

Hiring Decisions based on Optimal Stopping Policies

Stochastic Models and Optimization Project

Emile Naude, Anneke Speijers, Harihara Subramanyam Sreenivasan, Max van Esso

April 1, 2016

The secretary problem is a classic optimal stopping problem that has a long history in mathematics literature. Known also as the marriage problem, the dowry problem and the best choice problem, amongst others, it goes as follows. A finite number of items is presented one-by-one in random order to an observer. After each item has been presented, the observer ranks all items that have been presented so far in terms of desirability. A decision must be made to accept or reject an item immediately after it has been presented and the process stops once an item is accepted. The aim is to determine an optimal stopping rule that maximises the probability of accepting the most desirable item.

Many variations to the standard problem exist and in this project we consider some of them. We contextualise the problem by imagining that we work for a hiring consultancy. Our goal is to determine the optimal stopping policy for offering a job to the best candidate or candidates. The following assumptions are common to all of the scenarios we discuss:

- 1) All candidates have fulfilled the prerequisites for the position. The worst will still fulfill the basic requirements but nothing more. The best possible candidate is such that the company will substantially benefit from the hire.
- 2) The candidates are evaluated on various quantitative scales which are combined to arrive at a single score. No two candidates have the same score. This allows for them to be ranked.
- 3) Candidates have offers from other companies so we must decide to offer them the job or reject them immediately after the interview.
- 4) A rejected candidate cannot be recalled.
- 5) The position(s) must be filled. If we get to the end of the process without offering the job to any candidate, we must offer it to the last candidate(s).
- 6) Candidates offered a position are obliged to take it.

The Three Scenarios

- 1) We wish to pick the best out of N candidates.
- 2) We wish to pick one of the best k candidates out of N .
- 3) We wish to pick all of the best k candidates out of N .

1) The best out of N

This is the standard secretary problem, whereby we have N candidates and we want to maximise our probability of choosing the best one. The optimal strategy is known to be of the form: Let a certain number of candidates pass, only observing their scores. After some critical number of candidates, pick the first one

that has a higher score than those already observed. Our goal is to determine this critical number. Our methodology follows that put forward by Beckmann (1990) [1].

Let us start by defining the following terminology:

Win \sim choosing the best among all candidates.

Value \sim the expected probability of winning.

Viable Candidate \sim any candidate who is the best so far.

The dynamic program can be defined as follows:

Let m be the number of candidates seen and n be the number of candidates that are remaining, such that,

$$m + n = N$$

m is the variable that represents the state of our dynamic program.

The cost associated with this problem can be viewed as the cost of not hiring the best candidate. Therefore the cost increases over time with reduction in choices. The constraint is that one of these candidates must be hired.

We define two value functions:

- 1) v_m is the value when m candidates have been interviewed and the m^{th} candidate has not been chosen.
- 2) u_m is the value when m candidates have been observed and m is the best candidate so far.

When the last observed candidate is not viable, then an optimal policy requires that the search be continued so that v_m applies. A decision rule is needed for when the last candidate is a viable candidate. This rule can only depend on the state variable m and the total number of candidates N .

The value functions update recursively in the following manner:

$$1) v_m = \frac{m}{m+1}v_{m+1} + \frac{1}{m+1}u_{m+1}$$

Given that the m^{th} candidate is not chosen, the $m+1^{th}$ candidate will be either a viable candidate with probability $\frac{1}{m+1}$, or not with probability $\frac{m}{m+1}$. If not, the value will become v_{m+1} , but if he/she is a viable candidate the value will become u_{m+1} .

$$2) u_m = \max[\frac{m}{N}, v_m]$$

Given that the m^{th} candidate is a viable candidate, there is a decision to choose this candidate, or to continue. If chosen, the value is $\frac{m}{N}$ (just the probability that the chosen one is best among all N). If not chosen, the value is v_m , as already established.

Starting with $m = N$, we will solve the above equations recursively. By definition $v_N = 0$ and $u_N = 1$.

At $m = N - 1$

$$v_{N-1} = \frac{N-1}{N}v_N + \frac{1}{N}u_N$$

$$v_{N-1} = \frac{1}{N}$$

using this, we get

$$u_{N-1} = \max \left[\frac{N-1}{N}, v_{N-1} \right]$$

$$u_{N-1} = \frac{N-1}{N}$$

At $m = N - 2$

$$\begin{aligned} v_{N-2} &= \frac{N-2}{N-1} \cdot v_{N-1} + \frac{1}{N-1} \cdot u_{N-1} \\ &= \frac{N-2}{N-1} \cdot \frac{1}{N} + \frac{1}{N-1} \cdot \frac{N-1}{N} \\ &= \frac{1}{N} \left(\frac{N-2}{N-1} + \frac{N-2}{N-2} \right) \\ &= \frac{N-2}{N} \cdot \left(\frac{1}{N-2} + \frac{1}{N-1} \right) \end{aligned}$$

using, v_{N-1} , we get

$$\begin{aligned} u_{N-2} &= \max \left[\frac{N-2}{N}, \frac{N-2}{N} \left(\frac{1}{N-2} + \frac{1}{N-1} \right) \right] \\ u_{N-2} &= \frac{N-2}{N} \max \left(1, \frac{1}{N-2} + \frac{1}{N-1} \right) \end{aligned}$$

We can generalize this pattern to the form,

$$v_m = \frac{m}{N} \left(\frac{1}{m} + \frac{1}{m+1} + \dots + \frac{1}{N-1} \right)$$

and

$$u_m = \frac{m}{N} \max \left[1, \frac{1}{m} + \frac{1}{m+1} + \dots + \frac{1}{N-1} \right]$$

Let m^* be the critical candidate.

At this state, the values of v and u will be as follows:

$$u_{opt} = v_{opt} = v_{m^*-1}$$

$$v_{opt} = v_{m^*-1} = \frac{m^*-1}{m^*} v_{m^*} + \frac{1}{m^*} \max \left[\frac{m^*}{N}, v_{m^*} \right]$$

$$v_{opt} = v_{m^*-1} = \frac{m^*-1}{m^*} \cdot \frac{m^*}{N} \left(\frac{1}{m^*} + \dots + \frac{1}{N+1} \right) + \frac{1}{m^*} \cdot \frac{m^*}{N}$$

$$v_{opt} = \frac{m^* - 1}{N} \left(\frac{1}{m^* - 1} + \frac{1}{m^*} + \frac{1}{m^* + 1} + \dots + \frac{1}{N - 1} \right) = u_{opt}$$

Our goal now is to approximate m^* and determine the optimal stopping policy for a sufficiently large N .

From our definitions of v_m and u_m , it can be determined that no secretary who is best so far will be accepted as long as

$$\sum_{i=m}^{N-1} \frac{1}{i} < 1$$

The first viable candidate will be chosen when $m \geq m^*$. It follows that m^* is the solution of:

$$\sum_{i=m^*}^{N-1} \frac{1}{i} \leq 1 < \sum_{i=m^*-1}^{N-1} \frac{1}{i}$$

For the determination of m^* , we observe that

$$\sum_{i=m}^{N-1} \frac{1}{i} < \int_m^N \frac{dx}{x}$$

The approximate solution for m^* is therefore

$$\sum_{i=m}^{N-1} \frac{1}{i} = \int_{m^*}^N \frac{dx}{x} = \log(N) - \log(m) = 1$$

Which gives us:

$$m^* \approx \frac{N}{e}$$

Using the above rule, the algorithm for the optimal strategy is as follows:

- 1) Based on the N candidates in the particular case, derive the approximation for m^* and round it up to the nearest integer.
- 2) Up until and including state $m^* - 1$, interview all the candidates and keep in mind the best candidate that you have interviewed.
- 3) From state m^* onwards, pick the first viable candidate.

Using this approach, we will pick the best candidate with a probability of 36.79 %, .

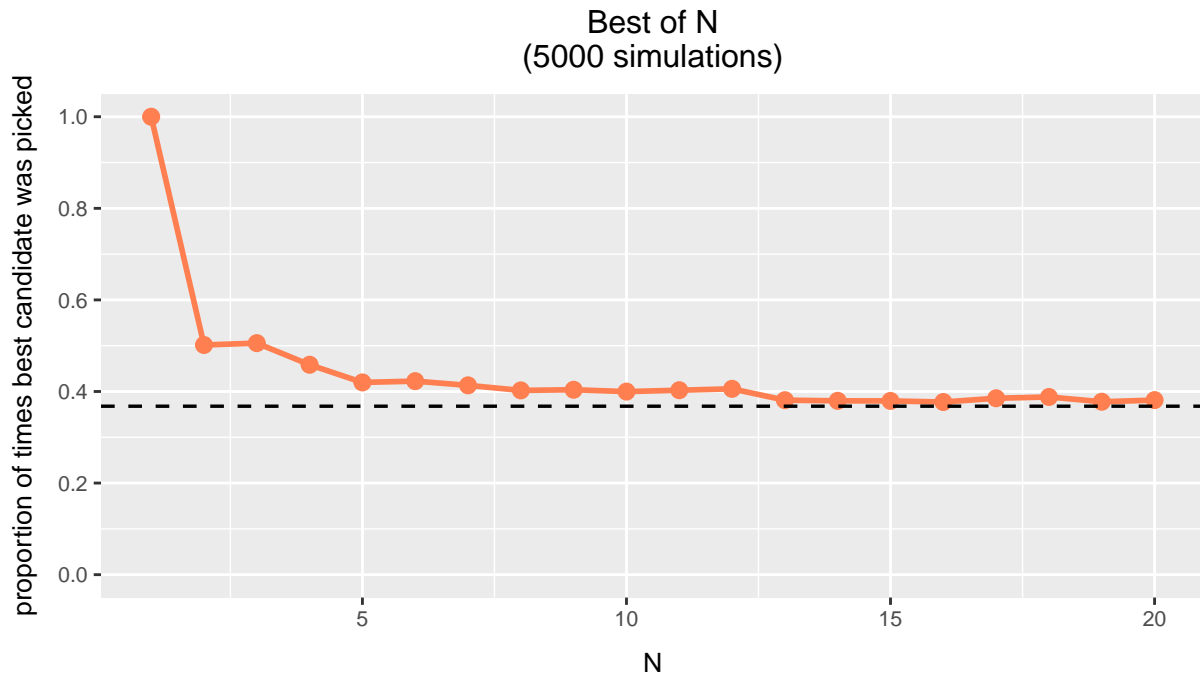


Figure 1. The algorithm was run 5000 times for different values of N . Clearly with $N=1$, the best candidate is always picked. The graph above shows that as N is increased above one, the probability of picking the best candidate decreases to a limit of approximately $1/e$, represented by the dashed black line.

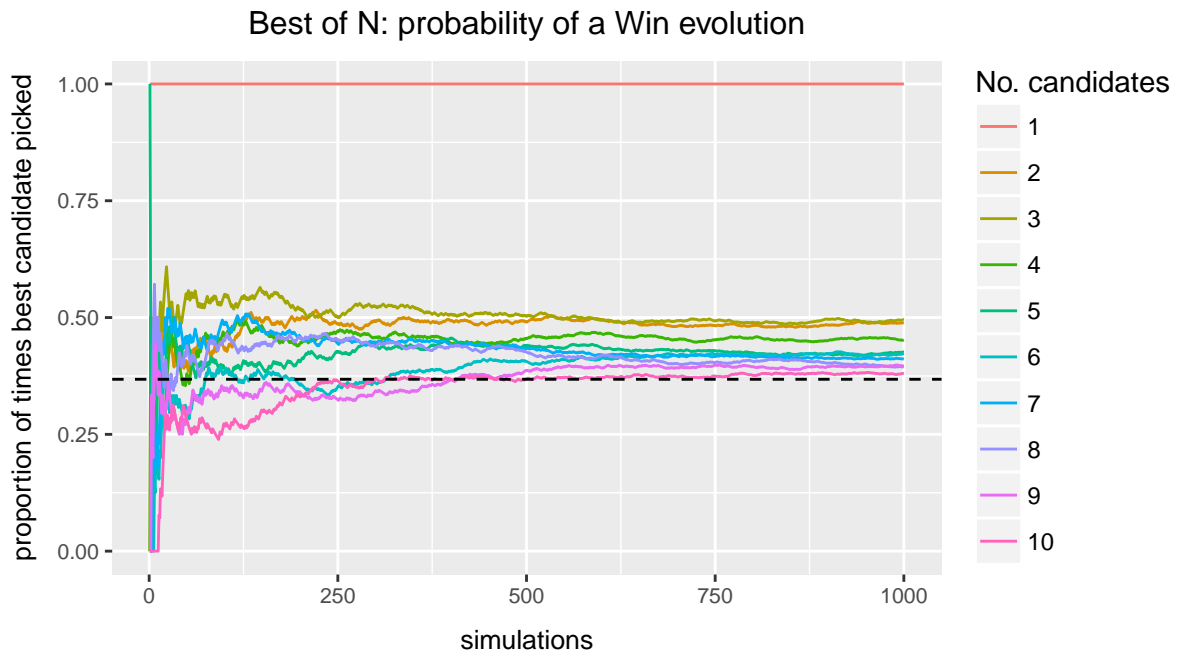


Figure 2. The above graph shows the evolution of the proportion of times you get a Win for different values of N . For few simulations, the results have high variance and the proportion of times you Win may not even be close to the expected values shown in Figure 1. An equilibrium is reached after somewhere around 750 simulations.

2) One from the best k out of N

As for the previous scenario, we let m be our state variable, such that we have observed and rejected $m - 1$ candidates.

Let us define a new finite set of elements A . A is defined as we progress through the interviews. At state m , we will have $m - 1$ elements all ranked according to quality.

Our interviews can be viewed as independent trials, which we will represent with A_m . Where $m = 1, 2, \dots, N$

At each state m , we define a variable i , which is equal to the rank of the m^{th} candidate relative to the previous $m - 1$ candidates.

Now we will determine the pay-off at each time period. If we make the decision to hire at state m , we will receive the pay-off $g_m(x_m)$. If we interview all the candidates and do not hire anyone, we will receive a pay-off of 0.

Let us define two probability values:

$p_m(i)$ is the probability of a candidate of quality i . It is easy to see that this can be calculated in the following manner.

$$p_m(i) = \mathbf{P}\{x_m = i\} = \frac{1}{m}$$

The above value is defined for $i \leq m$

$g_m(i)$ is the probability that stopping at the element which is i^{th} in quality among the first m , we will make a successful choice (a candidate among the best k). We can define it in the following manner.

$$g_m(i) = \sum_{j=i}^k \frac{\binom{i-1}{j-1} \binom{m-i}{N-j}}{\binom{m}{N}}$$

The above value is for $i \leq k$ because for $i > k$ the probability of success is 0.

We start by proving that:

$$g_{m-1}(i) \leq g_m(i) \leq g_{m+1}(i)$$

Note that g_m is computed recursively by the following formula:

$$g_{m-1}(i) = P_m(i)g_m(i+1) + [1 - P_m(i)]g_m(i)$$

where

$$P_m(i) = \mathbf{P}\{x_m \leq i\} = \sum_{j=1}^i p_m(j)$$

Let us now keep in mind that in the above formula if a_j is i^{th} in a_1, a_2, \dots, a_{m-1} , then it is either i^{th} or $i+1^{th}$ among a_1, a_2, \dots, a_m . Let:

- 1) Event O be the event where a_j is i^{th} in quality in a_1, a_2, \dots, a_{m-1} .
- 2) Event I_1 be the event where a_j is i^{th} in quality in a_1, a_2, \dots, a_m .
- 3) Event I_2 be the event where a_j is $i+1^{th}$ in quality in a_1, a_2, \dots, a_m .

We can now define the following probabilities:

$$\begin{aligned}\mathbf{P}\{I_2|O\} &= P_m(i) \\ \mathbf{P}\{I_1|O\} &= 1 - P_m(i)\end{aligned}$$

From our definition, $g_m(i) > 0$ for all $i \leq k$ and $g_m(i) = 0$ for $i > k$. For any m , let

$$g_m(i) \leq g_m(i-1)$$

By combining the above inequality and the recurrence formula, we have

$$g_{m-1}(i) = P_m(i)g_m(i+1) + [1 - P_m(i)]g_m(i) \leq P_m(i-1)g_m(i) + [1 - P_m(i) + 1]g_m(i-1) = g_{m-1}(i-1)$$

Further, we can conclude

$$g_{m-1}(i) = P_m(i)g_m(i+1) + [1 - P_m(i)]g_m(i) \leq g_m(i)$$

Now, for a valid stopping rule, we need to define a monotonically decreasing function $E(m)$ such that it is recommended to stop at the first state m where $E(m) \leq g_m(i)$

Additionally, when we consider two states m_1 and m_2 such that $m_1 \leq m_2$ and $g_{m_1}(i') \leq g_{m_2}(i'')$. Then if we should stop at m_1 for i' , then we should also stop at m_2 for i'' .

When we determine a stopping set $(m, i) \in \Gamma_o$ implies that one must stop when we have a candidate of quality i at m . From our earlier definition with m_1 and m_2 , we can extend this further to state that if $(m_o, i_o) \in \Gamma_o$ then $(m, i) \in \Gamma_o$ for all $m \geq m_o, i \leq i_o$.

We can assert that depending on our values of k and N , there exist numbers

$$1 \leq s_1 \leq s_2 \leq \dots \leq s_k \leq \dots \leq s_n \leq n$$

The optimal stopping rule can be described in the following manner:

- 1) We pass all the elements $a_1, a_2, \dots, a_{s_1-1}$
- 2) We select the first of the elements $a_{s_1}, a_{s_1+1}, \dots, a_{s_2-1}$ which is better than the preceding ones. If no such element exists, then move on to step 3.
- 3) We select the first of the elements $a_{s_2}, a_{s_2+1}, \dots, a_{s_3-1}$ which is surpassed by not more than one element that we have observed before. If no such element exists, move on to the next set of candidates.

We continue this process until we can find an element that satisfies the requirement. For expressing in a general step:

Find the element in $a_{s_n}, a_{s_n+1}, \dots, a_{s_{n+1}-1}$ that is surpassed by a maximum of $n-1$ elements observed before it.

Therefore, we will stop at the first state $m \geq s_i$. This by extension will prove that $s_{k+1} = s_{k+2} = \dots = s_n$

We will now move on to prove the existence of the function $E(m)$ and also study the limit behaviour of the numbers s_1, s_2, \dots, s_k as $n \rightarrow \infty$

For defining a stopping rule, for each of the states m_1, m_2, \dots, m_N , we need to exhibit values i_1, i_2, \dots, i_m for which we are stopping at state m . Therefore can define a set Γ of pairs (m, i) such that we stop at the first m such that $(m, i_m) \in \Gamma$.

There are two possible cases at state m :

- 1) $g_m \geq E(m)$. In this case, stopping at m gives a result no worse than continuing the interviews.
- 2) $g_m < E(m)$. In this case, it is advantageous not to stop and to use the optimal rule.

Therefore if at m , $g_m \geq E(m)$, then it is optimal to stop. If $g_m < E(m)$, then we would gain more by continuing to $m + 1$. Therefore:

$$E(m - 1) = E(\max(E(m), g_m(i_m)))$$

This clearly proves:

$$E(m) \leq E(m - 1)$$

Which implies that $E(m)$ is a monotonically decreasing function.

Since:

- 1) $E(m) \geq p_N(1)g_N(1) = \frac{1}{N}$ for $m < N$
- 2) $g_m(i) = 0$ for $i > k$

For $m < k$ and $m > N$: $(m, i) \notin \Gamma_o$

Hence $s_{k+1} = s_{k+2} = \dots = s_N = N$

Our final objective is to derive the probability of success for the optimal strategy.

We will denote the probability with p_N^k .

Let us consider a function $m_o = m_o(n)$ satisfying the condition $m_o(N)/N \rightarrow \sqrt[k]{\frac{1}{k}}$.

We let $(m, i) \in \Gamma$ if $i \leq k, t > t_o(N)$. The set Γ defines a stopping rule. Relying on the relation $\sum_{i=1}^k g_m(i) = \frac{km}{N}$, we make a successful choice with the probability:

$$\tilde{p}_N^k = \sum_{m=m_o+1}^N \frac{kt_o!(m-k-1)!}{N(t_o-k)!(m-1)!}$$

Hence, for $k > 1$

$$\lim_{N \rightarrow \infty} \tilde{p}_N^k = \frac{k}{k-1} \sqrt[k]{\frac{1}{k}} - \frac{1}{k-1}$$

Since $1 \geq p_N^k \geq \tilde{p}_N^k$ and $p_N^{k+1} \geq p_N^k$, we have $\lim_{k, N \rightarrow \infty} p_N^k = 1$

We set $p_k = \tilde{p}_N^k, \epsilon_i(k) = \lim_{N \rightarrow \infty} \frac{s_i(N, k)}{N}$, if these limits exist.

For any k

$$E(s_k - 1) > g_{s_k-1}(k), E(s_k) \leq g_{s_k}(k)$$

For $m + 1 \geq s_k(N, k)$

$$E(m) = \sum_{s=m+1}^N \frac{kt!(s-k-1)!}{N(m-k)!(s-1)!}$$

$$g_m(k) = \frac{m!(N-k)!}{(m-k)!(N)!}$$

Passing these to the limits, we obtain the following results

For $k = 1$

$$\epsilon_1(1) = e^{-1}$$

For $k > 1$

$$\epsilon_k(k) = \sqrt[k-1]{\frac{k}{2k-1}}$$

Therefore, the algorithm for picking the one of the best k out of N candidates is as follows:

- 1) Determine the initial observation period from state $m = 1$ to s_1 by picking by determining s_1 as:

$$s_1 = \frac{N}{e}$$

- 2) Determine s_k for $k \geq 2$ in by computing:

$$s_k = \left(\sqrt[k-1]{\frac{k}{2k-1}} \right) * N$$

- 3) Moving from state $m = 1$ use that stopping rule that after s_i for $i \geq 1$, you will pick a candidate only if they are better than at least one of the i best seen before. If no such candidate can be picked until N , then you hire the N^{th} candidate.

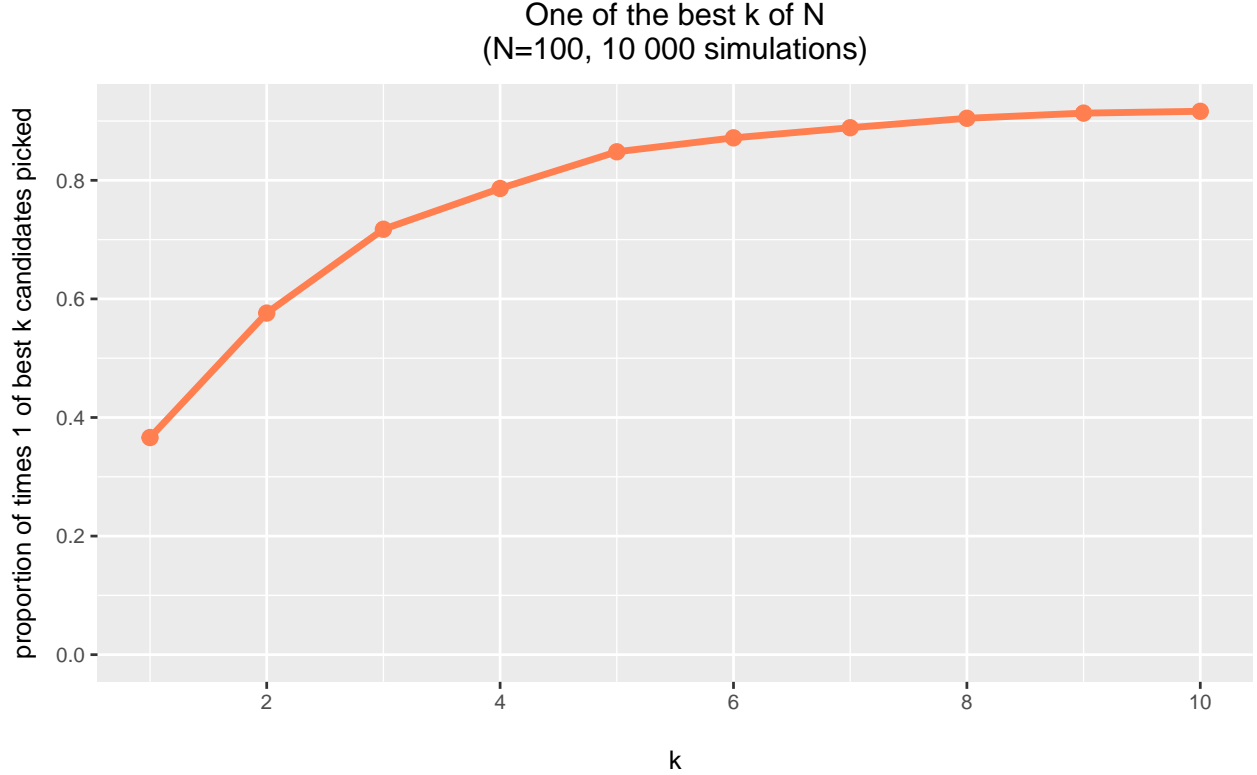


Figure 3. The above graph shows the probability of a Win for picking one of the best k candidates out of the total of 100. For $k = 1$, the probability is the same as that produced from the solution to the standard secretary problem. As k increases, the probability of a Win increases to 1. Note that these values are larger than asymptotic values since \hat{p}_N^k is the expected probability of success as $N \rightarrow \infty$.

3) The best k out of N

In this case, we have N candidates and our objective is to hire the top- k candidates. Under this scenario, a Win is defined as hiring exactly these candidates and we try to maximise the probability of this occurring. There are various approaches to solving this problem. We have decided to implement a fixed-threshold method, as put forward by Girdhar & Dudek (2009) [3].

Under this approach, we use a single threshold in order to optimally select the top k highest scoring candidates. The threshold is the maximum score obtained in the first m candidates. Our objective, is to determine the optimum value of m , depending on k and N . The best value of m for the observation period is determined by maximizing the probability of success $\Phi(m)$.

Let S_i^k be the event that we have successfully selected all k highest scoring candidates with the i^{th} being the last one. Then the probability of success $\Phi^k(m)$ can be written as:

$$\Phi^k(m) = \sum_{i=m+k}^N P(S_i^k)$$

$$\Phi^k(m) = \sum_{i=m+k}^N \frac{k}{N} \frac{m}{i-1} \frac{\binom{i-m-1}{k-1}}{\binom{N}{k-1}}$$

$$\Phi^k(m) = \frac{k}{N} \frac{m}{\binom{N}{k-1}} \sum_{i=m+k+1}^{N-1} \frac{\binom{i-m}{k-1}}{i}$$

Girdhar and Dudek [3] use simulations and heuristics to come up with the following expression for the optimal value of m :

$$\frac{N}{ke^{1/k}}$$

The algorithm for picking the top-k using a fixed threshold is therefore as follows:

- 1) Let $m^*(k) = \lceil \frac{N}{ke^{1/k}} \rceil$ be the observation or training interval.
- 2) Find the threshold $t^* = \max(\{x_1, \dots, x_{m^*-1}\})$
- 3) Select each candidate from the interval $\{m^*, \dots, N\}$ whose score is higher than the threshold until k candidates have been selected or we run out of candidates.

The probability of success using this method as $N \rightarrow \infty$ has been numerically approximated as:

$$\Phi(m) = \frac{1}{ek}$$

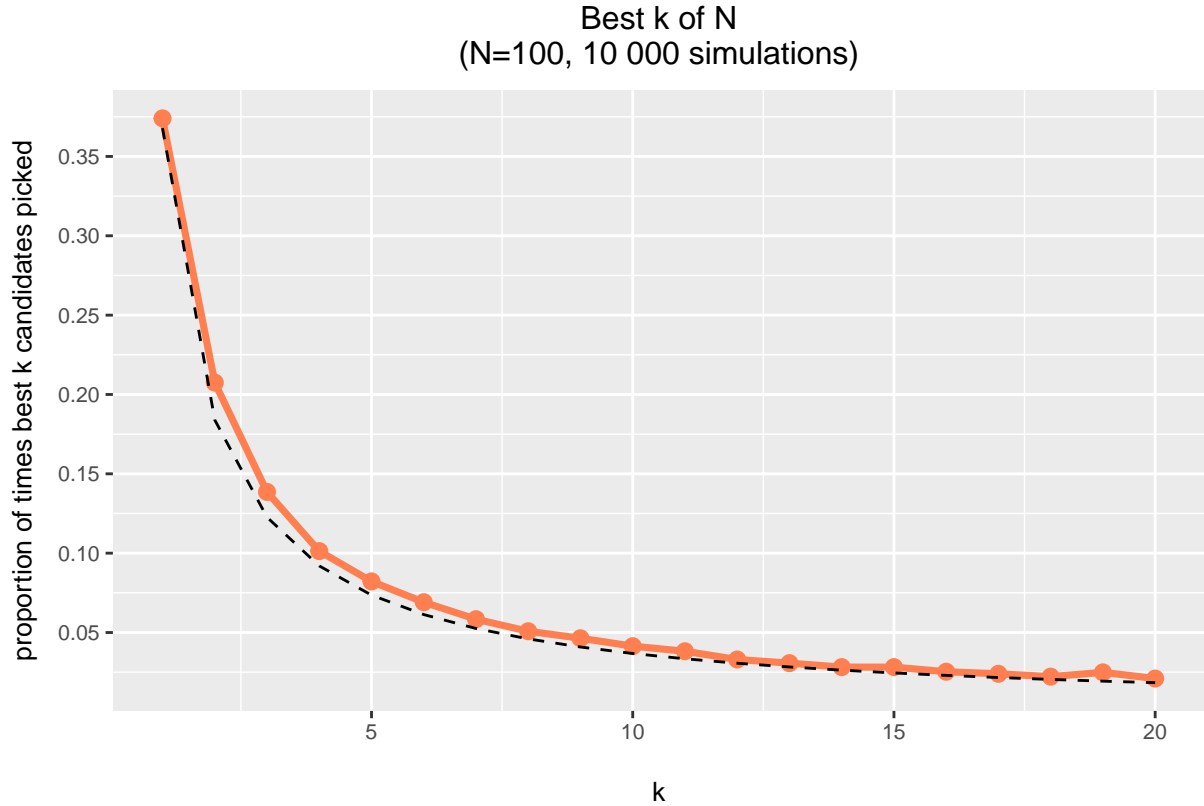


Figure 4. The graph above displays the asymptotic (after 10,000 simulations) probability of a Win for $N=100$ and varying values of k . $\Phi(m)$ is represented by the dashed black line. The graph demonstrates that for a particular N , the probability of a Win decreases exponentially with increasing k .

References

- [1] M.J. Beckmann. Dynamic programming and the Secretary Problem. Computers and Mathematics with Applications, Pergamon Press , Vol. 19, No. 11, pages 25-28, 1990.
- [2] Gusein-Zade, S.M. 1966. The problem of choice and the optimal stopping rule for a sequence of independent trials. Theory of Probability and its Applications 11, 472-476.
- [3] Girdhar, Y. , & Dudek, G. (2009). Optimal Online Data Sampling or How to Hire the Best Secretaries. In CRV 09: Proceedings of the 2009 Canadian Conference on Computer and Robot Vision (pp. 292-298). Kelowna, British Columbia: IEEE Computer Society.