Dismissing Decision Tasks:
The Optimality of the $\mathcal{M}$–Shaped Structure

Saed Alizamir Francis de Véricourt Peng Sun

May 8, 2013

Abstract

We consider a sequential hypothesis-testing problem where the Decision Maker (DM) faces a random stream of decision tasks that accumulate over time, creating congestion. As in the classical set-up, the agent needs to dynamically choose when to terminate the information collection process and make a final decision. In our set-up, however, unattended tasks accumulate in a queue and incur additional delay-related costs. This gives the DM an incentive to dismiss tasks from the queue, in the sense that the DM makes her decision a priori without running any test. In this note, we examine when it is desirable to dismiss decision tasks such as these. To that end, we model the problem as a Partially Observed Markov Decision Process and fully characterize the optimal policy which maximizes the long-run average profit. Our analysis reveals the optimality of an $\mathcal{M}$-shaped structure. This structure implies that dismissing tasks can mitigate inefficiencies in the decision process that have been reported in the literature.

1. Introduction

Sequential hypothesis-testing is concerned with situations where the Decision Maker (DM) performs tests (e.g., sampling) sequentially prior to making a terminal decision about the prevailing state of the world. The DM's main objective, therefore, is to dynamically balance the cost of eliciting additional information against the benefit of improving the accuracy of her decision. The study of sequential hypothesis-testing problems dates back to DeGroot (1970), and has provided the foundation for various decision analysis models, from technology adoption (McCardle 1985) to commitment decisions on influenza vaccine composition (Kornish and Keeney 2008). Despite the large literature on how to perform a single decision task in this setting, little is known about the DM's optimal strategy when she faces a stream of arriving tasks, which accumulate until they are attended to and create time pressure in the form of congestion. Accordingly, the cost of eliciting additional information depends on the size of the accumulated workload. As a result, the DM will sometimes has an incentive to dismiss tasks from the system, in the sense that she makes an a priori decision, without spending any time searching for information.

The objective of this note is to determine how the DM should make dismissing decisions such as these. To that end, we consider a simple setup based on Alizamir, de Vericourt, and Sun (2013), where the DM faces a stream of decision tasks. Each task is either in state s or in its complement, state $\bar{s}$. The DM does not know the state, and sequentially observes binary (positive or negative) signals produced by imperfect and identical tests. The DM's subjective belief about the current task's actual state is fully captured by the intensity of preference, which is defined as the difference between the number of positive and negative test results. Arriving tasks accumulate in a queue, and the DM needs to decide (i) when to stop the search on the current task and (ii) whether or not to dismiss some of the tasks from the queue in order to manage the size of the accumulated workload.

We formulate this problem as a Markov Decision Process with the objective of maximizing the long-run average value. The state of the system is two-dimensional and consists of the number of tasks awaiting completion (n) and the DM's intensity of preference (d). We characterize the structure of the optimal policy, which takes the form of two nested intervals on the DM's intensity of preference. More specifically, the DM should continue to collect observations for the current task as long as d falls into a search interval (we call it the *Outer Interval*). When, however, d falls into the dismiss interval (the *Inner Interval*), gathering additional observation for the current task must be preceded by removing some unprocessed tasks from the system. Furthermore, the Outer Interval shrinks as congestion intensifies whereas the Inner Interval expands with workload size. This means that the optimal policy possesses an $\mathcal{M}$-shaped structure in the (n, d) space.

Our analysis extends the model proposed by Alizamir, de Vericourt, and Sun (2013), in the case of identical and symmetrical tests, by allowing the DM to dismiss some decision tasks. In Alizamir, de Vericourt, and Sun (2013), tasks also accumulate but cannot be dismissed from the queue without processing. The corresponding optimal policy is characterized by a search interval that shrinks as congestion intensifies. This structure also implies that when congestion intensifies, the DM may sometimes interrupt the current task's search process and make a choice that goes *against* the information that has been collected thus far. For example, the DM should sometimes decide state $\bar{s}$ for a task, even though d is positive. This effect imposes an undesirable inefficiency, as the DM sometimes needs to incur the cost of collecting information, which she subsequently has to ignore. By contrast, our analysis reveals that allowing the DM to dismiss tasks eliminates this type of inefficiency. More generally, we show that the structure of a search interval needs to be augmented by a dismissal interval which *expands* as congestion intensifies.

Our paper bears relevance to the considerable body of work on admission control in the queueing literature (see Glazebrook et al. 2009; Deo and Gurvich 2011; Lin and Ross 2003; Ormeci and Burnetas 2004; Turhan et al. 2012; Cui et al. 2009; Yildirim and Hasenbein 2010 for recent work in this field; also see Shmueli et al. 2003; Song-Hee et al. 2013 for empirical studies of admission decisions at hospital ICUs). This paper, however, appears to be the first one to consider an admission control problem when the service consists in making a sequential decision. Further, the decision task we study naturally corresponds to a diagnostic service in a healthcare setting, and our work is also related to recent models of triage nurses (see, for example, Argon and Ziya 2009; Dobson and Sainathan 2011; Shumsky and Pinker 2003; Lee et al. 2012). The main focus of our model in this context is on the initial triage (the gatekeeper or router) where patient's needs may need to be determined with varying levels of accuracy (including no information gathering at all). On the other hand, we account for the costs of the subsequent treatments in expectation instead of dynamically.

The remainder of this paper is as follows: We describe the model in the next section. Characterization of the optimal policy and all analytical results are presented in Section 3. The conclusion is provided in Section 4.

2. Model Formulation

We consider a DM to whom tasks arrive randomly over time, according to a Poisson process with rate λ, and are accumulated in a queue until they are processed. Each task is either in state s or in its complement, $\bar{s}$, which is unknown to the DM a priori. The collectable observations are the binary signals produced by a sequence of imperfect and identical tests, so that a positive (resp. negative) signal supports state s (resp. $\bar{s}$). The time required to run a test (or collect a sample) is exponentially distributed with rate μ, and the process is preemptive so that a test can be stopped at any time.

The DM assigns prior probability p_0 to each arriving task being in state s, which is revised in process after each observation is gathered. Tests are symmetrical in the sense that the conditional probability of a positive outcome given state s equals the conditional probability of a negative outcome given state $\bar{s}$, and is denoted by $\beta > 0.5$. We define the *intensity of preference*, d, as the difference between the number of positive and negative signals obtained thus far. Note that d is a sufficient statistic assuming the DM has access to infinite number of identical and symmetrical tests. We therefore denote p_d to represent the DM's subjective belief about the task in process being in state s when the intensity of preference equals d. If an additional observation is collected at this point, Bayes' rule is applied to update the subjective belief as

$$\Pr\{s|d,+\} = p_{d+1} = \frac{\beta p_d}{\beta p_d + (1-\beta)(1-p_d)}, \quad \text{and} \quad \Pr\{s|d,-\} = p_{d-1} = \frac{(1-\beta)p_d}{(1-\beta)p_d + \beta(1-p_d)},$$

for a positive and negative signal, respectively.

Correct decisions generate value and incorrect decisions incur losses. We do not assume the value and cost structure to be symmetric. In particular, correctly determining that a task is in state s (resp. $\bar{s}$) generates value v (resp. $\bar{v}$) whereas a wrong judgment about its state imposes the loss c (resp. $\bar{c}$). Moreover, the delay penalty $w(n)$ is incurred per unit time if there are n tasks accumulated in the workload. In connection with two-level service processes, these parameters implicitly include the value/cost of the subsequent treatments, which depend on the accuracy of the diagnosis by the gatekeeper.

If the DM decides to stop and commit to a terminal decision on the current task when the intensity of preference is d, her expected reward becomes

$$r(p_d) = p_d v - (1 - p_d)\bar{c}, \qquad \text{and} \qquad \bar{r}(p_d) = (1 - p_d)\bar{v} - p_d c,$$

for favoring states s and $\bar{s}$, respectively. It follows immediately that state s is preferable over state $\bar{s}$ if p_d exceeds the critical fraction

$$\theta = \frac{\bar{v} + \bar{c}}{v + c + \bar{v} + \bar{c}},$$

and state $\bar{s}$ is favored otherwise. Without loss of generality, we assume that $p_0 < \theta$, and normalize $\bar{r}(p_0)$ to equal zero. Further, we can define d_θ as the largest d for which it is better to favor state $\bar{s}$, that is $d_\theta = \max\{d; p_d < \theta\}$. As d approaches d_θ, the DM is increasingly indifferent between either alternatives.

3. Analysis and Results

The optimal policy must determine, at any point in time, the best action among the following four options: (i) terminate testing on the current task and identify state s, (ii) terminate testing on the current task and identify state $\bar{s}$, (iii) acquire at least one more observation on the current task before committing to any decision, (iv) reduce the size of the accumulated workload by dismissing a task from the queue (which should be identified as in state $\bar{s}$ since $p_0 < \theta$). The performance of a policy is evaluated as the long-run average profit, which includes reward for correct decisions and penalties for misidentifications and delays.

We formulate the problem as a Markov Decision Process where the state of the system is given by (n, d), the number of tasks awaiting completion and the intensity of preference for the task in process. As mentioned before, this is because d provides a sufficient statistic for all the observed signals thus far. We assume, without loss of generality, that $\lambda + \mu = 1$, and apply uniformization to obtain the corresponding Bellman's equation as,

$$
\begin{aligned}
g + J(n,d) \quad &= \quad \max\Big\{ -w(n) + \lambda J(n+1,d) + \mu\big(\beta p_d + (1-\beta)(1-p_d)\big) J\big(n,d+1\big) \\
&\qquad +\mu\big((1-\beta)p_d + \beta(1-p_d)\big) J\big(n,d-1\big), \\
&\qquad g + r(p_d) + J(n-1,0), \\
&\qquad g + \bar{r}(p_d) + J(n-1,0), \\
&\qquad g + J(n-1,d)\Big\}, \quad \text{for } n \geq 1 \text{ and any } d, \qquad (1) \\
g + J(0,0) \quad &= \quad \lambda J(1,0) + \mu J(0,0),
\end{aligned}
$$

where g represents the long-run average profit, and $J(.,.)$ is the bias function.

The characterization of the optimal policy is quite involved and follows the steps outlined in the appendix. This leads to the following result,

Theorem 1. *For any given n, there exists intervals $(\bar{d}(n), \underline{d}(n))$ and $(\hat{d}(n), \check{d}(n))$ so that $\bar{d}(n) \leq \hat{d}(n) \leq d_\theta < \check{d}(n) \leq \underline{d}(n)$. When the system is at state (n,d), it is optimal to*

- *terminate testing in favor of state $\bar{s}$ if $d \leq \bar{d}(n)$,*
- *terminate testing in favor of state s if $d \geq \underline{d}(n)$,*
- *dismiss a task from the queue if $\hat{d}(n) < d < \check{d}(n)$, and*
- *continue testing on the current task otherwise.*

Hence, for any fixed queue size, the optimal policy takes the form of two nested intervals on the intensity of preference, where the outer interval (search interval) determines the continuation of the process on the current task and the inner interval (dismiss interval) regulates the adjustment of the workload by dismissing unprocessed tasks from the queue. The next result shows how these thresholds vary with n.

Theorem 2. *Thresholds $\bar{d}(n)$ and $\check{d}(n)$ increase in n, whereas thresholds $\hat{d}(n)$ and $\underline{d}(n)$ decrease with n.*

To derive the above results, we have to first consider the corresponding total discounted profit model and obtain the structural properties of the value function. The proof approach includes subtle use of value iteration and sample path arguments. We can then use SEN Conditions (Sennott 1999) to extend the results to the long-run average profit model.

Figure 1 illustrates the optimal policy in the (n,d) space for a numerical example. The upward (resp. downward) triangles are the states for which making the terminal decision in favor of s (resp. $\bar{s}$) is optimal. The triangles facing the d-axis, on the other hand, represent the states that correspond to dismissing a task from the queue. Continue observation

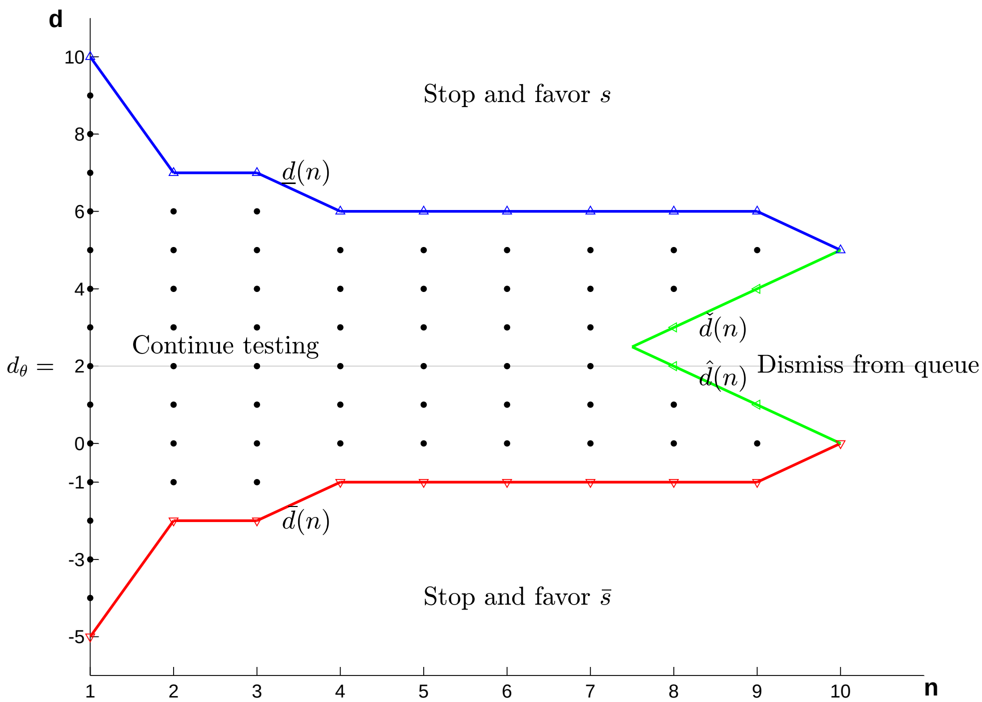

Figure 1: Illustration of the Optimal Policy for a Numerical Example: $\rho = \lambda/\mu = 0.1,\ p_0 = 0.3,\ \beta = 0.6,\ v = 1600,\ \bar{v} = 2000,\ c = \bar{c} = 0,\ w(n) = n.$

collection on the current task is optimal for the states depicted by dots. The optimal policy implies that terminal decision on the task has to be made only if the magnitude of d is large enough, i.e., certainty about its state is sufficiently high. On the other hand, at higher levels of ambiguity on the current task (d being close to d_θ), the optimal policy may prescribe dismissing a task from the workload because a long processing time on the current task is expected. Furthermore, as the size of the workload grows, the DM has two different levers to mitigate high delay penalties. She can reduce n by either removing the current task in service while possibly sacrificing its accuracy, or dismiss a task from the queue. Theorem 2 implies that both of these levers are exploited as congestion intensifies, so that the search and dismiss intervals shrink and expand, respectively, with congestion. Note that dismissing a task does not mean that the task is disregarded but rather that the decision is taken upfront. In a medical diagnostic setting, for example, this may mean that an arriving patient is treated as sick upon arrival without any diagnosis, when the triage nurse is overwhelmed.

An alternative representation of the optimal policy is in terms of thresholds on the maximum number of tasks allowed in the system. Then, the optimal policy can be described as thresholds $\bar{n}(d)$ so that when the intensity of preference is d, it is optimal to continue the process if $n < \bar{n}(d)$, and to release a task otherwise. The task which has to be dismissed in this case is either the task in process or a task from the queue depending on the value of d. With this interpretation, the threshold $\bar{n}(d)$ demonstrates an $\mathcal{M}$-shaped structure in d.

Finally, the optimal policy in our setting provides a remedy for an undesirable inefficiency described in Alizamir, de Vericourt, and Sun (2013), where tasks must be released in order of their arrivals. More specifically, with dismissing decisions, the DM never commits to a terminal decision that is against what the gathered information suggests. This is formally stated in the following corollary.

Corollary 1. *When dismissing tasks from the queue is allowed, the DM never commits to a terminal decision which is against the obtained information. In particular, it is never optimal to stop the process and choose state $\bar{s}$ (reps. s) for the current task when $d > 0$ (reps. $d < 0$).*

This is because when $d > 0$, the DM is always better off dismissing a task from the queue (by choosing $\bar{s}$) instead of terminating the search process on the current one, should she release a task due to high congestion.

4. Conclusion

In this note, we employ a simple paradigm to study dismissing decisions for a sequential hypothesis testing problem where tasks accumulate. The DM gathers information on each task, by performing a sequence of identical and symmetrical tests, prior to committing to a terminal decision. We show that the optimal policy in this setting can be characterized by two nested intervals so that the outer interval determines the treatment of the current task and the inner interval regulates the dismissing decisions as congestion intensifies. Furthermore, the former shrinks whereas the latter expands with workload size. Finally, we show that the flexibility of dismissing tasks from the workload eliminates the inefficiency identified by Alizamir, de Vericourt, and Sun (2013), by which the DM sometimes need to make choices against the gathered information.

Our assumption of identical tests allows us to model the DM's belief as her intensity of preference. This corresponds to situations where each test is a new sample from the same population, which applies to many practical applications (e.g., quality inspection; see Alizamir, de Vericourt, and Sun 2013 for more detail). This is also in parallel to the sequential decision making literature where an unknown state of nature is responsible for producing independent and identically distributed observations (see Rapoport and Burkheimer 1971 for one of the first works following this approach). The symmetry requirement on tests has also been dominant in previous research on psychological models of deferred decision making (Busemeyer and Rapoport 1988; Edwards 1965). On the other hand, further research is needed to explore the optimal policy when tests are not identical. The problem becomes a three dimensional Markov Decision Process and is therefore significantly more challenging. Another interesting extension is when the DM needs also to dynamically determine the order of the tests. Finally, building on the aforementioned psychology literature, we believe our approach constitutes a very promising framework for understanding how individuals make actual dismissal decisions such as those studied in this note, when tasks can accumulate.

References

Alizamir, S., F. de Vericourt, and P. Sun (2013). Diagnostic accuracy under congestion. *Management Science 59*(1), 157–171.

Argon, N. and S. Ziya (2009). Priority assignment under imperfect information on customer type identities. *Manufacturing & Service Operations Management 11*(4), 674–693.

Bertsekas, R. (2007). *Dynamic Programming and Optimal Control*, Volume II. Athena Scientific.

Busemeyer, J. R. and A. Rapoport (1988). Psychological models of deferred decision making. *Journal of Mathematical Psychology 32*, 91–134.

Cui, T., Y. Chen, and Z. Shen (2009). Optimal pricing, scheduling, and admission control for queueing systems under information asymmetry. *University of California at Berkeley working paper*.

DeGroot, M. (1970). *Optimal Statistical Decisions.* McGraw-Hill, New York.

Deo, S. and I. Gurvich (2011). Centralized vs. decentralized ambulance diversion: A network perspective. *Management Science 57*(7), 1300–1319.

Dobson, G. and A. Sainathan (2011). On the impact of analyzing customer information and prioritizing in a service system. *Decision Support Systems 51*, 875–883.

Edwards, W. (1965). Optimal strategies for seeking information: Models for statistics, choice reaction times, and human information processing. *Journal of Mathematical Psychology 2*(2), 312–329.

Glazebrook, K. D., C. Kirkbride, and J. Ouenniche (2009). Index policies for the admission control and routing of impatient customers to heterogeneous service stations. *Operations Research 57*(4), 975–989.

Kornish, L. and R. Keeney (2008). Repeated commit-or-defer decisions with a deadline: The influenza vaccine composition. *Operations Research 56*(3), 527–541.

Lee, H., E. Pinker, and R. Shumsky (2012). Outsourcing a two-level service process. *Management Science 58*(8), 1569–1584.

Lin, K. and S. M. Ross (2003). Admission control with incomplete information of a queueing system. *Operations Research 51*(4), 645–654.

McCardle, K. (1985). Information acquisition and the adoption of new technology. *Management Science 31*(11), 1372–1389.

Ormeci, E. L. and A. Burnetas (2004). Admission control with batch arrivals. *Operations Research Letters 32*, 448–454.

Rapoport, A. and G. J. Burkheimer (1971). Models for deferred decision making. *Journal of Mathematical Psychology 8*, 508–538.

Sennott, L. I. (1999). *Stochastic Dynamic Programming and the Control of Queueing Systems*. Wiley-Interscience, New York.

Shmueli, A., C. Sprung, and E. Kaplan (2003). Optimizing admissions to an intensive care unit. *Health Care Management Science 6*(3), 131–136.

Shumsky, R. and E. Pinker (2003). Gatekeepers and referrals in services. *Management Science 49*(7), 839–856.

Song-Hee, K., C. Chan, M. Olivares, and G. Escobar (2013). ICU Admission Control: An Empirical Study of Capacity Allocation and its Implication on Patient Outcomes. *Columbia Business School working paper*.

Turhan, A., M. Alanyali, and D. Starobinski (2012). Optimal admission control in two-class preemptive loss systems. *Operations Research Letters*. forthcoming.

Yildirim, U. and J. Hasenbein (2010). Admission control and pricing in a queue with batch arrivals. *Operations Research Letters 38*, 427–431.

Appendix

We start our analysis by first considering the corresponding model with total discounted value objective. After establishing the structural properties of the optimal value function and proving our results in this setting, we show that they can be extended to the model with long-run average value objective by letting the discount rate go to zero.

When the objective is to maximize the total discounted value, and assuming the discount rate to be γ, the optimal value function satisfies $\Gamma J^*(.,.) = J^*(.,.)$, where the operator Γ is defined in parallel to Bellman's Equation (1) as,

$$\begin{aligned}
\Gamma J(n,d) \quad &= \quad \max\Big\{ -w(n) + \lambda J(n+1,d) + \mu\big(\beta p_d + (1-\beta)(1-p_d)\big) J\big(n,d+1\big) \\
&\qquad +\mu\big((1-\beta)p_d + \beta(1-p_d)\big) J\big(n,d-1\big), \\
&\qquad r(p_d) + J(n-1,0), \\
&\qquad \bar{r}(p_d) + J(n-1,0), \\
&\qquad J(n-1,d)\Big\}, \quad \text{for } n \geq 1 \text{ and any } d, \qquad (2) \\
\Gamma J(0,0) \quad &= \quad \lambda J(1,0) + \mu J(0,0).
\end{aligned}$$

In the above optimality equation, we have assumed, without loss of generality, that $\lambda + \mu + \gamma = 1$.

Lemma 1. *The optimal value function, $J^*(.,.)$, satisfying Bellman's Equation (2) such that $\Gamma J^*(.,.) = J^*(.,.)$ uniquely exists, and can be obtained by the value iteration algorithm starting from any arbitrary function $J_0(.,.)$, i.e.,*

$$\lim_{k\to\infty} \Gamma^{(k)} J_0(.,.) = J^*(.,.) \; .$$

Proof: Since we have a maximization problem and the instantaneous costs $-w(n)$, $r(p_d)$ and $\bar{r}(p_d)$ are bounded from above, the Negativity Assumption holds and the result follows (Bertsekas 2007, Proposition 3.1.6).

■

Lemma 2. *For the optimal value function, $J^*(.,.)$, and any fixed d we have*

$$J^*(n,d) - J^*(n-1,0) \searrow n \; .$$

Proof: Suppose the system is at state $(n+1,d)$ and operates under the optimal policy. Let T be a random variable representing the time needed to either reach state $(n,0)$ or dismiss an unprocessed task for the first time. First, $T < \infty$ because otherwise, the queue length must always remain higher than n, which cannot be true under the optimal policy.

Next consider a policy u which is different than the optimal policy in the following manner. Instead of following the optimal policy for the current system, policy u follows the optimal policy for an otherwise identical system with one extra task in the queue. More precisely, when the system is at state (n', d'), policy u chooses the optimal action corresponding to state $(n'+1, d')$. Define $\hat{w} = \min_n \{w(n+1) - w(n)\}$. Now assume that the system is at state (n, d) and operates under policy u over a period of length T. After time T is elapsed, one of the following two scenarios has happened:

(i) The system is at some state (n', d'), and the optimal policy at state $(n'+1, d')$ prescribes dismissing an unprocessed task from the queue. In this case, policy u does not dismiss any task from the queue and coincides with the optimal policy afterwards. It follows that,

$$J^u(n,d) \geq \Big(J^*(n{+}1,d) {-} \mathbb{E}\left[e^{-\gamma T}\right] J^*(n'{+}1,d')\Big) {+} \mathbb{E}\left[e^{-\gamma T}\right] J^*(n',d') {+} \mathbb{E}\left[\int_0^T \hat{w} e^{-\gamma t} dt\right]$$

$$\Rightarrow J^u(n,d) \geq \Big(J^*(n{+}1,d) {-} \mathbb{E}\left[e^{-\gamma T}\right] J^*(n',d')\Big) {+} \mathbb{E}\left[e^{-\gamma T}\right] J^*(n',d') {+} \mathbb{E}\left[\int_0^T \hat{w} e^{-\gamma t} dt\right]$$

$$\Rightarrow J^u(n,d) \geq J^*(n+1,d) + \mathbb{E}\left[\int_0^T \hat{w} e^{-\gamma t} dt\right].$$

(ii) The system is at some state (n, d') and the optimal policy at state $(n{+}1, d')$ prescribes releasing the task in service. In this case, policy u also releases the task in service and continues following the optimal policy afterwards. It follows that,

$$J^u(n,d) \geq \Big(J^*(n{+}1,d) {-} \mathbb{E}\left[e^{-\gamma T}\right] J^*(n,0)\Big) {+} \mathbb{E}\left[e^{-\gamma T}\right] J^*(n{-}1,0) {+} \mathbb{E}\left[\int_0^T \hat{w} e^{-\gamma t} dt\right].$$

Noting that $J^*(n,d) \geq J^*(n-1,d)$, we can deduce in both cases that

$$J^*(n,d) \geq J^u(n,d) \geq \Big(J^*(n+1,d) - \mathbb{E}\left[e^{-\gamma T}\right] J^*(n,0)\Big) + \mathbb{E}\left[e^{-\gamma T}\right] J^*(n-1,0)$$

$$\Rightarrow J^*(n,d) - \mathbb{E}\left[e^{-\gamma T}\right] J^*(n-1,0) \geq J^*(n+1,d) - \mathbb{E}\left[e^{-\gamma T}\right] J^*(n,0).$$

Furthermore, $J^*(n,d) \nearrow n$ implies, for any finite T, that

$$-\mathbb{E}\left[1 - e^{-\gamma T}\right] J^*(n-1,0) \geq -\mathbb{E}\left[1 - e^{-\gamma T}\right] J^*(n,0).$$

Adding the above two inequalities gives us

$$J^*(n,d) - J^*(n-1,0) \geq J^*(n+1,d) - J^*(n,0).$$

■

Lemma 3. *For the optimal value function, $J^*(.,.)$, and any fixed n we have*

$$J^*(n,d) - \bar{r}(p_d) \nearrow d\ .$$

Proof: We prove that operator Γ propagates the property $J(n,d) - \bar{r}(p_d) \nearrow d$. This, together with Lemma 1 completes the proof.

For any $0 \leq p \leq 1$, let $p^+ = \beta p + (1-\beta)(1-p)$ and $p^- = (1-\beta)p + \beta(1-p)$. We start from the following equality (simple algebra shows that this equality holds):

$$\bar{r}(p_d) - p_d^+ \bar{r}(p_{d+1}) - p_d^- \bar{r}(p_{d-1}) = \bar{r}(p_{d-1}) - p_{d-1}^+ \bar{r}(p_d) - p_{d-1}^- \bar{r}(p_{d-2})$$

Replacing p^- with $1-p^+$ and some straightforward algebra:

$$\big(p_d^+ - p_{d-1}^+\big)\big(\bar{r}(p_{d-1}) - \bar{r}(p_d)\big) = p_d^+\big(\bar{r}(p_{d+1}) - \bar{r}(p_d)\big) - \bar{r}(p_d) + \bar{r}(p_{d-1}) + (1-p_{d-1}^+)\big(\bar{r}(p_{d-1}) - \bar{r}(p_{d-2})\big)$$

With the assumption that $J(.,.)$ holds the property:

$$\big(p_d^+ - p_{d-1}^+\big)\big(J(n,d-1) - J(n,d)\big) \leq p_d^+\big(\bar{r}(p_{d+1}) - \bar{r}(p_d)\big) - \bar{r}(p_d) + \bar{r}(p_{d-1}) + (1-p_{d-1}^+)\big(\bar{r}(p_{d-1}) - \bar{r}(p_{d-2})\big)$$

Adding $J(n,d-1)$ to both sides:

$$\begin{aligned} p_{d-1}^+ J(n,d) + (1-p_{d-1}^+)\big(J(n,d-1) - \bar{r}(p_{d-1}) + \bar{r}(p_{d-2})\big) - \bar{r}(p_{d-1}) \leq \\ p_d^+\big(J(n,d) + \bar{r}(p_{d+1}) - \bar{r}(p_d)\big) + (1-p_d^+)J(n,d-1) - \bar{r}(p_d) \end{aligned}$$

With the assumption that $J(.,.)$ holds the property:

$$p_{d-1}^+ J(n,d) + (1-p_{d-1}^+)J(n,d-2) - \bar{r}(p_{d-1}) \leq p_d^+ J(n,d+1) + (1-p_d^+)J(n,d-1) - \bar{r}(p_d)$$

Multiplying both sides by μ and another use of the assumption:

$$\begin{aligned} \lambda J(n+1,d-1) + \mu p_{d-1}^+ J(n,d) + \mu p_{d-1}^- J(n,d-2) - (\lambda+\mu)\bar{r}(p_{d-1}) \leq \\ \lambda J(n+1,d) + \mu p_d^+ J(n,d+1) + \mu p_d^- J(n,d-1) - (\lambda+\mu)\bar{r}(p_d) \end{aligned}$$

Adding the inequality $-\gamma\bar{r}(p_{d-1}) \leq -\gamma\bar{r}(p_d)$ to the above inequality:

$$\lambda J(n+1,d-1) + \mu p_{d-1}^{+} J(n,d) + \mu p_{d-1}^{-} J(n,d-2) - \bar{r}(p_{d-1}) \leq \\ \lambda J(n+1,d) + \mu p_d^{+} J(n,d+1) + \mu p_d^{-} J(n,d-1) - \bar{r}(p_d).$$

Another use of the assumption and noting that the maximum of a set of increasing functions is still increasing implies that,

$$\Gamma J(n,d-1) - \bar{r}(p_{d-1}) \leq \Gamma J(n,d) - \bar{r}(p_d)$$

$$\Rightarrow \Gamma J(n,d) - \bar{r}(p_d) \nearrow d\ .$$

■

Lemma 4. *For the optimal value function, $J^*(.,.)$, and any fixed n we have*

$$J^*(n,d) - r(p_d) \searrow d\ .$$

Proof: We prove that operator Γ propagates the property $J(n,d) - r(p_d) \searrow d$. This, together with Lemma 1 completes the proof.

We start from the following equality:

$$r(p_d) - p_d^{+} r(p_{d+1}) - p_d^{-} r(p_{d-1}) = r(p_{d-1}) - p_{d-1}^{+} r(p_d) - p_{d-1}^{-} r(p_{d-2}).$$

Following steps very similar to those in the proof of Lemma 3, we deduce that,

$$\Gamma J(n,d-1) - r(p_{d-1}) \geq \Gamma J(n,d) - r(p_d)$$

$$\Rightarrow \Gamma J(n,d) - r(p_d) \searrow d\ .$$

■

Lemma 5. *For the optimal value function, $J^*(.,.)$, and any fixed n and d, if $J^*(n+1,d) > J^*(n,d)$ then $J^*(n,d) > J^*(n-1,d)$.*

Proof: The proof proceeds using a sample path argument. Consider two systems, A and B, which are at states $(n+1,d)$ and (n,d), respectively, and let T be a random variable representing the time until the first task is released from system A. Also let N_T denote the number of new arrivals over T. Assume $J^*(n+1,d) > J^*(n,d)$ holds. This inequality implies that $J^*(n+1,d) > J^u(n,d)$ where policy u in system B works as follows: at each state, policy u adds an auxiliary task to the end of the queue and then follows the optimal policy until time T is elapsed. Thus, over time T, system B is able to collect all the rewards that is collected under the optimal policy in system A, while incurs a lower holding cost (because of holding one fewer task).

After time T, if the task which is released from system A is an unprocessed task, system B does not release a task and both systems transit into state $(n + N_T, d')$, where d' is the intensity of preference at time T. Along these sample paths, system B under policy u outperforms system A under the optimal policy because of incurring a lower holding cost. On the other hand, if system A releases the current task at time T, it collects $\max\{r(d'), \bar{r}(d')\}$ and transits into state $(n + N_T, 0)$. In this case, system B under policy u does not release a task and hence, remains at state $(n + N_T, d')$.

Next, let T' be a random variable representing the time at which system A releases the task in process and $N_{T'}$ the number of new arrivals during this period. During the $(T, T']$ period, system B under policy u does the same action as system A under the optimal policy. Note that during this period both systems incur the same holding cost and transit into state $(n+N_T+N_{T'}-1, 0)$ afterwards. Thus, $J^*(n+1, d) > J^u(n, d)$ implies that $\max\{r(d'), \bar{r}(d')\}$ plus the expected reward from a task with $d = 0$ is higher than the expected reward from a task with $d = d'$.

Now, consider two systems, C and D, which are at states (n, d) and $(n - 1, d)$, respectively. We intend to design a policy u' for system C so that it outperforms system D operating under the optimal policy. Under policy u', system C continues testing on the current task until the intensity of preference is d', and then it releases the task in process and collects the expected reward of $\max\{r(d'), \bar{r}(d')\}$. It then continues performing tests on the next task in line until system D releases its current task. From $J^*(n + 1, d) > J^*(n, d)$, we know that $\max\{r(d'), \bar{r}(d')\}$ plus the expected reward from a task with $d = 0$ is higher than the expected reward from a task with $d = d'$. It follows that $J^{u'}(n, d) > J^*(n - 1, d)$.

■

Lemmas 2-5 establish structural properties of the optimal value function when the objective is to maximize the total discounted value. An argument based on SEN Conditions (Sennott 1999), similar to that of Alizamir, de Vericourt, and Sun (2013), extends all these properties to the long-run average objective (see Alizamir et al. 2013 for complete details). Then, these properties are used to prove the main results of the paper, as follows.

Proof of Theorem 1

From Lemma 3, if $J^*(n, d) - \bar{r}(p_d) = J^*(n-1, 0)$ then $J^*(n, d-1) - \bar{r}(p_{d-1}) = J^*(n-1, 0)$. That is, if it is optimal to stop the search at state (n, d) and identify the current task in favor of $\bar{s}$, the same action is also optimal at state $(n, d - 1)$. Thus, there exists threshold $\bar{d}(n)$ so that it is optimal to terminate testing in favor of state $\bar{s}$ if $d \leq \bar{d}(n)$. Similarly, Lemma 4 implies that there exists a threshold $\underline{d}(n)$ so that it is optimal to terminate testing in favor of state s if $d \geq \underline{d}(n)$. It remains to characterize the optimal action at states (n, d) with $\bar{d}(n) < d < \underline{d}(n)$.

Our goal is to prove that if dismissing an unprocessed task from the queue is optimal at states (n, d') and (n, d'') for some $\bar{d}(n) < d' < d'' < \underline{d}(n)$, then the same action is also optimal at state (n, d) with $d' < d < d''$. First, note that Lemma 5 also implies that dismissing an unprocessed task is the optimal action at all states (n', d') and (n', d'') with $n' > n$. Hence, releasing the current task cannot be optimal at any state (n', d) with $n' > n$ and $d' < d < d''$. Putting all together, we deduce that starting from state (n, d) and along any possible sample path, one of the following happens:

(i) the system reaches state (n, d'),

(ii) the system reaches state (n, d''),

(iii) the system reaches state (n', d') or (n', d'') for some $n' > n$,

(iv) the system dismisses an unprocessed task prior to reaching any of the above states.

In all of cases (i)-(iv), an unprocessed task must be dismissed from the queue. In other words, along any sample path from state (n, d), a task is dismissed from the queue before reaching state $(0, 0)$. However, this cannot be optimal because the system would have been better off by dismissing that task now rather than incurring the holding cost over a non-zero time and then dismissing it without any processing. This means that it should be the optimal action at state (n, d) to dismiss a task from the queue.

■

Proof of Theorem 2

From Lemma 2, if $J^*(n, d) - J^*(n-1, 0) = \bar{r}(p_d)$ then $J^*(n+1, d) - J^*(n, 0) = \bar{r}(p_d)$. That is, if identifying the current task in favor of $\bar{s}$ is optimal at state (n, d), the same action is also optimal at state $(n+1, d)$. This implies that $\bar{d}(n)$ is increasing. Similarly, if $J^*(n, d) - J^*(n-1, 0) = r(p_d)$ then $J^*(n+1, d) - J^*(n, 0) = r(p_d)$. As a result, $\underline{d}(n)$ is decreasing.

Furthermore, from Lemma 5 it follows that if $J^*(n, d) - J^*(n-1, d) = 0$ then $J^*(n+1, d) - J^*(n, d) = 0$. That is, if dismissing a task from the queue is optimal at state (n, d), the same action is also optimal at state $(n+1, d)$. This completes the proof about the monotonicity of $\hat{d}(n)$ and $\check{d}(n)$.

■

Proof of Corollary 1

From Equation (1), note that the two actions of dismissing a task from the queue and releasing the task in process in favor of state $\bar{s}$ are identical at any state $(n, 0)$, which implies that $\bar{d}(n) \leq 0 \leq \hat{d}(n)$. Therefore, dismissing a task from the queue always dominates releasing the task in process in favor of state $\bar{s}$ when $d > 0$.

■