Stochastic scheduling with abandonment: Necessary and sufficient conditions for the optimality of a strict priority policy.

Gang Chen
School of Management, Guangzhou University, Guangzhou 510006, China, chengang5@mail.sysu.edu.cn

Jean-Philippe Gayon
Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France, j-philippe.gayon@uca.fr

Pierre Lemaire
Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France, pierre.lemaire@grenoble-inp.fr

We consider a stochastic scheduling problem in clearing systems with two types of jobs, each characterized by a general service time distribution, an exponentially distributed lifetime and a reward. A job abandons the system if its waiting time in the queue is larger than its lifetime. Preemption is not allowed. The objective is to maximize the total expected reward. When service times are homogeneous, we provide a set of necessary and sufficient conditions for the optimality of a strict priority policy. When service times are heterogeneous and exponentially distributed, we conjecture a set of necessary conditions, which would also be sufficient when one parameter is identical (reward or lifetime rate) for the two types of jobs.

Key words: Stochastic scheduling, Abandonment, Strict priority policy, Dynamic programming.

1. Introduction

In many service or production systems, customers or tasks may become impatient and leave the system, without having received service. This behavior is called an abandonment, which generates a cost of dissatisfaction, or a loss of revenue.

In this paper, we consider a stochastic scheduling problem where jobs can be categorized into two types, with n_i jobs of type $i = 1, 2$ waiting for service. All jobs are available at time zero and there are no arrivals. The lifetime of type i jobs is a random variable D_i exponentially distributed (with mean $1/\gamma_i$). The service time of type i jobs is a random variable X_i (with mean $1/\mu_i > 0$). The service times and lifetimes are independent continuous random variables. A job abandons the system if its waiting time in the queue is larger than its lifetime and can not abandon once in service. A positive reward w_i is earned when a job of type i is taken into service. The service is performed in a nonpreemptive manner. The decision epochs for our dynamic control problem are time zero and service completion instants. The objective is to maximize the expected total reward in the set of non-idling and nonpreemptive dynamic scheduling policies. Note that an idling policy, i.e., a policy under which the server may idle in the presence of jobs, is suboptimal in the sense of this optimization problem (Jacobson et al. 2012).

A strict priority (SP) policy is a policy which always gives priority to the same type of job, in any system state and decision epoch. The policy which gives a strict priority to type i jobs will be denoted by SP$_i$. We are interested in finding a set of necessary and sufficient conditions under which an SP policy is optimal for every initial state (n_1, n_2). In Section 2, we derive such conditions for homogeneous service times. In Section 3, we suggest a conjecture for the case with heterogeneous and exponentially distributed service times. To the best of our knowledge, this paper is the first to provide necessary and sufficient conditions for the optimality of strict priority policies for a stochastic scheduling problem with abandonment. On this topic, Jacobson et al. (2012) highlight that their results provide "sufficient conditions that lead to the optimality of index policies. They are not necessary conditions however, and index policies might be optimal even when none of these conditions holds".

Our paper is closely related to two streams of literature, one dealing with stochastic scheduling with abandonment and the other with priority queues with abandonment. In stochastic scheduling problems, there is a finite number of jobs to schedule. One can see a priority queuing problem as a scheduling problem with stochastic release dates (see e.g. Pinedo (2008)).

In the first stream of literature, Argon et al. (2008) consider a problem close to ours, but where rewards are job independent. They show that if lifetimes and service times can be ordered in the sense of likelihood ratio ordering, then priority is given to the job with shortest lifetime and shortest service time. For the special case with two types of jobs and exponential distributions, they partially characterize the optimal policy and develop two state-dependent heuristic policies. Li and Glazebrook (2010) also consider a formulation that is very similar to that of Argon et al. (2008), which aims at developing a near-optimal real time solution. Jacobson et al. (2012) extend the model of Argon et al. (2008) to the case with job-dependent rewards. They derive the following set of sufficient conditions under which an index policy is optimal: Priority should be given to the job with shortest lifetime (in the sense of hazard rate order), shortest service time (in the sense of likelihood ratio order) and largest reward if such a job exists. They also obtain additional results in the Markovian case with two types of jobs. When lifetimes are exponentially distributed and service times are identically distributed, Ross (2015) provides another set of sufficient conditions for the optimality of an index policy. Cao (2017) considers a multiple server version with exponential distributions. He provides conditions under which an index policy is optimal.

In the other related stream of literature, dedicated to priority queues with abandonment, Down et al. (2011) consider an $M/M/1+M$ queueing problem with two classes of customers and preemption, where processing times are identically distributed for both classes of customers. They provide a set of sufficient conditions for a strict priority rule to be optimal. Atar et al. (2010) relax the assumption of identically distributed processing times. Let c_j, μ_j and γ_j be respectively the waiting cost per unit of time, service rate and impatience rate of a class j customer. They show that scheduling jobs with the highest index $c_j\mu_j/\gamma_j$ is asymptotically optimal, under overload conditions and many-server fluid scaling. For the same problem, Bhulai et al. (2019) provide a set of sufficient conditions under which an index policy is optimal. Ayesta et al. (2017) study a variant of Atar et al. (2010) where customers in service are charged a waiting cost. Some of the other papers in the literature provide the asymptotic optimality of simple rules for closely related problems. The solution to the approximating fluid control problem in Puha and Ward (2019) motivates using a static priority scheduling rule that gives priority to class 1 customers when $w_1\mu_1 \geq w_2\mu_2$ where w_i

denotes the abandonment cost for a type i job. Under the additional assumption that $\mu_1 = \mu_2$, Kim et al. (2018) recommend using the aforementioned static priority scheduling rule when $\gamma_1 \leq \gamma_2$ and $w_1\gamma_1 \geq w_2\gamma_2$ for the approximating diffusion control problem. Long et al. (2020) study the fluid model of a many-server queue with multiple customer classes. They propose a fixed-priority policy which generalizes the $c\mu/\gamma$ rule of Atar et al. (2010).

2. General homogeneous service times

We assume in this section that both types of jobs have the same distribution for service times, i.e. $X_1 =_{st} X_2 =_{st} X$ (with mean $1/\mu$).

Theorem 1. *Policy SP_1 is optimal for every initial state (n_1, n_2) if and only if*

$$(S): \begin{cases} w_1 \geq w_2 \ \textit{and} \\ w_1\mathbb{P}(D_1 < X) \geq w_2\mathbb{P}(D_2 < X). \end{cases}$$

The complete proof of Theorem 1 is provided in Appendix, and here is its outline. We prove that (S) is a set of sufficient conditions by combining results from Ross (2015) and Jacobson et al. (2012). The necessary condition $w_1 \geq w_2$ appears when there are many jobs in the system. To prove its necessity, we formulate the problem as a stochastic dynamic program and study the asymptotic behavior of the value function when n_1 and n_2 go to infinity. The necessity of condition $w_1\mathbb{P}(D_1 < X) \geq w_2\mathbb{P}(D_2 < X)$ is shown by considering two jobs in the system.

Note that the second condition of (S) simplifies as follows for three specific service time distributions.

- Deterministic service times ($X = 1/\mu$): $w_1\left(1 - e^{-\frac{\gamma_1}{\mu}}\right) \geq w_2\left(1 - e^{-\frac{\gamma_2}{\mu}}\right)$;
- Exponentially distributed service times with mean $1/\mu$: $\frac{w_1\gamma_1}{\mu+\gamma_1} \geq \frac{w_2\gamma_2}{\mu+\gamma_2}$;
- Uniform service times on $[0, 2/\mu]$: $w_1\gamma_2\left(2\gamma_1 + \mu e^{-\frac{2\gamma_1}{\mu}} - \mu\right) \geq w_2\gamma_1\left(2\gamma_2 + \mu e^{-\frac{2\gamma_2}{\mu}} - \mu\right)$.

3. Conjectures for heterogeneous exponential service times

In this section, we suggest some avenues for research for the problem with heterogeneous service times. We assume that the service times of jobs of type i, $(i = 1, 2)$ are exponentially distributed with rate μ_i.

First, we conjecture that the following set of conditions is necessary for SP_1 to be optimal for any initial state when considering heterogeneous exponential service times:

$$(C): \begin{cases} w_1\mu_1 \geq w_2\mu_2 \text{ (many jobs in the system);} \\ \frac{w_1\gamma_1}{\mu_2+\gamma_1} \geq \frac{w_2\gamma_2}{\mu_1+\gamma_2} \text{ (two jobs in the system).} \end{cases}$$

The first condition simply states that priority should be given to the job type that brings the highest reward by unit of time. It is consistent with the index rule of Atar et al. (2010). The second necessary condition appears when there are two jobs in the system.

The next key question is whether the set of conditions (C) is sufficient to give priority to type 1 jobs for every initial state. In an extensive numerical study, we have observed that the answer is no in general but is true if one parameter is identical for the two types of jobs. We summarize these observations in the following conjecture.

Conjecture 1 **(Necessary and sufficient conditions)**. *Consider two types of jobs with exponentially distributed service times.*

(i) The set of conditions (C) is necessary for SP_1 to be optimal for every initial state;

(ii) The set of conditions (C) is sufficient for SP_1 to be optimal for every initial state if the two types of jobs share at least one common parameter ($\mu_1 = \mu_2$ or $\gamma_1 = \gamma_2$ or $w_1 = w_2$).

For homogeneous service times ($\mu_1 = \mu_2 = \mu$), this conjecture has been proved in Section 2. For homogeneous lifetimes ($\gamma_1 = \gamma_2 = \gamma$), no such conjecture has been stated, to the best of our knowledge. For equal weights ($w_1 = w_2 = w$), Argon et al. (2008) have already conjectured those conditions as sufficient, but did not state their necessity.

Moreover, we have observed numerically that if both $w_1 \geq w_2$ and $\mu_1 \geq \mu_2$, then (C) is still a set of sufficient conditions for SP_1 to be optimal. Conjecture 1 (ii) cannot be extended to cases where $w_1 > w_2$ or $\mu_1 > \mu_2$ or $\gamma_1 > \gamma_2$. Table 1 provides such examples where the set of conditions (C) holds, yet SP_1 is not optimal.

Case	μ_1	μ_2	γ_1	γ_2	w_1	w_2
$w_1 > w_2$	0.7146	0.8717	0.4053	0.4541	0.6819	0.5551
$\mu_1 > \mu_2$	0.1426	0.1243	0.7188	0.4059	0.3377	0.3787
$\gamma_1 > \gamma_2$	0.3900	0.3413	0.7804	0.6238	0.7225	0.8112

Table 1 Counter-examples where (C) is not a set of sufficient conditions for SP_1 to be optimal

4. Conclusion

We have provided the first set of necessary and sufficient conditions for the optimality of strict priority policies for a stochastic scheduling problem with abandonment. We have also identified several avenues for research. Proving Conjecture 1 is the next step. It would be also interesting to see how our results can be extended when considering stochastic releases of jobs, or more than two types of jobs. When lifetime distributions are not memoryless, the problem is considerably more complex as the state of the system should be described by the elapsed lifetime for each job.

Appendix. Proof of Theorem 1

A. Sufficient conditions

By Theorem 1 in Ross (2015), the following set of conditions is sufficient for the optimality of SP_1 :

$$(S_1) : \begin{cases} \gamma_1 \leq \gamma_2 \text{ and} \\ w_1 \mathbb{P}(D_1 < X) \geq w_2 \mathbb{P}(D_2 < X). \end{cases}$$

Corollary 1 in Jacobson et al. (2012), translated into our context, provides another set of sufficient conditions for the optimality of SP_1 :

$$(S_2) : \begin{cases} \gamma_1 \geq \gamma_2 \text{ and} \\ w_1 \geq w_2. \end{cases}$$

We prove in this paragraph that (S) holds if and only if either (S_1) or (S_2) holds. Trivially, we have (S) implying (S_1) when $\gamma_1 \leq \gamma_2$ and (S) implying (S_2) when $\gamma_1 \geq \gamma_2$. Reciprocally, assume that (S_1) is satisfied. As $\gamma_1 \leq \gamma_2$, we have $\mathbb{P}(D_1 < X) \leq \mathbb{P}(D_2 < X)$ which in turn implies, by the second inequality of (S_1), that $w_1 \geq w_2$. Thus (S) is satisfied. Assume now that (S_2) is satisfied. As $\gamma_1 \geq \gamma_2$, we have $\mathbb{P}(D_1 < X) \geq \mathbb{P}(D_2 < X)$ which in turn implies, by the second inequality of

(S_2), that $w_1\mathbb{P}(D_1 < X) \geq w_2\mathbb{P}(D_2 < X)$. Thus (S) is satisfied. In conclusion, (S) holds if and only if either (S_1) or (S_2) holds.

The first part of Theorem 1 (sufficient conditions) follows: if an instance satisfies (S), then it satisfies (S_1) or (S_2). In both cases, SP_1 is optimal (Ross 2015, Jacobson et al. 2012).

B. Necessary conditions

Stochastic dynamic program

We begin by formulating our optimization problem as a stochastic dynamic program.

System state Let $\mathbf{n} = (n_1, n_2)$ with n_i the number of type i jobs waiting for service in the system. Denote by u the status of the server where $u = j$ indicates that the server is busy processing a job of type j, $j \in \{1, 2\}$, and $u = 0$ indicates that the server is ready to process a new job. The state of the system can be fully described by $(\mathbf{n}, u)$. The decision epochs are the service completion times and time zero. At every decision epoch, the server can take a job of type 1 or 2 into service if some are available at that time. If the system is in state $(\mathbf{n}, 0)$, with $\mathbf{n} = (n_1, n_2)$, an action a is selected from the available action set $\Phi(\mathbf{n}) = \{i \in \{1, 2\} : n_i > 0\}$.

Transition law Let $U_i^k(u)$ be the Bernoulli random variable which equals 1 if the k-th job of type i does not abandon during the service time of the job u in process, and 0 otherwise.

When in state $(\mathbf{n}, u)$, $u = 1, 2$, the next state will be $(\mathbf{n}', 0)$ with probability

$$P_{\mathbf{n}\mathbf{n}'}(u) = \mathbb{P}\left\{\sum_{k=1}^{n_1} U_1^k(u) = n_1', \sum_{k=1}^{n_2} U_2^k(u) = n_2'\right\},$$

where $\mathbf{n}' \in \Omega(\mathbf{n}) = \{(n_1', n_2') \in \mathbb{N}^2 \mid n_1' \leq n_1, n_2' \leq n_2\}$.

Reward A reward w_i is obtained when we begin to process a job of type i.

Optimality equations We denote by $R(\mathbf{n}, u)$ the maximum expected total reward when the initial state is $(\mathbf{n}, u)$. Then we have the following optimality equations, when preemption is forbidden:

$$\begin{aligned} R(\mathbf{n}, 0) &= \max_{u \in \Phi(\mathbf{n})} \{w_u + R(\mathbf{n} - e_u, u)\} \quad \text{if } |\mathbf{n}| > 0, \\ R(\mathbf{n}, u) &= \sum_{\mathbf{n}' \in \Omega(\mathbf{n})} P_{\mathbf{n}\mathbf{n}'}(u) R(\mathbf{n}', 0) \quad \text{if } |\mathbf{n}| > 0, u > 0, \\ R(\mathbf{0}, u) &= 0, \end{aligned} \tag{1}$$

where $|\mathbf{n}| = n_1 + n_2$ and e_u is the 2-dimensional vector with 1 in the u-th coordinate and 0 elsewhere.

Value function properties

In what follows, we provide two properties of the value function that will be used to prove Theorem 1.

Lemma 1. *For all $\mathbf{n} \in \mathbb{N}^2$, we have $R(\mathbf{n}, 1) = R(\mathbf{n}, 2)$.*

Proof. This property is immediate from (1), as the transition probabilities do not depend on the job in service. This is due to the assumption that jobs of type 1 and jobs of type 2 have the same service time distribution. □

Let $T_i(\mathbf{n}, u)$ be the time to clear all jobs of type i, except the one with the lowest priority (i.e. the last job of type i to be served), under the optimal policy when the initial state is $(\mathbf{n}, u)$.

Lemma 2. *For $\mathbf{n} \in \mathbb{N}^2$, for $i = 1, 2$, $u = 1, 2$, we have*

$$R(\mathbf{n}, u) \leq R(\mathbf{n} + e_i, u) \leq R(\mathbf{n}, u) + w_i \mathbb{P}(T_i(\mathbf{n} + e_i, u) < D_i).$$

Proof. When the initial state is $(\mathbf{n}, u)$, the optimal expected reward is $R(\mathbf{n}, u)$. When adding an extra job of type i, a possible policy is to ignore this job and to apply the same policy as when starting in state $(\mathbf{n}, u)$. This policy will have an expected reward equal to $R(\mathbf{n}, u)$. Hence $R(\mathbf{n} + e_i, u) \geq R(\mathbf{n}, u)$.

Without loss of generality, assume that the additional job is the last job of type i to be served. This additional job can be seen as the one with the lowest priority as any policy can only specify which class of job to serve (not which job within a class to serve). The additional job cannot be served before the other jobs of type i have been cleared, i.e. before time $T_i(\mathbf{n} + e_i, u)$ under the optimal policy when the initial state is $(\mathbf{n} + e_i, u)$. Hence the probability to serve this additional job is smaller than or equal to $\mathbb{P}(T_i(\mathbf{n} + e_i, u) < D_i)$ and the expected reward related to this job is smaller than or equal to $w_i \mathbb{P}(T_i(\mathbf{n} + e_i, u) < D_i)$. Moreover, the expected reward to be made from the n_1 type 1 and n_2 type 2 jobs (excluding the additional job) is bounded by the maximum expected reward $R(\mathbf{n}, u)$. It follows that $R(\mathbf{n} + e_i, u) \leq R(\mathbf{n}, u) + w_i \mathbb{P}(T_i(\mathbf{n} + e_i, u) < D_i)$. □

Limit cases

If policy SP_1 is optimal for every initial state, it must be optimal in particular when there are only two jobs in the system, or when there are many jobs. We will consider in what follows these two situations to derive necessary conditions.

First, assume that there remains in the system two jobs of types 1 and 2. It is optimal to give priority to type 1 job if and only if $w_1 \mathbb{P}(D_1 < X) \geq w_2 \mathbb{P}(D_2 < X)$, which proves that the second condition in Theorem 1 is necessary.

Second, when considering a large number of jobs, priority should be given to the job type with the largest reward in order to maximize the per-unit-of-time reward. We state and prove it rigorously with the following lemma.

Lemma 3 **(Necessary condition, large number of jobs)**. *If policy SP_1 is optimal for any initial state, then $w_1 \geq w_2$.*

Proof. Assume that $w_1 < w_2$. Let $\epsilon = (w_2 - w_1)/2$. From optimality equations (1), it is not optimal to give priority to a job of type 1 (and optimal to give priority to a job of type 2) in state $(\mathbf{n}, 0)$ when the following quantity is negative:

$$\begin{aligned}\Delta &= w_1 - w_2 + R(\mathbf{n} - e_1, 1) - R(\mathbf{n} - e_2, 2)\\ &= -2\epsilon + R(\mathbf{n} - e_1, 1) - R(\mathbf{n} - e_2, 2).\end{aligned} \tag{2}$$

From Lemma 1, we know that $R(\mathbf{n}, 1) = R(\mathbf{n}, 2)$ and we can rewrite (2) as

$$\begin{aligned}\Delta &= -2\epsilon - [R(\mathbf{n}, 1) - R(\mathbf{n} - e_1, 1)] + [R(\mathbf{n}, 2) - R(\mathbf{n} - e_2, 2)]\\ &\le -2\epsilon + 0 + w_2\mathbb{P}(T_2(\mathbf{n}, 2) < D_2).\end{aligned} \tag{3}$$

Inequality 3 follows from Lemma 2. The probability that an extra job of type 2 being served, $\mathbb{P}(T_2(\mathbf{n}, 2) < D_2)$, goes to 0 when n_2 goes to infinity. Hence, given n_1, there exists M such that, when $n_2 \ge M$, we have $w_2\mathbb{P}(T_2(n_2 - 1) < D_2) < \epsilon$. Together with (3) we obtain that, for $n_2 > M$, we have $\Delta < -\epsilon$. Hence, for $n_2 > M$, it is not optimal to give priority to a job of type 1 and it is optimal to give priority to a job of type 2.

We conclude that, when $w_1 < w_2$, SP_1 is not optimal when the number of jobs of type 2 is sufficiently large. Hence, $w_1 \ge w_2$ is a necessary condition for SP_1 to be optimal for every initial state. □

Acknowledgments

The authors thank the associate editor and the two referees, whose thorough and accurate comments greatly improved this article. They also thank Alexandre Salch who participated in the beginning of this research work during his PhD.

Bios

Gang Chen is associate professor in the School of Management at Guangzhou University. His research interests include queueing games, Markov decision processes, and sensitivity-based optimization.

Jean-Philippe Gayon is full professor in the Department of Computer Science at Clermont Auvergne INP. His research interests are in operations management and operations research, with a special focus on stochastic models.

Pierre Lemaire is associate professor at Univ. Grenoble-Alpes - Grenoble INP. His main interests are operations research and data sciences with applications to scheduling and industrial or energy systems.

References

Argon NT, 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(3):301–332.

Atar R, Giat C, Shimkin N (2010) The cμ/θ rule for many-server queues with abandonment. Operations Research 58(5):1427–1439.

Ayesta U, Jacko P, Novak V (2017) Scheduling of multi-class multi-server queueing systems with abandonments. Journal of Scheduling 20(2):129–145.

Bhulai S, Blok H, Spieksma F (2019) K competing queues with customer abandonment: optimality of a generalised $c\mu$-rule by the smoothed rate truncation method. Annals of Operations Research 1–30.

Cao Y (2017) Multiple server preemptive scheduling with impatience. Probability in the Engineering and Informational Sciences 31(2):226–238.

Down DG, Koole G, Lewis ME (2011) Dynamic control of a single-server system with abandonments. Queueing Systems 67(1):63–90.

Jacobson EU, Argon NT, Ziya S (2012) Priority assignment in emergency response. Operations Research 60(4):813–832.

Kim J, Randhawa RS, Ward AR (2018) Dynamic scheduling in a many-servermulti-class system: The role of customer impatience in large systems. Manufacturing andService Operations Management 20(2):285–301.

Li D, Glazebrook KD (2010) An approximate dynamic programing approach to the development of heuristics for the scheduling of impatient jobs in a clearing system. Naval Research Logistics 57(3):225–236.

Long Z, Shimkin N, Zhang H, Zhang J (2020) Dynamic scheduling of multiclass many-server queues with abandonment: The generalized cμ/h rule. Operations Research 68(4):1218–1230.

Pinedo ML (2008) Scheduling: Theory, algorithms, and systems .

Puha AL, Ward AR (2019) Tutorial paper: scheduling an overloaded multiclass many-server queue with impatient customer. In INFORMS TutORials in Operations Research. 189–217.

Ross SM (2015) A sequential scheduling problem with impatient jobs. Naval Research Logistics 62(8):659–663.