Strategies When Server can Serve One Type of Jobs Together *

1st Xinyi Zhao
Institute for Interdisciplinary Information Sciences

Tsinhua University
Beijing, China
xyzhao18@mails.tsinghua.edu.cn

***Abstract*—In reality,many jobs of the same type can be done together. Sever spends less time doing them together than doing them one by one. The paper considers the queuing problem about the jobs that can be done together. We consider the situations with waste and without waste. We introduce three different strategies:MASTFS,FCFS and Mechanic and discuss the limitations and benefits of them. Also, we do some experiments trying to compare them. And find that MASTFS perform well when there's no waste and FCFS and Mechanic performs well when there's waste. And each strategy perform well on some extreme situations.**

I. Introduction

Many papers have covered the waiting time for the customers outside the restaurants or other shops. And they have often supposed that the service time of one customer follows one special distribution, like exponential distribution. But in reality, some sellers will smartly do some things to decrease the mean response(waiting) time for the sellers. For example, if in the restaurant, two customers require the same dishes, the chief will smartly do these two same dishes together and save the time. Therefore when there's only one chief, the second customer can wait for less time. This strategy not only decreases the possibility of customers feeling angry, but also can decrease some costs.

In this paper, we will consider a special queueing problem. All the jobs of the same type can be done together. And the server spends less time doing them together than doing them one by one. In fact, finding the best strategies for this problem is of great importance for the sellers in reality. With take-out becoming more and more popular, the sellers should cook as quickly as possible trying to decrease the waiting time for the customers. But the sellers don't know the arrival orders beforehand. In our problem, we don't care about the special restaurants where people must order the dishes beforehand or the expensive restaurants with good service.

In the second part of the paper, we will explicitly give our model. And in the second part of the paper, we will discuss some strategies. We first consider the offline policy and then consider the policy without waste ans waste. And we will give some conclusions, some of which have closed forms and some don't have. So in the fourth part of the paper, we will do some experiments and give a more intuitive explanation of every strategy. In the fifth part, we will give some conclusions and bring about more questions that can be finished in the future.

II. Model

To make the work more explicitly, we do some assumptions. We have n types of work. First, we suppose the interarrival time between each work follows exponentially distribution, i.e.,$v_i \sim Exp(\lambda_i)$. And the service time for each work is deterministic, i.e., the service time for type i job is S_i. And in the following sections, we will do some more assumptions about the service time. For example, the service time for doing k jobs of the same type i is also S_i. Or the service time for doing k jobs of the same type i is the function $f(k,i)$. To satisfy the condition that doing work together is better than doing them one by one, $\frac{f(k,i)}{k} < \frac{f(k-1,i)}{k-1}, f(1,i) = S_i$.

And for the server, we suppose we have just one server(machine). And the service time is just the same as the service time of jobs. And our goal is to design the strategy for the server trying to satisfy the requirements of the customers(jobs).

III. Strategies Under Different Assumptions

In this section, we will give some special conditions and design and analyze the strategies under these conditions. And then we will compare the strategies and give the strategies given by the conditions.

A. *Offline Policy*

We first consider one extreme condition. All the works appear at the same time. Then we should minimize the mean response time. Suppose we now have just two types of work, a and b. And the number of works of type a and b is x_a and x_b. The service time of a and b is S_a and S_b. Then if we can finish all the work of type a or type b in just S_a and S_b time. Then for the strategy that we first do work a and then do work b, the mean response time is

$$\begin{aligned} E[T]^1 &= \frac{1}{x_a + x_b}\left(x_a E[T_a] + x_b E[T_b]\right) \\ &= (S_a x_a + S_a x_b + S_b x_b) \end{aligned}$$

And the mean response time for the strategy that we first do type b and then do type a is

$$E[T]^2 = \frac{1}{x_a + x_b} (x_a E[T_a] + x_b E[T_b]) \\ = \frac{1}{x_a + x_b} (x_a S_b + x_a S_a + x_b S_b)$$

Then by comparing the two strategies, we can find that when $S_a X_b < S_b X_a$, such that $\frac{S_a}{X_a} < \frac{S_b}{X_b}$, strategy 1 has better mean response time.

Now if we have k types of job together. The optimal solution is to sort the works by $\frac{S_i}{X_i}$ from small to big. The reason is that we for two types i and j, i will influence j if and only if we do type i before type j. And the total mean waiting time is the total influence of all the pairs of jobs. Thus if we optimize each pair, we can optimize the total waiting time. And try to understand the $\frac{S_i}{X_i}$, it's just the average service time per work. So we just serve the job with minimal average service time.

Also, if we change the service time from S_i to $f(k, i)$, the solution is also the same. We just sort the jobs by $\frac{f(k_i, i)}{x_i}$. But this is just the simple condition. In the following sections, we will consider the on-line policy. All the jobs come following exponential distribution

B. Online Policy with No Waste

Sometimes the servers will do more jobs than the customers require. For example, if in the queue, five people each require one piece of bread. For the seller, how much bread should he toast? If he toasts just 5 pieces, then the new buyer will wait for the next time he toasts. If he toasts more than 5 pieces, it is possible that no buyer will come before the bread cools down and the seller will waste the bread he has toasted. In this section, we will consider the situation that the seller will not do anything that may cause wasting. And in the next section, we will consider the situation that the seller will use the possibility of wasting to increase profits.

In this section, we will give several strategies. Regretfully, some strategies don't have closed form. But we can discuss the situation of two types and also use some intuitions trying to analyze the strategies.

1) MASTFS(Mean Average Service Time Service First): The intuition of the strategy comes from the analysis from the off-line policy. We can easily generate a strategy that every time we finish one job, we choose to do the work with mean average service time, i.e. we first choose to do the work with minimal $\frac{S_i}{X_i}$.

The problem with this strategy is very trivial. If there is one type of job with high service time, but with very low arrival rate, this type of work will possibly not be served. So the variance of the response time of the jobs is very high.

For example, if there are two types of work a and b. If we just use expectation to calculate the number of jobs of type a and b. The way we deal with this work is to do several a and do several b and so on. So we can writhe the inequality equation as

$$\frac{S_a}{\lambda_a S_a} > \frac{S_b}{\lambda_b (T1 + S_a)}$$

$$\frac{S_b}{\lambda b S_b} > \frac{S_a}{\lambda_a (T2 + S_b)}$$

And we have $T_1 \geq S_b, T_2 \geq S_a$, thus we have

$$\frac{\lambda_b}{\lambda_a} > \frac{S_b}{S_a + S_b}$$

$$\frac{\lambda_a}{\lambda_b} > \frac{S_a}{S_a + S_b}$$

$$\frac{S_a}{S_a + S_b} < \frac{\lambda_a}{\lambda_b} < 1 - \frac{S_b}{S_a + S_b}$$

And the situation can be changed to the situation of k types of job. For any two types of job i and j, should have

$$\frac{S_i}{\lambda_i T} > \frac{S_j}{\lambda_j (T + S_i)}$$

Where $T \geq S_j$. So in this strategy, we require that the mean service time for each type of job is approximately the same.

2) Do work Mechanically: The main problem with MASTFS is that some customers will not be served and leave the restaurant angrily. Now we consider a more naive way to do the work. The seller don't know anything about math. So he just follows some procedures to the work. For example, if there are two types of work, he just serves A, serves B, serves A, serves B...... If there are no A in the queue, he just wait for the time of doing A and then doing B(this is stupid, but you can consider that he only receives the message at time $S_a, S_a + S_b, 2S_a + S_b, 2S_a + S_b \ldots$).

Thus we can calculate the mean waiting time for each type of work. For type A, the probability that the server is doing A when the new job arrive is $\frac{S_a}{S_a + S_b}$, the probability that the server is doing B is $\frac{S_b}{S_A + S_b}$. Therefore, we can calculate

$$E[T_Q^A] = \frac{S_a}{S_a + S_b}(E[A_e] + S_b) + \frac{S_b}{S_a + S_b} E[B_e] \\ = \frac{S_a}{S_a + S_b}(\frac{1}{2} S_a + S_b) + \frac{S_b}{S_a + S_b} \frac{1}{2} S_b \\ = \frac{1}{2(S_a + S_b)}(S_a^2 + 2 S_a S_b + S_b^2)$$

And also we can get

$$E[T_Q^B] = \frac{1}{2(S_a + S_b)}(S_a^2 + 2 S_a S_b + S_b^2)$$

They are the same. So $E[T_Q] = \frac{1}{2(S_a + S_b)}(S_a^2 + 2 S_a S_b + S_b^2)$.We can consider the situation that we do like this $A, A, B, A, A, B, A, A, B, \ldots$. And we can rewrite the equation as

$$E[T_Q^A] = \frac{S_a}{2S_a + S_b}(E[A_e]) + \frac{S_a}{2S_a + S_b}(E[A_e] + S_b) + \\ \frac{S_b}{2S_a + S_b} E[B_e] \\ = \frac{1}{2(2S_a + S_b)}(S_a^2 + S_a^2 + 2 S_a S_b + S_b^2) \\ = \frac{1}{2(2S_a + S_b)}(2 S_a^2 + 2 S_a S_b + S_b^2)$$

$$\begin{aligned}E[T_Q^B] = & \frac{S_a}{2S_a+S_b}(E[A_e]+S_a)+ \\ & \frac{S_a}{2S_a+S_b}(E[A_e]) + \frac{S_b}{2S_a+S_b}(E[B_e]+2S_a) \\ & = \frac{1}{2(2S_a+S_b)}(3S_a^2+S_a^2+S_b^2+4S_aS_b) \\ & = \frac{1}{2(2S_a+S_b)}(4S_a^2+4S_aS_b+S_b^2)\end{aligned}$$

And we have

$$E[T_Q] = \frac{\lambda_a}{\lambda_a+\lambda_b}E[T_Q^A] + \frac{\lambda_b}{\lambda_a+\lambda_b}E[T_Q^B]$$

And we solve the inequality that $E[T_Q]^{A-A-B} < E[T_Q]^{A-B}$, we can get that

$$\begin{aligned}\frac{\lambda_a}{\lambda_b} & > \frac{2S_a^3+3S_a^2+S_b^2S_a}{S_aS_b(S_a+S_b)} \\ \Rightarrow \frac{\lambda_a}{\lambda_b} & > 1+\frac{2S_a}{S_b}\end{aligned}$$

If $S_a = S_b$, λ_a should be at least three times λ_b. And we can write the $E[T_Q]$ for $\underbrace{A-A-A-\ldots-A}_{k}-B$.

$$\begin{aligned}E[T_Q^A] = & \frac{(k-1)S_a}{kS_a+S_b}(E[A_e]) + \frac{S_a}{kS_a+S_b}(E[A_e]+S_b) \\ & + \frac{S_b}{kS_a+S_b}E[B_e] \\ & = \frac{1}{2(kS_a+S_b)}((k-1)S_a^2+S_a^2+2S_aS_b+S_b^2) \\ & = \frac{1}{2(kS_a+S_b)}(kS_a^2+2S_aS_b+S_b^2)\end{aligned}$$

$$\begin{aligned}E[T_Q^B] = & \frac{S_b}{kS_a+S_b}(E[B_e]+kS_a)+ \\ & \frac{S_a}{kS_a+S_b}\sum_{i=1}^{k}E[A_e]+(k-i)S_a \\ & = \frac{1}{2(kS_a+S_b)}(S_b^2+2kS_aS_b+kS_a^2+k(k-1)S_a^2) \\ & = \frac{1}{2(kS_a+S_b)}(k^2S_a^2+2kS_aS_b+S_b^2)\end{aligned}$$

Then we can write the mean response time as

$$E[T_Q] = \frac{\lambda_a}{\lambda_a+\lambda_b}E[T_Q^A] + \frac{\lambda_b}{\lambda_a+\lambda_b}E[T_Q^B]$$

And then we can calculate the differentiate of $E[T_Q]$ and find the best k for certain problems. The situation of $B-B-B-\ldots-B-A$ is similar.

And then for k types of work. We can just do as one by one. The advantages of these strategies is that it can decrease the variance. But to find how to arrange the orders and how many works we should do for just one type of work is difficult(we can only solve the equation for finite k). So for any k, we should just consider do the work as $1-2-3-4-\ldots-k-1-2-3-\ldots-k-1-\ldots$. And this strategy is called mechanically doing work.

3) First Come First Serve(FCFS): This is a little bit different from traditional first come first serve. We just do the work in the first of the queue and finish all the work of the same type in the queue. Here is one Markov chain for the situation of just 2 types. This is just like a M/D/1 Markov chain, where $\mu_a = 1/S_a, \mu_b = 1/S_b$

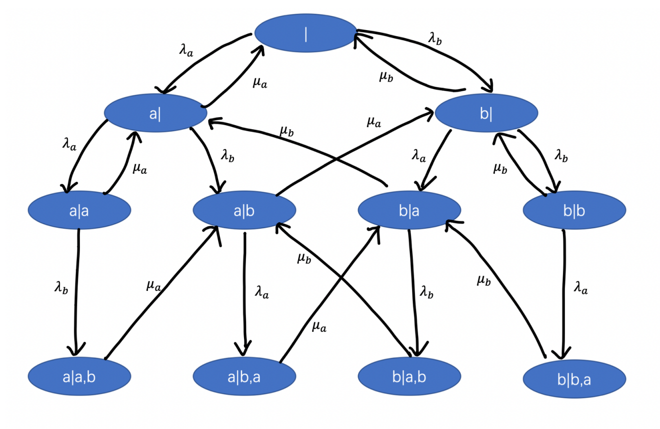

Fig. 1. the State Space for FCFS of 2 types

And we can solve this using the technique of M/D/1. If we change the serve time to the time following exponentially distribution. We can use the property of Markov Chain to solve the balance equations and then calculate the mean response time. But in this paper, we just consider M/D/1 model.

Now if we calculate the situations for k types, there are overall $k \times k!(1+1+\frac{1}{2!}+\frac{1}{3!}+\ldots)+1$ states. And for the state $(X|x_1,x_2,\ldots,x_t)$. We have $(X|x_1,x_2,\ldots,x_t) \rightarrow (x_1|x_2,x_3,\ldots,x_t)$ with rate $\mu_1 = 1/S_1$. And $(X|x_1,x_2,\ldots,x_t) \rightarrow (X|x_1,x_2,\ldots,x_t,x_{t+1})$ with rate λ_{t+1}.

Compare with mechanically doing work method, the method gives more chances to the work of high arrival rate and can also give the chance to the work with large service time. In section 4(experiments), we will try to find under which condition, FCFS is better than mechanically working or MASTFS.

C. Online policy with waste

In this section, we will consider the situation that the seller will do more jobs than the number of jobs in the queue. And all the jobs have a quality guarantee period. Now we just consider one type of job.

First we suppose that the quality guarantee period is 0, i.e, if the server has finished this job and the number of new coming people is less than the jobs we do more than requirements, the more jobs will be wasted. So smartly doing more jobs is important because people don't want to wait for a very long time. If one arrives and find that you have just started a new job, he may leave and don't buy the work since he has to wait for twice the service time if you don't do more. Now first, we suppose that the seller just do one more work. And the patience time for the customer is $\frac{3}{2}S$(this is from reality that people will not want to waste for half time of the true service time). Then the probability that no one come(wasting) is

$$P(\text{waste}) = P(\text{no come in time S}) = e^{-\lambda S}$$

And the expecting number of people leaving the queue. This is just the number of people that exceeds one in time $S/2$.

$$\begin{aligned} E[N^{drop}]^{waste} &= \int_2^\infty (x-1)P(\text{x people arrive in time } S/2)dx \\ &= \lambda\frac{S}{2} - (1 - e^{-\lambda S/2}) \end{aligned}$$

And consider the traditional way we serve the job, we don't do more job. Then the expectations of people dropped is

$$E[N^{drop}]^{tradition} = \frac{1}{2}\lambda S$$

Then suppose the prime cost is c, the profit of one person is r. Then under such situation wasting strategy is better than traditional one

$$cP(\text{waste}) + rE[N^{drop}]^{waste} < rE[N^{drop}]^{tradition}$$

$$\frac{c}{r} < \frac{1 - e^{-\lambda S/2}}{e^{-\lambda s}}$$

If c is small or r is big, the waste strategy is excellent.

Then if consider the situation that we do more than k jobs every time, and the waiting time for people is $(1 + uS)$,$0 \geq u \geq 1$.Then we rewite the equations above

$$\begin{aligned} E[\text{waste}] &= \int_0^k (k-x)P(\text{x people come in time S})dx \\ &= \int_0^k (k-x)\frac{e^{-\lambda S}(\lambda S)^x}{x!}dx \end{aligned}$$

$$\begin{aligned} E[N^{drop}] &= \int_k^\infty (x-1)P(\text{x people arrive in time uS})dx \\ &= \int_k^\infty (x-1)\frac{e^{-\lambda uS}(\lambda uS)^x}{x!}dx \end{aligned}$$

And we can also solve the equation $cP(\text{waste}) + rE[N^{drop}]^{waste} < rE[N^{drop}]^{tradition}$, and get

$$\frac{c}{r} < \frac{\lambda uS - \int_k^\infty (x-1)\frac{e^{-\lambda uS}(\lambda uS)^x}{x!}dx}{\int_0^k (k-x)\frac{e^{-\lambda S}(\lambda S)^x}{x!}dx}$$

Then we can use c and r to determine the maximum of k. On the other hand, we can just calculate the wasting cost as

$$\begin{aligned} E[\text{cost}] &= cE[waste] + rE[N^{drop}] \\ &= c\int_0^k (k-x)\frac{e^{-\lambda S}(\lambda S)^x}{x!}dx + \\ & r\int_k^\infty (x-1)\frac{e^{-\lambda uS}(\lambda uS)^x}{x!}dx \end{aligned}$$

Obviously, the function of $E[cost]$ is convex in the view of integer. So we can always find a k that minimize $E[cost]$

For the situation of the queue of different types of work. We can just calculate k_i for each job and then apply the method of no waste situation. Since it doesn't have closed form we just do some simulations in the next section trying to prove the our analysis and give more intuitions about the problem.

IV. Experiments and Analysis

The models given before just provide some analysis about the model. But we can't get the accurate solution for some models. In this section, we will simulate the process. And try to compare the performance of different strategies. We use programs to simulate the Poisson process and then run for 1,000,000 times. And try to compare under which condition it is better.

A. Two Type with No Waste

In the experiments of two types of jobs, we experiment on three different strategies: FCFS, mechanically working, MASTFS. we try different $\lambda_a, \lambda_b, S_a, S_b$.

First we check that when the two jobs are very similar to each other. We set $\lambda_a = \lambda_b = S_a = S_b$, and change from 1 to 20.

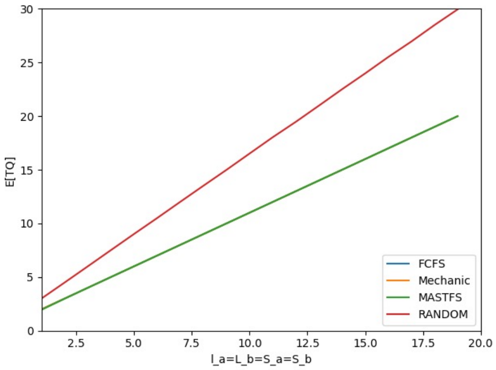

Fig. 2. $\lambda_a = \lambda_b = S_a = S_b$

From the picture, we can see that, three strategies are similar, and all are better than random. So do some changes is better than just random.

Then we set $\lambda_a = \lambda_b = S_a = 1$ and then change S_b from 1 to 100. Here is the result.

And we do the experiments with different λ_b. We also set $\lambda_a = S_a = S_b = 1$.

From these two pictures, we can find that if one job has higher service time, Mechanic way and MASTFS way is the best. And if one has higher arrival rate, Mechanic way has no influence. But FCFS way is better than MASTFS way when the arrival rate is large. But when the arrival rate is not too big, MASTFS is better than FCFS. This satisfies our analysis in the last section, MASTFS do well when the two jobs have the same properties.

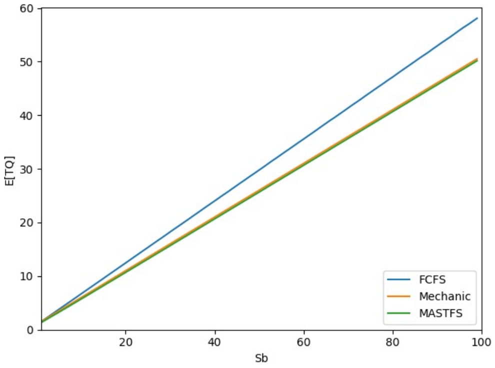

Fig. 3. $\lambda_a = \lambda_b = S_a = 1$, and S_b changes

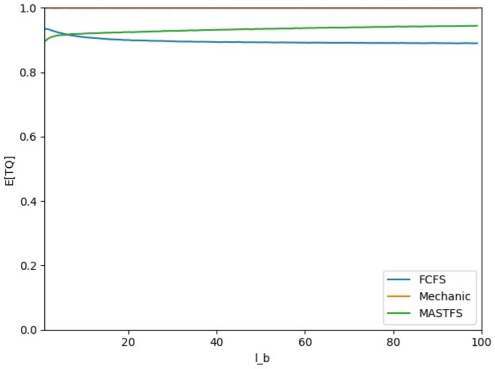

Fig. 4. $\lambda_a = S_a = S_b = 1$, and λ_b changes

So when there are only two types of job, MATFS performs well on non-extreme situations. And FCFS or Mechanic well performs well on some special situations.

B. *Strategies with Waste*

First let we consider the two types strategies MASTFS, FCFS and Mechanic. And we first suppose that the seller can only bear at most one wasting every service, which means he will do one more job every service time.

And we first see the situations when $\lambda_a = \lambda_b = s_a = s_b$, the difference between them is not obvious(shown in the Fig 5).

And if we use $\lambda_a = \lambda_b = 1$,and change $s_a = s_b$, we can see that the all the three strategies behave approximately the same, but after adding one waste job, the situation becomes this(Fig 6)

In fact, FCFS and Mechanic performs nearly well but MASTFS performs less good. This is because that MASTFS depends on the current situation and consider less about the future situation. It is a good offline policy, but it doesn't consider anything about the waste situation. On the contrary, doing things just according to some procedure will depend less on the current situation.

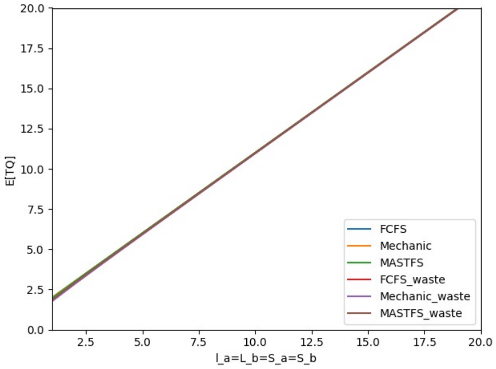

Fig. 5. $\lambda_a = \lambda_b = S_a = S_b$,with one waste

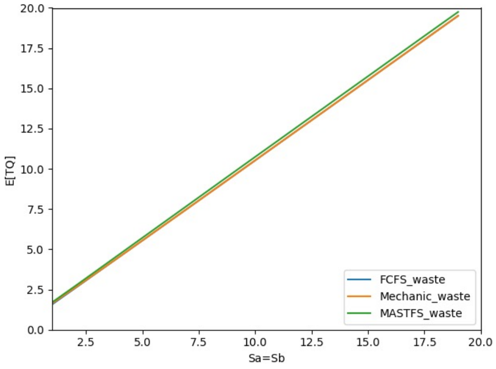

Fig. 6. $\lambda_a = \lambda_b = 1$,$S_a = S_b$ changes ,with one waste

V. Conclusion

Doing work together can decrease the mean waiting time. And it is meaningful in reality. The three strategies introduced before all have advantages and disadvantages. The Mechanic can guarantee that every customer will not wait for too much time. The FCFS, which is accepted by many restaurants, can somehow reduce the mean response time. MASTFS, consider more about the current situation, but may have high variance.

And the situations with and without waste have many differences. This is discussed less in the paper because we don't know much about cost and profits of every job. And if we can bear waste, FCFS and Mechanic performs much better than MASTFS.

But because of the lack of time, there's a lot of can be discussed in the future. For example, we don't consider the situations when there are k types of jobs. And the situations that the customers will leave when they leave for too much time. And if different jobs have different profits, which one should be done first in order to gain more profits ans satisfy as many customers as possible.