ANALYSIS OF DECENTRALIZED ALGORITHM FOR ONLINE SECRETARY PROBLEM

ABSTRACT

Inspired by the famous TV shows these years – "The Voice of China". We investigate a new generalization of classical online secretary problem: given weights between a list of firms and applicant, the goal is to design an online algorithm for each firm to find a maximum weighted matching in an online setting with the assumption that the incoming order of applicants is uniform at random. One additional restriction is that, the algorithm should be decentralized. During the execution no communication is allowed between the firms and there is no supervisor doing such an assignment. In this paper we proposed two decentralized algorithms based on the classical stopping rule which optimally solved the online secretary problem. Then we examined their performances in general case respectively and showed their capabilities and shortnesses in several different scenarios. Which gives us the insight on how firms would perform in different market settings and a framework of the firms' strategies.

Keywords: Secretary Problem, Online Matching, Decentralized Algorithm, Competitive Analysis, Stopping Rule

Contents

Chapter 1 Introduction

Recent years, the TV show “The Voice of China” has grown its interests among the people. The show which intends to find the most talented vocalist among a list of contestants has several rounds. In the first round, the *blind audition*, contestants come into the stage one by one and give their performances to show their gifts on singing. At first all coaches are faced towards the audience during the performance. If some of them get impressed, they could hit the button and turn the chair around. In the end of one performance, the contestant cound choose one of the coaches who turn the chair around to be the mentor and continue the competition and even win the show. For coaches, they have a quota on how many team members they can have so they have to be very careful making their decisions on whether to turn the chair or not. It is an interesting problem for each of the coaches that, by adopting what strategies they can find the most talented vocalists to form his/her team so that they can win the show.

A problem of this kind, which intends to find one or several maximal elements from an unknown input stream, has a origination of “online secretary problem” and has been under intensive studies for many years (see Ferguson *et al.* (1989) and Freeman (1983) for a survey).

In the original version of online secretary problem, these is only one firm (just as the coach in blind audition) who wants to hire one secretary. And the secretary comes in for an interview one by one in a random order. After one interview the firm should make a instant decision on whether to accept the applicant as its secretary. All the decisions are non-revokable. It is known that the best strategy for original online secretary problem is stopping rule and with a probability of at least $\frac{1}{e}$ the firm could hire the best secretary.

Since then many variations of online secretary problem have been brought out. Such as

- The *multiple-choice secretary problem* which allows the firm to hire more than one secretaries, and it is nearly optimally solved in Kleinberg (2005). It also adopts a simple algorithm in Babaioff et al. (2007) which has a good performance.

- The *knapsack secretary problem* which assume that each applicant has a cost if the firm hires her and the firm could hire as many secretaries as possible if the budget is feasible. This problem also has a simple but effective algorithm in approximation scheme Babaioff et al. (2007).

- The *matroid secretary problem* which generalized the feasible set of applicants being chosen to the matroid structure. Several subproblems of it have been found that there exists an elegent solution in Dimitrov and Plaxton (2008) Babaioff et al. (2007), but for the general case the problem remains open.

- *Online secretary problem with time discounting* which involves the opportunity cost as studied in **?** Rasmussen and Pliska (1975) Gershkov and Moldovanu (2007). It assume that the weights of applicants may discount as the time grows.

- The *incentive compatibility* is also an interesting topic.

In this paper we will be considering the model as the blind audition in television show The Voice of China. For simplicity we assume that the quota for each coach is just 1. And we will present our algorithms for it and analysis whether they could achieve good performances in different scenarios. For analyzing the performance of online algorithm, we use the powerful tool "competitive analysis" proposed in Sleator and Tarjan (1985) by comparing its outcome to the optimal offline algorithm's outcome.

In another perspective, we could also describe this problem in an online matching way. You have an underline weighted bipartite graph with edge weights unknown to us. In an online setting you have to find a matching with the largest possible weight while revealing the vertex one by one. By revealing one vertex it means that you can see the weights of all the edges incident to it. With no assumption, in Khuller et al. (1994) they proposed an online deterministic algorithm for online weighted matching and claimed this is optimal. And if randomization is allowed, the optimal algorithm for unweighted case has been found out in Karp et al. (1990). Even more if we assume that the revealing order is uniform at random, many algorithms have been proposed in recent years and they did achieve a better performance than the old ones. Such as Aggarwal et al. (2011), Feldman et al. (2009), Mahdian and Yan (2011), Mehta et al. (2007) and Bahmani and Kapralov (2010). Most of them was based on linear programming.

One common problem of those algorithms is, they require the firms (or the coaches) to cooperate or there should be one supervisor doing the assignment. In this paper, we are going to propose some decentralized algorithms. By decentralization it means that the firms could not communicate with each other and have no global information during the execution, and there are no supervisor doing the assignment.

In chapter 2, we are going to introduce some basic notions. From the formal definition of graph matching to its online version, and many variations of online secretary problem with their solutions, as well as the details about competitive analysis.

In chapter 3, we are going give a formal description of our model and the objective functions. Then we will propose two simple decentralized algorithms and check whether they could solve this problem in general case.

In chapter 4, we are going to propose some models for more specific scenarios. And see how could the two algorithms proposed in chapter 3 could manage these situations.

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 2.1 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).

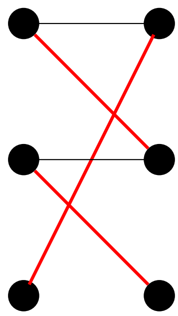

Figure 2.1 The red edges forms a maximum matching in graph

	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 2.2 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 2.2, 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

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

□

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 Matroid 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 algebraic 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 solution forms 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 at the same time.

- *Gammoid*. For example, given a graph, some travelers start their journey from an identical source. And each of them has her own destination. The goal is to find pathes for a subset of them with largest possible value such that the capacities on the edges are not violated.

- *Graphic matroid* which is less natural motivated. With a edge-weighted graph, by revealing the edge weight one by one, the goal is to find an acyclic subgraph whose total weight is maximized.

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 the problem of whether general matroid secretary problem has a constant competitive approach still remains open. The best postitive result so far 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 based on the classical stopping rule. Similar to the well known stopping rule, this algirthm consists of a observation phase and a selection phase. Starting by observing half of applicants, let t be the best score of them. Then choose a scale parameter j uniformly at random from $\{1, \ldots, \log_2 k\}$. In the selection phase pick all applicants whose score is at least $\frac{t}{2^j}$ if it is still feasible. The intuition behind the effectiveness is, given the optimal solution $(v_1, \ldots, v_k)$, the probability of threshold $\frac{v}{2^j}$ lies below v_i is $\Omega(\frac{1}{\log_2 k})$, then it follows by the exchange property of matroid and linearity of expectation.

2.3.3 Time Discounting

The previous discussion has treated the value of an element as independent of when it was selected. However, there are certain natural cases which the classical secretary problem does not address: e.g., it does not capture the opportunity cost incurred due to delay in selecting an element. For example, when seeking to purchase a house, we might think of choosing a slightly suboptimal house at the beginning of the experiment (and being able to occupy it for the entire period) as being more desirable than a long wait to pick the most desirable house.

We model such a problem as the *discounted secretary problem*, where we are given a time-dependent "discount" function $d(t)$ for every time step t: the benefit derived from choosing an element/secretary e with value $v(e)$ at time t is the product $d(t) \cdot v(e)$. The value of a set S is thus $\sum_{x \in S} v(x) \cdot d(\pi(x))$, where $\pi(x)$ denotes the arrival time of x in the random ordering of bids. In the example above, the discount function $d(t)$ is a monotone decreasing function of t, but in general the discount function may be more complicated due to other considerations. For example, our financial situation may improve over time and waiting longer may get us a better mortgage rate, so our discount function $d(t)$ may increase up to some point in time, and then decrease. We assume here that the discount fucntion $d(t)$ are known beforehand to the algorithm designer; they are not revealed online.

Firstly, even the original secretary problem (i.e., choosing a single element from a sequence of n elements) becomes significantly more difficult with arbitrary discount factors. In Babaioff et al. (2009) they proved that no online algorithm for the resulting discounted secretary problem can achieve a better competitive ratio than $\Omega(\frac{\log n}{\log \log n})$. Their main idea is to construct a discount function d and a family of instances $\mathcal{I}_1, \ldots, \mathcal{I}_{2c}$ such that no randomized algorithm can be $\Theta(\frac{\log n}{\log \log n})$-competitive on all the $\mathcal{I}_t$'s.

On the positive side, In Babaioff et al. (2009) they also presented an online algorithm as follows. Let d_{max} be the maximum discount and let v_{max} be the maximum value. For $c \geq 1$ let $I_c = (2^{-c} d_{max}, 2^{-(c-1)} d_{max}]$ be the interval defining the c-th discount class, and let $P_c = \{i \in [n] : d(i) \in I_c\}$ be the set of times that have discount value in class c. Our algorithm $\mathcal{A}$ chooses a $c \in [3 \log n + 2]$ uniformly at random, and then runs the classical secretary in the time steps $i \in P_c$. Note that $\mathcal{A}$ uses no knowledge of $E[OPT]$. This algorithm turns out to be $O(\log n)$-competitive for the discounted secretary problem, nearly matching the lower bound.

Surprisingly, the problem becomes much easier with a small amount of information about the input — merely knowing the expected maximal benefit in advance (as is implied, for example, by knowing the unordered set of values) leads to dramatic improvements. In Babaioff et al. (2009) they showed that knowing a good estimate Z of $E[OPT]$ enables one to design a constant competitive algorithm: Given that $Z \leq E[OPT]$, pick the first element e seen (say, at time j) that satisfies $v(i) \cdot d(j) \geq Z/2$. This is a simple 4-competitive algorithm.

However, In Babaioff et al. (2009) they also exhibited that even knowing the entire unordered set $\{v(x)|x \in \mathcal{U}\}$ of values in advance, an online algorithm cannot be better than $\sqrt{2}$-competitive. To be specific, for any $\epsilon > 0$, every algorithm $\mathcal{A}$ for the discounted secretary problem has $E[OPT] + E[\mathcal{A}] \geq \sqrt{2} - \epsilon$, even when $\mathcal{A}$ knows the set of values (and hence $E[OPT]$) in advance.

Note that, all the algorithms discussed above are monotone, and thus can be converted to truthful mechanisms. Extensions of the discounted secretary problem to various matroid domains are also discussed in Babaioff et al. (2009).

Additional positive results have been obtained for the discounted secretary problem with specific "well-behaved" discount functions like $d(t) = \beta^t$ in Rasmussen and Pliska (1975) or $d(t) = \sum_{i=1}^{n} \beta^t$ in M. Mahdian and Pennock (2008) for some fixed $\beta < 1$, as well as in a continuous-time model in which the values of the elements are independently and identically distributed and element arrivals are given by some renewal stochastic process in Gershkov and Moldovanu (2007).

2.3.4 Incentive Compatibility

Our motivating examples have been drawn from various online auction scenarios. Yet we have so far ignored incentive issues: the problem that the bidders may not reveal their values or other parameters correctly if a misrepresentation stands to benefit them. Fortunately, all of the algorithms described so far are *value-monotonic*: each time a new element x is about to be observed, there is a predetermined threshold $u(x)$ (possibly infinite) such that x will be selected if and only if $v(x) \geq u(x)$. Value-monotonicity directly implies that the algorithms are truthful, so long as bidders can only manipulate their declared value, and not — for example — their arrival time. To convert the algorithm into a truthful mechanism, one simply charges each bidder x a price of $u(x)$ if he is selected, and 0 otherwise.

The situation becomes notably more complex if the agents also have some control over their arrival times. For example, a person shopping for airline tickets may choose not to purchase them at the earliest possible time, instead preferring to wait until a later date in hopes of a price reduction. To model agents' ability to strategize about their arrival time, we assume that each agent has an arrival time $a(x)$ and departure time $d(x)$, such that the agents are randomly ordered by arrival time. The agent's reported arrival and departure times $\hat{a}(x), \hat{d}(x)$ may differ from the actual arrival and departure times. If agent x is selected during the time interval $[a(x), d(x)]$, he receives a value of $v(x)$; otherwise, he receives no value for being selected. The agent's ability to manipulate his reported arrival and departure times is constrained by his true arrival and departure times; some common assumptions are *no early arrivals* $(\hat{a}(x) \geq a(x))$, *no late departures* $(\hat{d}(x) \leq d(x))$, or both. These assumptions are justified if the agent's presence can be directly verified, or (in the case of "no early arrivals") if we think of $a(x)$ as the time at which the agent first becomes aware of the existence of the auction or of his own desire to participate in it.

Some of the algorithms discussed earlier in this article can be transformed into truthful online mechanisms in the no-early-arrivals model, with little or no loss in the competitive ratio. Specifically, a variant of Dynkin's algorithm is truthful and e-competitive in the no-early-arrivals model Hajiaghayi et al. (2004), and Kleinberg's $(1+O(k^{-1/2}))$-competitive algorithm for the k-choice secretary problem in Kleinberg (2005) can be transformed into a truthful and $(1+O(k^{-1/2}))$-competitive mechanism for the k-choice secretary problem in the no-early-arrivals model. The basic idea for both of these results is to modify the original algorithm so that whenever it sets a new threshold price, it scans through the list of agents who have not yet departed and selects those whose value exceeds the new threshold, even if they arrived much earlier. At the departure time of any agent who was selected as a winner of the auction, the agent is charged a price equal to the minimum threshold price attained during the agent's reported arrival-departure interval, even if this price is lower than the threshold price at the time the agent was selected as a winner.

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 **??**, 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

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

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 $\{v_1, v_2, \ldots, v_n\}$.

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

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

To sum up:

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

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

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

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

In our daily lives, the score that firm u grades v is often dependent on (i) the needs of u and (ii) the inherent quality of v. While the first one is usually uncertain and hard to quantify. So in the following part, we are going to discuss how the second term would affect firms' strategy.

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 ranking model discussed in the previous section. Here we assume that D_i is a gaussian distribution $N(q_i, \sigma^2)$ where q_i is the inherent quality of applicant v_i and the standard deviation σ is a fixed constant.

4.2.1 Simultaneous stopping rule vs. Gaussian Model

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 $\frac{E[ALG]}{E[OPT]} \geq \alpha$.

First we have to define some parameters: 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.

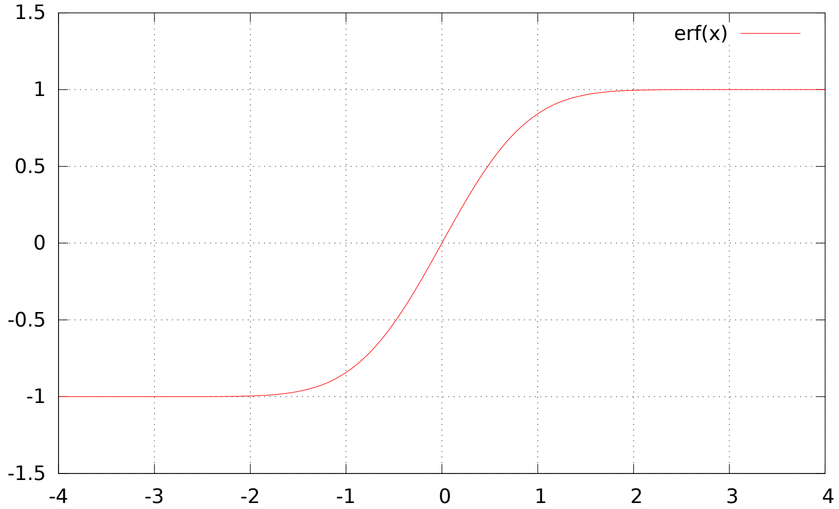

Figure 4.1 Plot of error function

Together with the events which is defined in notation 4.2, 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.

The proof of this theorem shares the same idea with the one for Theorem 4.1. We are going to argue that each firm has a large probability to send its best applicant an offer, and with large probability no other firm would propose to this special applicant. The first two propositions below are established to ease the calculation, and then follows the key lemma which directly implies the main theorem.

First of all we will introduce the error function we use in the analysis:

Definition 4.4. *The error function*

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

This is a monotonically increasing function over $\mathbb{R}$ with value range $(-1, 1)$ as shown in figure 4.1. It has some interesting properties. For example, it is related to the cumulative distribution Φ for gaussian distribution with mean δ and standard deviation σ:

$$\Phi(x) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{x - \delta}{\sqrt{2}\sigma}\right)$$

And the derivative of error function is

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

Then it is time to begin our proof.

Proposition 4.5. *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 $x_i = w(u, v_i)$ be sampled from $N(q_i, \sigma^2)$. WLOG assume that $q_i = 0$.

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

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

Now take derivative with respect to q_j:

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

which implies the result. □

Proposition 4.6. *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

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

Knowing that

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

by Mean Value Theorem, we have

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

Therefore

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

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

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

Which implies the result.

□

Lemma 4.7. *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. Fix 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 4.5 and 4.6 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 $c' = 1+\frac{c}{\sqrt{2\pi}}$. Assume that $m \leq \alpha n$ 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 when n goes to infinity,

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

Therefore we have shown that, for each firm u, it has at least a constant probability to get its best applicant. □

With Lemma 4.7 we could know that simultaneous stopping rule is almost an optimal strategy for each of the firms. It grantees that each firm 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 Simultaneous stopping rule with m slot vs. Gaussian Model

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. Here preference list of a firm u stands for the order of applicants according to the scores u graded them. 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 are going to claim that simultaneous stopping rule with m slots can solve the problem.

With the same model define before, and correspondingly, we define some parameters that $\delta_{min} = \min_{i \neq j} |q_i - q_j|$, and $\psi = \frac{\delta_{min}}{\sigma}$. We claim that

Theorem 4.8. *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. We will break the proof of this theorem into two parts. The first we are going to state that most of the time, firms will have the same preference list according to the scores the graded applicants. And the second, in this situation, the best m applicants would have large possibility of getting an offer.

Lemma 4.9. *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 are going 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 $\Pr(x_1 > x_2 > \cdots > x_n)$.

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

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. □

In the following part we assume that each firms has the same preference list as $\{q_j\}_{i=1}^n$. Then we are going to show that in this situation, the best m applicants could have a very good response – being matched to their desired firms – with high probability.

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 according to the protocol. Since the preference list of all firms are identical. Note that, in the algorithm every firm sends out offers at most m times, thus no more than m applicants would have received offers. Which implies that once the applicant received offers, someone of them must be "non-virtual" and the applicant could choose the best offer among them.

Denote the set of all the applicants who receive offers by S. According to lemma 2.3 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 4.10. *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 incoming 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.

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 happens 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 $\sum_{j=1}^{m} \Pr(s_{\tau,j} = v_i|v_i \in S) = 1$. If we can show that $\Pr(s_{\tau,j} = v_i|v_i \in S)$ is also decreasing by the growth of j, 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 $\Pr(v \in S) \geq \frac{r}{n}\ln(\frac{n}{r})$ in lemma 2.3, and it's done.

Now let's proof that $\Pr(s_{\tau,j} = v_i|v_i \in S)$ will decrease when j becomes greater. 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 this new incoming 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 $j > 1$. □

Proof of Theorem 4.8. With Lemma 4.9, 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 4.10, 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 implies the result. □

Corollary 4.11. *Each firm has a probability of $\Omega(\frac{1}{m})$ to obtain the best applicant.*

Proof. By Lemma 4.9, 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. □

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.

4.3 Future works

In the section of Gaussian model, we discussed two ways on how the edge weights are generated. In section 4.2.1 we showed that if the "correlation" between the scores of one applicant is weak enough – i.e. the preference lists of firms are highly distinguished – simultaneous stopping rule would do a good job. On the other hand, in section 4.2.2, if the distributions of one applicant's scores are highly correlated – i.e. the preference lists of firms are quite similar – by extending the size of threshold set simultaneous stopping rule with m slots could achieve a good performance.

So intuitively we believe that there is a relation between how similar the firms' preference lists are and how many thresholds each firm should set. For example, 1 for all preference lists are totally independent, m for they are the nearly the same to obtain a constant competitive ratio for both social welfare and individuals.

It will be interesting to dig out a correlation parameter between the firms' preference lists. And hopefully the following conjecture will hold.

Conjecture 4.12. *Under the framework of simultaneous stopping rule, there exists a function f and a parameter ρ which depends only on the distributions of the edge weights such that, simultaneous stopping rule with $f(\rho)$ slots could achieve a constant competitive ratio.*

Chapter 5 Conclusion

In this paper we presented one variation of the classical online secretary problem where the idea originated from the famous television shows. We also expressed this problem in the perspective of weighted online bipartite matching.

The main contribution of this paper is the analysis of online decentralized algorithm. We have proposed two simple decentralized algorithms to solve the weighted online bipartite matching problem based on the optimal algorithm for classical online secretary problem – stopping rule.

For more specific models, we use the simultaneous stopping rule to solve the problem where the correlation between edge weights are nearly independent and use the simultaneous stopping rule with m slots where the firms' view on applicants are nearly identical. And we have shown respectively that these two algorithms are nearly optimal within a constant factor in different scenarios for firms and applicants. Then we intuitively describe the relation between the market's setting and the strategy that each firm adopts under this framework.

REFERENCE

[1] FERGUSON T S, et al. Who solved the secretary problem?[J]. Statistical science, 1989, 4(3):282–289.

[2] FREEMAN P. The secretary problem and its extensions: A review[J]. International Statistical Review/Revue Internationale de Statistique, 1983:189–206.

[3] 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.

[4] 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.

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

[6] 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.

[7] RASMUSSEN W T, PLISKA S R. Choosing the maximum from a sequence with a discount function[J]. Applied Mathematics and Optimization, 1975, 2(3):279–289.

[8] GERSHKOV A, MOLDOVANU B. The dynamic assignment of heterogenous objects: A mechanism design approach[M].[S.l.]: Centre for Economic Policy Research, 2007.

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

[10] KHULLER S, MITCHELL S G, VAZIRANI V V. On-line algorithms for weighted bipartite matching and stable marriages[J]. Theoretical Computer Science, 1994, 127(2):255–267.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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

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

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

[19] 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.

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

[21] BABAIOFF M, DINITZ M, GUPTA A, et al. Secretary problems: weights and discounts[C]//Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms. .[S.l.]: [s.n.] , 2009:1245–1254.

[22] M MAHDIAN P M, PENNOCK D. The secretary problem with durable employment[J]. personal communication, 2008.

[23] HAJIAGHAYI M T, KLEINBERG R, PARKES D C. Adaptive limited-supply online auctions[C]//Proceedings of the 5th ACM conference on Electronic commerce. .[S.l.]: [s.n.] , 2004:71–80.

Acknowledgements

First of all, I have to thank to my research supervisor, Dr. Pinyan Lu from Microsoft Research Asia and Dr. Ning Chen from Nanyang Technological University. Without their assistance this paper would have never been accomplished.

Thanks to my supervisor in Shanghai Jiao Tong University – Prof. Yong Yu who has always been nice and kind for his dedications to my study and other affairs. It was he who guide me to the research area of computer science.

Thanks to my classmates Peihan Miao, Youer Pu, Xiangru Huang, Xinchen Yan as well as Xiaohui Bei and Andrea Pisoni for their company during my internship at Singapore where most of the work was done. We have had a wonderful time there.

And I'll give my special thanks to Peihan Miao who was my partner. We had been discussing the massive details and she has always been patient while listenning and correcting my mistakes.