Probability in the Engineering and Informational Sciences, **31**, 2017, 226–238.
doi:10.1017/S0269964816000176

MULTIPLE SERVER PREEMPTIVE SCHEDULING WITH IMPATIENCE

YANG CAO
Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA 90089, USA
E-mail: cao573@usc.edu

There are n customers that need to be served by m parallel servers ($n \geq m$). Customer i will only wait in queue for an exponentially distributed time with rate λ_i before departing the system. The service time on server j is exponentially distributed with rate μ_j for all customers, and upon completion of service of customer i a positive reward r_i is earned. The non-preemptive problem is to choose, after each service completion, which currently in queue customer to serve next. The preemptive problem is to decide when to preempt a service, and to choose, after each service completion or preemption, which currently in queue customer to serve next. The objective of both problems is to maximize the expected total return. We give conditions under which a list policy is optimal for both problems.

1. INTRODUCTION

Consider a queueing model in which there are n customers in the system that need to be served by m parallel servers ($n \geq m$). Customer i can only wait a random time, called its impatience time, which is exponentially distributed with rate λ_i in queue before entering service, and it will depart the queue if its impatience time expires. A customer cannot depart the system while in service, and a departed customer cannot be served. The service time on server j is exponentially distributed with rate μ_j for all customers, and upon completion of service of customer i a positive reward r_i is earned and i departs the system. In the non-preemptive case, servers are not allowed to preempt customers in service. Each time a service is completed, a decision must be made as to which job in queue should enter service next. On the other hand, in the preemptive case, servers are allowed to preempt customers in service and serve other customers which have not departed the system. The preempted customers then rejoin the queue and their remaining impatience times are still exponentially distributed. For example, it can be considered as a model for upgrading machines which will fail in an exponentially distributed amount of time if the upgrade has not started by then. Technicians may preempt current services and upgrade machines with higher emergency level or higher reward. The remaining impatience times of the preempted machines start over again because of the lack of memory property of exponential distributions, and the upgrades of such machines should be restarted. Therefore, the preemptive problem is to decide when to preempt a service, and to choose, after each service completion or preemption, which currently in queue customer to serve next. However, since the impatience times and service

 0269-9648/16 $25.00

times of customers are exponentially distributed, we only need to consider policies that only preempt when there is a service completion or a customer in queue departs the system. Let R denote the total return of all customers. The objective of both problems is to maximize the expected total return $E[R]$.

For $i_1, \ldots, i_n$ being a permutation of $1, \ldots, n$, we define the list policy $(i_1, \ldots, i_n)$ as the non-preemptive, non-idling policy that initially serves customer i_j on server j, for $j = 1, \ldots, m$, and then after each service completion, elects to serve the remaining customer i_k if $i_1, \ldots, i_{k-1}$ are not in queue. In Section 2, we provide a literature review on needed results. In Section 3, we give a sufficient condition that results in the list policy $(1, \ldots, n)$ being optimal (in the sense of maximizing expected total return) for the multiple server non-preemptive problem. In Section 4, we further study the multiple server preemptive problem and give a sufficient condition that results in the list policy $(1, \ldots, n)$ being optimal. Interestingly, it turns out that in addition to maximizing the expected total return objective function, the list policy $(1, \ldots, n)$ also has the property of stochastically minimizing the makespan under certain conditions for both the non-preemptive (in Section 3) and the preemptive problem (in Section 4). Although idling is not considered in this paper, it can be shown by a sample path coupling argument that a policy that maximizes the expected total return would not idle.

2. LITERATURE REVIEW

Many papers have studied stochastic scheduling, optimal control and performance measures of queuing systems with impatient customers; see, for example, Ross [12], Glazebrook et al. [3], Ward and Kumar [16], Movaghar [7,8], Zeltyn [17], Mandelbaum and Momcilovic [6], and Argon, Ziya and Righter [1]. Among these papers, the most related study is conducted by Ross [12]. In this section, we first review the work by Ross [12] with adaptation to our model, and then provide a survey on the literature that has relevance to our work.

Ross [12] studied the single server non-preemptive problem with service times having general distributions. Results of [12] applying to the case where service times are exponentially distributed gives the following Theorem 1 and Proposition 2.

THEOREM 1: *Suppose there is a single server with rate μ and preemptions are not allowed. If*

$$\lambda_i \uparrow i \text{ and } \frac{r_i \lambda_i}{\mu + \lambda_i} \downarrow i$$

then the list policy $\pi = (1, 2, \ldots, n)$ maximizes the expected total return $E[R]$.

Let the makespan, call it T, be the time until all customers have departed the system.

PROPOSITION 2: *Suppose there is a single server with rate μ and preemptions are not allowed. If $\lambda_i \uparrow i$ then the list policy $\pi = (1, 2, \ldots, n)$ stochastically minimizes the makespan T.*

There are several other papers in the literature that have some relevance to our work. Pandelis and Teneketzis [9] studied the scheduling problem where tasks belonging to N priority classes (higher priorities correspond to higher penalty costs when a loss occurs) arrive to a single or multi server facility. They considered impatience times that are either known to the scheduler or have known probability distributions, and determined properties

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

of dynamic, non-idling, non-preemptive policies that minimize the infinite horizon expected cost due to task losses. Both the groups, that is, Panwar, Towsley and Wolf [10], and Zhao, Panwar and Towsley [18] assumed impatient jobs arrive randomly, and considered scheduling problems with the objective of maximizing the rate of jobs being served within their deadlines. Atar, Giat and Shimkin [2] assumed that customers can be partitioned into classes where customers of each class have the same stochastic characteristics such as arrival rate, impatience rate and service rate. With the objective of minimizing the overall long run average holding cost, they presented and analyzed an asymptotically optimal scheduling policy called the $c\mu/\theta$ rule, where θ corresponds to λ and c is the holding cost. Salch, Gayon and Lemaire [13] studied the class of static list policies for single server queuing models. With the objective of minimizing the expected weighted number of late jobs, they provided sufficient conditions for determining optimal policies among the class of static policies.

There are also several articles in the general context of scheduling with stochastic processing times or stochastic due dates. Pinedo [11] considered the scheduling problems of minimizing the expected weighted sum of completion times and the expected weighted number of late jobs. Towsley and Panwar [15] studied the $G/M/c$ queue in which customers have deadlines and only certain stochastic relationships between the deadlines of eligible customers are known to the scheduler. Two cases were considered: The non-preemptive case where deadlines are until the beginning of service, that is, servers are not allowed to preempt and a customer's deadline is missed if the customer has not entered service before the deadline; and the preemptive case where deadlines are until the end of service, that is, servers are allowed to preempt customers in service (maybe a customer whose deadline is missed while in service) and a customer's deadline is missed if the customer has not completed service before the deadline. They proved that in such two cases, the policy that stochastically minimizes the number of customers missing their deadlines by time t never schedules a customer which is known to have a deadline that is stochastically larger than that of another customer also in the queue. Seo, Klein and Jang [14] assumed that jobs have normally distributed processing times and a common deterministic due date, and presented a non-linear integer programming model that generates near optimal solutions for minimizing the expected number of tardy jobs. Both Jang and Klein [5] and Jang [4] assumed stochastic processing times and deterministic due dates. They considered the effect of variance of job processing time for determining optimal policies with the objective of minimizing the expected number of tardy jobs.

Throughout this paper, suppose there are m servers ($m \leq n$) and the service time on server j is exponentially distributed with rate μ_j for all customers. In addition, let $\mu = \sum_{j=1}^{m} \mu_j$ and let $E_\pi[R]$ denote the expected total return under the list policy $\pi = (1, \ldots, n)$.

3. MULTIPLE SERVER NON-PREEMPTIVE PROBLEM

We now extend results of the single server non-preemptive problem to the multiple server non-preemptive problem. Preemptions are not allowed in this section. For $\{i_1, \ldots, i_k\}$ being a subset of $\{1, \ldots, n\}$ ($m \leq k \leq n$), consider a non-preemptive system that begins with customers $i_1, \ldots, i_k$. Let $\pi(i_1, \ldots, i_k)$ be the list policy $(i_1, \ldots, i_k)$, and let $R(i_1, \ldots, i_k) = E_{\pi(i_1,\ldots,i_k)}[R]$ be the expected total return under $\pi(i_1, \ldots, i_k)$.

We now show in the multiple server case, as an immediate corollary of Theorem 1, that under the condition of that theorem the list policy $\pi = (1, \ldots, n)$ maximizes the expected total return.

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

COROLLARY 3: *If*

$$\frac{r_i\lambda_i}{\mu+\lambda_i} \downarrow i \text{ and } \lambda_i \uparrow i$$

then the list policy $\pi = (1, \ldots, n)$ *maximizes the expected total return* $E[R]$.

PROOF: The times between service completions are independently and identically distributed exponential random variables with rate μ. Thus, after the initial m customers enter service, the system is equivalent to a single server system with service rate μ and it follows from Theorem 1 that the list policy $\pi = (1, \ldots, n)$ should then be applied. Now, to show that it is optimal to initially serve customers $1, \ldots, m$, consider an arbitrary policy that initially serves customers $i_1, \ldots, i_m$, where $\{i_1, \ldots i_m\} \neq \{1, \ldots, m\}$ and then follows policy π after the initial choice. Call this policy π' and let $E_{\pi'}[R]$ denote the expected total return under π'. In addition, let $k = \text{argmax}_{j\in\{1,\ldots,m\}}\, i_j$ and $w = \text{argmin}_{j\in\{m+1,\ldots,n\}}\, i_j$, so $i_w \leq m < i_k$ and $i_w = \min\{i_k, i_{m+1}, \ldots, i_w, \ldots, i_n\}$. Let π'' denote the policy that initially serves customers $i_1, \ldots, i_{k-1}, i_{k+1}, \ldots i_m, i_w$ and then follows policy π after the initial choice, and let $E_{\pi''}[R]$ denote the expected total return under π''. Then it follows from Theorem 1 and $i_w = \min\{i_k, i_{m+1}, \ldots, i_w, \ldots, i_n\}$ that $E_{\pi''}[R] \geq E_{\pi'}[R]$. In addition, if $\{i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_m, i_w\} \neq \{1, \ldots, m\}$, we can use the same argument to improve policy π''. Therefore, by repeating the preceding argument we will finally obtain a policy that initially serves customers $1, \ldots, m$ and has expected total return greater than or equal to $E_{\pi'}[R]$. The optimality of the list policy π then follows. ■

Moreover, it can be shown with the same argument as in Corollary 3 and based on Proposition 2, that the list policy π also has the property of stochastically minimizing the makespan.

COROLLARY 4: *If* $\lambda_i \uparrow i$ *then the list policy* $\pi = (1, 2, \ldots, n)$ *stochastically minimizes the makespan* T.

4. MULTIPLE SERVER PREEMPTIVE PROBLEM

Preemptions are allowed in this section. When a server completes service or a customer in queue departs the system, servers can choose to either continue or preempt the current service. The preempted customers will rejoin the queue.

There are cases where preemptions are useful. For example, suppose that $m = 1, n = 3$ and that for a small positive number ϵ,

$$\frac{r_1\lambda_1}{\mu+\lambda_1} = \frac{r_2\lambda_2}{\mu+\lambda_2} - \epsilon.$$

In addition, suppose $\lambda_1 < \lambda_2$ and λ_3 is small so that customer 3 should be served last. Since $\lambda_1 < \lambda_2$, the time it takes to serve customer 1 then customer 2 (if it has not departed yet) is stochastically smaller than that of serving customer 2 then customer 1. Therefore, when all customers are in the system, the optimal policy should serve customer 1 first because customer 3 will then be more likely to be served. However, if customer 3 departs the queue before the service completion of customer 1 and the departure from queue of customer 2, the optimal policy should then preempt customer 1 and serve customer 2 because $r_1\lambda_1/(\mu+\lambda_1) < r_2\lambda_2/(\mu+\lambda_2)$.

In the remainder of this section, we first present several needed lemmas, and then show that when preemptions are allowed, the list policy $\pi = (1, \ldots, n)$ maximizes the expected

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

total return and has the property of stochastically minimizing the makespan under certain conditions.

The next two lemmas are immediate.

LEMMA 5: *For $j_1, \ldots, j_m$ being a permutation of $i_1, \ldots, i_m$,*

$$R(i_1, \ldots, i_m, i_{m+1}, \ldots, i_k) = R(j_1, \ldots, j_m, i_{m+1}, \ldots, i_k).$$

LEMMA 6: $r_j + R(i_1, i_2, \ldots, i_k) = r_{i_1} + R(j, i_2, \ldots, i_k)$.

LEMMA 7: *If $r_i\lambda_i/(\mu + \lambda_i) \downarrow i$ and $\lambda_i \uparrow i$ then $R(1, \ldots, n-1) = \max_j R(1, \ldots, j-1, j+1, \ldots, n)$.*

PROOF: Consider an $n-1$ customer non-preemptive problem that begins with customers $1, \ldots, n-1$ in system. Based on Corollary 3, for any $j = 1, \ldots, n-1$, we have $R(1, \ldots, n-1) \geq R(1, \ldots, j-1, j+1, \ldots, n-1, j)$. Moreover, since $\lambda_j \leq \lambda_n$ and $r_j \geq r_n$, a standard coupling argument gives $R(1, \ldots, j-1, j+1, \ldots, n-1, j) \geq R(1, \ldots, j-1, j+1, \ldots, n-1, n)$. Thus for any $j = 1, \ldots, n-1$, $R(1, \ldots, n-1) \geq R(1, \ldots, j-1, j+1, \ldots, n-1, j) \geq R(1, \ldots, j-1, j+1, \ldots, n-1, n)$, which yields the result. ■

Based on Lemma 7, we have

LEMMA 8: *If $r_i\lambda_i/(\mu + \lambda_i) \downarrow i$ and $\lambda_i \uparrow i$ then for all $j = 1, \ldots, n$,*

$$r_1 + R(2, \ldots, n) - R(1, \ldots, j-1, j+1, \ldots, n) \geq \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}}.$$

PROOF: For $\{i_1, \ldots, i_k\}$ being a subset of $\{1, \ldots, n\}$, define $P^{i_k}_{i_1,\ldots,i_k}$ as the probability that customer i_k is served under the non-preemptive list policy $\pi(i_1, \ldots, i_k)$ when the system begins with customers $i_1, \ldots, i_k$. Then

$$\begin{aligned}
&r_1 + R(2, \ldots, n) - R(1, \ldots, j-1, j+1, \ldots, n) \geq r_1 + R(2, \ldots, n) - R(1, \ldots, n-1) \\
&\quad = r_1 + r_2 + \cdots + r_{m+1} + \sum_{j=m+2}^{n} r_j P^j_{2,\ldots,j} \\
&\qquad - \left(r_1 + r_2 + \cdots + r_m + r_{m+1}\frac{\mu}{\mu + \lambda_{m+1}} + \sum_{j=m+2}^{n-1} r_j P^j_{1,\ldots,j} \right) \\
&\quad = \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} + r_n P^n_{2,\ldots,n} + \sum_{j=m+2}^{n-1} r_j (P^j_{2,\ldots,j} - P^j_{1,\ldots,j}) \\
&\quad \geq \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}},
\end{aligned}$$

where the last inequality is because $P^j_{2,\ldots,j} \geq P^j_{1,\ldots,j}$ (can be easily proven with a coupling argument). ■

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

Using the preceding lemmas, we can find the optimal policy in a special case.

THEOREM 9: *If*

$$\lambda_i \uparrow i, \min_{1\le j\le m} \mu_j \ge \max_{1\le i\le n} \lambda_i, \text{ and } \frac{r_i\lambda_i}{\mu+\lambda_i} \downarrow i,$$

then the list policy $\pi = (1, 2, \ldots, n)$ *maximizes the expected total return* $E[R]$.

PROOF: The proof is by induction on n. When $n = m + 1$, we do not need to consider preemptions because after a service completion or someone departs the queue, there are m customers in system to be served by m servers. Thus, based on Corollary 3, the list policy $\pi = (1, 2, \ldots, n)$ maximizes the expected total return $E[R]$.

Assume it true whenever there are at most $n - 1$ customers whose parameters satisfy the condition of the theorem, and now suppose there are n customers. First consider any policy that initially serves customers $1, \ldots, m$. By the induction hypothesis, the best policy of this type will switch to π (preempting customers in service if needed) after the first service completion or when someone departs the queue. Thus, such policy is equivalent to policy π, and the expected total return under such policy is the same as under π. Therefore, to show the optimality of the list policy π, it suffices to show that for any arbitrary policy, there exists a policy which has greater expected total return and initially serves customers $1, \ldots, m$.

Consider any policy that starts by serving customer i_j on server j $(j = 1, \ldots, m)$ and leaving customers $i_{m+1}, \ldots, i_n$ in queue, where $i_1, \ldots, i_n$ is an arbitrary permutation of $1, \ldots, n$. Then, by the induction hypothesis, the best policy of this type will switch to π (preempting customers in service if needed) after the first service completion or when someone departs the queue. Call this policy π' and let $E_{\pi'}[R]$ denote the expected total return under π'. If $\{i_1, \ldots, i_m\} = \{1, \ldots, m\}$, then policy π' is equivalent to policy π and $E_{\pi'}[R] = E_\pi[R]$. If $\{i_1, \ldots, i_m\} \ne \{1, \ldots, m\}$, then let $k = \text{argmax}_{j\in\{1,\ldots,m\}} i_j$ and $w = \text{argmin}_{j\in\{m+1,\ldots,n\}} i_j$ and it follows that $i_w \le m < i_k$. Let π'' be the policy that starts by serving customer i_j on server j, for $j = 1, \ldots, k-1, k+1, \ldots, m$ and customer i_w on server k, and then switches to π after the first service completion or when someone departs the queue. Let $E_{\pi''}[R]$ denote the expected total return under π''. We now show that $E_{\pi''}[R] \ge E_{\pi'}[R]$.

Let $\mu^{(k)} = \sum_{\substack{j=1\\ j\ne k}}^{m} \mu_j$ and $\lambda^{(w)} = \sum_{\substack{j=m+1\\ j\ne w}}^{n} \lambda_{i_j}$. Condition on the first thing that happens:

$$\begin{aligned}
E_{\pi'}[R] &= \frac{1}{\mu + \sum_{j=m+1}^{n} \lambda_{i_j}} \left\{ \sum_{j=1}^{m} \mu_j [r_{i_j} + R(1, \ldots, i_j - 1, i_j + 1, \ldots, n)] \right. \\
&\quad \left. + \sum_{j=m+1}^{n} \lambda_{i_j} R(1, \ldots, i_j - 1, i_j + 1, \ldots, n) \right\} \\
&= \frac{1}{\mu + \lambda^{(w)} + \lambda_{i_w}} \left\{ \sum_{\substack{j=1\\ j\ne k}}^{m} \mu_j [r_{i_j} + R(1, \ldots, i_j - 1, i_j + 1, \ldots, n)] \right. \\
&\quad + \mu_k [r_{i_k} + R(1, \ldots, i_k - 1, i_k + 1, \ldots, n)] \\
&\quad \left. + \sum_{\substack{j=m+1\\ j\ne w}}^{n} \lambda_{i_j} R(1, \ldots, i_j - 1, i_j + 1, \ldots, n) + \lambda_{i_w} R(1, \ldots, i_w - 1, i_w + 1, \ldots, n) \right\}.
\end{aligned}$$

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

For fixed k, w, define

$$\alpha' = \frac{\mu_k + \lambda_{i_w}}{\mu^{(k)} + \lambda^{(w)}},$$

$$A = r_1 + R(2, \ldots, n),$$

$$B = \frac{\begin{array}{c}\sum_{\substack{j=1\\ j\neq k}}^{m} \mu_j [r_{i_j} + R(1, \ldots, i_j - 1, i_j + 1, \ldots, n)] \\ + \sum_{\substack{j=m+1\\ j\neq w}}^{n} \lambda_{i_j} R(1, \ldots, i_j - 1, i_j + 1, \ldots, n)\end{array}}{\mu^{(k)} + \lambda^{(w)}}.$$

From Lemma 6, we have

$$\begin{aligned} E_{\pi'}[R] &= \frac{1}{1+\alpha'}\left\{ B + \frac{\mu_k}{\mu^{(k)} + \lambda^{(w)}} [A + R(i_k, 2, \ldots, i_k - 1, i_k + 1, \ldots, n) - R(2, \ldots, n)] \right. \\ &\quad \left. + \frac{\lambda_{i_w}}{\mu^{(k)} + \lambda^{(w)}} [A + R(i_w, 2, \ldots, i_w - 1, i_w + 1, \ldots, n) - R(2, \ldots, n) - r_{i_w}] \right\} \\ &= \frac{B + \alpha' A}{1 + \alpha'} + \frac{\mu_k}{\mu + \lambda^{(w)} + \lambda_{i_w}} [R(i_k, 2, \ldots, i_k - 1, i_k + 1, \ldots, n) - R(2, \ldots, n)] \\ &\quad - \frac{\lambda_{i_w} r_{i_w}}{\mu + \lambda^{(w)} + \lambda_{i_w}}, \end{aligned}$$

where the last equality is from Lemma 5 and $i_w \leq m$.

Likewise,

$$\begin{aligned} E_{\pi''}[R] &= \frac{1}{\mu + \lambda^{(w)} + \lambda_{i_k}} \left\{ \sum_{\substack{j=1\\ j\neq k}}^{m} \mu_j [r_{i_j} + R(1, \ldots, i_j - 1, i_j + 1, \ldots, n)] \right. \\ &\quad + \mu_k [r_{i_w} + R(1, \ldots, i_w - 1, i_w + 1, \ldots, n)] + \sum_{\substack{j=m+1\\ j\neq w}}^{n} \lambda_{i_j} R(1, \ldots, i_j - 1, i_j + 1, \ldots, n) \\ &\quad \left. + \lambda_{i_k} R(1, \ldots, i_k - 1, i_k + 1, \ldots, n) \right\}. \end{aligned}$$

For fixed k,w, define

$$\alpha'' = \frac{\mu_k + \lambda_{i_k}}{\mu^{(k)} + \lambda^{(w)}}.$$

From Lemma 6, we have

$$\begin{aligned} E_{\pi''}[R] &= \frac{1}{1+\alpha''}\left\{ B + \frac{\mu_k}{\mu^{(k)} + \lambda^{(w)}} [A + R(i_w, 2, \ldots, i_w - 1, i_w + 1, \ldots, n) - R(2, \ldots, n)] \right. \\ &\quad \left. + \frac{\lambda_{i_k}}{\mu^{(k)} + \lambda^{(w)}} [A + R(i_k, 2, \ldots, i_k - 1, i_k + 1, \ldots, n) - R(2, \ldots, n) - r_{i_k}] \right\} \\ &= \frac{B + \alpha'' A}{1 + \alpha''} + \frac{\lambda_{i_k}}{\mu + \lambda^{(w)} + \lambda_{i_k}} [R(i_k, 2, \ldots, i_k - 1, i_k + 1, \ldots, n) - R(2, \ldots, n)] \\ &\quad - \frac{\lambda_{i_k} r_{i_k}}{\mu + \lambda^{(w)} + \lambda_{i_k}}, \end{aligned}$$

where the last equality is from Lemma 5 and $i_w \leq m$.

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

Therefore

$$E_{\pi''}[R] - E_{\pi'}[R] = \frac{(\alpha'' - \alpha')(A - B)}{(1+\alpha'')(1+\alpha')} + \frac{\lambda_{i_w} r_{i_w}}{\mu + \lambda^{(w)} + \lambda_{i_w}} - \frac{\lambda_{i_k} r_{i_k}}{\mu + \lambda^{(w)} + \lambda_{i_k}} + \left(\frac{\mu_k}{\mu + \lambda^{(w)} + \lambda_{i_w}} - \frac{\lambda_{i_k}}{\mu + \lambda^{(w)} + \lambda_{i_k}} \right) [R(2, \ldots, n) - R(i_k, 2, \ldots, i_k - 1, i_k + 1, \ldots, n)].$$

By the induction hypothesis, $R(2, \ldots, n) - R(i_k, 2, \ldots, i_k - 1, i_k + 1, \ldots, n) \geq 0$. Also since $\mu_k/(\mu + \lambda^{(w)} + \lambda_{i_w}) \geq \lambda_{i_k}/(\mu + \lambda^{(w)} + \lambda_{i_k})$, we have

$$\begin{aligned} E_{\pi''}[R] - E_{\pi'}[R] &\geq \frac{(\alpha'' - \alpha')(A - B)}{(1+\alpha'')(1+\alpha')} + \frac{\lambda_{i_w} r_{i_w}}{\mu + \lambda^{(w)} + \lambda_{i_w}} - \frac{\lambda_{i_k} r_{i_k}}{\mu + \lambda^{(w)} + \lambda_{i_k}} \\ &= \frac{(\lambda_{i_k} - \lambda_{i_w})(A - B)}{(1+\alpha'')(1+\alpha')(\mu^{(k)} + \lambda^{(w)})} + \frac{\lambda^{(w)}(\lambda_{i_w} r_{i_w} - \lambda_{i_k} r_{i_k})}{(\mu + \lambda^{(w)} + \lambda_{i_w})(\mu + \lambda^{(w)} + \lambda_{i_k})} \\ &\quad + \frac{\lambda_{i_w} r_{i_w}(\mu + \lambda_{i_k}) - \lambda_{i_k} r_{i_k}(\mu + \lambda_{i_w})}{(\mu + \lambda^{(w)} + \lambda_{i_w})(\mu + \lambda^{(w)} + \lambda_{i_k})} \\ &\geq \frac{\lambda^{(w)}(\lambda_{i_k} - \lambda_{i_w})}{(1+\alpha'')(1+\alpha')(\mu^{(k)} + \lambda^{(w)})^2} \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} \\ &\quad + \frac{\lambda^{(w)}(\lambda_{i_w} r_{i_w} - \lambda_{i_k} r_{i_k})}{(\mu + \lambda^{(w)} + \lambda_{i_w})(\mu + \lambda^{(w)} + \lambda_{i_k})} \\ &\quad + \frac{\lambda_{i_w} r_{i_w}(\mu + \lambda_{i_k}) - \lambda_{i_k} r_{i_k}(\mu + \lambda_{i_w})}{(\mu + \lambda^{(w)} + \lambda_{i_w})(\mu + \lambda^{(w)} + \lambda_{i_k})}, \end{aligned}$$

where the last inequality is based on the following Lemma 10.

For fixed k, w, define

$$C = (\mu + \lambda^{(w)} + \lambda_{i_w})(\mu + \lambda^{(w)} + \lambda_{i_k}).$$

It follows that

$$\begin{aligned} E_{\pi''}[R] - E_{\pi'}[R] &\geq \frac{\lambda^{(w)}}{C} \left[(\mu + \lambda_{i_k}) \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} - (\mu + \lambda_{i_w}) \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} \right] \\ &\quad + \frac{\lambda^{(w)}(\lambda_{i_w} r_{i_w} - \lambda_{i_k} r_{i_k})}{C} + \frac{1}{C}(\mu + \lambda_{i_k})(\mu + \lambda_{i_w}) \left(\frac{\lambda_{i_w} r_{i_w}}{\mu + \lambda_{i_w}} - \frac{\lambda_{i_k} r_{i_k}}{\mu + \lambda_{i_k}} \right) \\ &\geq \frac{\lambda^{(w)}}{C} \left[(\mu + \lambda_{i_k}) \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} - (\mu + \lambda_{i_w}) \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} \right] \\ &\quad + \frac{\lambda^{(w)}(\lambda_{i_w} r_{i_w} - \lambda_{i_k} r_{i_k})}{C} \\ &= \frac{\lambda^{(w)}}{C} \left[(\mu + \lambda_{i_k}) \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} - \lambda_{i_k} r_{i_k} + \lambda_{i_w} r_{i_w} - (\mu + \lambda_{i_w}) \frac{r_{m+1}\lambda_{m+1}}{\mu + \lambda_{m+1}} \right] \\ &\geq 0, \end{aligned}$$

where the last two inequalities follow from $r_i\lambda_i/(\mu + \lambda_i) \downarrow i$ and $i_w < m + 1 \leq i_k$.

With the preceding argument, we have proven that π'' (initially serving customers $i_1, \ldots, i_{k-1}, i_w, i_{k+1}, \ldots, i_m$) is better than π' (initially serving customers

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

$i_1, \ldots, i_m)$ in the sense of maximizing expected total return $E[R]$. In addition, if $\{i_1, \ldots, i_{k-1}, i_w, i_{k+1}, \ldots, i_m\} \neq \{1, \ldots, m\}$, we can use the same argument to improve policy π''. Therefore, by repeating the preceding argument we will finally obtain a policy that initially serves customers $1, \ldots, m$, and has the same expected total return as π does. The optimality of the list policy $\pi = (1, \ldots, n)$ then follows. ■

LEMMA 10: *Under the condition of Theorem 9,*

$$A - B \geq \frac{\lambda^{(w)} r_{m+1} \lambda_{m+1}}{(\mu^{(k)} + \lambda^{(w)})(\mu + \lambda_{m+1})}.$$

PROOF:

$$A - B = \frac{\begin{array}{c}\sum_{\substack{j=1 \\ j \neq k}}^{m} \mu_j [A - r_{i_j} - R(1, \ldots, i_j - 1, i_j + 1, \ldots, n)] \\ + \sum_{\substack{j=m+1 \\ j \neq w}}^{n} \lambda_{i_j} [A - R(1, \ldots, i_j - 1, i_j + 1, \ldots, n)]\end{array}}{\mu^{(k)} + \lambda^{(w)}}.$$

For $j = 1, \ldots, k-1, k+1, \ldots, m$, it follows from Lemma 6 and Corollary 3 that

$$\begin{aligned} & A - r_{i_j} - R(1, \ldots, i_j - 1, i_j + 1, \ldots, n) \\ & \quad = r_1 + R(2, \ldots, n) - r_{i_j} - R(1, 2, \ldots, i_j - 1, i_j + 1, \ldots, n) \\ & \quad = r_1 + R(2, \ldots, n) - r_1 - R(i_j, 2, \ldots, i_j - 1, i_j + 1, \ldots, n) \\ & \quad = R(2, \ldots, n) - R(i_j, 2, \ldots, i_j - 1, i_j + 1, \ldots, n) \\ & \quad \geq 0. \end{aligned}$$

For $j = m+1, m+2, \ldots, w-1, w+1, \ldots, n$, it follows from Lemma 8 that

$$\begin{aligned} A - R(1, \ldots, i_j - 1, i_j + 1, \ldots, n) &= r_1 + R(2, \ldots, n) - R(1, \ldots, i_j - 1, i_j + 1, \ldots, n) \\ &\geq \frac{r_{m+1} \lambda_{m+1}}{\mu + \lambda_{m+1}}. \end{aligned}$$

Therefore

$$A - B \geq \frac{\lambda^{(w)} r_{m+1} \lambda_{m+1}}{(\mu^{(k)} + \lambda^{(w)})(\mu + \lambda_{m+1})}.$$ ■

Moreover, it turns out that the list policy $(1, \ldots, n)$ also has the property of stochastically minimizing the makespan under the condition of $\lambda_i \uparrow i$ and $\min_{1 \leq j \leq m} \mu_j \geq \max_{1 \leq i \leq n} \lambda_i$ which we show in the remainder of this section.

For $\{i_1, \ldots, i_k\}$ being a subset of $\{1, \ldots, n\}$ $(0 \leq k \leq n - m)$, consider an m server non-preemptive system that begins with all servers busy and customers $i_1, \ldots, i_k$ in queue. Note that the makespan does not depend on which customers are initially served. Thus, for such a system we let $T_B(i_1, \ldots, i_k)$ denote the makespan under the non-preemptive policy that after each service completion elects to serve the remaining customer i_l if $i_1, \ldots, i_{l-1}$ are no longer in queue.

The following lemma can be easily proven with a standard coupling argument.

LEMMA 11: *If $\lambda_i \uparrow i$, then for any $\{i_1, \ldots, i_{n-m-1}\} \subset \{1, \ldots, n\}$, $T_B(i_1, \ldots, i_{n-m-1})$ is stochastically larger than $T_B(m+2, \ldots, n)$.*

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

For $i_1, \ldots, i_n$ being an arbitrary permutation of $1, \ldots, n$, consider two scenarios of an m server preemptive problem that begins with customers $1, \ldots, n$: the first which follows the list policy $\pi = (1, \ldots, n)$; the second which starts by serving customer i_j on server j ($j = 1, \ldots, m$) and leaving customers $i_{m+1}, \ldots, i_n$ in queue, and then switches to π (preempting customers in service if needed) after the first event, where an event occurs either at a service completion or a departure from the queue. Let D_1 and D_2 denote the respective remaining makespans in scenario 1 and 2 after the first event. To derive the expressions of D_1 and D_2, let $X_1 = \{1, \ldots, m\}, Y_1 = \{m+1, \ldots, n\}$. In addition, for fixed $i_1, \ldots, i_n$, let $X_2 = \{i_1, \ldots, i_m\}, Y_2 = \{i_{m+1}, \ldots, i_n\}$, and

$$U = X_1 \cap X_2, V = X_1 \cap Y_2, \quad W = Y_1 \cap Y_2, \quad Z = X_2 \cap Y_1,$$
$$Q = \{j : 1 \le j \le m, i_j \in U\}, \quad R = \{j : 1 \le j \le m, i_j \in Z\}.$$

With the preceding notation, we derive the expressions of D_1 and D_2 as follows:

$$D_1 = \begin{cases} T_B(m+2, \ldots, n) & \text{w.p. } \frac{\mu}{\mu + \sum_{l \in Y_1} \lambda_l} \\ T_B(m+1, \ldots, i_j - 1, i_j + 1, \ldots, n) & \text{w.p. } \frac{\lambda_{i_j}}{\mu + \sum_{l \in Y_1} \lambda_l}, \text{ for each } j \in R \\ T_B(m+1, \ldots, i-1, i+1, \ldots, n) & \text{w.p. } \frac{\lambda_i}{\mu + \sum_{l \in Y_1} \lambda_l}, \text{ for each } i \in W \end{cases}$$

and

$$D_2 = \begin{cases} T_B(m+2, \ldots, n) & \text{w.p. } \frac{\sum_{j \in Q} \mu_j + \sum_{l \in V} \lambda_l}{\mu + \sum_{l \in Y_2} \lambda_l} \\ T_B(m+1, \ldots, i_j - 1, i_j + 1, \ldots, n) & \text{w.p. } \frac{\mu_j}{\mu + \sum_{l \in Y_2} \lambda_l}, \text{ for each } j \in R \\ T_B(m+1, \ldots, i-1, i+1, \ldots, n) & \text{w.p.} \frac{\lambda_i}{\mu + \sum_{l \in Y_2} \lambda_l}, \text{ for each } i \in W \end{cases}$$

where the notation "w.p." means "with probability".

LEMMA 12: *If*

$$\lambda_i \uparrow i \text{ and } \min_{1 \le j \le m} \mu_j \ge \max_{1 \le i \le n} \lambda_i$$

then D_1 is stochastically smaller than D_2.

PROOF: For any $t \ge 0$,

$$\begin{aligned} P\{D_1 > t\} = {} & P\{T_B(m+2, \ldots, n) > t\} \frac{\mu}{\mu + \sum_{l \in Y_1} \lambda_l} \\ & + \sum_{j \in R} P\{T_B(m+1, \ldots, i_j - 1, i_j + 1, \ldots, n) > t\} \frac{\lambda_{i_j}}{\mu + \sum_{l \in Y_1} \lambda_l} \\ & + \sum_{i \in W} P\{T_B(m+1, \ldots, i-1, i+1, \ldots, n) > t\} \frac{\lambda_i}{\mu + \sum_{l \in Y_1} \lambda_l}, \end{aligned}$$

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

and

$$P\{D_2 > t\} = P\{T_B(m+2,\ldots,n) > t\}\frac{\sum_{j\in Q}\mu_j + \sum_{l\in V}\lambda_l}{\mu + \sum_{l\in Y_2}\lambda_l} + \sum_{j\in R} P\{T_B(m+1,\ldots,i_j-1,i_j+1,\ldots,n) > t\}\frac{\mu_j}{\mu + \sum_{l\in Y_2}\lambda_l} + \sum_{i\in W} P\{T_B(m+1,\ldots,i-1,i+1,\ldots,n) > t\}\frac{\lambda_i}{\mu + \sum_{l\in Y_2}\lambda_l}.$$

From Lemma 11, for any $t \geq 0$ and any $\{i_1,\ldots,i_{n-m-1}\} \subset \{1,\ldots,n\}$,

$$P\{T_B(m+2,\ldots,n) > t\} \leq P\{T_B(i_1,\ldots,i_{n-m-1}) > t\}.$$

In addition, it follows from $\lambda_i \uparrow i$ and $\min_{1\leq j\leq m}\mu_j \geq \max_{1\leq i\leq n}\lambda_i$ that

$$\frac{\mu}{\mu + \sum_{l\in Y_1}\lambda_l} \geq \frac{\sum_{j\in Q}\mu_j + \sum_{l\in V}\lambda_l}{\mu + \sum_{l\in Y_2}\lambda_l},$$

$$\frac{\lambda_{i_j}}{\mu + \sum_{l\in Y_1}\lambda_l} \leq \frac{\mu_j}{\mu + \sum_{l\in Y_2}\lambda_l}, \text{ for all } j \in R,$$

$$\frac{\lambda_i}{\mu + \sum_{l\in Y_1}\lambda_l} \leq \frac{\lambda_i}{\mu + \sum_{l\in Y_2}\lambda_l}, \text{ for all } i \in W.$$

Therefore, from the following Lemma 13, for any $t \geq 0$,

$$P\{D_1 > t\} \leq P\{D_2 > t\},$$

and the result follows. ■

The following is immediate.

LEMMA 13: *If* $a_1,\ldots,a_n$, $p_1,\ldots,p_n$, $q_1,\ldots,q_n$ *are non-negative,* $a_1 = \min_i a_i$, $\sum_i p_i = \sum_i q_i = 1$, $p_1 \geq q_1$, *and* $p_i \leq q_i, i = 2,\ldots,n$, *then* $\sum_i p_i a_i \leq \sum_i q_i a_i$.

Using the preceding lemmas, we now show that the list policy $(1,\ldots,n)$ has the property of stochastically minimizing the makespan under certain conditions.

PROPOSITION 14: *If*

$$\lambda_i \uparrow i \text{ and } \min_{1\leq j\leq m}\mu_j \geq \max_{1\leq i\leq n}\lambda_i$$

then the list policy $\pi = (1,\ldots,n)$ *stochastically minimizes the makespan* T.

PROOF: The proof is by induction on n. When $n = m+1$, we do not need to consider preemptions because after a service completion or someone departs the queue, there are m customers in system to be served by m servers. Thus based on Corollary 4, the list policy $\pi = (1,\ldots,n)$ stochastically minimizes the makespan T.

Assume it true whenever there are at most $n-1$ customers whose parameters satisfy the condition of the proposition, and now consider the n case. Consider a policy that starts by serving customer i_j on server j $(j = 1,\ldots,m)$ and leaving customers $i_{m+1},\ldots,i_n$ in queue, where $i_1,\ldots,i_n$ is an arbitrary permutation of $1,\ldots,n$. Then, by the induction hypothesis,

https://doi.org/10.1017/S0269964816000176 Published online by Cambridge University Press

the best policy of this type will switch to π (preempting customers in service if needed) after the first event, where an event occurs either at a service completion or a departure from the queue. Call this policy π' and let $T_{\pi'}$ denote the makespan under π'. In addition, let T_π denote the makespan under π. We now show that T_π is stochastically smaller than $T_{\pi'}$.

Define $X \oplus Y$ as the sum of two independent random variables X and Y. Condition on the first event that happens:

$$T_\pi =_{\text{st}} \text{Exp}(\mu + \sum_{l \in Y_1} \lambda_l) \oplus D_1,$$

$$T_{\pi'} =_{\text{st}} \text{Exp}(\mu + \sum_{l \in Y_2} \lambda_l) \oplus D_2,$$

where D_1 and D_2 are the respective remaining makespans after the first event under policy π and π'. Then, it follows from Lemma 12 that D_1 is stochastically smaller than D_2.

In addition,

$$\text{Exp}\left(\mu + \sum_{l \in Y_1} \lambda_l\right) \leq_{\text{st}} \text{Exp}\left(\mu + \sum_{l \in Y_2} \lambda_l\right).$$

Therefore,

$$\begin{aligned} T_\pi &=_{\text{st}} \text{Exp}\left(\mu + \sum_{l \in Y_1} \lambda_l\right) \oplus D_1 \\ &\leq_{\text{st}} \text{Exp}\left(\mu + \sum_{l \in Y_2} \lambda_l\right) \oplus D_2 \\ &=_{\text{st}} T_{\pi'}, \end{aligned}$$

which yields the result. ■

Remark: Proposition 14 will not hold without the condition $\min_{1 \leq j \leq m} \mu_j \geq \max_{1 \leq i \leq n} \lambda_i$. For example, when $m = 1, n = 3$, let T_1 denote the makespan under the list policy $\pi = (1, 2, 3)$, and let T_3 denote the makespan under a policy that first puts customer 3 in service, and then switches to π after the first event. The distributions of T_1 and T_3 can be analytically derived. Suppose $\mu_1 = 1, \lambda_1 = 3, \lambda_2 = 4$ and $\lambda_3 = 10$, then $E[T_1] = 1.2303 > 1.2295 = E[T_3]$ and it follows that T_1 is not stochastically smaller than T_3. On the other hand, however, we are not sure if the condition $\min_{1 \leq j \leq m} u_j \geq \max_{1 \leq i \leq n} \lambda_i$ is necessary for Theorem 9, although it is needed in the given proof. When $m = 1, n = 3$ or $m = 1, n = 4$, Theorem 9 can be algebraically proven without such condition by comparing $E_\pi[R]$ with $E_{\pi'}[R]$.

Acknowledgments

This material is based upon work supported by, or in part by, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-14-1-0166.

References

1. Argon, N.T., Ziya, S., & Righter, R. (2008). Scheduling impatient jobs in a clearing system with insights on patient triage in mass casualty incidents. *Probability in the Engineering and Informational Sciences*, 22: 301–332.

2. Atar, R., Giat, C., & Shimkin, N. (2010). The cμ/θ rule for many-server queues with abandonment. *Operations Research*, 58(5): 1427–1439.
3. Glazebrook, K.D., Ansell, P.S., Dunn, R.T., and Lumley, R.R. (2004). On the optimal allocation of service to impatient tasks. *Journal of Applied Probability*, 41(1): 51–72.
4. Jang, W. (2002). Dynamic scheduling of stochastic jobs on a single machine. *European Journal of Operational Research*, 138(3): 518–530.
5. Jang, W. & Klein, C.M. (2002). Minimizing the expected number of tardy jobs when processing times are normally distributed. *Operations Research Letters*, 30(2): 100–106.
6. Mandelbaum, A. & Momcilovic, P. (2012). Queues with many servers and impatient customers. *Mathematics of Operations Research*, 37(1): 41–65.
7. Movaghar, A. (1998). On queueing with customer impatience until the beginning of service. *Queueing Systems*, 29(2–4): 337–350.
8. Movaghar, A. (2006). On queueing with customer impatience until the end of service. *Stochastic Models*, 22(1): 149–173.
9. Pandelis, D.G. & Teneketzis, D. (1993). Stochastic scheduling in priority queues with strict deadlines. *Probability in the Engineering and Informational Sciences*, 7(02): 273–289.
10. Panwar, S.S., Towsley, D., & Wolf, J.K. (1988). Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service. *Journal of the ACM (JACM)*, 35(4): 832–844.
11. Pinedo, M. (1983). Stochastic scheduling with release dates and due dates. *Operations Research*, 31(3): 559–572.
12. Ross, S.M. (2015). A sequential scheduling problem with impatient jobs. *Naval Research Logistics (NRL)*, 62(8): 659–663.
13. Salch, A., Gayon, J.-P., & Lemaire, P. (2013). Optimal static priority rules for stochastic scheduling with impatience. *Operations Research Letters*, 41(1): 81–85.
14. Seo, D.K., Klein, C.M., & Jang, W. (2005). Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Computers & Industrial Engineering*, 48(2): 153–161.
15. Towsley, D. & Panwar, S.S. (1991). Optimality of the stochastic earliest deadline policy for the g/m/c queue serving customers with deadlines. Technical paper available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.3124&rep=re%p1&type=pdf.
16. Ward, A.R. & Kumar, S. (2008). Asymptotically optimal admission control of a queue with impatient customers. *Mathematics of Operations Research*, 33(1): 167–202.
17. Zeltyn, S. (2005). Call centers with impatient customers: exact analysis and many-server asymptotics of the M/M/n+ G queue. Ph.D. thesis, Technion-Israel Institute of Technology, Faculty of Industrial and Management Engineering.
18. Zhao, Z.X., Panwar, S.S., & Towsley, D. (1991). 1991 Queueing performance with impatient customers. In *INFOCOM '91. Proceedings. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking in the 90s, IEEE*, vol. 1, April, pp. 400–409.