得长足进展 [18, 17, 39]。在过去几十年中，围绕秘书问题开展的系统性研究极大地激发了众多领域的兴趣，并产生了广泛影响，为在各种约束条件下寻找“合适的人选”提供实践指导 [19, 15, 45, 1, 37, 41, 29, 7, 32, 46, 9, 16, 13, 6]。此外，秘书问题还广泛应用于在线交易 [27], 在线拍卖 [24], 和客户行为 [48]。

1961 年，Lindley 发表了标准问题的首个解 [30]。此后，秘书问题以多种方式得到扩展和推广，其中包括面试成本 [4, 31], 有限记忆长度 [38, 44], 就业不确定性 [43], 观测误差 [42], 以及重复秘书问题 [22]。研究者还从不同的数学视角研究了该问题，例如线性规划 [10],次模函数 [5],随机游走 [25]以及图论方法 [28]。

更具体地说，一类重要的扩展允许多重选择 [34, 36], , 或者一般的d 重选择 [21]。Chow研究了一种选择模型，旨在最小化预期入选候选人的排名，使其达到 ≈ 3.87 [14], , 并且

*通讯作者地址：美国新罕布什尔州汉诺威市，达特茅斯学院数学系，凯梅尼楼 6188 室，主街 27 号，邮编 03755。电话： +1 (603) 646 2293电子邮箱：roberto.brera.24@dartmouth.edu (Roberto Brera), fufeng@gmail.com (Feng Fu)

Smith详细研究了不确定就业的情形 [43], : 每位候选人仅以固定概率p接受录用邀请。此外, Buchbinder、Jain 和 Singh 研究了J选择K佳秘书问题 [10], : 决策者可以选择J 位候选人，并需要从排名前K 位的候选人中选出一位，以取得成功。Albers 还探讨了其他多重选择变体 [2] （决策者必须最大化给定数量选择的定性和）; Glasser [21], 则计算了在有d 次选择机会时，选中全部d 位最佳候选人的概率。Smith 放宽了不同的标准，允许随时从最近$m \geq 1$ 位接受面试的候选人中进行选择，同时保留其他所有标准规则 [44]。Freeman [18] 和 Samuels [40]也研究了这些问题变体及其他若干变体。

在先前研究的基础上，本文探讨秘书问题的一种变体。在原始问题中，我们需要从n 位候选人中选出最佳候选人，每位候选人只能面试一次，并且必须立即接受或拒绝。我们为这个问题的变体额外设定一个输入——容忍度d；若从n 位候选人中选出排名前d 的任意一位，即为成功。

为指导我们研究广义秘书问题的方法，我们首先考察标准问题的解法，该问题也称为最优停止问题。为了最大限度地提高选中最佳候选人的可能性，我们采用一种策略：自动拒绝前$r-1$ 位候选人（以获取候选人群体的样本），然后接受第一位胜过所有已拒绝候选人的候选人。我们可以计算出使从候选人群体中选出最佳候选人的成功概率达到最大的r 值。

对于r的某个特定取值，该概率为:

$$\begin{aligned}
P(r) &= \sum_{i=r}^{n} P(i \text{ is the best candidate}) \cdot P(\text{success at selecting } i \mid i \text{ is the best candidate}) \\
&= \sum_{i=r}^{n} \frac{1}{n} \cdot P(\text{best candidate before } i \text{ lies in the } r-1 \text{ group}) \\
&= \frac{1}{n} \sum_{i=r}^{n} \frac{r-1}{i-1} \\
&= \frac{r-1}{n} \sum_{i=r}^{n} \frac{1}{i-1}.
\end{aligned}$$

当$n \to \infty$时，我们可以用积分近似求和，其中$x = \lim_{n\to\infty} r^{-1}n$ ，$t = \lim_{n\to\infty} i^{-1}n$ ，且$dt = \lim_{n\to\infty} {}^{1}n$。然后，我们可以对$x$ 求导，以求得使$x \approx r^{-1}n$ 处$P(x)$ 达到最大值的近似值:

$$P(x) = x \int_{x}^{1} \frac{1}{t} dt = x \ln(x)$$

$$\frac{d}{dx} P(x) = \ln(x) + 1 = 0.$$

因此，当n 较大时，最优选择策略的成功概率约为e^{-1} ，并且在$r \approx \left\lfloor \frac{n}{e} \right\rfloor + 1$.时达到。

I与成功标准严格（仅找到最佳候选人）的标准情形不同，只要找到排名前d 的候选人之一，求职通常就会被视为成功。本研究聚焦于这个“满足即可”的秘书问题。我们提出一种具有多个停止标准的自适应搜索策略(r_1,r_2, $\cdots$,r_d)。对于每个区间 $[r_s, r_{s+1})$，面试官（决策者）会逐步降低预期，继而尝试找到相对于先前r_s 名被拒候选人而言排名第s的候选人；如果在先前整个区间 $[r_{s-1}, r_s)$的搜索中没有选中任何人，则继续如此。对于给定的参数选择，我们能够以闭式推导成功概率，即找到排名前d 的可接受候选人之一；参数包括r_i的取值、$i = 1, \cdots$和d。

下文将介绍本文的组织结构。第2节阐述满意化秘书问题的模型细节。我们首先介绍简单情形$d = 2$ ，由此为推导

一般情形$d \geq 2$的闭式解提供启发式方法。我们还将解析结果与计算机模拟进行比较。第3节介绍使用有效的模拟退火计算方法优化多重停止准则$(r_1, \cdots, r_d)$的具体结果，并将其与暴力搜索方法进行比较。最后，我们在第4节总结并讨论这项研究。

2. 模型与方法

2.1. 面向前 d 名可接受候选人的满意化秘书问题

当我们放宽成功标准，允许从一组d 名同等可接受的候选人中任选一人时，放宽后的选择标准会提高成功概率。因此，我们采用一种策略：自动拒绝前$r_1 - 1$ 名候选人；随后，对于一组停止点$(r_1, r_2, ..., r_d)$，在每个r_i与$r_{i+1} - 1$（含）之间，我们愿意接受当前已拒绝候选者池中排名前i 的候选人之一。

本节将说明，使用这种策略和一组停止点$(r_1, r_2, ..., r_d)$，从总共n 名候选人中选出前d 名候选人之一的概率为

$$P(r_1, r_2, ..., r_d) = \sum_{s=1}^{d} \prod_{i=1}^{s-1} \frac{r_i - i}{r_s - i} \cdot P(s, r_s, r_{s+1}),$$

其中

$$P(s, r, r') = d! \prod_{l=0}^{d-1} \frac{1}{n-l} \left(\sum_{k=0}^{s-1} \binom{r-1}{k} \sum_{i=r}^{r'-1} \binom{n-i}{d-k-1} \prod_{j=k+1}^{s} \frac{r-j}{i-j} + \sum_{k=s}^{d-1} \binom{r-1}{k} \sum_{i=1}^{d-k} \binom{r'-r}{i} \cdot \binom{n-r'+1}{d-k-i} \cdot S_s(k+i, i) \right)$$

$$S_s(k, m) = 1 - \prod_{i=0}^{s-1} \frac{k-m-i}{k-i}.$$

出于启发式考虑，本节首先考察$d = 2$这一简化情形。

2.2. 两位可选候选人的情形：概率计算

作为推广的第一步，我们考虑成功标准较宽松的第一种情形：与标准情形遵循相同规则，但只要求从候选人池中选出两位最佳候选人之一。为了最大化成功概率，我们自然会将标准策略扩展为两种策略；它们的概率分布将为第三种自适应搜索策略的实施提供依据，而这种策略有望带来更高的成功概率。

2.2.1. 第一种策略：停止于迄今面试过的最佳候选人

乍看之下，我们或许会想采用与标准情形完全相同的策略，并利用较宽松的成功标准：

策略 1： 拒绝前 $r - 1$ 位候选人（$r \geq 2$），然后选出第一位优于被拒绝候选人群体的候选人。

将“成功”定义为选中最佳或第二佳候选人，并假设这两位候选人分别位于n_1 和n_2, 的位置，那么成功可能发生的子情形只有三种：

$$\begin{aligned} &\text{Subcase 1:} \quad r \leq n_1 < n_2 \leq n \\ &\text{Subcase 2:} \quad r \leq n_2 < n_1 \leq n \\ &\text{Subcase 3:} \quad n_2 < r \leq n_1 \leq n \end{aligned}$$

我们可以通过以下方式计算$P($成功 $\cap$ 子情况$1)$：先考虑两位相关候选人分别位于固定位置n_1 和n_2, 时的成功概率，再让n_1 和n_2 遍历**子情况 1**中的所有可能位置：

$$
\begin{aligned}
P(\text{Success} \cap \text{Subcase 1}) &= \sum_{n_1=r}^{n-1} \sum_{n_2=n_1+1}^{n} P(\text{Success} \cap \text{Best two candidates at } n_1 \text{ and } n_2) \\
&= \sum_{n_1=r}^{n-1} \sum_{n_2=n_1+1}^{n} P(\text{Best two candidates at } n_1 \text{ and } n_2) \\
&\qquad \cdot P(\text{Success} \mid \text{Best two candidates at } n_1 \text{ and } n_2) \\
&= \sum_{n_1=r}^{n-1} (n-n_1) \cdot \frac{1}{n} \cdot \frac{1}{n-1} \cdot P(\text{Best candidate before } n_1 \text{ lies in the } r-1 \text{ group}) \\
&= \frac{1}{n} \cdot \frac{1}{n-1} \sum_{n_1=r}^{n-1} (n-n_1) \cdot \frac{r-1}{n_1-1}.
\end{aligned}
$$

通过类似的推理，我们可以得到P(成功 $\cap$ 子情况 2)的相同表达式。此外，由于r 和n_1 之间没有候选人能够优于n_2,，因此

$$
\begin{aligned}
P(\text{Success} \cap \text{Subcase 3}) &= P(\text{Subcase 3}) \cdot P(\text{Success} \mid \text{Subcase 3}) \\
&= (r-1) \cdot (n-r+1) \cdot \frac{1}{n} \cdot \frac{1}{n-1}.
\end{aligned}
$$

由于只有在这三种子情形中才可能成功，因此**策略 1** 的总成功概率为

$$
\begin{aligned}
P(r) &= P(\text{Success} \cap \text{Subcase 1}) + P(\text{Success} \cap \text{Subcase 2}) + P(\text{Success} \cap \text{Subcase 3}) \\
&= \left(\frac{r-1}{n(n-1)}\right)\left((n-r+1) + 2\sum_{i=r}^{n-1} \frac{n-i}{i-1}\right) \\
&= \left(\frac{r-1}{n}\right)\left(2\sum_{i=r}^{n} \frac{1}{i-1} - \left(\frac{n-r+1}{n-1}\right)\right) \\
&\approx 2 \cdot \left(\frac{r-1}{n}\right) \sum_{i=r}^{n} \frac{1}{i-1}.
\end{aligned}
$$

请注意，这一成功概率大致是第一节严格情形中计算出的成功概率的两倍。

2.2.2. 第二种策略（满意化）：当目前面试过的候选人中出现最佳或次佳候选人时停止

或者，我们也可以采用以下策略：

策略 2： 拒绝前 $r-1$ 位候选人（$r \geq 3$），然后选择第一个相对于他们而言是最佳或次佳的候选人。

使用这一策略时，我们还需要考虑另一种可能成功的情况：

$$
\text{Subcase 4}: \quad n_1 < r \leq n_2 \leq n
$$

P(成功 $\cap$ 子情况 1) 可采用与**策略 1**类似的方法计算：

$$
\begin{aligned}
P(\text{Success} \cap \text{Subcase 1}) &= \sum_{n_1=r}^{n-1} \sum_{n_2=n_1+1}^{n} P(\text{Best two candidates at } n_1 \text{ and } n_2) \\
&\qquad \cdot P(\text{Success} \mid \text{Best two candidates at } n_1 \text{ and } n_2) \\
&= \sum_{n_1=r}^{n-1} (n-n_1) \cdot \frac{1}{n} \cdot \frac{1}{n-1} \cdot P(\text{Best two candidates before } n_1 \text{ lie in the } r-1 \text{ group}) \\
&= \frac{1}{n} \cdot \frac{1}{n-1} \sum_{n_1=r}^{n-1} (n-n_1) \cdot \frac{r-1}{n_1-1} \cdot \frac{r-2}{n_1-2}.
\end{aligned}
$$

通过类似的推理，我们得到P(成功 $\cap$ 子情况 2)的相同表达式。

此外，用同样的方法：

$$
\begin{aligned}
P(\text{Success} \cap \text{Subcase 3}) &= \sum_{n_2=1}^{r-1} \sum_{n_1=r}^{n} \frac{1}{n} \cdot \frac{1}{n-1} \cdot P(\text{Second-best candidate before } n_1 \text{ lies in the } r-1 \text{ group} \mid \\
&\qquad n_2 \text{ lies in the } r-1 \text{ group}) \\
&= \frac{1}{n} \cdot \frac{1}{n-1} \cdot (r-1) \sum_{n_1=r}^{n} \frac{r-2}{n_1-2}.
\end{aligned}
$$

同样，通过类似的推理，我们得到 P(成功 $\cap$ 子情况 4) 的相同表达式。

因此，**策略 2** 的成功总概率为

$$
P(r) = \left(\frac{2(r-1)(r-2)}{n(n-1)} \right) \left(\sum_{i=r}^{n} \frac{1}{i-2} + \sum_{i=r}^{n-1} \frac{n-i}{(i-1)(i-2)} \right).
$$

2.2.3. 比较两种策略

利用第 2.2.1 节和第 2.2.2 节推导出的公式，我们绘制了两种策略的成功概率随r值变化的曲线，并将其与对 100,000 次试验取平均值的计算机模拟结果进行了比较（有关所实现代码的更多详情，请参见附录），如图 1 所示。

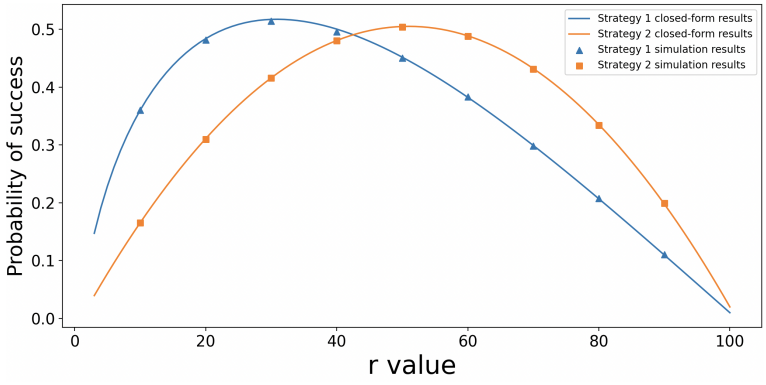

图 1：对于 $d = 2$ 和 $n = 100$，策略1和策略2的成功概率。解析结果（实线）与计算机模拟（符号）吻合良好。

2.2.4. 第三种策略（自适应搜索策略）

从图 1 所示的数据可以清楚地看出，在r值较低时，**策略 1** 明显优于**策略 2**；而在r值较高时，情况则恰恰相反。因此，我们可以将这两种策略结合起来，进一步优化其表现：

自适应搜索策略（策略3）： 实施策略 ***1***，拒绝前 $r_1 - 1$ 位候选人。如果到达第$(r_2 - 1)^{th}$ 位候选人时仍未作出选择，就对剩余候选人采用策略 ***2***。

采用此策略的成功概率为

$$\begin{aligned} &P(\text{Success with } \textbf{Strategy 1} \text{ on first } r_2 - 1 \text{ candidates}) \\ &\quad + P(\text{No selections made with } \textbf{Strategy 1} \cap \text{ Success with } \textbf{Strategy 2}). \end{aligned}$$

对于第二项，我们注意到：

$$\begin{aligned} &P(\text{No selections made with } \textbf{Strategy 1} \cap \text{ Success with } \textbf{Strategy 2}) \\ &\qquad = P(\text{No selections made with } \textbf{Strategy 1}) \cdot P(\text{Success with } \textbf{Strategy 2}) \\ &\qquad = P(\text{Best candidate before } r_2 \text{ lies before } r_1) \cdot P(\text{Success with } \textbf{Strategy 2}) \\ &\qquad = \left(\frac{r_1 - 1}{r_2 - 1}\right) \cdot P(\text{Success with } \textbf{Strategy 2}), \end{aligned}$$

要计算P(采用**策略 2**)时的成功概率，只需在第2.2.2节末尾的公式中令$r = r_2$。对于第一项，我们需要对第2.2.1节中的公式稍作推广：

2.2.5. 在受限样本上采用策略 *1*

本节的目标是计算在总样本中有n 名候选人的情况下，通过**策略 1** 在包含$r_2 - 1$ 名候选人的受限样本中取得成功的概率。与第2.2.1节类似，我们首先找出所有可能成功的子情形，并考虑最佳候选人与第二佳候选人分别位于n_1 和n_2 的位置：

$$\begin{aligned} &\text{Subcase 1:} \quad r_1 \le n_1 < n_2 \le r_2 - 1 \\ &\text{Subcase 2:} \quad r_1 \le n_2 < n_1 \le r_2 - 1 \\ &\text{Subcase 3:} \quad n_2 < r_1 \le n_1 \le r_2 - 1 \\ &\text{Subcase 4:} \quad r_1 \le n_2 \le r_2 - 1 < n_1 \\ &\text{Subcase 5:} \quad r_1 \le n_1 \le r_2 - 1 < n_2 \end{aligned}$$

对于前两个子情形，只需重复与第2.2.1节的**子情况 1** 和**子情况 2** 类似的证明，并限制n_1 和n_2的取值范围：

$$\begin{aligned} P(\text{Success} \cap \text{ Subcase 1 or 2}) &= 2 \sum_{n_1=r_1}^{r_2-2} \sum_{n_2=n_1+1}^{r_2-1} P(\text{Success} \cap \text{ Best two candidates at } n_1 \text{ and } n_2) \\ &= \frac{2}{n} \cdot \frac{1}{n-1} \sum_{n_1=r_1}^{r_2-2} (r_2 - 1 - n_1) \cdot \frac{r_1 - 1}{n_1 - 1}. \end{aligned}$$

类似地，$P(\text{成功} \cap \text{子情形 3}) = \frac{1}{n} \cdot \frac{1}{n-1} \cdot (r_1 - 1) \cdot (r_2 - r_1)$，并且根据类似的推理：

$$P(\text{Success} \cap \text{Subcase 4 or 5}) = \frac{2}{n} \cdot \frac{1}{n-1} \cdot (n - r_2 + 1) \sum_{n_1=r_1}^{r_2-1} \frac{r_1 - 1}{n_1 - 1}.$$

因此，化简后：

$$P(\text{Success with } \textbf{Strategy 1} \text{ on first } r_2 - 1 \text{ candidates}) = \left(\frac{r_1 - 1}{n(n-1)}\right) \left((r_2 - r_1) + 2 \sum_{i=r_1}^{r_2-1} \frac{n-i}{i-1}\right).$$

2.2.6. 策略*3*的概率计算

接下来，我们计算策略3的总成功概率 $P(r_1,r_2)$：

$$\begin{aligned}
P(r_1,r_2) &= \left(\frac{r_1 - 1}{n(n-1)}\right) \left(2(r_2 - 2) \left(\sum_{i=r_2}^{n} \frac{1}{i-2} + \sum_{i=r_2}^{n-1} \frac{n-i}{(i-1)(i-2)}\right) + (r_2 - r_1) + 2 \sum_{i=r_1}^{r_2-1} \frac{n-i}{i-1}\right) \\
&= \left(\frac{r_1 - 1}{n(n-1)}\right) \left(2(r_2 - 2) \left(\sum_{i=r_2}^{n} \frac{n-1}{(i-1)(i-2)}\right) + (r_2 - r_1) + 2 \sum_{i=r_1}^{r_2-1} \left(\frac{n-1}{i-1} - 1\right)\right) \\
&= \left(\frac{r_1 - 1}{n(n-1)}\right) \left(2(r_2 - 2)(n-1) \left(\sum_{i=r_2}^{n} \frac{1}{i-2} - \sum_{i=r_2}^{n} \frac{1}{i-1}\right) + (r_1 - r_2) + 2 \sum_{i=r_1}^{r_2-1} \left(\frac{n-1}{i-1}\right)\right) \\
&= \left(\frac{2(r_1 - 1)}{n(n-1)}\right) \left((n - r_2 + 1) + \frac{(r_1 - r_2)}{2} + \sum_{i=r_1}^{r_2-1} \left(\frac{n-1}{i-1}\right)\right).
\end{aligned}$$

图2展示了该搜索自适应策略的成功概率随（r_1,r_2）值变化的热力图，等高线表示迄今讨论的两种策略所达到的最大值：

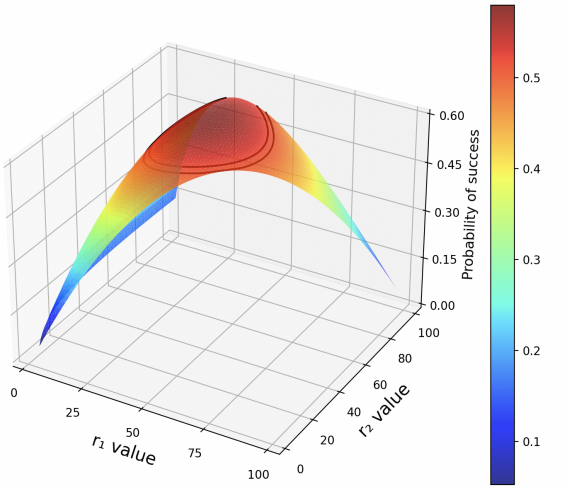

图2：采用自适应搜索策略时，成功概率随停止阈值 (r_1,r_2) 的变化，其中 $d = 2$ 且 $n = 100$。等高线表示先前策略1和策略2给出的最大成功概率。

表1比较了通过第2.2.6节推导出的公式得到的相应结果，以及对100,000次试验取平均值的计算机模拟结果（有关所实现代码的更多详情，请参阅附录）：

(r_1, r_2)	闭式解的成功概率(策略3)	模拟的成功概率(策略3)
(20, 50)	0.50439	0.50497
(20, 60)	0.51804	0.52063
(20, 70)	0.52043	0.51970
(30, 50)	0.54855	0.54934
(30, 60)	0.56939	0.56910
(30, 70)	0.57304	0.57300
(40, 50)	0.54252	0.54046
（40, 60）	0.57056	0.57028
(40, 70)	0.57546	0.57539

表1：闭式成功概率与模拟结果的比较

2.3. 可接受候选人数为 d 的情形：概率计算

在本节中，我们旨在推广上一节的结果，建立一个公式，用于计算在包含n 名候选人的候选池中，要求选出d 名最佳候选人之一时的总成功概率 $P(r_1, r_2,...、 r_d)$。与第2.2节类似，我们将首先分步确定在候选人池的受限样本中，d 种标准策略各自的总成功概率公式。随后，我们将基于互斥事件把这些策略组合成一个最终策略，从而最大化总体成功概率。

2.3.1. 候选人受限样本上的一般标准策略

当要求从d 名最佳候选人中选出一人时，共有d 种可能实施的策略。因此，对于每个取正整数值的 $2 \le s \le d$，我们可以定义：

Str策略 s： 拒绝前 $r-1$ 名候选人，然后选择相对于被拒绝候选人的候选人池而言排名至少为 s 的第一名候选人。

本小节旨在确定，对于每个 $s \le d$，在含有$r'-1 \le n$ 位候选人的受限样本上实施策略 s，并拒绝前$r-1$ 位候选人时，成功的总概率 $P(s,r,r')$。

假设排名不计，最优秀的d 位候选人占据$n_1 < n_2 < ... < n_d \le n$这些位置，那么按照已拒绝候选者池中最优秀的$d$ 位候选人的人数对可能成功的子情形进行分类会很方便，因此我们考虑以下d 种一般子情形：

子情况 0：$r \le n_1 < n_2 < ... < n_d$子情况 1：$n_1 < r \le n_2 < n_3 < ... < n_d$子情况 2：$n_1 < n_2 < r \le n_3 < n_4 < ... < n_d$... 子情形k：$n_1 < n_2 < ... < n_k < r \le n_{k+1} < ... < n_d$... 子情形d-1：$n_1 < n_2 < ... < n_{d-1} < r \le n_d$

在继续之前，先定义两个相关表达式会很有帮助：

2.3.2. 连续候选项的排列

给定某个n，$x \in \mathbb{Z}^+$，其中$x \geq n$，注意到

$$\frac{1}{n!}\prod_{i=0}^{n-1}(x-i) = \binom{x}{n}.$$

那么，位于位置a和b之间的$k \in \mathbb{Z}^+$个连续候选人的排列总数，其中第i个候选人占据位置n_{k-i+1}，且$a \leq n_k < n_{k-1} < \ldots < n_1 \leq b$，可表示为

$$\sum_{n_k=a}^{b-k+1}\sum_{n_{k-1}=n_k+1}^{b-k+2}\ldots\sum_{n_i=n_{i+1}+1}^{b-k+i}\ldots\sum_{n_1=n_2+1}^{b}\cdot(1) \;=\; \frac{1}{k!}\prod_{i=-1}^{k-2}(b-a-i) \;=\; \frac{1}{k!}\prod_{i=0}^{k-1}\big((b-a+1)-i\big) \;=\; \binom{b-a+1}{k}.$$

上述等式分别可对k和b运用归纳法证明，其中$b \geq (a+k-1)$:

$$\sum_{n_k=a}^{b-k+1}\sum_{n_{k-1}=n_k+1}^{b-k+2}\ldots\sum_{n_i=n_{i+1}+1}^{b-k+i}\ldots\sum_{n_1=n_2+1}^{b}\cdot(1) \;=\; \sum_{n_k=a}^{b-k+1}\left(\frac{1}{(k-1)!}\prod_{i=-1}^{k-3}(b-(n_k+1)-i)\right) \;\big[\textit{induction hypothesis on } (k-1)\big]$$

$$= \frac{1}{(k-1)!}\prod_{i=-1}^{k-3}(b-a-i-1) \;+\; \frac{1}{(k-1)!}\sum_{n_k=a}^{(b-1)-k+1}\prod_{i=-1}^{k-3}\big((b-1)-(n_k+1)-i\big)$$

$$= \frac{1}{(k-1)!}\prod_{i=-1}^{k-3}(b-a-i-1) \;+\; \frac{1}{k!}\prod_{i=-1}^{k-2}\big((b-1)-a-i\big) \quad \big[\textit{induction hypothesis on } (b-1)\big]$$

$$= \frac{1}{k!}\left(((b-1)-a+2)\prod_{i=-1}^{k-3}\big((b-1)-a-i\big)\right)$$

$$= \frac{1}{k!}\prod_{i=-1}^{k-2}(b-a-i). \quad \blacksquare$$

2.3.3. 选择

给定某个$k \in \mathbb{Z}^+$，假设有k个候选人所处的位置严格低于r'，其中m个候选人的位置大于或等于r，且$m \leq k \leq r'-1$。在本小小节中，我们计算概率$S_s(k,m)$：给定某个$s \leq k-m$，在m个候选人中，至少有一人的排名相对于k个候选人而言不低于第s名。

注意，当且仅当最优秀的s个候选人全部位于严格低于r的$k-m$个位置之中时，这种情况才不会发生；此时剩余的$k-s$个候选人可以不受限制地排列。因此，通过排列计数可得:

$$1 - S_s(k,m) \;=\; \frac{\frac{(k-m)!}{(k-m-s)!}\cdot(k-s)!}{k!},$$

使得

$$S_s(k,m) \;=\; 1-\prod_{i=0}^{s-1}\frac{k-m-i}{k-i}.$$

定义上述表达式后，我们现在将通过把总共$d-1$个子情形划分为两种类型（以一般子情形k为依据），来计算$P(s,r,r')$，并在接下来的两个子小节中分别计算它们的概率。

2.3.4. k $0 \leq s$ $\leq$ 的子情形 -1

请注意，对于这类子情形，成功概率与排名最高的d位候选人的具体排名无关（因为所采用的策略保证，只要面试到排名最高的d 位候选人中的任何一位，就会选择他/她），但取决于n_{k+1}位之前候选人的排名（因为**策略 s** 并不要求候选人必须位列排名最高的d 位才能被选中）。

因此，采用与第2.2.1节类似的方法：

$$
\begin{aligned}
& P(\text{Success} \cap \text{Subcase } k \leq s-1) \\
= & \sum_{n_1=1}^{r-k} \sum_{n_2=n_1+1}^{r-k+1} \cdots \sum_{n_k=n_{k-1}+1}^{r-1} \sum_{n_{k+1}=r}^{r'-1} \sum_{n_{k+2}=n_{k+1}+1}^{n-(d-(k+2))} \cdots \sum_{n_d=n_{d-1}+1}^{n} P(\text{Success} \cap \text{Best } d \text{ candidates at } n_1, \ldots, n_d) \\
= & \binom{r-1}{k} \cdot \sum_{n_{k+1}=r}^{r'-1} \binom{n-n_{k+1}}{d-k-1} \cdot P(\text{Best } d \text{ candidates at } n_1, \ldots, n_d) \cdot P(\text{Success} \mid \text{Best } d \text{ candidates at } n_1, \ldots, n_d) \\
= & \binom{r-1}{k} \cdot d! \prod_{i=0}^{d-1} \frac{1}{n-i} \cdot \sum_{n_{k+1}=r}^{r'-1} \binom{n-n_{k+1}}{d-k-1} \\
& \qquad \cdot P(\text{Best } s \text{ candidates before } n_{k+1} \text{ in the } r-1 \text{ group} \mid \text{Best } d \text{ candidates at } n_1, \ldots, n_d) \\
= & \binom{r-1}{k} \cdot d! \prod_{i=0}^{d-1} \frac{1}{n-i} \cdot \sum_{n_{k+1}=r}^{r'-1} \binom{n-n_{k+1}}{d-k-1} \cdot \prod_{j=k+1}^{s} \frac{r-j}{n_{k+1}-j}.
\end{aligned}
$$

2.3.5. s $\leq k$ $\leq d$ -1 的子情形

与前一个子小节相反，请注意，对于这类子情形，成功概率取决于排名最高的d 名候选人的具体排名（因为选择性更强的策略甚至可能拒绝这样的候选人，除非他们相对于已拒绝候选者池的排名足够高），但与第n_{k+1}个位置之前候选人的排名无关（因为**策略 s**在这种情况下要求候选人的排名高于排名最高的k 名被拒绝候选人中的至少一人才能被选中，因此该候选人必须是所有候选人中排名最高的d 名之一）。

那里 因此，由于最佳d 位候选人的排名存在依赖关系，我们需要采用稍微不同的方法 ：

$$
\begin{aligned}
& P(\text{Success} \cap \text{Subcase } k \geq s) \\
= & \sum_{i=1}^{d-k} P\left((\text{Success} \cap \text{Subcase } k \geq s) \cap n_1 < \ldots < n_k < r \leq n_{k+1} < \ldots < n_{k+i} < r' \leq n_{k+i+1} < \ldots < n_d \leq n\right) \\
= & \sum_{i=1}^{d-k} P(n_1 < \ldots < n_k < r \leq n_{k+1} < \ldots < n_{k+i} < r' \leq n_{k+i+1} < \ldots < n_d \leq n) \\
& \qquad \cdot P\left((\text{Success} \cap \text{Subcase } k \geq s) \mid n_1 < \ldots < n_k < r \leq n_{k+1} < \ldots < n_{k+i} < r' \leq n_{k+i+1} < \ldots < n_d \leq n\right) \\
= & \sum_{i=1}^{d-k} \frac{\binom{r-1}{k} \cdot \binom{r'-r}{k-i} \cdot \binom{n-r'+1}{d-}}{\binom{n}{d}} \cdot P(n_{k+1}, \ldots, n_{k+i} \text{中的至少一位候选人相对于} n_1, \ldots, n_{k+i} \text{中的候选人排名至少为第} s \text{名}) \\
= & \binom{r-1}{k} \cdot d! \prod_{j=0}^{d-1} \frac{1}{n-j} \cdot \sum_{i=1}^{d-k} \binom{r'-r}{i} \cdot \binom{n-r'+1}{d-k-i} \cdot S_s(k+i, i).
\end{aligned}
$$

2.3.6. 计算受限候选人样本中任意给定策略 s 的成功概率

利用前两个子小节的结果，并考虑到成功只能发生在第2.3.1节列出的d种子情形中，我们终于可以计算出，在介于r和$r'-1$之间的受限候选人样本中，任意给定策略s的成功概率：

$$
\begin{aligned}
P(s,r,r') &= \sum_{k=0}^{d-1} P(\text{Success} \cap \text{Subcase } k) \\
&= \sum_{k=0}^{s-1} P(\text{Success} \cap \text{Subcase } k \le s-1) \; + \; \sum_{k=s}^{d-1} P(\text{Success} \cap \text{Subcase } k \ge s) \\
&= d! \prod_{l=0}^{d-1} \frac{1}{n-l} \cdot \\
&\left(\sum_{k=0}^{s-1} \binom{r-1}{k} \sum_{i=r}^{r'-1} \binom{n-i}{d-k-1} \prod_{j=k+1}^{s} \frac{r-j}{i-j} \; + \; \sum_{k=s}^{d-1} \binom{r-1}{k} \sum_{i=1}^{d-k} \binom{r'-r}{i} \cdot \binom{n-r'+1}{d-k-i} \cdot S_s(k+i,i) \right).
\end{aligned}
$$

2.3.7. 一般自适应搜索策略

沿用第 2.2.4 节的思路，我们现在可以一般性地定义：

策略 *d+1*： 拒绝前 $r_1 - 1$ 位候选人。对于每个 $s \le d$，在尚未做出选择之前，实施策略 $\boldsymbol{s}$，范围从第 r_s^{th}位到第$(r_{s+1}-1)^{th}$ 位候选人，其中 $r_{d+1} = n + 1$。

借助上一小节的结果，我们现在可以计算该自适应搜索策略的总成功概率。请注意，对于$s = 1$，下面的乘积算子取值为 1，从而得到**策略 1** 在r_1,r_2, 之间的成功概率，正如所需。

$$
\begin{aligned}
P(r_1,r_2,...,r_d) &= \sum_{s=1}^{d} \prod_{i=1}^{s-1} P(\text{Strategy } i \text{ made no selections}) \cdot P(s,r_s,r_{s+1}) \\
&= \sum_{s=1}^{d} \prod_{i=1}^{s-1} P(\text{best } i \text{ candidates before position } r_{i+1}-1 \text{ in the first } r_i - 1 \text{ positions}) \cdot P(s,r_s,r_{s+1}) \\
&= \sum_{s=1}^{d} \prod_{i=1}^{s-1} \prod_{j=1}^{i} \frac{r_i - j}{r_{i+1} - j} \cdot P(s,r_s,r_{s+1}) \\
&= \sum_{s=1}^{d} \prod_{i=1}^{s-1} \frac{r_i - i}{r_s - i} \cdot P(s,r_s,r_{s+1}),
\end{aligned}
$$

其中，最后一步可以通过归纳法证明（该命题对 $s = 1$ 显然成立）：

$$
\begin{aligned}
\prod_{i=1}^{s-1} \prod_{j=1}^{i} \frac{r_i - j}{r_{i+1} - j} &= \left(\prod_{j=1}^{s-1} \frac{r_{s-1} - j}{r_s - j} \right) \cdot \left(\prod_{i=1}^{s-2} \prod_{j=1}^{i} \frac{r_i - j}{r_{i+1} - j} \right) \\
&= \left(\frac{r_{s-1} - (s-1)}{r_s - (s-1)} \cdot \prod_{j=1}^{s-2} \frac{r_{s-1} - j}{r_s - j} \right) \cdot \left(\prod_{i=1}^{s-2} \frac{r_i - i}{r_{s-1} - i} \right) \\
&= \frac{r_{s-1} - (s-1)}{r_s - (s-1)} \cdot \prod_{i=1}^{s-2} \frac{r_i - i}{r_s - i} \; = \; \prod_{i=1}^{s-1} \frac{r_i - i}{r_s - i}.
\end{aligned}
$$

因此，对于每个新增的阈值 r_s，该策略都会使总体成功概率随 d 单调上升。虽然这并不令人意外，但它确保了在选择标准之上找到次优选择的概率更高。

3. 搜索成功率优化的计算结果

本节通过计算机模拟验证上一节推导出的结果，并提出一种算法，可在合理的计算时间内最大化成功概率。

3.1. 模拟任意数量试验的结果

下面的算法针对总共n 名候选人，在给定选择阈值$(r_1, r_2, ..., r_d)$的情况下，对寻找最佳d 名候选人的问题模拟 *numtrials* 次。通过模拟，可以用它验证第2.3.7节推导出的公式的解析结果。该算法的完整 Python 代码见附录。

算法1：模拟选择过程

参数： n、d、*numtrials*、选择阈值$(r_1, r_2, ..., r_d)$，其中$r_{d+1} = n + 1$

对于试验 $= 1, \cdots,$ 试验次数 **执行**
  生成一个包含n 名候选人的列表，并赋予随机排名顺序；
  将可接受的最低标准 设为列表中排名最高的第d位；
  将选择轮次s 设为 1；
  对于 $s = 1$到 d **执行**
    对于 $i = r_s$ 至 $r_{s+1} - 1$ **执行**
      如果 第 i 个候选人的排名高于迄今为止最佳候选人的排名 **则**
        如果 第 i 个候选人的排名高于可接受的最低标准 **则**
          记录成功试验
        else
          记录失败试验
  记录失败试验
计算成功试验百分比：成功试验数除以*numtrials* 试验总数。

例如，选择阈值 我们针对 100 名候选人中排名前 5 的候选人，将该问题模拟一百万次，使用的为 50、60、65、75、80（具体实现的代码详情请参见附录）。

在这种情况下，第2.3.7节推导出的最终公式计算得出的成功概率为 0.829427（精确到小数点后 6 位）；进一步进行类似测试后发现，结果非常吻合，概率差异通常低于 10^{-3}。

3.2. 最大化成功概率：蛮力法

为了确定使成功概率达到最大的 $r_1, r_2, \ldots\ldots$、r_d 的取值，我们可以使用以下直接程序。完整的 Mathematica 代码见附录。

1) 生成一个列表 $rlist[d, n]$，其中包含给定 d、n 值时所有可能的组合 $r_1, r_2, ..., r_d$。2) 使用第2节末尾推导出的公式，计算给定 d,n 值以及包含当前选择 $r_1, r_2, ..., r_d$的列表 *rlist* 对应的成功概率。3) 对于给定的 d、n 值，通过穷举遍历 $rlist[d,n]$，计算 $r_1, r_2, ..., r_d$ 的最大概率和最优选择。

尽管这种方法能够保证找到问题的最优解，但由于运行时间较长，效率很低。事实上，如下方公式所示，随着 d,n 值增大，算法需要遍历的列表 $L(d,n)$ 总数会迅速增加：

$$L(d,n) \;=\; \frac{1}{d!}\prod_{i=1}^{d}(n-i) \;=\; \frac{n-d}{n}\binom{n}{d}.$$

例如，如果我们想从总共$n \;=100$ 位候选人中，选出$d \;=6$ 位最佳候选人中的一位，那么仅$r_1, r_2,..., r_6$，就需要遍历超过十亿种可能组合，这使得该算法在实际应用中非常不便。

这一问题促使我们在下一节介绍模拟退火方法。

3.3. 最大化成功概率：模拟退火

该方法旨在给定d、n 值的情况下，在总搜索空间$rlist[d、n]$中进行任意次数$numIterations$ 的搜索，以找到可能的最大成功概率。

我们从初始状态$rlist0[d、n]$（对$r_1, r_2,...、r_d$的最优选择作出的最佳猜测）开始。随后，给定当前状态$rlistCurrent$，接受概率函数P 会根据各状态成功概率的相对差异以及退火日程温度T，确定转移到随机邻近状态$rlistCandidate$的概率。给定状态的成功概率p 由第2节末尾推导出的公式确定（计算方式与第3.2节步骤2完全相同）。T 取决于当前迭代次数i 相对于允许的总迭代次数$numIterations$ 的比例，以及用于放大邻近状态之间相对较小概率差异的$RescaleFactor$ 。通常，随着算法接近最大迭代次数，T 会逐渐趋近于零，从而降低状态改变的概率。

下方伪代码展示了该算法。实现该算法的完整 Mathematica代码、$rlist0[d、n]$, T 和 P 的完整定义，以及邻居生成过程，均见附录。

算法2： 用于求解最优解的模拟退火

参数： d , n, 迭代次数
设置当前状态 $s \leftarrow rlist0[d,n]$;
对于 $i \;=1$到迭代次数 **执行**
  设置温度$T \leftarrow \left(\frac{n-i}{n}\cdot RescaleFactor\right)$;
  选择一个随机邻居s_{new} ;
  设置接受概率$P(s,s_{new}) \leftarrow \frac{1}{1+e^{\frac{p(s)-p(snew)}{T}}}$;
  如果$P(s, s_{new}) \;>$ 随机$(0,1)$[1] **则**
    设置$s \leftarrow s_{new}$;
输出最终状态 s;

尽管如此　模拟退火并不能保证找到问题的最优解，但它可以找到极其接近的估计值 b　，只需采样总搜索空间中极小的一部分。

表2通过比较该方法在可忽略不计的计算时间内（仅需100次迭代）得出的结果与暴力搜索在长得多的时间内计算出的最大概率，展示了该方法的效率；后者所需时间与上文所示公式 $L(d,n)$ 成正比：

[1]此函数调用会生成一个在 (0,1) 上均匀分布的随机数。

参数 $(d,\ n)$	模拟退火	最优 r_i 值	蛮力法	最优 r_i 值	差异
2, 20	0.604616	(8, 14)	0.604616	(8、14)	0.000000
3, 20	0.747476	(7, 13, 16)	0.747476	(7, 13, 16)	0.000000
4, 20	0.842694	(7, 12, 15, 17)	0.842694	(7, 12, 15, 17)	0.000000
2, 30	0.594123	(11, 21)	0.594123	(11, 21)	0.000000
3, 30	0.734922	(11, 18, 24)	0.734922	(11, 18, 24)	0.000000
2, 50	0.585779	(18, 34)	0.585779	(18, 34)	0.000000
2, 100	0.579554	(36, 68)	0.579561	(35, 67)	0.000007

表2：模拟退火与直接暴力法的比较

尽管暴力搜索方法的运行时间复杂度目前尚不允许我们验证，对于更大的d,n 值，偏离最大概率的结果是否同样有希望，但这些初步结果表明，只要选择合适的参数，模拟退火就能高效地快速找到最优解。

虽然这并不令人意外，但值得注意的是，对于固定的n，随着d增大，成功概率也会单调增加。第2.2.1节从解析角度阐释了这一概念，其中指出，对于固定的n 和单一停止点r，当$d = 2$时，成功概率几乎是$d = 1$这一标准情形下的两倍。类似地，对于任意$d > 1$，只需采用策略 1 的严格选择标准（其成功概率低于最优自适应搜索策略），即可发现，与标准情形相比，我们能够保证获得更高的成功概率，因为在选择标准相同的情况下，我们的成功标准会更加宽松。

4. 讨论与结论

本研究通过将次优选择也视为成功聘用，拓展了经典秘书问题；同时采用具有多个停止标准的自适应搜索策略，从而能够闭式计算聘用任意一位前d 名候选人的概率。我们的方法为原始秘书问题提供了更全面、更实用的解决方案，因为我们考虑了现实场景：在这些场景中，未必总能聘用到最佳候选人。此外，与暴力穷举搜索法相比，将模拟退火法用作替代计算工具，能够更快速、更有效地进行优化。我们的研究结果对多个领域均有启示，因为这些领域普遍面临不确定性下的决策挑战 [1, 41, 43], , 包括人员选拔和资源分配。

为了进一步完善本文提出的问题解决方案，一个关键步骤是确定能否通过代数方法推导出闭式表达式，直接求得任意d、n取值下的最优选择阈值，进而得到最大成功概率。推导出这样的公式将大幅简化第3节所述问题的计算过程，并有望帮助我们更全面地理解其解法。然而，由于变量数量不定（这是较宽松的选择标准所致），且第2.3.7节计算的一般概率公式较为复杂，因此难以沿用标准情形下求解最优策略的方法 [26]。

此外，另一个相关的次要方面是对第3.3节提出的概率最大化方法的有效性进行更严格的验证。这可以通过解决上述更根本的问题来实现，即提供一种能够针对任意参数直接计算最优选择阈值和最大成功概率的方法；或者，也可以通过增强计算能力来评估较不简单情形下的偏差。如果此类验证发现当前方法在d、n取值较大时并不适用，则可以通过优化模拟退火的温度函数（例如，采用算法2所示的*RescaleFactor*）并增加迭代总数来改进该方法。另一种做法是采用粒子群优化，以扩大搜索空间的覆盖范围 [26]。

受第2节中策略1和策略2比较的启发，我们的研究探讨了一种具有任意多个d 截断准则的自适应搜索策略。此前，Gusein-Zade [23] 以及 Gilbert 和 Mosteller [20],分别研究了$d = 2$ 的基本情形，并计算出当$n \to \infty$ 趋于

≈ 0.574 时的成功概率极限，以及一般d 取值下的渐近行为。更重要的是，我们能够推导出获得排名前d 的任一候选人的成功概率的闭式解。与参考文献 [8], 不同，我们的研究不会偏向排名前d 的任何一位候选人：选择最佳候选人、第二佳候选人或排名第d的候选人，都被视为同等成功；而未能选中排名前d的候选人，无论其最终排名如何，都被视为同等失败。此外，与参考文献 [47], 不同，我们的研究将所有候选人（包括排名前d的候选人）都视为可以直接比较的对象。

未来研究的一个有前景的拓展方向，是将博弈论 [35]引入秘书问题，以考虑选拔委员会中 [3]具有独特偏好的多位决策者 [12]。此类拓展有助于制定公平、高效的选拔实践，同时减轻内群体偏好 [33, 11]等人类偏见的影响。

总之，我们利用多重停止准则和闭式解，计算了录用前d 位候选人中任意一人的概率。为避免耗费大量计算时间，我们采用模拟退火法进行优化；与穷举搜索相比，该方法表现良好，并显著节省了计算时间。我们的研究结果强调，采用自适应搜索策略和计算方法，对于在满意化秘书问题中最大限度地提高获得次优结果的可能性至关重要。

致谢

F.F. 衷心感谢比尔及梅琳达·盖茨基金会（奖项编号 OPP1217336）、美国国立卫生研究院 COBRE 计划（资助编号 1P20GM130454）以及 Neukom CompX 教师资助项目提供的支持。Roberto Brera 感谢 Ian Gill 在项目初期作出的重要贡献。

参考文献

[1] AR Abdel-Hamid、JA Bather 和 GB Trustrum。《候选人数未知时的秘书问题》。Journal of Applied Probability，19(3):619–630，1982。[2] Susanne Albers 和 Leon Ladewig。《k-秘书问题的新结果》。Theoretical Computer Science，863:102–119，2021。[3] Steve Alpern 和 Vic Baston。《设有遴选委员会的秘书问题：从众型委员会能否聘用更好的秘书？》。Manage- ment Science，63(4):1184–1197，2017。[4] R Bartoszy´nski 和 Z Govindarajulu。《有面试成本的秘书问题》。Sankhy¯a: The Indian Journal of Statistics, Series B，第11–28页，1978。[5] MohammadHossein Bateni、Mohammadtaghi Hajiaghayi 和 Morteza Zadimoghaddam。《子模秘书问题及其扩展》。ACM Transactions on Algorithms (TALG)，9(4):1–23，2013。[6] L Bay´on、P Fortuny Ayuso、JM Grau、AM Oller-Marc´en 和 MM Ruiz。《多次返回秘书问题》。Discrete Applied Mathemat- ics，320:33–46，2022。[7] J Neil Bearden。《一种基于排名选择并采用基数收益的新型秘书问题》。Journal of Mathematical Psychology，50 59, 2006 1 58( ): – 。[8] J Neil Bearden、Amnon Rapoport 和 Ryan O Murphy。《基于排名的收益下的序贯观察与选择：一项实验研究》。Management Science，52(9):1437–1449，2006。[9] Domagoj Bradac、Anupam Gupta、Sahil Singla 和 Goran Zuzic。《秘书问题的鲁棒算法》。arXiv预印本 arXiv:1911.07352，2019。[10] Niv Buchbinder、Kamal Jain 和 Mohit Singh。《通过线性规划求解秘书问题》。Mathematics of Operations Research，39(1):190–206，2014。[11] Ho-Chun Herbert Chang 和 Feng Fu。《数学中的精英主义与不平等》。Humanities and Social Sciences Co mmunications，8 8, 2021 1 1( ): – 。[12] Robert W Chen、Burton Rosenberg 和 Larry A Shepp。《有两位决策者的秘书问题》。Journal of Applied Probability，34(4):1068–1074，1997。[13] Robert Chin、Jonathan E Rowe、Iman Shames、Chris Manzie 和 Dragan Ne˘si´c。《序数优化与离线多重噪声秘书问题》。arXiv预印本 arXiv:2106.01185，2021。[14] YS Chow、Sigaiti Moriguti、Herbert Robbins 和 SM Samuels。《基于相对排名的最优选择（“秘书问题”？）》。Israel Journal of mathematics，2(2):81–90，1964。[15] Ruth M Corbin。《作为选择模型的秘书问题》。Journal of Mathematical Psychology，21(1):1–29，1980。[16] Jos´e Correa、Andr´es Cristi、Laurent Feuilloley、Tim Oosterwijk 和 Alexandros Tsigonias-Dimitriadis。《独立抽样下的秘书问题》。载于《2021年ACM-SIAM离散算法研讨会论文集（SODA）》，第2047–2058页。SIAM，2021。[17] Thomas S Ferguson。《谁解决了秘书问题？》。Statistical science，4(3):282–289，1989。[18] PR Freeman。《秘书问题及其扩展：综述》。International Statistical Review/Revue Internationale de Statistique，第189–206页，1983。[19] Jacqueline Gianini 和 Stephen M Samuels。《无限秘书问题》。The Annals of Probability，4(3):418–432，1976。

[20] John P Gilbert 和 Frederick Mosteller。《识别序列最大值》。Selected Papers of Frederick Mosteller，第355–398页，2006。[21] Kenneth S Glasser 和 Richard Holzsager Austin Barron。《d选择秘书问题》。Sequential Analysis，2(3):177–199，1983。[22] Daniel G Goldstein、R Preston McAfee、Siddharth Suri 和 James R Wright。《重复秘书问题中的学习》。arXiv预印本 arXiv:1708.08831，2017。[23] SM Gusein-Zade。《选择问题与独立试验序列的最优停止规则》。Theory of Probability & Its Applications，11(3):472–476，1966。[24] Greg Harrell、Josh Harrison、Guifen Mao 和 Jin Wang。《在线拍卖与秘书问题》。载于《国际科学计算会议（CSC）论文集》，第241页。The Steering Committee of The World Congress in Computer Science, Computer ?，2015。[25] M Hlynka 和 JN Sheahan。《随机游走的秘书问题》。Stochastic Processes and their applications，28(2):317–325，1988。[26] James Kennedy 和 Russell Eberhart。《粒子群优化》。载于《ICNN'95国际神经网络会议论文集》，第4卷，第1942–1948页。IEEE，1995。[27] Elias Koutsoupias 和 Philip Lazos。《作为秘书问题的在线交易》。载于《算法博弈论国际研讨会》，第201–212页。Springer，2018。[28] Grzegorz Kubicki 和 Michal Morayne。《秘书问题的图论推广：有向路径情形》。SIAM Journal on Discrete Mathematics，19(3):622–632，2005。[29] M Lee、T Gregory 和 Matthew Welsh。《完全信息秘书问题中的决策》。第二十六届认知科学学会会议论文集，第819–824页，2005。[30] Denis V Lindley。《动态规划与决策理论》。Journal of the Royal Statistical Society: Series C (Applied Statistics)，10(1):39–51，1961。[31] Thomas J Lorenzen。《存在抽样成本时的最优停止：秘书问题》。The Annals of Probability，9(1):167–172，1981。[32] Mohammad Mahdian、Randolph Preston McAfee 和 David Pennock。《满足风险率条件的秘书问题》。载于《互联网与网络经济国际研讨会》，第708–715页。Springer，2008。[33] Naoki Masuda 和 Feng Fu。《内群体偏好的进化模型》。F1000prime reports，7，2015。[34] TAMAS F M´ori。《多重选择下的随机秘书问题》。Annales Univ. Sci. Budapestinensis de Rolando Eotvos Nominatae V，第91–102页，1984。[35] Matjaˇz Perc 和 Attila Szolnoki。《协同进化博弈？简要综述》。BioSystems，99(2):109–125，2010。[36] J Preater。《多重选择秘书问题》。Mathematics of Operations Research，19(3):597–602，1994。[37] Peter A Rogerson。《秘书问题中选中任意排名申请者的概率》。Journal of applied probability，24(2):527–533，1987。[38] H Rubin 和 SM Samuels。《有限记忆秘书问题》。The Annals of Probability，第627–635页，1977。[39] Stephen M Samuels。 [谁解决了秘书问题？]：评论：谁将解决秘书问题？Statistical Science，4(3):289–291，1989。[40] Stephen M Samuels。《作为基准界来源的秘书问题》。Lecture Notes-Monograph Series，第371–387页，1992。[41] Darryl A Seale 和 Amnon Rapoport。《相对排名下的最优停止行为：总体规模未知时的秘书问题》。Journal of Behavioral Decision Making，13(4):391–411，2000。[42] Marek Skarupski。《观测可能出错时的秘书问题》。Mathematics，8(10):1639，2020。[43] MH Smith。《不确定就业下的秘书问题》。Journal of applied probability，12(3):620–624，1975。[44] MH Smith 和 JJ Deely。《有限记忆秘书问题》。Journal of the American Statistical Association，70(350):357–361，1975。[45] Theodor J Stewart。《选项数量未知时的秘书问题》。Operations Research，29(1):130–145，1981。[46] Krzysztof Szajowski 和 Mitsushi Tamaki。《广义秘书问题中候选人的有效期》。Operations Research Letters，44(4):498–502，2016。[47] Xinlin Wu 和 Haixiong Li。《秘书问题的满意化策略：理论与模拟》。Communications in Statistics-Theory and Methods，第1–14页，2020。[48] Tong Zhao、Mandy Hu、Razieh Rahimi 和 Irwin King。《时机至关重要！将每日特惠网站中的客户行为建模为秘书问题》。载于《2017年国际联合神经网络会议（IJCNN）》，第3670–3679页。IEEE，2017。