BitTorrent Protocol and a Modified Secretary Problem

J. Cummings and J. Ezaki

Abstract - BitTorrent is a popular file sharing protocol in which a user downloads a file from a large collection of *seeders*, each of which has the file. The user downloads from b of these seeders, but repeatedly drops its worst connection and establishes a new one in search of the best b connections. In this paper we model the BitTorrent protocol using a modified secretary problem, and in doing so find precise stopping times for an arbitrary b. We show that every b gives the same success probability, and that this probability asymptotically approaches $1/e$. We then extend this result to the case of securing a connection with c of the top b peers.

Keywords : BitTorrent protocol; secretary problem

Mathematics Subject Classification (2010) **:** 68R05

1 Introduction

BitTorrent is one of the most popular file sharing protocols available en masse. In fact, it can currently account for 3.35% of worldwide bandwidth usage [12]. In this paper we describe its function and prove new results on the optimization of its "tit for tat" procedure, which we will soon describe. In particular, we explore how this tit for tat system of peer acquisition is modeled by a modified version of the classic secretary problem.

The BitTorrent protocol, pioneered and implemented by Bram Cohen in the early 2000's, is a peer-to-peer (P2P) file sharing protocol. By definition, P2P protocols ditch the traditional client-server model used by the majority of the internet in order to maximize download speeds, hence minimizing download times. In the BitTorrent protocol, a user downloads a .torrent file which contains a list of trackers, hash numbers, block sizes, the number of blocks, and other necessary information about the file to be downloaded.

When a user decides to upload a file using BitTorrent, the file is broken up into many (typically) standard-size blocks of 256 kB; these blocks are the same for all users. The user would then distribute the .torrent file over the internet to those who wish to download it, and also announce the presence to a chosen tracker. The tracker is simply a server whose purpose is to keep track of all the users in a given swarm (the name given to a group of users downloading/uploading a single file). As users arrive, the *initial seeder* (the user who uploaded the file) begins to distribute the file to those requesting it, and will continue to do so indefinitely.

Users trying to download this file are known as *leechers*. They download the .torrent file, and then use the contents contained therein to find the proper tracker and join the swarm. Once in the swarm, a single leecher requests specific blocks from its peers. Those providing this leecher with blocks of the file are known as *seeders*. Once a leecher has a complete block, it too can become a seeder, simultaneously seeding and leeching.

It is at this point where the tit for tat protocol takes place. BitTorrent clients (the programs that run the BitTorrent protocol for users) are designed to reciprocate what they receive from their peers. That is, if a peer uploads at the maximum rate to you, or *unchokes* you, then you upload at your maximum rate to the peer, unchoking that peer. Similarly, if a peer *chokes* you, or stops uploading to you, you reciprocate by choking that peer. It would be easy to imagine a situation in which nobody unchokes anyone else, creating a stalemate. Luckily BitTorrent solves this problem by using what is called an *optimistic unchoke*, where you unchoke a peer from the swarm and hope that the peer reciprocates.

Each peer in the swarm generally maintains a relationship with (a standard) four other peers simultaneously. Ideally, these are the four best peers (those that upload at the highest rate). However, there is often a better peer remaining in the swarm. In an attempt to find that better peer, a user's client will reserve one of its four connections as its optimistic unchoke connection. In other words, three of its connections will be maintained, but every thirty seconds the fourth connection will be choked and another one will be made and unchoked with a random peer from the swarm [2]; for the purposes of this paper we will assume that each new connection is with a peer that has not been seen before (a multitude of other factors determines the extent to which this actually happens in practice). The hope is that this new peer will also unchoke the user. If it turns out one of these optimistically unchoked peers is better than one of the user's three maintained connections, then one of those three connections will become the connection that is choked and transferred to another random peer. It is important to note that it is impossible to know if a random peer from the swarm will be better than a current peer without first establishing a connection. Furthermore, it is impossible to make a new connection without dropping a current one.

In short, the goal of a BitTorrent client is to find the four best peers for a given user, and does so by repeatedly swapping its worst peer for a new one, slowly searching through the typically-large swarm; in 2010, the average swarm size of the most popular torrents was 691.14 [8]. Since there is typically no good way to know whether your current set of clients is optimal, a BitTorrent client will typically never stop swapping. In this paper, though, we analyze when one should stop the search to maximize the chance that they have collected the top four.

1.1 Other Considerations

Part of the BitTorrent protocol is the selection of which blocks to request from peers first. The standard protocol is called "rarest first," whereby clients request from their peers the least prevalent block within their connected peer list. In other words, the goal is to minimize the standard deviation of all blocks among the swarm in order to maximize the probability that a given peer will have the block needed by a particular client.

While this is an issue, it will not be considered in the scope of this paper.

2 The Classic Secretary Problem and its Generalizations

The rules for the classic secretary problem are as follows [3]:

1. A collection of applicants apply for one secretarial position.
2. There are n applicants and n is known.
3. The applicants can be linearly ranked from best to worst without ties.
4. You interview the applicants in a random order, with each of the $n!$ orderings being equally likely.
5. At the conclusion of each interview, you must either offer the applicant the job and end the search, or reject the candidate and call in the next.
6. The decision to accept or reject an applicant must be based only on the relative ranks of the applicants interviewed thus far.
7. Once a candidate is rejected, they cannot later be recalled.
8. The objective is to select the best applicant.

So, the goal of this problem, and the definition of success, is to select the objectively best applicant. The challenge is, of course, to identify the proper applicant. As it turns out, the best way to do this is to rely on probability.

To begin, consider the set of applicants $\{x_1, x_2, \ldots, x_n\}$ where each index represents the order in which the applicant is interviewed. As each applicant is interviewed, the interviewer has a relative ranking of the current applicant as compared to the previous candidates. So, if applicant x_j receives the best relative ranking (of the previously interviewed applicants), then the probability that applicant x_j is the objectively best applicant is the probability that the best candidate is among the first j applicants. That is, the probability applicant x_j is the best is simply j/n. Note that as $j \to n$ the probability applicant x_j is the objectively best applicant, given that they are the relatively best applicant, approaches 1. As such, as $j \to n$, the probability that the objectively best applicant is after applicant x_j significantly decreases, meaning that the interviewer risks more and more by *not* selecting applicant x_j.

As such, there must be a threshold T such that maximizes the probability that the next relatively best applicant is the objectively best candidate. That is, the optimal strategy is for the interviewer to reject the first $T-1$ applicants and then select the next relatively best applicant. It turns out that the optimal threshold value is $T = \lceil n/e \rceil$. In addition, the probability of success by using this strategy is approximately $e^{-1} \approx 37\%$.

These ratios arrive from the following formula. Let $p_n^1(T)$ be the probability of success with threshold T given n applicants. Then,

$$p_n^1(T) = \begin{cases} \frac{1}{n} & , T = 1 \\ \frac{1}{n}\displaystyle\sum_{j=T}^{n} \frac{T-1}{j-1} & , T \in \{2, 3, ..., n\}. \end{cases} \tag{1}$$

Also, if T_n is the optimal threshold (maximizing $p_n^1(T)$) for a particular n, then [3]

$$\lim_{n\to\infty} \frac{T_n}{n} = \frac{1}{e}.$$

Of note in (1) is the term $\frac{T-1}{j-1}$. This term measures the probability that, given the best applicant is x_j, that the best among the first $j-1$ applicants (the best one which precedes x_j) was one of the first $T-1$ applicants (implying that this next-best was not selected, and also that nothing before x_j will end the search). That is, the strategy with threshold T will successfully select the best candidate if and only if this condition is met.

Generalizations

There are many different generalizations to the classic secretary problem that have been studied. Smith [17] studied the variant in which a secretary, when accepted, has a probability of not being available and will therefore must be passed over. Yang [21] allowed the manager to attempt to hire an applicant that they have already dismissed; however, with a certain probability this candidate will no longer be available. Petrucelli [13] studied what happens when these two situations happen simultaneously.

Rubin and Samuels [15] considered the "finite memory" variant, where the manager may only remember the previous candidate's abilities.

Gianini and Samuels [5] introduced an infinite version of the problem where infinitely many rankable candidates (rank 1 is the best) arrive at times which are i.i.d., uniform on $(0, 1)$. The goal is then to minimize the mean of a prescribed increasing function. Gianini [4] and Lorenzen [10] showed that this problem is the limit of corresponding finite problems.

Other generalizations involve the expected rank of the chosen candidate [1], limited recall of previous candidates [7], an oberservation cost for each additional interview [9], and a game version in which one player gets some control on the order in which the applicants are interviewed [6].

A final variant that has been studied in which a single candidate is hired was considered by Smith and Deely [18]. In this paper, they allow the manager to at any point hire one of the last m applicants. This in some ways mirrors our own situation in which we will allow the "manager" to maintain a "pool" of potential candidates, who have been interviewed but not yet fired.

In our paper, though, we study the situation where one wants to select more than one "secretary." This has also been studied before, but in the literature the set-up is again that applicants are interviewed one at a time, and at the end of the interview must either be hired or not hired. In Nikolaev [11], the task was to hire the best two applicants; in Tamaki [19], it was to hire at least one of the top two. In Rose [14], the problem of filling two positions, P_1 and P_2, was discussed. Here, applicants are considered one at a time, and at the end of the interview must either be offered the top position or the second position, or be dismissed without an offer. Tamaki [20] then extended this to case where m positions need to be staffed, in order. But still, they enter sequentially, and one at a time are offered a position or dismissed.

In the below we ask what happens when you want to hire the top b secretaries, in no particular order, and you may keep a pool of b candidates at a time which you have interviewed but have not yet been fired.

3 Applying the Secretary Problem to BitTorrent Protocol

In the case of BitTorrent protocol, we are not interested in the single best peer, but in the four best peers. In addition, there will continuously be three active connections and a fourth used for optimistic unchoking. We term these four connections as the "bank." We will now examine the "standard" BitTorrent protocol bank size of four and determine the probability of success in selecting the top four peers. We will then generalize the bank size, and finally generalize the success conditions.

3.1 The Setup

First, some definitions. Let:

- n be the total number of peers in the swarm.
- P be the set of all peers, and p_i be the i^{th} objectively ranked peer, for $i = 1, 2, ..., n$, where a lower value means a better rank.
- x_j be the peer that arrives in location j, regardless of objective rank, for $j = 1, 2, ..., n$.
- b be the bank size.
- K be the set of the best b peers, and k_i be its elements for $i = 1, 2, ..., b$ such that any k_i is the i^{th} element of K to be selected, not necessarily its objective rank. Note that the objective rank for all $k \in K$ is better than $p_i \in P$ for $i = (b+1), (b+2), ..., n$. Furthermore, note that k_b is the final peer from K to be selected.
- T be the threshold.
- ℓ_0 be the b^{th} best peer that arrives prior to k_b.

3.2 The Rules

In order to apply the secretary problem solution to BitTorrent, we must first translate the rules of the secretary problem to be in terms of BitTorrent protocol.

1. There are four connections available which constitute a "bank." That is, at all times, other than transition periods, four connections must be maintained.

2. There are n peers in the swarm; n is known and $n \geq 4$.

3. It is assumed that you can rank the peers linearly from best to worst without ties.

4. Connections with peers are tested sequentially in a random order with each of the $n!$ orderings being equally likely.

5. As each connection is tested, you must either accept it as the last of the top four connections and end the search, accept it as one of the top three current connections and drop your current worst connection, or drop it as your worst connection and attempt the next connection, if any exist.

6. The decision to accept or reject a connection must be based only on the relative ranks of the connections tested so far.

7. A connection, once rejected, cannot later be re-established.

8. The objective is to select the four best peers.

Clearly the setup for this problem is very similar to the secretary problem. However, there are some important subtleties in the procedure to note.

Notice that, regardless of their objective ranks, the first four peers to arrive are the first four peers in the bank. Then, until the x_{T-1} arrives, continue to drop the worst connection in the bank and establish a new connection. Once the j^{th} connection is made, for $j \geq T$, if peer x_j is the b^{th} best peer to arrive, accept x_j as the final connection in the bank and stop the search.

3.3 Win Conditions

In order for this process to succeed—that is, for all $k \in K$ to be selected—the three following conditions must be met. The converse also holds too: if these conditions are met, then the process will succeed.

1. Exactly $(b-1)$ peers from K must arrive in the first $(T-1)$ peers.

2. k_b arrives in position j, for some $j \geq T$. Say, $k_b = x_j$.

3. ℓ_0 must arrive in the first $(T-1)$ peers.

We justify these now, with a particular emphasis on the case $b = 4$.

—Condition 1—

First, exactly $(b-1)$ peers from K must arrive in the first $(T-1)$ peers. For the purpose of this demonstration, let $b = 4$. Clearly the process fails if all b peers from K arrive before the threshold, because then the worst such connection will be dropped before the threshold and will never be picked back up again. So, consider the case where only 2 members of K arrive before the threshold, T:

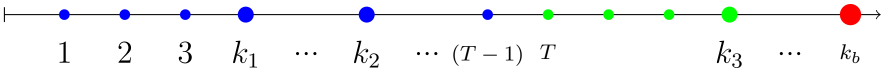

Figure 1: Only 2 members of K arrive before T.

The blue dots represent all peers that arrive prior to the threshold, the best 4 of which (because $b = 4$) will remain in the bank into the green side. The green dots represent the peers considered as candidates for stopping the search. Recall that the process will stop once the relatively best peer (as compared to those that have been discarded) is found after the threshold. Thus, in the case where only 2 members of K arrive prior to the threshold, it will never be the case that k_b is found. Similarly, this can easily be extended to the cases of 1 or 0 members of K arriving prior to the threshold.

Clearly, then, if the process succeeds, then 3 members of K arrived before the threshold. More generally, if the process succeeds in finding the top b peers, then $(b-1)$ members of K arrived before the threshold.

—Condition 2—

Second, k_b must arrive in position j, for some $j \geq T$. This condition is a corollary from Condition 1, but worthwhile to point out.

—Condition 3—

The third and final condition for success is that, with an arbitrary value of T, and $k_b = j$, for $j \geq T$, it must be the case that ℓ_0 arrives before T.

The easiest way to understand the proof to this condition is in the classic case when $b = 1$. Consider the case where ℓ_0 arrives after T.

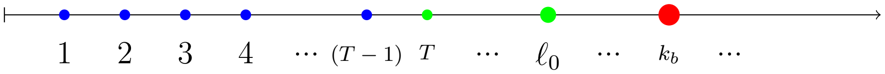

Figure 2: The case where ℓ_0 arrives after T.

By the definition of ℓ_0, it is the next relatively best peer that arrives before k_b. In other words, k_b is the only relatively better peer than ℓ_0 that arrives after T. However, in the above case, the process will stop once ℓ_0 arrives because it is better than all of the previous peers and arrives after the threshold. Therefore, if ℓ_0 arrives after T, then the process will fail. Notice the difference when ℓ_0 arrives before T.

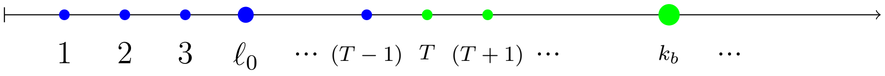

Figure 3: The case where ℓ_0 arrives before T.

In this case, as the only peer better than ℓ_0 is k_b, and ℓ_0 is automatically discarded (because it is before the threshold), then k_b becomes the only peer for which the process will stop.

Now, for cases where $b > 1$, assuming Condition 1 and 2 are also met, it can easily be seen that Condition 3 must also be met. So, indeed, Condition 1, 2, and 3 are necessary and sufficient for success.

3.4 The b=4 Case

Proposition 3.1 *For $4 < T \leq n$, we have*

$$p_n^4(T) = \frac{\binom{T-1}{3}}{\binom{n}{4}} \cdot \sum_{j=T}^{n} \frac{T-4}{j-4}.$$

Proof. Observe that

$$\begin{aligned} p_n^4(T) &= P(C_1, C_2, C_3) \\ &= P(C_1)P(C_2|C_1)P(C_3|C_1, C_2). \end{aligned}$$

We proceed by determining these probabilities.

Probability of Condition 1. The event space in which only k_1, k_2, k_3 arrive in the first $T-1$ peers, meaning k_b must arrive after, can be determined as follows.

- Choose the three locations in the first $T-1$: $\binom{T-1}{3}$

  Note this does not guarantee *exactly* 3 of 4 members of K arrive prior to T.

- Choose the location for k_b such that $k_b = x_j$ for $j \geq T$: $\binom{n-T+1}{1}$

- Possible orders of K: $4!$

- Possible orders of the remaining peers: $(n-4)!$

Let C_1 be the case that three of the top four peers are among the first $T-1$ peers. Then, with a sample space of $n!$, the probability of C_1 occurring is:

$$\begin{aligned} P(C_1) &= \frac{\binom{T-1}{3} \cdot \binom{n-T+1}{1} \cdot 4! \cdot (n-4)!}{n!} \\ &= \frac{\binom{T-1}{3} \cdot \binom{n-T+1}{1}}{\binom{n}{4}}. \end{aligned}$$

Probability of Condition 2, given Condition 1. Note that in a random ordering, $\frac{4}{n}$ is the probability that $x_j \in K$ for $j = 1, 2, ..., n$. This term strongly correlates with the $\frac{1}{n}$ term in (1). However, unlike the secretary problem where the best applicant may be anywhere, in this problem, recall Condition 1 for success is that three of the top four are among the first $T-1$ peers. As a consequence of this, it must be the case that k_b arrives after the threshold. As such, given Condition 1, the probability that any peer $x_j = k_b$, for $j \geq T$ is given by

$$P(k_b = x_j) = \frac{1}{n-(T-1)}.$$

Probability of Condition 3, given Condition 1 and 2. Then, to calculate the probability of the third condition for success, that ℓ_0 arrives in the first $(T-1)$ peers, note there are exactly $(T-1)-3 = T-4$ locations which we would consider successful placement of ℓ_0, but a total of $(j-1)-3 = j-4$ total possible locations, where $k_b = x_j$. So, assuming $k_b = x_j$,

$$P(\ell_0 \text{ arrives in first } (T-1)|k_b = x_j) = \frac{(T-1)-3}{(j-1)-3} \text{ for } j \geq T.$$

The arrival of k_b is definitely after the first $(T-1)$ peers as dictated by Conditions 1 and 2, but its actual arrival time is otherwise unknown. As such, it is necessary to calculate the probability ℓ_0 arrives in the first $(T-1)$ peers for each possible arrival time of k_b.

$$\begin{aligned} P(\text{Finding } k_b \text{ using } T) &= \sum_{j=T}^{n} P(k_b = x_j) \cdot P(\ell_0 \text{ arrives in first } (T-1)|k_b = x_j) \\ &= \sum_{j=T}^{n} \frac{1}{n-T+1} \cdot \frac{(T-1)-3}{(j-1)-3}. \end{aligned}$$

Total Probability of Success. Let $p_n^4(T)$ be the probability of success with threshold T given n peers. Note that the probability of success is only dependent on the probability that Conditions 1, 2, and 3, denoted C_1, C_2, C_3, respectively, are met.

$$\begin{aligned}
p_n^4(T) &= P(C_1, C_2, C_3) \\
&= P(C_1)P(C_2|C_1)P(C_3|C_1, C_2) \\
&= \left[\frac{\binom{T-1}{3} \cdot \binom{n-T+1}{1}}{\binom{n}{4}}\right] \cdot \sum_{j=T}^{n} \frac{1}{n-T+1} \cdot \frac{(T-1)-3}{(j-1)-3} \\
&= \frac{\binom{T-1}{3}}{\binom{n}{4}} \cdot \sum_{j=T}^{n} \frac{T-4}{j-4}\ , \text{ for } 4 < T \leq n.
\end{aligned}$$

□

Observe that $p_n^4(T) = 0$, for $n < 4$ or $T \leq 4$, and that success is guaranteed for $n = 4$. Figure 4 shows the plot of $p_{10^4}^4(T)$. Note that the highest probability of success occurs at $T = 7789$ with a probability $p_{10^4}^4(7789) \approx 36.8\%$.

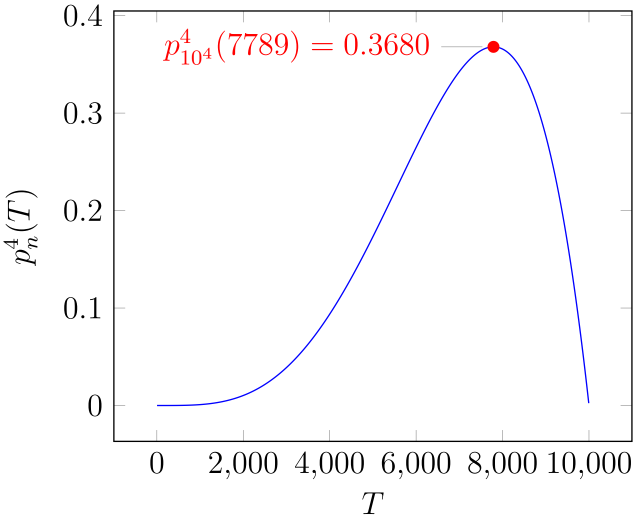

Figure 4: A graph of threshold values versus their probability of success for $n = 10,000$ and $b = 4$.

Interestingly, while the optimum ratio of T/n is much larger for $b = 4$ than in the original secretary problem, the maximum probability of success is the same.

Peers n	Optimal threshold T_n	Ratio T_n/n	Optimal probability $p_n^4(T_n)$
10	9	0.9000	0.4889
20	17	0.8500	0.4170
40	32	0.8000	0.3899
80	64	0.8000	0.3786
160	126	0.7875	0.3732
320	250	0.7813	0.3705
640	500	0.7813	0.3692
1280	998	0.7800	0.3685
2560	1995	0.7792	0.3682
10000	7789	0.7789	0.3680

Table 1: Probabilities of success for certain n with $b = 4$.

As can be seen from Table 1, the ratio of the optimal threshold to the number of peers is given by $T_n/n \approx 77.89\%$, for large n and $b = 4$. In addition, given T_n and a large n, the optimal probability is given by $p_n^4(T_n) \approx e^{-1} \approx 36.79\%$.

Theorem 3.2 *The optimal stopping time for the $b = 4$ case approaches $T = \frac{n}{\sqrt[4]{e}}$. The probability of success with this T approaches $1/e$.*

Proof. By Proposition 3.1, we wish to optimize

$$p_n^4(T) = \frac{\binom{T-1}{3}}{\binom{n}{4}} \cdot \sum_{j=T}^{n} \frac{T-4}{j-4}$$

as $n \to \infty$. As $n \to \infty$, if the optimal T does not also diverge to infinity, then clearly by the above formula this optimal T gives probabilities approaching zero infinitely often. Now assume we do have $T \to \infty$ as $n \to \infty$. As we show below, this gives a higher probability, and therefore the optimal probability in the below case will be the optimal probability for $p_n^4(T)$.

Observe that

$$\begin{aligned}\frac{\binom{T-1}{3}}{\binom{n}{4}} \cdot \sum_{j=T}^{n} \frac{T-4}{j-4} &\sim \frac{T^3/(3!)}{n^4/(4!)} \cdot \sum_{j=T}^{n} \frac{T-4}{j-4}\\ &\sim \frac{4T^4}{n^4} \cdot \sum_{j=T}^{n} \frac{1}{j-4}\\ &\sim \frac{4T^4}{n^4} \cdot \sum_{j=T}^{n} \frac{1}{j}.\end{aligned}$$

By setting $x = T/n$, $t = j/n$ and dt for $1/n$, as n grows this sum is approximated by the integral

$$\sim 4x^4 \cdot \int_{xn}^{n} \frac{1}{t}\, dt,$$

which after applying the substitution $u = t/n$, we reach

$$\begin{aligned} &\sim 4x^4 \cdot \int_x^1 \frac{1}{u}\, du \\ &= -4x^4 \ln(x). \end{aligned}$$

To optimize $p_n^4(T)$ we take its derivative, $p_n'(T) = -16x^3 \ln(x) - 4x^3$. This function is zero when $\ln(x) = -1/4$, which means that $p_n^4(T)$ is maximed when $x = 1/\sqrt[4]{e}$. Recall that $x = T/n$, and so the optimal threshold occurs at $T = n/\sqrt[4]{e}$.

To find the asymptotic likelihood of success at this T, first recall that since the harmonic series grows with the natural logarithm (see [16]),

$$\frac{4T^4}{n^4} \cdot \sum_{j=T}^{n} \frac{1}{j} \sim \frac{4T^4}{n^4} \cdot (\ln(n) - \ln(T)) = \frac{4T^4}{n^4} \cdot \ln\left(\frac{n}{T}\right).$$

Plugging in our choice of T,

$$\frac{4(n/\sqrt[4]{e})^4}{n^4} \cdot \ln\left(\frac{n}{(n/\sqrt[4]{e})}\right) = \frac{4}{e} \cdot \ln(\sqrt[4]{e}) = \frac{1}{e}.$$

As this T gives an asymptotically positive probability, T must indeed grow with n, and so $T = n/\sqrt[4]{e}$ is indeed the optimal threshold. □

3.5 General Case

Now that the case of $b = 4$ has been investigated, we move to a general b. We again begin by computing the precise probability function.

Proposition 3.3 *For $b < T \leq n$, we have*

$$p_n^b(T) = \frac{\binom{T-1}{b-1}}{\binom{n}{b}} \cdot \sum_{j=T}^{n} \frac{T-b}{j-b}.$$

Proof. Again appealing to the three conditions which characterize success, observe that

$$\begin{aligned} p_n^b(T) &= P(C_1, C_2, C_3) \\ &= P(C_1)P(C_2|C_1)P(C_3|C_1, C_2). \end{aligned}$$

We proceed by determining these probabilities.

Probability of Condition 1. In the general case, Condition 1 is that $(b-1)$ members of K arrive before the threshold, T. In other words, $k_1, k_2, ..., k_{b-1}$ arrive in the first $T-1$ peers, see Section 3.3 for clarification. So, the probability of C_1 is as follows.

$$\begin{aligned} P(C_1) &= \frac{\binom{T-1}{b-1} \cdot \binom{n-(T-1)}{1} \cdot (b)! \cdot (n-b)!}{n!} \\ &= \frac{\binom{T-1}{b-1} \cdot \binom{n-(T-1)}{1}}{\binom{n}{b}}. \end{aligned} \tag{2}$$

Probability of Condition 2, given Condition 1. Recall from Section 3.3 that Condition 2 requires k_b to arrive after the threshold. As it turns out, because this fact is highly dependent on Condition 1, that $P(C_2)$ remains the same.

$$\begin{aligned} P(C_2|C_1) &= P(x_j = k_b|(b-1) \text{ of } K \text{ arrive before } T) \\ &= \frac{1}{(n-T+1)} \text{ , for } j \geq T. \end{aligned} \tag{3}$$

Probability of Condition 3, given Conditions 1 and 2. The method for achieving an equation to find the probability of Condition 3 for any $b \in \mathbb{N}$ is much the same as that of the case $b = 4$. As such, ℓ_0 must arrive in the first $(T-1)$ peers, excluding the $(b-1)$ known to be in the first $(T-1)$, in accordance with Condition 1. Furthermore, as the exact value of j, for $x_j = k_b$, is not known, the probability of ℓ_0 arriving in the first $T-1$ peers must be summed over all possible locations of k_b.

$$P(C_2|C_1)P(C_3|C_1, C_2) = \sum_{j=T}^{n} \frac{1}{(n-(T-1))} \cdot \frac{(T-1)-(b-1)}{(j-1)-(b-1)}. \tag{4}$$

Total Probability of Success. Let $p_n^b(T)$ be the probability of success with threshold T given n peers, and note that the probability of success is only dependent on the probability that Conditions 1, 2, and 3 are met.

$$\begin{aligned} p_n^b(T) &= P(C_1, C_2, C_3) \\ &= P(C_1)P(C_2|C_1)P(C_3|C_1, C_2) \\ &= \frac{\binom{T-1}{b-1} \cdot \binom{n-(T-1)}{1}}{\binom{n}{b}} \cdot \sum_{j=T}^{n} \frac{1}{(n-(T-1))} \cdot \frac{(T-1)-(b-1)}{(j-1)-(b-1)} \\ &= \frac{\binom{T-1}{b-1}}{\binom{n}{b}} \cdot \sum_{j=T}^{n} \frac{T-b}{j-b} \text{ , for } b < T \leq n. \end{aligned} \tag{5}$$

□

3.6 Generalized Secretary Theorem

Theorem 3.4 *The optimal stopping time for a general* b *approaches* $T = \frac{n}{\sqrt[b]{e}}$. *The probability of success with this* T *approaches* $1/e$.

Proof. From our earlier work, we wish to optimize

$$p_n^b(T) = \frac{\binom{T-1}{b-1}}{\binom{n}{b}} \cdot \sum_{j=T}^{n} \frac{T-b}{j-b}$$

as $n \to \infty$. Recall from the proof of Theorem 1 that if the optimal T does not also diverge to infinity as n does, that we get a probability tending toward zero. Thus, we take $T \to \infty$ as $n \to \infty$. Observe that

$$\begin{aligned}
\frac{\binom{T-1}{b-1}}{\binom{n}{b}} \cdot \sum_{j=T}^{n} \frac{T-b}{j-b} &\sim \frac{T^{b-1}/((b-1)!)}{n^b/(b!)} \cdot \sum_{j=T}^{n} \frac{T-b}{j-b} \\
&\sim \frac{bT^b}{n^b} \cdot \sum_{j=T}^{n} \frac{1}{j-b} \\
&\sim \frac{bT^b}{n^b} \cdot \sum_{j=T}^{n} \frac{1}{j}.
\end{aligned}$$

By setting $x = T/n$, $t = j/n$ and dt for $1/n$, as n grows this sum is approximated by the integral

$$\sim bx^b \cdot \int_{xn}^{n} \frac{1}{t}\, dt,$$

which after applying the substitution $u = t/n$, we reach

$$\begin{aligned}
&\sim bx^b \cdot \int_{x}^{1} \frac{1}{u}\, du \\
&= -bx^b \ln(x).
\end{aligned}$$

To optimize $p_n^b(T)$ we take its derivative, $p_n'(T) = -b^2 x^{b-1} \ln(x) - bx^{b-1}$. This function is zero when $\ln(x) = -1/b$, which means that $p_n^b(T)$ is maximized when $x = 1/\sqrt[b]{e}$. Recall that $x = T/n$, and so the optimal threshold occurs at $T = n/\sqrt[b]{e}$.

To find the asymptotic likelihood of success at this T, first note that since the harmonic series grows with the natural log (see [16]),

$$\frac{bT^b}{n^b} \cdot \sum_{j=T}^{n} \frac{1}{j} \sim \frac{bT^b}{n^b} \cdot (\ln(n) - \ln(T)) = \frac{bT^b}{n^b} \cdot \ln\left(\frac{n}{T}\right).$$

Plugging in our choice of T,

$$\frac{b(n/\sqrt[b]{e})^b}{n^b} \cdot \ln\left(\frac{n}{(n/\sqrt[b]{e})}\right) = \frac{b}{e} \cdot \ln(\sqrt[b]{e}) = \frac{1}{e}.$$

As this T gives an asymptotically positive probability, T must indeed grow with n, and so $T = n/\sqrt[b]{e}$ is indeed the optimal threshold. □

4 Data Analysis

Clearly this mathematical model would only be sound if the case of $b = 1$, which corresponds exactly to the classic secretary problem, produces the same outcome. Indeed this is the case as can be seen in Table 2.

Bank Size b	Optimal threshold ratio $\frac{T_n}{n}$	Probability of success $p_n^b(T_n)$
1	0.3680	0.36791
2	0.6066	0.36793
3	0.7167	0.36794
4	0.7789	0.36796
5	0.8189	0.36798
6	0.8466	0.36800
7	0.8670	0.36802
8	0.8826	0.36804
9	0.8950	0.36805
10	0.9050	0.36807
50	0.9803	0.36881

Table 2: Probability of success for various b with $n = 10,000$.

Notice in Table 2 that as $b \to n$, the probability of success tends toward $e^{-1} \approx 37\%$. However, also note that the ratio of the optimal threshold to n approaches 1 as the bank size increases.

Furthermore, the optimal stopping times also approach our asymptotic theoretical result. Observe in Table 2 that the optimal threshold for $b = 4$ is $T = (10,000)(0.7789) = 7789$ and $\lceil \frac{10,000}{\sqrt[4]{e}} \rceil = 7789$. For comparison, we can see from Table 1 that the optimal threshold for $n = 2560$ is $T = 1995$ and $\lceil \frac{2560}{\sqrt[4]{e}} \rceil = 1994$. So, our experimental results show that even by $n = 10,000$ the optimal threshold is only 0.01% off from the asymptotic limit.

5 Further Extensions

So far we have only considered the case in which we want to secure connections with all of the top b peers. However, it may be of use to consider the possibility of only securing a certain number, c, of the top b peers, for $c < b$. The calculation would be largely the same as before.

5.1 Probability of Selecting Exactly c Members of K

The same rules as the previous situation apply, however the conditions must be altered in order to fit this more generalized case.

5.1.1 Condition 1 and its Probability

Recall from Section 3.3 that Condition 1 requires $(b-1)$ members of K to arrive prior to the threshold. In a similar fashion, in order to select c members of K, $(c-1)$ members must arrive prior to the threshold. Thus, the probability of Condition 1 is as follows:

$$P(C_1) = \frac{\binom{T-1}{c-1} \cdot \binom{n-T+1}{b-c+1} \cdot (b!) \cdot (n-b)!}{n!} = \frac{\binom{T-1}{c-1} \cdot \binom{n-T+1}{b-c+1}}{\binom{n}{b}}. \tag{6}$$

5.1.2 Condition 2 and its Probability Given Condition 1

In this case, Condition 2 is still that $x_j = k_c$ for some $j \geq T$, but note that we are looking for the c^{th} member of K rather than the b^{th}. The probability of Condition 2 in this case is also quite similar to Equation 3. However, instead of only having one remaining member of K after the threshold, there are $b - (c - 1)$. Thus, we have the following probability that $x_j = k_c$:

$$P(C_2|C_1) = \frac{b - (c-1)}{(n - T + 1)}, \quad \text{for } j \geq T. \tag{7}$$

5.1.3 Condition 3 and its Probability Given Condition 1 and 2

Indeed the idea behind Condition 3 for this extended case remains the same. In terms of probability, we must change the $(b - 1)$ term of Equation 4 to $(c - 1)$. We must do this because $(c - 1)$ locations have already been chosen by Condition 1. Thus, the probability of C_3 is:

$$\begin{aligned} P(C_2|C_1)P(C_3|C_1, C_2) &= \sum_{j=T}^{n} \frac{b - (c-1)}{(n - (T-1))} \cdot \frac{(T-1) - (c-1)}{(j-1) - (c-1)} \\ &= \sum_{j=T}^{n} \frac{b - (c-1)}{(n - (T-1))} \cdot \frac{T - c}{j - c}. \end{aligned} \tag{8}$$

5.1.4 Combining the Three Conditions

Our final equation is:

$$\begin{aligned} p_n^b(T) &= P(C_1, C_2, C_3) \\ &= P(C_1)P(C_2|C_1)P(C_3|C_1, C_2) \\ &= \frac{\binom{T-1}{c-1} \cdot \binom{n-(T-1)}{b-c+1}}{\binom{n}{b}} \cdot \sum_{j=T}^{n} \frac{b - (c-1)}{(n - (T-1))} \cdot \frac{T - c}{j - c}. \end{aligned} \tag{9}$$

Note that when $c = b$, Equation 9 is the same as Equation 5, consistent with the previous model. Table 3 shows optimal threshold values and maximum probabilities using $b = 4$ and $c = 3$.

Peers n	Optimal threshold T_n	Ratio T_n/n	Optimal probability $p_n^b(T_n)$
10	7	0.7000	0.6510
20	12	0.6000	0.5899
40	23	0.5750	0.5619
80	46	0.5750	0.5498
160	91	0.5688	0.5436
320	180	0.5625	0.5406
640	360	0.5625	0.5391
1280	718	0.5609	0.5384
2560	1435	0.5605	0.5380
10000	5604	0.5604	0.5377

Table 3: Probabilities of success for certain n with $b = 4$ and $c = 3$.

Comparing Table 1 and Table 3, we can see that relaxing the conditions under which we consider the search a win requires much less searching and results in a much higher probability of success.

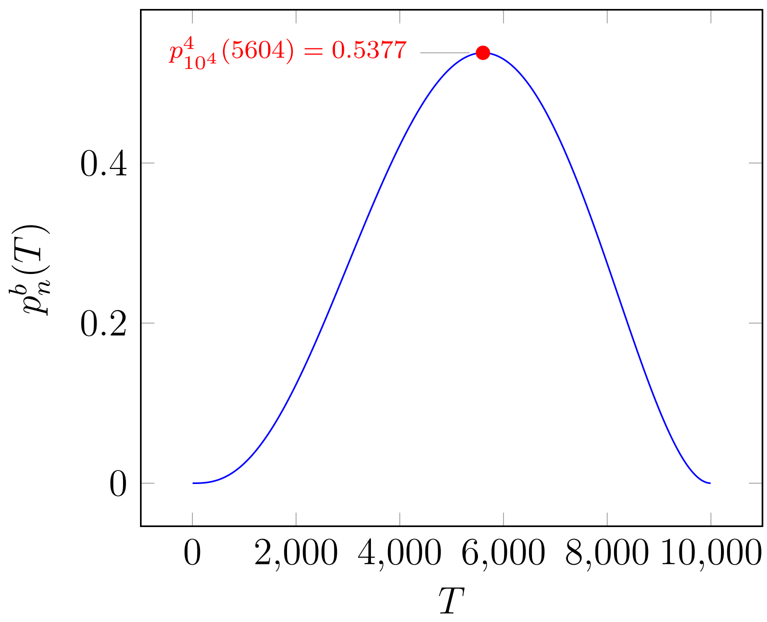

Figure 5: A graph of threshold values versus their probability of success for $n = 10,000$, $b = 4$, and $c = 3$.

6 Conclusion

A modified secretary problem does seem to be a good model for understanding BitTorrent protocol. As it turns out, there is an optimal stopping point when searching for the best possible seeders. While this stopping point varies based on the size of the bank and the swarm, it does provide the very same maximum probability of success as the secretary problem, a remarkable discovery. In addition, we proved that the optimal threshold for a bank size of b approaches $\frac{n}{\sqrt[b]{e}}$ as $n \to \infty$.

While these results are very interesting, it may not be feasible to implement in the real world because of the proportion T_n/n. In other words, because the optimal threshold value is a large portion of n, you would need to make a connection with almost all peers in the swarm before stopping the search. Keep in mind that there is overhead involved in establishing connections, thus, the more connections made, the more time spent not transferring the file. However, we have shown that slightly modifying the win conditions significantly improves the odds of wining as well as shortens the search time.

Despite the potential problems with real world implementation, this work does show some interesting and promising results. In particular, this model may still have further extensions. For example, how could we calculate the probability of selecting at least c members of K? If this were the case, what threshold value should be chosen? In addition, the model provided in this paper may serve as the basis for a more general secretary problem.

Acknowledgments

The authors would like to thank the anonymous referee, whose excellent suggestions improved the quality of this paper.

References

[1] Y.S. Chow, S. Moriguti, H. Robbins, S.M. Samuels, Optimal selection based on relative rank (the "secretary problem"), *Israel J. Math.*, **2** (1964), 81–90.

[2] B. Cohen, Incentives build robustness in BitTorrent, *Workshop on Economics of Peer-to-Peer Systems*, **6** (2003).

[3] T. Ferguson, Who solved the secretary problem?, *Statist. Sci.*, **4** (1989), 282–289.

[4] J. Gianini, The infinite secretary problem as the limit of the finite problem, *Ann. Probab.*, **5** (1977), 636–644.

[5] J. Gianini, and S. Samuels, The infinite secretary problem, *Ann. Probab.*, **4** (1976), 418–432.

[6] J. Gilbert, and F. Mosteller, Recognizing the maximum of a sequence, *Selected Papers of Frederick Mosteller*, Springer, New York, NY, (2006), 355–398.

[7] B. GoMys, The secretary problem-the case with memory for one step, *Demonstr. Math.*, **11** (1978).

[8] T. Hoßfeld, D. Hock, S. Oechsner, F. Lehrieder, Z. Despotovic, W. Kellerer, M. Michel, Measurement of BitTorrent swarms and their AS topologies, *Computer Networks* (2009).

[9] T. Lorenzen, Generalizing the secretary problem, *Adv. in Appl. Probab.*, **11** (1979), 384–396.

[10] T. Lorenzen, Optimal stopping with sampling cost: the secretary problem, *Ann. Probab.*, **9** (1981), 167–172.

[11] M.L. Nikolaev, On a generalization of the best choice problem, *Theory Probab. Appl.*, **22** (1977), 187–190.

[12] Palo Alto Networks, *Palo Alto Networks Blog*, (2019), available online at the URL: `http://researchcenter.paloaltonetworks.com/app-usage-risk-report-visualization/#sthash.3AmpVFlp.3Uo35YUm.dpbs`

[13] J. Petruccelli, Best-choice problems involving uncertainty of selection and recall of observations, *J. Appl. Probab.*, **18** (1981), 415–425.

[14] J. Rose, Optimal sequential selection based on relative ranks with renewable call options, *J. Amer. Statist. Assoc.*, **79** (1984), 430–435.

[15] H. Rubin, and S.M. Samuels, The finite-memory secretary problem, *Ann. Probab.*, **5** (1977), 627–635.

[16] W. Rudin, *Principles of mathematical analysis*, McGraw-hill, New York, NY, **3** (1976).

[17] M.H. Smith, A secretary problem with uncertain employment, *J. Appl. Probab.*, **12** (1975), 620–624.

[18] M.H. Smith, and J.J. Deely, A secretary problem with finite memory, *J. Amer. Statist. Assoc.*, **70** (1975), 357–361.

[19] M. Tamaki, A secretary problem with double choices, *J. Oper. Res. Soc. Japan*, **22** (1979), 257–265.

[20] M. Tamaki, The secretary problem with optimal assignment, *Oper. Res.*, **32** (1984), 847–858.

[21] M. Yang, Recognizing the maximum of a random sequence based on relative rank with backward solicitation, *J. Appl. Probab.*, **11** (1974), 504–512.

Jay Cummings
California State University, Sacramento
6000 J Street, Sacramento, CA 95819
E-mail: `jay.cummings@csus.edu`

Joe Ezaki
California State University, Sacramento
6000 J Street, Sacramento, CA 95819
E-mail: `ezaki@csus.edu`

Received: November 3, 2018 **Accepted**: January 7, 2019
Communicated by Laura Smith Chowdhury