ANALYSIS OF DECENTRALIZED ALGORITHM FOR ONLINE SECRETARY PROBLEM

ABSTRACT

This is an abstract.

Keywords: Secretary Problem, Online Algorithm, Decentralized Algorithm, Competitive Analysis

Contents

Chapter 1 Introduction

Variants of online secretary problem have be

emphTODO: Need revision

TODO: connection and distinction with online matching

Consider a complete weighted bipartite graph $G = (U, V, w), w : U \times V \to R^+$. $U = \{u_1, u_2, \ldots, u_m\}$ and $V = \{v_1, v_2, \ldots, v_n\}$ ($m \leq n$) are commonly known as firms and applicants in the market, respectively. To avoid ties, we assume that no two edges have the same weight.

In an online setting, all applicants arrive one by one in a random order. Each firm decides whether to send an offer immediately or not, and then each applicant can choose an offer from what she has got. And the edge weight $w(u, v)$ is considered to be the benefit u and v will get if the applicant v accepts the offer from the firm u, in other words, $w(u, v)$ is both the value of u for v and the value of v for u. Note that each firm can hire at most one applicant as its secretary. All decisions can not be revoked. Firms and applicants can only see weights of edges which are incident to them. Our goal is to design decentralized algorithms for each firm such that (i) the resulting overall social welfare is nearly optimal and (ii) each firm by adopting the proposed algorithm can get the nearly optimal applicant. By "decentralized" it means there is no supervisor doing the assignments, and each firm runs its own algorithm independently with no communication.

Chapter 2 Preliminaries

2.1 Graph Matching

2.1.1 Bipartite Matching

A graph is called *bipartite* if its vertex set can be partitioned into two disjoint set U and V such that every edge connects a vertex in U with a vertex in V. Such a graph is often written as $G = (U, V, E)$ where U and V are two disjoint vertex set and E is the edge set.

A *matching* in a graph $G = (U, V, E)$ is a set of edges M such that no two edges shared a common vertex. And a matching M is called *maximal* if for every edge $e \in E \setminus M$ it satisfies that e shares some common vertices with some edges from E. Normally we are interested in the *maximum* matching, i.e. a matching containing the largest possible number of edges. And the size of the maximum matching is called the *matching number* of this graph. Figure **??** shows an example of a maximum matching in a bipartite graph. Note that the problem of finding a maximum matching can be solved in polynomial time by *Hungarian method*.

People show their great interests in finding the maximum matching since there are deep connections between the matching number and many other interesting properties in a given graph. For example, the *König's theorem* proved by Dénes Kőnig in 1931 states that, in bipartite graph, it is equivalent to find a maximum matching that to find a minimum set cover. Independently in the same year by Jenő Egerváry the same result was shown in the more general case of weighted bipartite graphs. Thus a lot of problems was shown to be NP-hard in general graphs have a polynomial time algorithm in bipartite graphs (e.g. minimum set cover).

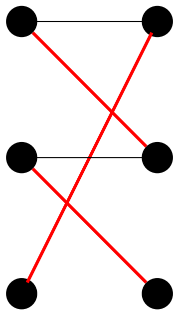

Figure 2.1 The red edges forms a maximum matching in graph

	potato	orange	banana
alice	**5**	3	1
bob	2	4	**7**
celine	3	**6**	4

Figure 2.2 An example of maximum weighted bipartite matching

2.1.2 Weighted Bipartite Matching

To generalize the maximum matching problem we could assign weights to those edges. And the goal is then changed to find a matching with the largest/smallest possible weight. This gives us the *maximum/minimum weighted bipartite matching problem*.

Formally speaking, given a weighted complete bipartite graph $G = (U, V, w)$ where U and V are two disjoint vertex sets and the edge set $E = U \times V$. w is the weight function maps every edge in E to its own weight in $\mathbb{R}$. We have to find a matching M such that its weight $w(M) \triangleq \sum_{e \in M} w(e)$ is maximized/minimized.

This maximum weighted bipartite matching problem characterized a lot of problems in our daily lives. For example we are selling some items (e.g. potato, orange and banana) and the customers (e.g. alice, bob and celine) provides their prices on each of these items. Every customer wants only one item and a item could only be sold to one of the customers. The problem is how are we going to sell these items so that we can earn the most money. Figure **??** shows an example of such a problem, and it is clear that we should sell potato to alice, orange to celine and banana to bob so that we can earn $5 + 6 + 7 = 18$ in total.

To solve the maximum/minimum weighted bipartite matching problem, a polynomial-time algorithm was proposed by Harold Kuhn (1955), who gives the name "*Hungarian method*" as this algorithm was largely based on the works done by two Hungarian mathematicians Dénes Kőnig and Jenő Egerváry.

2.2 Online Algorithms

But most of time in our real lives, our information about the graph is usually incomplete. And we have to make our decisions before the whole graph is shown to us.

For example in figure **??**, when Alice comes and wants to buy one item, we can not let her wait until Bob and Celine provide their prices. We have to make our decision right now on which one to sell before Alice gets angry. This scenario raises a new sort of problems in an online fashion – the problems we have to output our answers before the whole input was shown to us.

2.2.1 Secretary Problem

One of the most classical problem is the *online secretary problem*.

Suppose you are hiring a secretary for your firm, and there are n applicants who come to apply for this job. You have to interview these applicants one by one in a random order until you choose one of them as the secretary. After an interview you have to make your decision whether to offer her this secretarial position. Once the decision is made, it can not be revoked. Each applicant has a score on how good they can handle this job and of course you want the best one. The difficulty is, before a applicant is interviewed you can not know her score. How can you make a decision so that the probability of choosing the best applicant is maximized?

People have found out that the best strategy for this problem is *stopping rule*. This strategy contains two phases, the observation phase and the selection phase: In the *observation phase*, the firm would interview the first $r-1$ applicants and reject them, set t be the best score among them. Later in the *selection phase*, the firm would choose the first subsequent applicant who has a score better than t. Here r is set accordingly and proportional to n.

For example, suppose that 6 applicants are waiting for the interview and their score is $\{3, 4, 2, 5, 1, 6\}$ (in the coming order). We choose the secretary using stopping rule with $r = 3$. First we interview the first 2 applicants and reject them, record the best score as $t = 4$. When the third applicant comes in we found out that her score is no better than t, hence we reject her. Then the fourth applicant comes in, luckily she has a score 5 greater than t, so we offer her the secretarial position and terminate the protocol. Unfortuately we failed to pick the best one in the sixth place, but a score of 5 is not so bad to our real needs.

Now what I'm going to show that if the coming order of applicants is uniformly at random, then the probability to get the best applicant using stopping rule is at least a constant.

With a parameter r, the probability of choosing the best applicant can be easily calculated:

$$\begin{aligned}P(r) &= \sum_{i=1}^{n} \Pr(i\text{-th applicant is the best and chosen}) \\ &= \sum_{i=1}^{n} \Pr(i\text{-th applicant is chosen}|i\text{-th applicant is the best}) \times \frac{1}{n} \\ &= \sum_{i=r}^{n} \Pr(\text{no one is chosen before } i|i\text{-th applicant is the best}) \times \frac{1}{n} \\ &= \sum_{i=r}^{n} \frac{r-1}{i-1} \times \frac{1}{n}\end{aligned}$$

Note that in the previous equations, the event that no one is chosen before i means the second best among the first i applicant appears before r. Therefore
$\Pr(\text{no one is chosen before } i|i\text{-th applicant is the best}) = \frac{r-1}{i-1}$.

When n goes to infinity, $P(r)$ is approximately the integral $\frac{r}{n}\int_{\frac{r}{n}}^{1} \frac{1}{x}\mathrm{d}x = \frac{r}{n}\ln(\frac{n}{r})$.

In this problem we can choose $r \approx \frac{n}{e}$ which maximize the integral above so that $P(\frac{n}{e}) \approx \frac{1}{e} \approx 0.368$.

2.2.2 Online Matching

Another perspective of this paper's work originates from online bipartite matching. Given a weighted bipartite graph $G = (U, V, w)$, but at first you don't know any information about this graph. Each time one vertex in V is revealed and you can see the weights of all edges incident to it. Then you have to decide which vertex in U should be matched to it immediately before revealing the next vertex in V. When all vertex are shown to us you should grantee that the decisions you made form a matching and the weight of this matching is maximized. Note that the unweighted version of online bipartite matching can be viewed as the weighted one where the edge weight are limited in $\{0, 1\}$.

In fact, the online secretary problem is a special case of online weighted bipartite matching. Where the firm is the only one element in U and all applicants form V.

For the unweighted version, first we allow an adversery to decide the revealing order of V. The first landmark result was shown in Karp et al. (1990) that if randomization is allowed, there exists an algorithm which in expectation obtains a matching of size at least $(1 - \frac{1}{e})n$ assuming that $|U| = |V| = n$ and there exists a matching of size n in the underline graph. Recently in Birnbaum and Mathieu (2008) it provides a much simpler proof for this result and in Devanur et al. (2013) it gives a randomized primal-dual analysis for the algorithm they proposed. This algorithm first randomly rank the vertex in U and each time when one vertex v in V is revealed, it matches v to the a vertex in U with the highest possible rank. This algorithm does break the barrier that no deterministic protocol can find a matching of size better than $\frac{n}{2} + O(\log(n))$ in the worst case. And then they proved this *ranking algorithm* is indeed optimal for unweighted online bipartite matching problem.

In another direction, if we assume that the revealing order of V is uniformly at random, will there be some more powerful algorithms? It is quite a hot topic in recent years, and the answer is luckily positive. In Aggarwal et al. (2011), Feldman et al. (2009), Mahdian and Yan (2011), Mehta et al. (2007) and Bahmani and Kapralov (2010) they provide several new algorithms which achieve better performances and even work well in the weighted case. Most of these algorithms are based on linear programmming.

2.2.3 Competitive Analysis

Sometimes it may be a little bit complicated while analysing the performance of online algorithms since these analyses are often highly dependent on the actual computational model behind. So a new powerful tool called *competitive analysis* is invented to solve this problem.

Competitive analysis of an online algorithm compares its performance to the performance of its optimal offline algorithm that knows all the input data before making decisions. The outcome of the optimal offline algorithm is often called the God's result since it knows everything just as the God does. It is first brought out to analysis the protocols for dynamically maintaining a linear list in Sleator and Tarjan (1985). And we called an algorithm α-competitive if its *competitive ratio* – the ratio between its solution and the solution of its optimal offline algorithm – is bounded from above/below by α. More formally,

Definition 2.1. *For a maximization(or minimization) problem, an online algorithm is said to have a competitive ratio of α – α-competitive – if and only if its outcome, denoted by ALG, and the optimal offline algorithm's outcome, denote by OPT, satisfy that $\frac{ALG}{OPT} \geq \alpha$ (or respectively $\frac{ALG}{OPT} \leq \alpha$).*

Note that this definition may change accordingly subject to the actual computational model, such as involving randomness.

For example, the ranking algorithm which is mentioned above to solve the unweighted online bipartite matching, always output a matching of size at lease $(1 - \frac{1}{e})n$ in expectation. While we know that the underline graph has a maximum matching of size n and we can find it out using Hungarian algorithm (which is optimal offline algorithm). Therefore the competitive ratio of ranking algorithm is $(1 - \frac{1}{e})$ and we called it an $(1 - \frac{1}{e})$-competitive algorithm.

Unlike the classical worst-case analysis of an algorithm, which only focuses on the algorithm's performance for those "difficult" inputs. Competitive analysis requires the online algorithm to perform well on both "easy" and "difficult" inputs. Here "easy" and "difficult" are defined accordingly with respect to the performance of the optimal offline algorithm.

2.3 Generalized Secretary Problems

In previous section we have introduced the classicle online secretary problem. It requires one firm to hire one secretary among a list of applicants. And it adapts an optimal protocol called the stopping rule. But in our real life, the scenarios could be more complicated. Therefore we have to generalize this problem and give more refined analysis to them.

2.3.1 Multiple-choice Secretary Problem

One if its generalization is, instead of just hiring one secretary among the n applicants, the firm needs more secretaries (let's say k). Which gives us the *multiple-choice secretary problem*. It is natural to consider whether we could modify the optimal stopping rule for classical online secretary problem to solve this new problem.

For example, we keep the observation phase unchanged – reject the first $r-1$ applicants and record the best score among them as t. And in the selection phase we choose every applicant whose score is greater than t until we reach the quota k. Sadly this modified algorithm does not work so well as we expected. Difficulties show up when k grows large.

Here is a counterexample. Assume that $k = \alpha n$ for some constant α. Let p be any constant with in the range $(0, 1)$ and we will look at the best pk applicants. The probability that no one among the best pk applicants appears in the observations phase is $(1 - \frac{pk}{n})^r$. As for k and r are both proportional to n, this probability tends to 0 as n goes to infinity. Therefore it is almost sure that at lease one of the best pk applicants will be in the observation phase. Then in the selection phase, no more than pk applicants would be selected since the threshold t is set to be the score of one of the best pk

applicants according to the protocol. So the competitive ratio of this modified protocol is no better than p. Notice that p could be chosen arbitrarily as long as it's a constant, so this protocol can never achieve a constant competitive ratio.

In Kleinberg (2005) Robert Kleinberg gave a clever algorithm based on a simple recursion to solve the multiple-choice secretary problem. It achieves a competitive ratio of $1-O(\sqrt{1/k})$. This is even better than constant competitive ratio that origin stopping rule achieves. He did also proof that this result is already tight, i.e. no algorithm could achieve a competitive ratio of $1-\Omega(\sqrt{1/k})$.

Later in Babaioff et al. (2007), Moshe Babaioff, Nicole Immorlica, David Kempe and Robert Kleinberg proposed their modifications of the classical stopping rule to solve the multiple-choice secretary problem with a constant competitive ratio. In their paper they provides two modifications, the *virtual* algorithm and the *optimistic* algorithm. Both algorithms have the same observation phase: observe and reject the first $r-1$ applicants, but instead of setting just one threshold t, we keep a threshold set T to be the set of the best k applicants among them. Denote the score of applicant v as $s(v)$. Denote that $T=\{t_1,t_2,\ldots,t_k\}$ where $t_1,t_2,\ldots,t_k$ are sorted in decreasing order with respect to their score $s(t_i)$. Assume that v_i is the i-th applicant according to the coming order. Then there comes the selection phase:

Virtual: When an applicant v_i comes in, it is selected if and only if $s(v_i) > s(t_k)$ and t_k appears in the observation phase. If it happens that $s(v_i) > s(t_k)$, v_i is added to the threshold set T and t_k is removed. Thus, this algorithm always keeps the best k applicants having met so far in the threshold set T.

Optimistic: When an applicant v_i comes in, it is selected if and only if T is not empty and $s(v_i) > s(t_{|T|})$. After selecting v_i, $t_{|T|}$ is removed from the threshold set T while no other elements would be added to T. This is called "optimistic" since we always remove $t_{|T|}$ even if the score of v_i is better than, say, $s(t_1)$. Thus, it implicitly assumes that it will see additional more outstanding applicants in the future and then offer them these secretarial positions.

It's clear that both two algorithms select no more than k secretaries according to the protocol. Now it's sufficient to present the main theorem of their result on multiple-choice secretary problem and prove it in the way they did.

Theorem 2.2. *Both virtual algorithm and optimistic algorithm achieves a competitive ratio of e as n grows to infinity if we set $r \approx \frac{n}{e}$.*

Before proving the main theorem, two important lemmas are established below. Denote the set of applicants who are selected by S.

Lemma 2.3. *For applicant v whose score is among the best k applicants, using the virtual algorithm, the probability of v is selected as one of the secretaries is*

$$\Pr(v \in S) \geq \frac{r-1}{n} \ln(\frac{n}{r-1})$$

Lemma 2.4. *For applicant v whose score is among the best k applicants, using the optimistic algorithm, the probability of v is selected as one of the secretaries is*

$$\Pr(v \in S) \geq \frac{r-1}{n} \ln(\frac{n}{r-1})$$

The proof of lemma 2.4 is quite complicated and requires pages long calculation. But the proof of lemma 2.3 is surprisingly simple:

Proof of Lemma 2.3: According to the protocol, if applicant v – the applicant whose score happens to be among the best k applicants – comes in at the i-th place where $i > r-1$, it will be selected if and only if the last one in the threshold set T appears before time r. Since the coming order is uniformly at random, this event has a probability of $\frac{r-1}{i-1}$. Therefore the probability of v is selected as one of the secretaries is

$$\Pr(v \in S) \geq \sum_{i=r}^{n} \frac{1}{n} \frac{r-1}{i-1} = \frac{r-1}{n} \sum_{i=r}^{n} \frac{1}{i-1} \geq \frac{r-1}{n} \int_{r-1}^{n} \frac{1}{x} \mathrm{d}x = \frac{r-1}{n} \ln(\frac{n}{r-1})$$

$\square$

Then it is straightforward to proof the main theorem:

Proof of Theorem 2.2. Assume that v_i^* is the i-th best applicant among the n applicants applying for the secretarial positions. Clearly that the answer of the optimal offline algorithm is $OPT = \sum_{i=1}^{k} s(v_i^*)$ regardless what the incomming order is. The by the linearity of expectation, the answer obtained by the virtual/optimistic algorithm is

$$
\begin{aligned}
E[ANS] &= \sum_{i=1}^{n} s(v_i^*) \times \Pr(v_i^* \in S) \geq \sum_{i=1}^{k} s(v_i^*) \times \Pr(v_i^* \in S) \\
&\geq \sum_{i=1}^{k} s(v_i^*) \times \frac{r-1}{n} \ln(\frac{n}{r-1}) = OPT \times \frac{r-1}{n} \ln(\frac{n}{r-1})
\end{aligned}
$$

Then the theorem is obtained by setting $r \approx \frac{n}{e}$. □

2.3.2 More on Generalized Secretary Problem

Besides multiple-choice secretary problem, there are many other generalizations of online secretary problem. For example, the *knapsack secretary problem*, where each applicant has a cost – you have to pay them the salary. You can select as many applicants as possible so long as your budget is feasible.

It is known that the offline version of knapsack problem is NP-hard. But fortunately it adapts a full polynomial-time approximation scheme and a really simple 2-approximation algorithm as in Vazirani (2001). By assuming that the incoming order is uniformly at random, in Babaioff et al. (2007) they proposed a 2-approximation algorithm with a constant competitive ratio.

Another most interesting generalization is in a more algebic way – the *matroid secretary problem*, it requires that the applicants being chosen lie in a given matroid. There are many subproblems of it which are highly related to our daily lives:

- In multiple-choice secretary problem the feasible set is a *uniform matroid*.
- *Transversal matroid*. For example n customers are purchasing tickets for m movies and a bipartite indicating which movies each customer is interested in. Each movie has a capacity of seats and the goal is to find a subset of customers with maximal total values who can all see a movie simultaneously.

- *Gammoid*. Given a graph, each travelers are seeking path from one source to other destination. Then the goal is to route a subset of travelers with maximal profits without violating capacities being set on the edge.

- *Graphic matroid* which is less natural motivated. Each customers corresponds an edge, and the goal is to pick a acyclic subgraph with maximal total value.

Some special cases of matroid secretary problem have been found a constant-competitive solution. Babaioff et al. gave constant-competitive algorithms for several classes of matroid including graphic matroid and transversal matroid with bounded degree in Babaioff et al. (2007). Subsequently Dimitrov and Plaxton presented another constant-competitive algorithm for transversal matroid without bounded degree restriction in Dimitrov and Plaxton (2008).

While whether general matroid secretary problem has a constant competitive ratio approach remains open. The best postitive result is in Babaioff et al. Babaioff et al. (2007), they provide a $O(\log k)$-competitive algorithm where k is the rank of matroid. This algorithm is also similar to the classical stopping rule.

Chapter 3 General Model

3.1 Formal Definition

Before giving our results, we need to first define the model of our problem and its objective function. Generally speaking, instead of one, we have n firms and each of them wants to hire one secretary. There are m applicants waiting outside for their interview. The applicants will come in one by one and all n firms would interview her simultaneously. After one interview each of the firms would grade the applicant who has just been interviewed a score, and then decide whether to accept her or not. If the applicant receives offers, she could choose any one of them. In the following part we assume that the applicant would always choose the offer with the highest score, for it probably has the highest salary in real life.

More formally, in a bipartite matching perspective. For a given edge weighted complete bipartite $G = (U, V, w)$, where U corresponds to the set of m firms, V corresponds to the set of n applicants and w is the weight function corresponding to the score firm u graded applicant v. For simplicity we assume that no two edges have the same weights.

The goal is to design a decentralized algorithm for each firm to (i) maximize the social welfare, i.e. find a matching with the largest possible sum of scores and (ii) for each firm by adopting the proposed algorithm they can get their optimal applicants, i.e the best applicant they can get among all possible maximum weighted bipartite matching. By decentralization it means that, there is no supervisor doing such an assignment and each firm has to run the algorithm independently without any communication or any global information.

We have the value of optimal assignment (denoted by OPT), and the value of the algorithm's outcome (denoted by ALG). By saying "competitive ratio α" in the general case, we mean that for any instance G, $\frac{E[ALG]}{E[OPT]} \geq \alpha$. In the following sections, we will present more specific settings, and will define the objective functions accordingly.

3.2 Simultaneous stopping rule

Then we turn to the algorithms. In the following part we will provides two simple algorithms, and see how well they can solve this problem in different scenarios.

It is natural to think of whether we cound simply apply the classical stopping rule for online secretary problem here. So here comes our first approach – **simultaneous stopping rule**:

- The *observation phase*: each firm u observes and rejects the first $(r-1)$ applicants. Denote the value of the best applicant among them by t_u which is the threshold. Typically we set r to be a constant fraction of n such as $\lfloor \frac{n}{e} \rfloor + 1$.
- The *selection phase*: for each applicant who comes in later, each firm u who hasn't been matched before sends her an offer if and only if its value exceeds t_u.

This algorithm is simple and straightforward. But unfortunately it does not work very well in general case. Here is a counterexample: let $r = \lfloor \beta n \rfloor + 1$ for some constant $\beta \in (0, 1)$.

Example 3.1. *Let $m = \Theta(n)$. There are two kinds of applicants. First we have $a = \Theta(\log(n))$ "good" applicants with weight $2 + \epsilon_{u,v}$ for all edges incident to them. The rest applicants are marked as "bad" with weight $1+\epsilon_{u,v}$ for all edges incident to them ($|\epsilon_{u,v}| < \frac{1}{2}$ is only used to avoid ties and they could be arbitrarily small so we can ignore them for simplicity of the analysis).*

For any permutation of the applicants, it is clear that $OPT = m + a$ by choosing a "good" applicants and $m - a$ "bad" applicants. To show that simultaneous stopping rule fails to handle this setting, we are going to give a upperbound for $E[ALG]$. The upperbound is established by stating that (i) it is almost sure that the firms can witness a "good" applicant in the observation phase and (ii) in this case, only a few firms cound successfully find their secretaries.

In the first $(r-1)$ rounds, the probability that no firm observes a 'good' applicant is:

$$\prod_{i=0}^{\lfloor \beta n \rfloor} \frac{n-a-i}{n-i} \leq (1-\frac{a}{n})^{\lfloor \beta n \rfloor} \leq (1-\frac{a}{n})^{\beta n-1}$$

We first observes that this function $(1-\frac{a}{n})^{\beta n}$ is asymptotically $e^{-\beta a}$ as n goes to infinity. And we set $a = \Theta(\log(n))$, so this probability goes to 0 as n grows.

The second thing is, if the firms did observe a "good" applicant in the observation phase, no firm could give a offer to "bad" applicants in the selection phase according to the protocol since threshold for each firm is set to be near 2.

Therefore

$$\begin{aligned}
E[ALG] &\leq (1-\frac{a}{n})^{\beta n-1} \times (m+a) + (1-\prod_{i=0}^{\lfloor \beta n \rfloor} \frac{n-a-i}{n-i}) \times 2a \\
&\leq \frac{n}{n-a} e^{-\beta a} \times (m+a) + 2a, \\
\frac{E[ALG]}{E[OPT]} &\leq \frac{n}{n-a} e^{-\beta a} + \frac{2a}{m+a} = \frac{1}{n^{\Theta(1)}} + \Theta(\frac{\log n}{n}).
\end{aligned}$$

Corollary 3.2. *In the general case, simultaneous stopping rule cannot get better competitive ratio than $\Theta(\frac{\log n}{n})$ if we set $r = \lfloor \beta n \rfloor + 1$ for some constant $\beta \in (0,1)$.*

The problem encountered here is that, without global information, firms will be competing over a small set of elite applicants and left all the others aside. While their choices could be more flexible to avoid this kind of tragedy.

3.3 Simultaneous stopping rule with m slots

To tackle this problem, we slightly modify the algorithm above. Instead of only one threshold, we set m thresholds. Like the virtual algorithm for multiple choice secretary problem proposed in section 2.3.1, we call it **simultaneous stopping rule with m thresholds**:

- The *observation phase*: Each firm u observes and rejects the first $(r-1)$ applicants, then choose the best m applicants among them to form a threshold set T_u. Denote their scores by $t_{u,1}, t_{u,2}, \ldots, t_{u,m}$ in decreasing order. (Some of them are set to be $-\infty$ if not enough).

- The *selection phase*: For each applicant v who comes in later and each firm u, if the applicant v's value $w(u, v)$ exceed $t_{u,m}$, then add it to T_u and the least valuable one in T_u is removed. That is, firm u always keeps the best m values in T_u. And v will get an offer from u if and only if $w(u, v)$ is added to T_u and the one removed from T_u is either $-\infty$ or has an arrival time less than r.

Similar you can define a new algorithm based on the optimistic algorithm for multiple-choice secretary problem, and prove the results with this new algorithm. But it will be a lot more complicated, so here we do only focus on the variant of virtual algorithm.

As you can see the choices firms have by adopting simultaneous stopping rule with m slots are more flexible. They all have at most m chances to find their desired secretaries. It will be shown later that this simultaneous stopping rule works well for example 3.1 above. But the flexibility brought out more problems. Here is a simple bad example for simultaneous stopping rule.

Example 3.3. *Let $m = \Theta(n)$. The score that firms grade applicants are all $1 + \epsilon_{u,v}$ except for a special firm u^* and a special applicant v^*. The score u^* gives v^* is denoted by s, and s can be arbitrarily large. About $\epsilon_{u,v}$, the same as in example 3.1, $|\epsilon_{u,v}|$ could be arbitrarily small. One additional requirement is that, for any applicant v, $\epsilon_{u^*,v} > 0$ and for any firm $u \neq u^*$, $\epsilon_{u,v} < 0$.*

It is observed that, with the additional requirement, once u^* send its offer, the applicant who receives it would definitely choose u^*. Another observation is that, the performance of simultaneous stopping rule is dependent on the probability that v^* is matched to u^* since the score which u^* grade v^* can be arbitrarily large. Then we are going to show that this probability is relatively small.

Assume that v^* comes in for an interview at the i-th position where $i \geq r$. If we want u^* to hold v^*, two things have to be ensured: (i) u^* hasn't sent its offer yet, (ii) u^* has to send its offer to v^*. Clearly that condition (ii) directly follows from condition (i). As for the condition (i), it is equivalent to the situation that, for u^* the best m applicants who comes in before applicant v^* has an arrival time before r. And because the incoming order is uniformly at random, this happens with a probability of

$$\frac{r-1}{i-1}\frac{r-2}{i-2}\cdots\frac{r-m}{i-m} \le \left(\frac{r-1}{i-1}\right)^m$$

Therefore the probability of u^* getting v^* is no greater than

$$\frac{1}{n}\sum_{i=r}^{n}\left(\frac{r-1}{i-1}\right)^m$$

And this formula is asymptotically the integral

$$\int_{\frac{r-1}{n}}^{1}\left(\frac{r-1}{nx}\right)^m \mathrm{d}x = \frac{x}{-m+1}\left(\frac{r-1}{nx}\right)^m\Bigg|_{\frac{r-1}{n}}^{1} = \frac{1}{m-1}\left(\frac{r-1}{n} - \left(\frac{r-1}{n}\right)^m\right)$$

Because m and r are both proportional to n, this integral tends to 0 as n grows to infinity. Hence it is very unlikely that u^* will get v^*. So the performance of simultaneous stopping rule with m slots is relatively poor.

The problem of the simultaneous stopping rule with m slots is, the threshold that we set is not high enough to filter out those applicants who are not sufficiently good. While this is what the simultaneous stopping rule is good at.

Although this two algorithms does not work that well in general case. In the next chapter we are going to find some more specific models to show how powerful this two algorithms could be solving this problem.

Chapter 4 Specific Models

In the following part, we are going to propose several different ways on how the edge weights are generated. And see how the two algorithms proposed before can manage these situations.

4.1 Ranking Model

Consider the following model: all the edge weights are independently sampled from the same distribution D which is unknown to us. The concept "competitive ratio α" in this particular model means that for any instance $G = (U, V, w)$ and distribution D, by taking the expectation over all the possible weights and all the permutations, we have $\frac{E[ALG]}{E[OPT]} \geq \alpha$.

Then there comes our first result.

Theorem 4.1. *Simultaneous stopping rule achieves a constant competitive ratio.*

As stated before, the problem for simultaneous stopping rule is, firms have strong willings to compete with each other on those elite applicants and less focus on those applicants who are not so outstanding but have a safe position. So it's hard to grantee that each of them could successfully hire a secretary. Here we will show that in the ranking model, the chances of such collisions is rare.

In the following part, first we are going to state with high probability that each firm u would send an offer to its favorite applicant – the applicant with the highest score. Then bound the probability that other firms would compete over its own favorite applicant. Thus the probability for each firm to get its best applicant is relatively high (at least a constant), which implies the result.

Before the proof of this theorem, first we have to introduce some notations:

Notation 4.2.

- *Event $B(u_i, v_j)$ means that applicant v_j has the highest score with respect to firm u_i.*
- *Event $P(u_i, v_j)$ means that firm u_i sends an offer to v_j.*
- *Event $A(u_i, v_j)$ means that firm u_i sends an offer to v_j and v_j accepts it.*

Proof of Theorem 4.1. WLOG let's assume that the incoming order of the applicants is $\{v_1, v_2, \dots, v_n\}$.

Fix a particular firm $u \in U$, first bound the probability that it will get the best applicant with the highest score.

$$\begin{aligned}
\Pr(u \text{ gets the best}) &= \sum_{i=1}^{n} \Pr(B(u, v_i)) \times \Pr(A(u, v_i)|B(u, v_i)) \\
&= \sum_{i=r}^{n} \frac{1}{n} \times \Pr(P(u, v_i)|B(u, v_i)) \\
&\quad \times \Pr(A(u, v_i)|B(u, v_i), P(u, v_i))
\end{aligned}$$

Given v_i is the best applicant for u, once we ensure that the previous $(i - r)$ applicants are no better than the threshold, u will be free when v_i comes in and hence u will send v_i an offer. That is, we need to ensure that in the first $(i - 1)$ applicants, the best of them is among the first $(r - 1)$ ones. Given that all the applicants arrive in a random order, this probability would be simply $\frac{r-1}{i-1}$. Therefore

$$\Pr(P(u, v_i)|B(u, v_i)) \geq \frac{r-1}{i-1}$$

As for the second part, if v_i receives only one offer which is from u, v_i can only choose u. And the condition holds when the score of v_i for each $u' \neq u$ does not exceed the threshold set by u'. In other words, for each $u' \neq u$, over the first $r - 1$ applicants together with v_i, the best applicant for u' is not v_i. These $m - 1$ events (each with probability $\frac{r-1}{r}$) are independent from each other, and are all independent from $B(u, v_i)$ and $P(u, v_i)$. Thus

$$\Pr(A(u, v_i)|B(u, v_i), P(u, v_i)) \geq (\frac{r-1}{r})^{m-1}$$

To sum up:

$$\Pr(u \text{ gets the best}) \geq \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1} \times (\frac{r-1}{r})^{m-1}$$

Let $p = \frac{r}{n}$ be a constant, and assume that $m \leq \alpha n$ where $\alpha \in (0, 1]$ is a parameter. Then $(\frac{r-1}{r})^{m-1} \geq (1 - \frac{1}{r})^{\alpha n} = ((1 - \frac{1}{r})^r)^{\frac{\alpha}{p}}$ and

$$\Pr(u \text{ gets the best}) \geq ((1 - \frac{1}{r})^r)^{\frac{\alpha}{p}} \sum_{i=r}^{n} \frac{1}{n} \frac{r-1}{i-1}$$

Now let n goes to infinity and it is asymptotically the integral:

$$f(p) = e^{-\frac{\alpha}{p}} \int_p^1 \frac{p}{x} dx = -p \ln(p) e^{-\frac{\alpha}{p}}$$

which is a constant depending on the choice of p. Thus, for each firm u, it has at least a constant probability $f(p)$ to get its best applicant, then we have

$$\begin{aligned} E[\text{score of the applicant } u \text{ gets in } ALG] &\geq f(p) \times E[\text{score of the best applicant for } u] \\ &\geq f(p) \times E[\text{score of the applicant } u \text{ gets in } OPT] \end{aligned}$$

Therefore, it is clear to see that

$$\frac{E[ALG]}{E[OPT]} \geq f(p),$$

and we did prove that the simultaneous stopping rule achieves a constant competitive ratio. □

Note that, in the analysis above, it is not necessary that all the weights of edges are independently and identically distributed. All we need is that firms' preference lists – the order of applicants according to the firm's score – are sampled independently from each other. That is, the rank of an applicant v for a firm u_i has nothing to do with the rank of v for another firm u_j. So we named this model the *ranking model*.

To generalize the ranking model above, we could weaken the restriction by adding correlations between the weights of edges incident to the same applicant, and introduce some new model.

4.2 Gaussian Model

4.2.1 Normal distribution direction 1

TODO: Give it a name: generative model?

Assume that each applicant has a quality q_i, and the weights of edges incident to a certain applicant v_i are generated independently from a distribution D_i with mean q_i. As we can see, if all qualities are equal and all the distributions are the same, then it is equivalent to the random rank model discussed previously.

As before, we formally define "competitive ratio α" in this model — for any given G, $\{q_i\}_{i=1}^n$, and $\{D_i\}_{i=1}^n$, by taking the expectation over all the possible weights and all the permutations, we have $\frac{E[ALG]}{E[OPT]} \geq \alpha$.

Typically, we consider normal distributions. Assume D_i is a normal distribution $N(q_i, \sigma^2)$ where q_i is the quality of applicant v_i and σ is a fixed constant. Denote that $\delta_{max} = \max_{i \neq j} |q_i - q_j|$ and $\varphi = \frac{\delta_{max}}{\sigma}$. The following theorem holds.

Theorem 4.3. *Simultaneous stopping rule acheives a constant competetive ratio when $\varphi \leq O(\frac{1}{n^2})$.*

WLOG we assume that applicants arrive in the order of $\{v_1, v_2, \ldots, v_n\}$, and that for given qualities $\{q_i'\}_{i=1}^n$, $\{q_i\}_{i=1}^n$ is a random permutation of $\{q_i'\}_{i=1}^n$. This is equivalent with the model where applicants arrive in a random order.

For $i \geq r$, let $D(u, v_i)$ be the event that $w(u, v_i)$ does not exceed the threshold set by u.

Proposition 4.4. *For each $u \in U$ and $i \geq r$,*

$$\Pr(D(u, v_i) | \{q_i\}_{i=1}^n) \geq \Pr(D(u, v_i) | \forall 1 \leq j \leq r-1, q_j = q_i - \delta_{max}).$$

Proof. For convience, let $x_i = w(u, v_i)$ sampled from $N(q_i, \sigma^2)$. WLOG let $q_i = 0$.

$D(u, v_i)$ means that there exists some x_j such that $1 \leq j \leq r-1$ and $x_i < x_j$.

$$\begin{aligned}
\Pr(D(u,v_i)|\{q_i\}_{i=1}^n) &= 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} \prod_{j=1}^{r-1} (\int_{-\infty}^{x_i} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_j-q_j)^2}{2\sigma^2}} \mathrm{d}x_j) \mathrm{d}x_i \\
&= 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} \prod_{j=1}^{r-1} (\frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{x_i - q_j}{\sqrt{2}\sigma})) \mathrm{d}x_i,
\end{aligned}$$

where $\mathrm{erf}(\cdot)$ is the error function:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t.$$

Now take derivative with respect to q_j:

$$\begin{aligned}
&\frac{\partial \Pr(D(u,v_i)|\{q_i\}_{i=1}^n)}{\partial q_j} \\
=& \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-q_j)^2}{2\sigma^2}} \prod_{k<r, k\neq j} (\frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{x_i - q_k}{\sqrt{2}\sigma})) \mathrm{d}x_i \geq 0,
\end{aligned}$$

which concludes the result. □

Proposition 4.5. *For each $u \in U$ and $i \geq r$,*

$$\Pr(D(u,v_i)|\forall 1 \leq j \leq r-1, q_j = q_i - \delta_{max}) \geq 1 - \frac{1}{r} - \frac{r-1}{\sqrt{2\pi}}\varphi.$$

Proof. First we have

$$\begin{aligned}
&\Pr(D(u,v_i)|\forall 1 \leq j \leq r-1, q_j = q_i - \delta_{max}) \\
=& 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} (\frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{x_i + \delta_{max}}{\sqrt{2}\sigma}))^{r-1} \mathrm{d}x_i.
\end{aligned}$$

Knowing that

$$\begin{aligned}
&\frac{\mathrm{d}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x}{\sqrt{2}\sigma}))^{r-1}}{\mathrm{d}x} \\
=&\frac{r-1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x}{\sqrt{2}\sigma}))^{r-2} \\
\le&\frac{r-1}{\sqrt{2\pi}\sigma},
\end{aligned}$$

by Mean Value Theorem, we have

$$(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i+\delta_{max}}{\sqrt{2}\sigma}))^{r-1} \le (\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i}{\sqrt{2}\sigma}))^{r-1}+\frac{r-1}{\sqrt{2\pi}\sigma}\times\delta_{max}.$$

Therefore

$$\begin{aligned}
&\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i+\delta_{max}}{\sqrt{2}\sigma}))^{r-1}\mathrm{d}x_i \\
\le&\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i}{\sqrt{2}\sigma}))^{r-1}\mathrm{d}x_i \\
&+\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma}\mathrm{d}x_i.
\end{aligned}$$

Note that the first term is exactly the probability that x_i is the highest value among $\{x_i\}_{j=1}^{r-1}\cup\{x_i\}$, which equals $\frac{1}{r}$, and the second term

$$\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma}\mathrm{d}x_i=\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}\mathrm{d}x_i=\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma},$$

thus

$$\begin{aligned}
&\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i+\delta_{max}}{\sqrt{2}\sigma}))^{r-1}\mathrm{d}x_i \\
=&\frac{1}{r}+\frac{r-1}{\sqrt{2\pi}}\frac{\delta_{max}}{\sigma} \\
=&\frac{1}{r}+\frac{r-1}{\sqrt{2\pi}}\varphi.
\end{aligned}$$

Hence concludes the result.

$\square$

Lemma 4.6. *If every firm adopts simultaneous stopping rule, each of them can get its best applicant with constant probability.*

Proof. This is almost the same as Theorem 4.1. Fixing a particular firm $u \in U$, we estimate the probability that u gets its best applicant.

$$\Pr(u \text{ gets its best}|\{q_i\}_{i=1}^n) \geq \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1} \times \Pr(A(u,v_i)|B(u,v_i), P(u,v_i), \{q_i\}_{i=1}^n)$$

Again if no other firm would send an offer to v_i, $A(u,v_i)$ must be true if $P(u,v_i)$ holds. For any $u' \neq u$, $P(u',v_i)$ only depends on the value of $\{q_i\}_{i=1}^n$. Thus, given $\{q_i\}_{i=1}^n$, all the events $\{P(u',v_i)|u' \neq u\}$ are independent from each other, and are all independent from $B(u,v_i)$ and $P(u,v_i)$. Therefore

$$\begin{aligned}
&\Pr(A(u,v_i)|B(u,v_i), P(u,v_i), \{q_i\}_{i=1}^n) \\
\geq &\Pr(\bigcap_{u' \neq u} \overline{P(u',v_i)}|B(u,v_i), P(u,v_i), \{q_i\}_{i=1}^n) \\
= &\prod_{u' \neq u} \Pr(\overline{P(u',v_i)}|\{q_i\}_{i=1}^n) \\
\geq &\prod_{u' \neq u} \Pr(D(u',v_i)|\{q_i\}_{i=1}^n)
\end{aligned}$$

Let $p = \frac{r}{n}$ be a constant, $\varphi \leq \frac{c}{r(r-1)}$ for some constant c since $\varphi \leq O(\frac{1}{n^2})$. Then by proposition 4.4 and 4.5 we have

$$\begin{aligned}
\Pr(A(u,v_i)|B(u,v_i), P(u,v_i), \{q_i\}_{i=1}^n) &\geq (1 - \frac{1}{r} - \frac{r-1}{\sqrt{2\pi}}\varphi)^{m-1} \\
&\geq (1 - (1 + \frac{c}{\sqrt{2\pi}})\frac{1}{r})^{m-1}
\end{aligned}$$

Denote that $c' = 1 + \frac{c}{\sqrt{2\pi}}$. $m \leq \alpha n$ where $\alpha \in (0,1]$ is a parameter. To sum up, we have

$$\begin{aligned}\Pr(u \text{ gets its best}|\{q_i\}_{i=1}^n) &\geq \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1} \times (1-\frac{c'}{r})^{m-1} \\ &\geq ((1-\frac{c'}{r})^r)^{\frac{\alpha}{p}} \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1}\end{aligned}$$

When n goes to infinity, the summation above can be approximated by integral:

$$f(p) = e^{-\frac{c'\alpha}{p}} \int_p^1 \frac{p}{x} \mathrm{d}x = -p\ln(p)e^{-\frac{c'\alpha}{p}}.$$

which is a constant.

Go back to the very beginning, where we are given a quality sequence $\{q'_i\}_{i=1}^n$, and $\{q_i\}_{i=1}^n$ is a random permutation of it. Thus we have

$$\begin{aligned}&\Pr(u \text{ gets its best}|\{q'_i\}_{i=1}^n) \\ =&\sum \Pr(\{q_i\}_{i=1}^n|\{q'_i\}_{i=1}^n) \times \Pr(u \text{ gets its best}|\{q_i\}_{i=1}^n) \\ \geq&\sum \Pr(\{q_i\}_{i=1}^n|\{q'_i\}_{i=1}^n) \times f(p) \\ =&f(p)\end{aligned}$$

when n goes to infinity. □

With Lemma 4.6 we could know that simultaneous stopping rule is almost an optimal strategy for firms. It grantees that each of them could have a very good response with constant probability. Now using the same analysis in Theorem 4.1, it's sufficient to complete the proof of our main theorem.

4.2.2 Normal distribution direction 2

In the previous sections, we have got a rough idea that simultaneous stopping rule works well when the preference lists of all firms are "different enough" from each other. Recall that in Example 3.1, this algorithm can get no better than $\Theta(\frac{\log n}{n})$−competitive ratio when the firms' view on the applicants are nearly identical to each other. In this situation, we claim that simultaneous stopping rule with m slots can solve the problem.

In this section, we consider the same model as Section **??**. And correspondingly, we define that $\delta_{min} = \min_{i \neq j} |q_i - q_j|$, and $\psi = \frac{\delta_{min}}{\sigma}$. We claim that

Theorem 4.7. *Simultaneous stopping rule with m slots achieves a constant competitive ratio when $\psi \geq \omega(n)$ with large probability.*

Here by saying "achieves a constant competitive ratio with large probability" we mean: with probability approaching 1 over all possible weights, $\frac{E[ALG]}{E[OPT]} \geq c$ for some constant $c > 0$ where the expectation is taken over all possible coming order of applicants. This theorem follows directly from the following two lemmas.

Lemma 4.8. *When $\psi \geq \omega(n)$, for a given sequence $\{q_i\}_{i=1}^n$, with probability approaching 1 that each firm will have the same preference list of applicant as $\{q_i\}_{i=1}^n$*

Proof. For a particular firm u, denote $w(u, v_i)$ by x_i. Note that x_i is sampled from $N(q_i, \sigma^2)$. What we hope to calculate is the probability that for any pair of x_i and x_j where $i \neq j$, $x_i < x_j$ iff $q_i < q_j$.

WLOG, we assume $q_1 > q_2 > ... > q_n$. And we are going to give a lowerbound for $\Pr(x_1 > x_2 > \cdots > x_n)$.

$$\begin{aligned}\Pr(Pr(x_1 > x_2 > \cdots > x_n) &= \Pr(\bigcap_{i=2}^{n}(x_{i-1} > x_i)) \\ &= 1 - \Pr(\bigcup_{i=2}^{n}(x_{i-1} > x_i)) \\ &\geq 1 - \sum_{i=2}^{n}\Pr(x_{i-1} \leq x_i)\end{aligned}$$

Given that $q_i - q_{i-1} \geq \delta_{min}$, we have

$$\begin{aligned}\Pr(x_{i-1} \leq x_i) =& \Pr(x_i - x_{i-1} \geq 0) \\ =& \Pr((x_i - q_i) - (x_{i-1} - q_{i-1}) \geq q_{i-1} - q_i) \\ =& Pr(a - b \geq q_{i-1} - q_i) \\ \leq& \Pr(a - b \geq \delta_{min}),\end{aligned}$$

where a and b are sampled independently from $N(0, \sigma^2)$.

Considering the random variable $a - b$, it's the same as the random variable $a + b$, i.e., the sum of two identical normal distributions. Thus $a - b$ follows another normal distribution $N(0, 2\sigma^2)$. By Chebyshev's inequality:

$$\Pr(x_i \leq x_{i-1}) \leq \Pr(b - a \geq \delta_{min}) = \frac{1}{2}\Pr(|b - a| \geq \delta_{min}) \leq \frac{\sigma^2}{\delta_{min}^2} = \frac{1}{\psi^2}$$

To sum up we have

$$\Pr(x_1 > x_2 > \cdots > x_n) \geq 1 - \frac{n-1}{\psi^2}$$

Thus the probability that each firm has the same preference list as $\{q_i\}_{i=1}^n$ is no less than $(1 - \frac{n-1}{\psi^2})^m$. Given that $m \leq n$ and $\psi \geq \omega(n)$, this probability approaches 1 when n goes to infinity. $\square$

In the following part we assume that each firms has the same preference list as $\{q_j\}_{i=1}^n$.

For convenience, we allow each firm to send offers even after it has been matched. That is to say, it can still send virtual offers (although virtual offers will be rejected all the time). By our assumption, if an applicant receives an offer from a firm, every other firm would also send her an offer which might be virtual. Note that, in the algorithm every firm sends out offers at most m times, thus no more than m applicants would receive offers. This follows that once receiving offers, the applicant can always see "real" offers, and choose the best from them.

Denote the set of all the applicants who receive offers by S. In *TODO* paper it is proved that for each applicant v who is among the best m applicants, $\Pr(v \in S) \geq \frac{r}{n}\ln(\frac{n}{r})$ which is a constant.

Lemma 4.9. *If all firms have the same preference list as $\{q_i\}_{i=1}^n$, with constant probability, each of the best m applicants (with highest qualities) will be matched to her best firm.*

Proof. Assume the coming order of applicants is τ, let $s_{\tau,i}$ be the i-th applicant who receives offers. Fix an applicant v who is one of the best m applicants.

First, for every τ where $s_{\tau,j} = v$ and $j > 1$, by swapping the position between $s_{\tau,j-1}$ and v we can obtain a new order τ'. In the new coming order, v becomes the $(j-1)$-th to receive offers, i.e., by algorithm $s_{\tau',j-1} = v$. Clearly, for two different coming order τ_1 and τ_2 with $s_{\tau_1,j} = s_{\tau_2,j} = v$, the corresponding new orders τ_1' and τ_2' are also different. Thus $|\{\tau|s_{\tau,j-1} = v\}| \geq |\{\tau|s_{\tau,j} = v\}|$. Therefore $\Pr_\tau(s_{\tau,j-1} = v|v \in S) \geq \Pr_\tau(s_{\tau,j} = v|v \in S)$ for all $j > 1$.

Now given that $s_{\tau,j} = v$, among m offers v has received, $j-1$ of them are virtual and must be rejected. If the best offer for v is among the left $m-j+1$ ones, then v will get her best offer. Since all the weights of edges incident to v are generated independently from the same distribution, this event occurs with probability $\frac{m-j+1}{m}$ and it is decreasing by the growth of j, therefore

$$\Pr(v_i \text{ gets her best}|v_i \in S) = \sum_{j=1}^{m} \Pr(s_{\tau,j} = v_i|v_i \in S) \times \frac{m-j+1}{m}$$

We know that $\sum_{j=1}^{m} \Pr(s_{\tau,j} = v_i|v_i \in S) = 1$, by Chebyshev's sum inequality:

$$\Pr(v_i \text{ gets her best}|v_i \in S) \geq \frac{1}{m}\sum_{j=1}^{m} \frac{m-j+1}{m} \geq \frac{1}{2}$$

Combine this with $\Pr(v \in S) \geq \frac{r}{n}\ln(\frac{n}{r})$, and it's done. □

Proof of Theorem 4.7. With Lemma 4.8, what we need to show is that given all firms have the same preference list as $\{q_i\}_{i=1}^{n}$, $\frac{E[ALG]}{E[OPT]} \geq c$ for some constant $c > 0$.

Denote the set of the best m applicant by T. In this situation, $E[OPT] \leq \sum_{v_i \in T} \max_{u \in U} w(u, v_i)$.

According to Lemma 4.9, the algorithm grantees that for every $v_i \in T$, she will be matched to her best firm with constant probability. Which means $E[ALG] \geq \sum_{v_i \in T} c \times \max_{u \in U} w(u, v_i)$ for some constant $c > 0$.

Which concludes the result. □

Corollary 4.10. *Each firm has a probability of $\Omega(\frac{1}{m})$ to obtain the best applicant.*

Proof. By Lemma 4.8, with probability approaching 1 that all firms consider the same applicant as the best. Denote the best applicant by v. By the fact that $Pr(v \in S) \geq \frac{r}{n} \ln(\frac{n}{r})$ is a constant, v would be matched to some firm with constant probability. Since there is no difference between the firms, each firm has a probability of $\frac{1}{m}$ to be chosen. $\square$

Note that in this setting all firms are facing nearly the same situation. Because every firm wants good applicants and here 'good' means almost the same for them. When competing with other firms, the result relies more on applicant's choice instead of their own strategies.

Chapter 5 Conclusion

REFERENCE

[1] KUHN H W. The Hungarian method for the assignment problem[J]. Naval research logistics quarterly, 1955, 2(1-2):83–97.

[2] KARP R M, VAZIRANI U V, VAZIRANI V V. An optimal algorithm for on-line bipartite matching[C]//Proceedings of the twenty-second annual ACM symposium on Theory of computing. .[S.l.]: [s.n.] , 1990:352–358.

[3] BIRNBAUM B, MATHIEU C. On-line bipartite matching made simple[J]. ACM SIGACT News, 2008, 39(1):80–87.

[4] DEVANUR N R, JAIN K, KLEINBERG R D. Randomized Primal-Dual analysis of RANKING for Online BiPartite Matching.[C]//SODA. .[S.l.]: [s.n.] , 2013:101–107.

[5] AGGARWAL G, GOEL G, KARANDE C, et al. Online vertex-weighted bipartite matching and single-bid budgeted allocations[C]//Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms. .[S.l.]: [s.n.] , 2011:1253–1264.

[6] FELDMAN J, MEHTA A, MIRROKNI V, et al. Online stochastic matching: Beating 1-1/e[C]//Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on. .[S.l.]: [s.n.] , 2009:117–126.

[7] MAHDIAN M, YAN Q. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps[C]//Proceedings of the 43rd annual ACM symposium on Theory of computing. .[S.l.]: [s.n.] , 2011:597–606.

[8] MEHTA A, SABERI A, VAZIRANI U, et al. Adwords and generalized online matching[J]. Journal of the ACM (JACM), 2007, 54(5):22.

[9] BAHMANI B, KAPRALOV M. Improved bounds for online stochastic matching[M]//Algorithms–ESA 2010.[S.l.]: Springer, 2010:170–181.

[10] SLEATOR D D, TARJAN R E. Amortized efficiency of list update and paging rules[J]. Communications of the ACM, 1985, 28(2):202–208.

[11] KLEINBERG R. A multiple-choice secretary algorithm with applications to online auctions[C]//Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms. .[S.l.]: [s.n.] , 2005:630–631.

[12] BABAIOFF M, IMMORLICA N, KEMPE D, et al. A knapsack secretary problem with applications[M]//Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques.[S.l.]: Springer, 2007:16–28.

[13] VAZIRANI V V. Approximation algorithms[M].[S.l.]: springer, 2001.

[14] BABAIOFF M, IMMORLICA N, KLEINBERG R. Matroids, secretary problems, and online mechanisms[C]//Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. .[S.l.]: [s.n.] , 2007:434–443.

[15] DIMITROV N B, PLAXTON C G. Competitive weighted matching in transversal matroids[M].[S.l.]: Springer, 2008.

Acknowledgements

Thx