Search Under Accumulated Pressure

Saed Alizamir
School of Management, Yale University, New Haven, CT, saed.alizamir@yale.edu
Francis de Véricourt
ESMT, Berlin, Germany, devericourt@esmt.org
Peng Sun
Fuqua School of Business, Duke University, Durham, NC, peng.sun@duke.edu

Arrow et al. (1949) introduced the first sequential search problem, "where at each stage the options available are to stop and take a definite action or to continue sampling for more information." We study how time pressure in the form of task accumulation may affect this decision problem. To that end, we consider a search problem where the decision maker (DM) faces a stream of random decision tasks to be treated one at a time, and accumulate when not attended to. We formulate the problem of managing this form of pressure as a Partially Observable Markov Decision Process, and characterize the corresponding optimal policy. We find that the DM needs to alleviate this pressure very differently depending on how the search on the current task has unfolded thus far. As the search progresses, the DM is less and less willing to sustain high levels of workloads in the beginning and end of the search, but actually increases the maximum workload she is willing to handle in the middle of the process. The DM manages this workload by first making a priori decisions to release some accumulated tasks, and later by aborting the current search and deciding based on her updated belief. This novel search strategy critically depends on the DM's prior belief about the tasks, and stems, in part, from an effect related to the decision ambivalence. These findings are robust to various extensions of our basic set-up.

1. Introduction

Information-search problems are concerned with situations where a Decision Maker (DM) elicits information overtime prior to making a final choice. In this setting, the DM dynamically balances the benefit of improving her belief about the best choice to make against the cost of acquiring more information. Many important decisions correspond to this search problem, including technology adoptions (McCardle 1985; Ulu and Smith 2009; Smith and Ulu 2012; Smith and Ulu 2017), decisions on vaccine composition (Kornish and Keeney 2008), research project management (McCardle et al. 2017), or medical diagnostic tasks (Alizamir et al. 2013). Despite the large literature on how to perform a single decision task in this set-up, little is known about the DM's best strategy when additional and unattended decision tasks build up overtime, creating time pressure in the form of accumulation.

Yet, congestion and task accumulation are pervasive. A triage nurse, for example, elicits different pieces of information to assess whether or not a patient should be admitted or treated (Gerdtz and

Bucknall 2001). However, new patients may show up in the meantime, creating long waits that could result in adverse outcomes (Travers 1999). Similar issues also frequently occur in non-diagnostic systems, such as support centers and help desks (de Vericourt and Zhoug 2005). Accumulation also takes the form of research project pipelines (Loch and Terwiesch 1999). Pharmaceutical companies, for instance, need to determine whether their new drugs will pass a series of clinical trials. However, facilities and volunteers for these trials are scarce, which may delay the development of other treatments and incur opportunity costs (Girotra et al. 2007).

These accumulation effects change the structure of a search problem in two very fundamental ways. First, the cost of acquiring more information is endogenous in this case. This cost depends on the size of the accumulated workload, and the more time is spent on a search, the higher the workload and thus the future search costs. Second, the workload may induce the DM to decide a priori, i.e. without any search. For a single task, the DM needs to abort the decision process (and commit to a choice) if the cost of acquiring more information is too high. With accumulation, by contrast, the DM can sometimes alleviate this cost by directly reducing the workload. Indeed, when additional decision tasks have accumulated, the DM can always release some of them immediately by deciding a priori.

The objective of this paper is to explore these effects and understand how a DM should sequentially acquire information for a decision, while additional decision tasks accumulate in the meantime. We seek to identify when and how the DM should alleviate this form of pressure.

To that end, we consider a simple set-up where the DM faces a stream of decision tasks. Each task consists of an elementary search problem, in which the DM sequentially collects observations to determine the better of two possible choices. Each observation either reveals the better option or is inconclusive. The DM has a prior belief about the optimal choice, which she updates as she collects observations. Gathering this information, however, takes time. As the search progresses, new decision tasks might accumulate until they are attended to. At any time instant, the DM needs to decide (i) whether to abort the current search and commit to a decision based on her current belief, or to continue the search and (ii) whether to release some of the accumulated tasks by deciding a priori, without collecting any observation.

We formulate this problem as a Partially Observable Markov decision process (POMDP), in which the true type of each task is unknown. The type determines which of two possible options is more suited for a given task. Correctly identifying this option brings value, but choosing the wrong one incurs penalty. The DM can collect additional observations to determine the task type, and the time to collect each observation is exponentially distributed. In essence, a decision task in our set-up corresponds to a special case of the sequential search problem pioneered by Arrow et al. (1949), and a discrete version of the project decision studied in McCardle et al. (2017). In our context, however, new tasks arrive to the system and accumulate until they are attended to. Hence, the state of the system consists of not only the number of observations the DM has collected thus far on the current search, but also the number of tasks awaiting completion. The objective is to maximize the expected total discounted profit.

We identify key properties of the optimal value function for this problem and fully characterize the structure of the corresponding optimal policy. Our proofs rely on an original induction approach on the number of accumulated tasks, which links the DM's optimal choices under a given workload level with those under higher levels.

This analysis reveals new insights on making decisions when the DM is subject to time pressure in the form of accumulation. We find that the DM needs to alleviate this pressure very differently depending on the stage of a given search. As the search progresses, the maximum number of accumulated tasks the DM is willing to handle decreases both in the beginning and end of the search, but actually *increases* in the middle of it. The DM does this by first making a priori decisions to release some accumulated tasks. Later, the DM aborts the current search and takes a decision based on her updated belief. The shape of this strategy critically depends on the DM's prior belief about which option is better.

More specifically, we identify two regimes for this problem, which depend on whether the DM's prior probability for the better option exceeds the critical level at which the DM is indifferent between the two options. In the first regime, the prior is below the indifference level. In this case, a search that remains inconclusive tends to falsify the DM's prior belief about which option is better. By contrast, in the second regime where the prior is above the indifference level, an inconclusive search tends to confirm the DM's prior choice.

In the first regime with a low prior belief, we find that the DM is less and less willing to sustain high levels of task accumulation at the start of a search. As a result, she decreases the maximum workload she is willing to handle, by completing some of the accumulated tasks based solely on her prior belief. If the current search remains inconclusive for some time, however, this effect is reversed. The DM becomes willing to sustain higher and higher accumulation levels and increases the maximum workload as the search progresses. Finally, the DM decreases this workload again when the search remains inconclusive for too long. In contrast to the previous phases, however, the DM reduces the workload by aborting the current search instead of releasing a task that has accumulated. In this case, she decides based on her updated belief about the better choice to make, and moves to the next task.

This strategy directly stems from the two ways in which accumulation affects a search problem: the endogeneity of search costs and the prevalence of a-priori decisions. To see this, consider first the beginning of a search task. Two different effects provide the DM with an incentive to reduce her workload: the decision problem's ambivalence and the search's inconclusiveness.

First, as the search progresses, the DM's belief that one option is better increases, while her belief about the other option decreases. In the first regime, her prior belief about the first option is sufficiently low, and thus the DM prefers the second option a priori. As the search progresses, however, her updated belief approaches the critical level at which she is indifferent between the two options. In other words, the decision problem becomes increasingly ambivalent. The more ambivalent the decision becomes, however, the more information the DM is willing to elicit, and thus the longer she expects the search to last. Hence, the DM has a stronger incentive to decrease future search costs and thus the workload.

Second, the longer the search fails to reveal the better choice in full, the less the DM is convinced that the next observation will be conclusive. In other words, the more the search progresses, the more likely it will remain inconclusive. As the search progresses, therefore, the DM is more inclined to reduce the current workload if she wants to pursue the search.

Both effects explain why the DM decreases the maximum workload she is willing to bear in the beginning of a search. Nonetheless, if this process continues for a period long enough, the DM's belief reaches the point at which she is indifferent between both options. Her preferred option switches from the second to the first one, and from now on, the decision becomes less and less ambivalent as the search progresses. The first effect is hence reversed.

At the same time, the second effect becomes stronger, and the DM's subjective probability that the search will reveal the better option diminishes. This, in turn, increases the DM's incentive to abort the search on the current task. Anticipating this, she is increasingly willing to replenish her workload (to generate future gains). Indeed, once the DM interrupts a search and starts a new task, the maximum workload she is willing to accept is reset to a high level. As a result, the DM starts to increase the workload as the search progresses past a certain point.

In what precedes, the DM decreases the workload by sometimes deciding without search on some of the accumulated tasks. In the final stage of the process, however, the DM becomes sufficiently confident about which option is better. At some point, therefore, the DM prefers to conclude the current search rather than foregoing future gains from accumulated tasks. In this final stage, the maximum workload again decreases as the search progresses, since the marginal value of collecting observations decreases. This goes on until the process stops and the DM moves to the next task.

By contrast, consider the second regime, where the DM's prior belief about the first option is high. She prefers this option a priori and becomes more confident about her choice as the search progresses. In this sense, an inconclusive search actually confirms the DM's prior choice. She thus always prefers to manage the workload by completing the current task based on her stronger updated belief, rather than by releasing an accumulated task based on a weaker prior. This situation corresponds exactly to the last phase of a search in the first regime, and hence the maximum workload decreases as the search progresses.

These underlying effects are in fact quite general. To show this, we test the robustness of our findings by extending our main model in many different directions. We consider tasks with heterogeneous prior probabilities or with a set-up cost, the possibility of delegating tasks (generating a fixed releasing value), observations that correspond to noisy signals and tasks with no decision (i.e. for which early termination is always treated as a failure). We fully characterize the structure of the optimal policy for each of these extensions, and show that our main insights hold in these various set-ups. In addition, we explore numerically the possibility of collecting observations for multiple tasks in parallel by switching among them. This study indicates that our main insights naturally generalize to this set-up as well.

The paper is organized as follows. Section 2 provides a literature review. Section 3 introduces the model, and, in particular, the two possible regimes corresponding to falsifying and confirming decision tasks, respectively. Section 4 analyzes the case where tasks do not accumulate (which also corresponds to a discrete version of McCardle et al. 2017). We then present our main findings in Section 5 and test the robustness of our insights in Section 6. Section 7 concludes the paper.

2. Literature Review

Arrow et al. (1949) is the first paper to propose the fundamental sequential search problem, which has been referred to and studied in classical textbooks on Bayesian statistics, sequential decision making and dynamic programming (Savage 1972; DeGroot 1970; Bertsekas 2005). While Arrow et al. (1949) "assume that the cost of experimentation depends only on the number [...] of observations taken," we assume that this cost also depends on the dynamic accumulation of tasks in the system. That is, instead of considering a single decision task as they do, we consider a random stream of decision tasks. This, in effect, makes the information search cost endogenous.

Specifically, each task that the DM needs to complete in our model is inspired by the search problem introduced in McCardle et al. (2017), a paper closely related to ours. McCardle et al. (2017) studies optimal stopping decisions when working on a project. The DM pursues the project until it either succeeds, fails, or is abandoned. In their setting, the DM updates her belief about the success/failure rates in a way that is similar to our Bayesian updating process. In fact, if we scale up the frequency of observations in our discrete set-up, while scaling down their accuracy, our information gathering process converges in the limit to a basic version of their search model.

In McCardle et al. (2017), however, the DM enjoys the benefit of a success only if she receives a conclusive positive signal. Otherwise, early termination is always treated as a failure. In other words, a task does not involve any choice in that set-up. By contrast, we consider situations in which the DM can stop the current search and still proceed with an option, even without being certain of the correctness of this choice. In this case, the DM enjoys an expected payoff, which accounts for the probability of making a correct choice. As a result, the early termination cost depends on the DM's current belief, and, thus, the search length, while this cost remains constant in McCardle et al. (2017). Nonetheless, we treat the later case in an extension of our main model in Section 6.5.

McCardle et al. (2017) provide closed form solutions for the maximum time the DM should work on the project before abandoning it, which they leverage to conduct different sensitivity analyses with respect to model parameters. Our focus, however, is different, as we consider the effect of accumulation when working on a project. Thus, we need to track the current workload faced by the DM (in addition to the progress of the current task), yielding a two dimensional state space. The decision set is also richer in our set-up, as the DM can release some accumulated tasks, without search. The price to pay for this additional complexity, however, is that we do not obtain closed form solutions as McCardle et al. (2017) do.

Our analysis further extends the work of Alizamir et al. (2013), which also studies a task accumulation problem. In that set-up, however, the DM cannot release a task without first working on it. One of the main results is that the DM may sometimes make a choice that goes *against* the information that has been collected thus far. This effect introduces undesirable inefficiencies, as the DM sometimes needs to ignore costly information. By contrast, we show that allowing the DM to release an accumulated task without search eliminates these inefficiencies. More generally, we explore when the DM should release a task, without search, based on her prior belief. This yields a different control policy structure, which is highly non-monotonic.

Finally, our work is related to the stream of research which studies the general problem of balancing congestion against generated values (see, in particular, Hopp et al. 2007; Bouns 2003; George and Harrison 2001). In this framework, the value provided to a customer is captured by an increasing function of the service time. This literature finds that, at optimality, the service time decreases with the workload. This intuitive result stems from the decreasing marginal value of the service time. We retrieve an equivalent structure, in a special setting where an inconclusive search confirms the DM's prior belief. By contrast, in a setting where the search tends to falsify this belief, we find that the maximum level of workload can actually increase as the search progresses.

3. Model Formulation

We consider a decision maker (DM) who faces tasks that arrive randomly over time, according to a Poisson process with rate λ, and are accumulated in a queue until they are processed. The

DM can process only one task at a time. That is, the DM needs to terminate the task before starting to work on a new one from the queue. (Section 6.4 explores the case where the DM can process multiple tasks by switching back and forth among them). For each task, the DM needs to decide whether option s or its complement, $\bar{s}$, is the better one. To make this choice, the DM collects a sequence of observations, which may be positive (+), negative (−), or inconclusive (∼). A positive (resp. negative) observation correctly reveals that the better option is s (resp. $\bar{s}$). An inconclusive observation does not reveal the true nature of the task, but changes the DM's belief about the better option through Bayesian updating. Observations are independent from each other conditioning on truth. The time required to collect an observation (or a sample) is exponentially distributed with rate μ, and the process is preemptive so that the information gathering process can be stopped at any time.

3.1. Decision Tasks

Probability p_0 denotes the DM's prior that option s is better for any given task, which the DM updates as she collects observations. Specifically, given that the better option for a task is s, the observation reveals it with probability α. Similarly, the observation reveals if the better option is $\bar{s}$ with probability β. Thus, we have conditional probabilities $P(+|s) = \alpha$, $P(\sim|s) = 1-\alpha$, $P(-|\bar{s}) = \beta$, and $P(\sim|\bar{s}) = 1-\beta$. We denote p_k to represent the DM's subjective belief that option s is better, after k inconclusive observations. Probability p_k further determines the DM's subjective belief that the next observation will be conclusive (and hence will not return ∼), denoted as c_k, which is equal to $c_k = \alpha p_k + \beta(1-p_k)$. If the next observation is inconclusive, Bayes' rule implies that

$$p_{k+1} = \frac{p_k(1-\alpha)}{p_k(1-\alpha)+(1-\beta)(1-p_k)}.$$

In particular, if $\alpha = \beta$, an inclusive observation never changes the DM's subjective probability, i.e., $p_k = p_0$ for all k. We solve this simple case at the end of this section. The rest of the paper focuses on the case with $\alpha \neq \beta$, and we assume that $\alpha < \beta$ without loss of generality.

For each task, the DM needs to decide which option is better. Correct decisions generate values and incorrect ones incur costs. In particular, correctly choosing option s (resp. $\bar{s}$) for a given task generates a value v (resp. $\bar{v}$), whereas wrongly choosing option s (resp. $\bar{s}$) incurs a cost c (resp. $\bar{c}$). Moreover, a delay penalty $w(n)$ is incurred per unit time when there are n tasks accumulated in the workload. Assume that $w(n)$ is non-decreasing convex and approaches infinity with n, which accounts for the special case of linear cost functions. Note that while $w(\cdot)$ is an exogenous cost function, workload level n, and hence penalty $w(n)$, are both endogenous.

If the DM stops and commits to a final decision when the search is inconclusive after k observations, her expected reward becomes

$$r_k \equiv p_k v - (1-p_k)\bar{c}, \text{ and } \bar{r}_k \equiv (1-p_k)\bar{v} - p_k c, \tag{1}$$

for choosing option s and $\bar{s}$, respectively. In this case, the optimal expected reward is equal to $r_k \vee \bar{r}_k := \max\{r_k, \bar{r}_k\}$. It follows immediately that option s is preferable over option $\bar{s}$ if and only if p_k exceeds the critical fraction

$$\theta := \frac{\bar{v}+\bar{c}}{v+c+\bar{v}+\bar{c}} \ .$$

Fraction θ is the indifference level corresponding to the DM's belief at which options s and $\bar{s}$ yield the same expected value, i.e., $\theta v - (1-\theta)\bar{c} = (1-\theta)\bar{v} - \theta c$.

The following lemma, which is immediate, determines how the DM's beliefs and expected rewards evolve as the search progresses.

LEMMA 1. *Probability p_k increases and probability c_k decreases in k. Further, reward r_k increases and reward $\bar{r}_k$ decreases in k.*

Lemma 1 reveals two essential effects that drive our main results, as discussed in the introduction. The first one corresponds to the *decision's ambivalence*. Assuming $p_0 < \theta$, the DM prefers option $\bar{s}$ first. As the search progresses without revealing the better option, however, probability p_k increases and approaches the critical fraction θ from below. As a result, the difference between rewards r_k and $\bar{r}_k$ shrinks and the preference between the two options becomes increasingly ambivalent. If the search continues and remains inconclusive, probability p_k crosses θ and the DM's preference shifts to option s. Consequently, the probability p_k increases away from the critical fraction θ, the difference in rewards increases, and the decision problem becomes less and less ambivalent.

The second effect is related to the *search's inconclusiveness*. Indeed, the lemma states that probability c_k decreases in k. This also means that the longer the search does not reveal the better option, the more likely the next observation, and hence the search process, will remain inconclusive. This effect also holds in the model of McCardle et al. (2017) in which, conditional on the success rate being higher than the failure rate, the likelihood of the project's success decreases with the working time.

Hence, Lemma 1, and the first effect in particular, suggests two different regimes depending on whether or not $p_0 < \theta$. As discussed, if $p_0 < \theta$, a search starts by increasing the ambivalence of the decision problem. The DM believes option $\bar{s}$ is better a priori but inconclusive observations tend to falsify this belief. Therefore, in this regime, we refer to the decision tasks as *falsifying* search problems. By contrast, if $p_0 \geq \theta$, the DM prefers option s from the start. Since probability p_k keeps increasing above θ, inconclusive observations tend to confirm the DM's prior choice. We refer to these decision tasks as *confirming* search problems in this regime.

Finally, note that condition $p_0 < \theta$ can be expressed in terms of number of observations. Define the critical number k_θ as $k_\theta := \min\{k : p_k \geq \theta\}$, which is the number of inconclusive observations required for the DM to prefer s. We have $k_\theta > 0$ if and only if $p_0 < \theta$.

3.2. The DM's problem

The DM must determine, at any point in time, the best action among the following three alternatives: (i) terminate the current search and decide between options s and $\bar{s}$, (ii) release without search some accumulated tasks and decide between options s and $\bar{s}$ for each one of them, and (iii) continue the search to acquire an additional observation on the current task. Note that the DM decides between options s and $\bar{s}$ based on updated belief p_k for alternative (i) but based on her prior p_0 for alternative (ii). The objective is to minimize the the total discounted profit, which includes rewards from correct identifications, penalties for misidentifications, and delay costs. The DM's discount rate is γ.

We formulate this continuous time problem as a semi-Markov Decision Process, where decisions are made at any point in time when there is an arrival or a search returns a signal.[1] The state of the system is given by $(n, k) \in \mathbb{S} := \{1, 2, \ldots\} \times \{0, 1, \ldots\} \cup (0, 0)$, such that n is the number of tasks awaiting identification and k the number of inconclusive observations collected thus far for the task in process. (In particular, the number of observations can be positive only if the system is non-empty). Following standard uniformization, and without less of generality, we assume that $\lambda + \mu + \gamma = 1$. The corresponding Bellman equations are,

$$J(n,k) = \max\Big\{ \max\big\{r_k \vee \bar{r}_k + J(n-1,0),\ r_0 \vee \bar{r}_0 + J(n-1,k)\big\},\ h_J(n,k)\Big\}, \text{ for } n \geq 1, k, \quad (2)$$

$$J(1,k) = \max\Big\{r_k \vee \bar{r}_k + J(0,0),\ h_J(1,k)\Big\}, \quad (3)$$

$$J(0,0) = \lambda J(1,0)/(\lambda + \gamma), \quad (4)$$

where the value for collecting an additional observation, $h_J(n,k)$, is equal to

$$h_J(n,k) = -w(n) + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v}\right] + \lambda J(n+1,k) + \mu c_k J(n-1,0) + \mu(1-c_k)\, J\big(n, k+1\big). \quad (5)$$

The first maximization problem in (2) determines whether to decrease the workload or to collect an additional observation (alternative (iii)). In the former case, the second maximization problem

[1] Although we do allow decisions to occur at any point in time, the memoryless property of Poisson process ensures that restricting decision epochs to arrival and signal times is without loss of generality.

further determines whether to terminate the current search (alternative (i)) or release an accumulated task (alternative (ii)). Maxizimations $r_k \vee \bar{r}_k$ and $r_0 \vee \bar{r}_0$ further indicate which option to choose.

We conclude this section by considering the simple case where $\alpha = \beta$, so that $\Pr(+|s) = \Pr(-|\bar{s}) = \alpha$. Recall that in this case, the DM's belief about the better option remains the same after each inconclusive observation, and thus $p_k = p_0$ for all k. As a result, the state space in the Bellman equations (2)-(5) collapse to a single dimension n. The corresponding optimal policy is of threshold type, as shown in the following proposition.

PROPOSITION 1. *Suppose $\alpha = \beta$. A threshold $\hat{n} \geq 0$ exists such that if $n < \hat{n}$, the DM elicits additional information for the current task. Otherwise, the DM releases a task from the system and decide between the two options based on prior p_0.*

Note that when $\alpha = \beta$, releasing an accumulated task and aborting the current search (alternatives (i) and (ii)) are equivalent, since the expected payoff is $r_0 \vee \bar{r}_0$ in both cases. By contrast, we assume in the remaining of the paper that $\alpha < \beta$ (w.l.o.g.), for which both alternatives are not equivalent.

4. Optimal search with no task accumulation

It is worth noting that a decision task in our model essentially corresponds to a discrete and elementary version of the search problem studied by McCardle et al. (2017).[2] That is, the DM works on a task until either the better option is revealed, or she voluntarily aborts the search and decides based on her updated belief. No additional tasks are accumulated in the meantime. But once the DM completes a task, it takes an exponentially distributed time for her to receive a new one. This set-up corresponds to our main model with a queuing capacity of one and shares essential features with the model in Section 4 of McCardle et al. (2017). Next, we study this benchmark, which helps identify the effects driving the main results of Section 5.[3]

Without accumulation, the search cost rate per unit of time is a constant, which equals to $w = w(1)$, since there is at most one task in the system. As such, the state space is a single dimensional value k, which is the number of inconclusive observations received so far. The corresponding Bellman equation becomes

$$J_s(k) = \max\Big\{ -w + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v} + (1-c_k)J_s\big(k+1\big) + c_k J_s(0)\right] + \lambda J_s(k),$$
$$r_k \vee \bar{r}_k + \lambda J(0)/(\lambda+\gamma)\Big\}. \tag{6}$$

[2] To see this, increase the search rate μ while scaling probabilities α and β accordingly. In the limit as μ approaches infinity, it can be shown that the (random) time it takes for a search to reveal the better option and stop follows exactly the distribution of the completion time in McCardle et al. (2017) (assuming their minimum time Δ is zero).

[3] Note that if arrival rate λ is high enough in a model with accumulation, keeping tasks in the queue becomes suboptimal and n is always less or equal than one. Indeed, in this case, holding a waiting task is unnecessary as a new one is almost guaranteed to arrive soon after the current task is completed. In effect, the problem introduced in Section 3.2 becomes a search problem without accumulation, which we study in this section.

The first term in the maximization corresponds to continuing the search by collecting one more observation, whereas the second term is the expected payoff if the DM aborts the search. In this case, the DM enjoys the optimal reward as well as future gains from the next task. As a special case, when $\lambda = 0$ there are no future tasks and the situation reduces to a classic single search problem (Bertsekas 2005, Example 5.4.1).

The next proposition characterizes the optimal policy for the problem without task accumulation.

PROPOSITION 2. *Consider the search problem without task accumulation given by optimality equation* (6)*. The optimal policy is either to i) decide option $\bar{s}$ for each task without any search, or ii) choose option s if the search has not revealed the better option after $\bar{k}$ observations, where threshold $\bar{k} \geq k_\theta$ uniquely exists.*

Hence without task accumulation, the optimal policy is to either (i) never search or (ii) always search but stop after a given number of inconclusive observations. Whether case (i) or (ii) is optimal depends on model parameters. In contrast, as the next section makes clear, the search cost $w(n)$, and the choice between deciding a priori versus based on an updated belief, are endogenously determined when tasks accumulate.

This optimal structure is driven by the effects stemming from the decision's ambivalence and search's inconclusiveness (see Section 3) in the falsifying task case ($p_0 < \theta$). Indeed, consider case (ii) mentioned in the last paragraph, for which starting the search rather than choosing option $\bar{s}$ upfront is optimal. Initially ($k < k_\theta$ and $p_k < \theta$), the search becomes increasingly more likely to remain inconclusive, which strengthens the DM's incentive to abort the search (search's inconclusiveness). At the same time, the decision problem becomes increasingly more ambivalent between the two options, which strengthens the DM's incentive to continue the search (decision's ambivalence). The DM does not stop the search yet, since the expected gain of the preferred option $\bar{s}$ is worse than before the search even started (otherwise case (i), i.e. not searching in the first place, would have been optimal).

Hence, the DM endures at least k_θ inconclusive observations before voluntarily aborting the search. After that ($k \geq k_\theta$), her preference switches and becomes less and less ambivalent towards option s. The incentives from both effects are then aligned, and the DM has increasingly weaker incentive to continue the search. Eventually the DM aborts the search when the marginal benefit of collecting an additional observation becomes lower than the associated search cost (i.e. when k reaches $\bar{k}$).

As we show next, these incentives are present and follow similar patterns for the problem with task accumulation. However, the DM can modulate these effects by directly regulating the workload, and hence the search costs, in this case.

5. Optimal search under accumulated pressure

In this section, we study the problem when unattended decision tasks build up overtime, creating time pressure in the form of accumulation. This changes the structure of the search problem of Section 4 in two fundamental ways. First, the cost of acquiring more information, $w(n)$, becomes endogeneous. Second, the DM can sometimes alleviate this search cost by directly reducing the workload. Indeed, when new decision tasks have accumulated, the DM can always release some of them immediately by deciding a priori. These additional features modulate the incentives of the DM uncovered in Section 4.

In the following, we characterize the resulting optimal policy, which constitutes our main result. We first establish the existence of an optimal value function.

LEMMA 2. *A function $J^*(\cdot,\cdot):\mathbb{S}\mapsto\mathbb{R}$ exists such that $J^*(\cdot,\cdot)$ satisfies the Bellman equations (2)-(4) and $J^*(n,k)$ is the optimal value function given initial state $(n,k)\in\mathbb{S}$.*

Several key properties of $J^*(\cdot,\cdot)$ shed lights on the structure of the corresponding optimal control policy. First, the marginal benefit of working on the current task (i.e., $J^*(n,k)-J^*(n-1,0)$) diminishes when the workload increases, as we show next. (Unless specified otherwise, all the proofs are in the appendix.)

LEMMA 3. *For any fixed k, $J^*(n,k)-J^*(n-1,0)$ is non-increasing in n.*

From the Bellman equation we know that $J^*(n,k)-J^*(n-1,0)\geq r_k\vee\bar{r}_k$ for any state (n,k), where the inequality holds as an equality if aborting the search is optimal. Lemma 3, therefore, implies that if it is optimal to stop searching after k inconclusive observations given a workload of n tasks (i.e., $J^*(n,k)-J^*(n-1,k)=r_k\vee\bar{r}_k$), it is also optimal to stop the search for higher levels of workload.

A similar result holds for the number of observations given a fixed workload.

LEMMA 4. *For any fixed n, $J^*(n,k)-r_k$ is non-increasing in k, and $J^*(n,k)-\bar{r}_k$ non-decreasing in k.*

Lemma 4 implies that if it is optimal to release the current task and choose option s after k observations (i.e., $J^*(n,k)-r_k=J^*(n-1,0)$), then it must also be optimal to do so with more observations under the same workload (i.e. in states (n,k') with $k'>k$). By contrast, if it is optimal to release the current task and choose option $\bar{s}$ after k observations (i.e., $J^*(n,k)-\bar{r}_k=J^*(n-1,0)$), the same must be true with *fewer* observations (i.e. in states (n,k') with $k'<k$).

This further implies that the DM should never abort an inconclusive search to choose option $\bar{s}$. Otherwise, the DM would always be better off choosing this option upfront without starting the search in the first place. (Alternatively, the lemma implies that $\bar{r}_k+J^*(n-1,0)\leq\bar{r}_0+J^*(n-1,k)\leq$

$r_0 \vee \bar{r}_0 + J^*(n-1,k)$, or, releasing an accumulated task without search dominates stopping to choose option $\bar{s}$.) This generalizes the corresponding property identified in Section 4 to the accumulation case.

If the DM never aborts a search to choose $\bar{s}$, however, she can still opt for this option when releasing an accumulated task directly from the queue if $p_0 < \theta$. In fact, the marginal benefit of doing so without a search (i.e., $J^*(n,k) - J^*(n+1,k)$) increases with the workload, as stated below.

LEMMA 5. *For any fixed k, the optimal value function $J^*(n,k)$ is concave in n with*

$$J^*(n+1,k) + J^*(n-1,k) \leq 2J^*(n,k)\,.$$

Lemma 5 implies that if it is optimal to release an accumulated task without search given a workload of n tasks (i.e., $J^*(n,k) - J^*(n-1,k) = r_0 \vee \bar{r}_0$), then it is also optimal to do so for higher levels of workload. (Indeed, we have $J^*(n+1,k) - J^*(n,k) \geq r_0 \vee \bar{r}_0$ from the Bellman equation, but $J^*(n+1,k) - J^*(n,k) \leq J^*(n,k) - J^*(n-1,k) = r_0 \vee \bar{r}_0$ from Lemma 5, and, therefore, $J^*(n+1,k) - J^*(n,k) = r_0 \vee \bar{r}_0$.)

Taken together, Lemmas 3 and 5 futher imply that the optimal decision to elicit an additional observation is of threshold type as the next proposition establishes.

PROPOSITION 3. *Threshold $\hat{n}(k)$ exists such that $J^*(n,k) = h_{J^*}(n,k)$ if and only if $n < \hat{n}(k)$.*

Intuitively, Lemma 3 implies that if it is optimal to stop searching a focal task in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Similarly, Lemma 5 implies that if it is optimal to dismiss a task from the queue in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Thus, if it is optimal to continue searching in state (n,k), the same decision must also be optimal in state $(n-1,k)$, which implies the threshold $\hat{n}(k)$.

The previous properties of the optimal value function also allow us to establish next another crucial result for our main finding. In essence, this result states that the DM should reduce the workload by releasing a task from the queue as soon as the number of inconclusive observations collected thus far belongs to an interval. Further, this interval shrinks as the workload diminishes.

More specifically, define set $\mathcal{K}(n)$ as,

$$\mathcal{K}(n) = \left\{ k \geq 0 : \; J^*(n,k) > J^*(n-1,0) + r_k \vee \bar{r}_k \text{ and } J^*(n,k) > h_{J^*}(n,k)/(\lambda+\mu+r) \right\}.$$

In other words, if state (n,k) is such that $k \in \mathcal{K}(n)$, the DM releases a task without search from the queue under the optimal policy.

We next show, by induction on the workload level n, that $\mathcal{K}(n)$ is a connected set, which shrinks as n decreases.

Proposition 4. *Two thresholds $\check{k}(n)$ and $\hat{k}(n)$ exist such that $k \in \mathcal{K}(n)$ if and only if $\check{k}(n) < k < \hat{k}(n)$. Furthermore, $\check{k}(n)$ is non-increasing and $\hat{k}(n)$ non-decreasing in n.*

Hence, given the workload level n, it is optimal to reduce this workload by directly releasing tasks from the queue if and only if $\check{k}(n) < k < \hat{k}(n)$. Further, the set $\mathcal{K}(n)$ shrinks as n decreases, and the DM relies less and less on completing tasks without search to regulate the workload. This property provides the key structure of the optimal policy, which we are now ready to fully characterize.

Theorem 1. *A threshold $\hat{n}(k) \geq 0$ for any $k \geq 0$, and two thresholds $\bar{k} \geq \underline{k} \geq 0$ with $\bar{k} \geq k_\theta$ uniquely exist, such that in state (n,k),*

- *if $n < \hat{n}(k)$, the DM elicits additional information for the current task;*
- *if $n \geq \hat{n}(k)$ and $k < \bar{k}$, the DM releases a waiting task and chooses option $\bar{s}$ without search,*
- *if $n \geq \hat{n}(k)$ and $k \geq \bar{k}$, the DM aborts the current search and chooses option s.*

Further, threshold $\hat{n}(k)$ is monotonically non-increasing in k for $k < \underline{k}$ or $k \geq \bar{k}$, but monotonically non-decreasing for $\underline{k} \leq k < \bar{k}$.

Proof: Defined $\bar{k}$ as the smallest value of k for which releasing the task in search and choosing option s is optimal at any workload level n. Proposition 3 ensures that for any $n \geq \hat{n}(k)$, the optimality between releasing a waiting task and aborting the current search is determined by the same threshold $\bar{k}$. Further, $\bar{k} \geq k_\theta$ follows from the fact that $r_{\bar{k}}$ needs to be no less than $\bar{r}_{\bar{k}}$ for all $k < k_\theta$.

Furthermore, Lemma 4 implies stopping to choose option $\bar{s}$ is never optimal. Therefore, for $k < \bar{k}$ and $n \geq \hat{n}(k)$, dismissing a task from the queue is the only decision that can be optimal.

Finally, the existence of threshold $\underline{k}$ immediately follows from Proposition 4, which also implies that $\hat{n}(k)$ is non-increasing for $k < \underline{k}$, and non-decreasing for $\underline{k} \leq k < \bar{k}$. The monotonicity of $\hat{n}(k)$ for $k \geq \bar{k}$ follows from Lemma 4, because if it is optimal to release the current task with option s when the system is at state (n,k), the same decision must also be optimal in state $(n, k+1)$.

■

Note that according to Lemma 3 (resp. Lemma 5), if it is optimal to terminate the search on the current task (resp. release an accumulated task from the queue) when the system is in state (n,k), the same decision must also be optimal when the system is more congested (i.e., is in state $(n+1,k)$). As such, it is never optimal to let the number of tasks in the system exceed $\hat{n}(k)$ when k inconclusive observations are collected on the current task. Threshold $\hat{n}(k)$ thus indicates the maximum workload the DM is ready to sustain, given the stage of the current search. Threshold $\bar{k}$ determines how the DM regulates this workload: by releasing an accumulated task or by completing the current search. Theorem 1, therefore, indicates that the DM needs to alleviate the accumulation pressure very differently depending on the stage of a search. The maximum number of accumulated

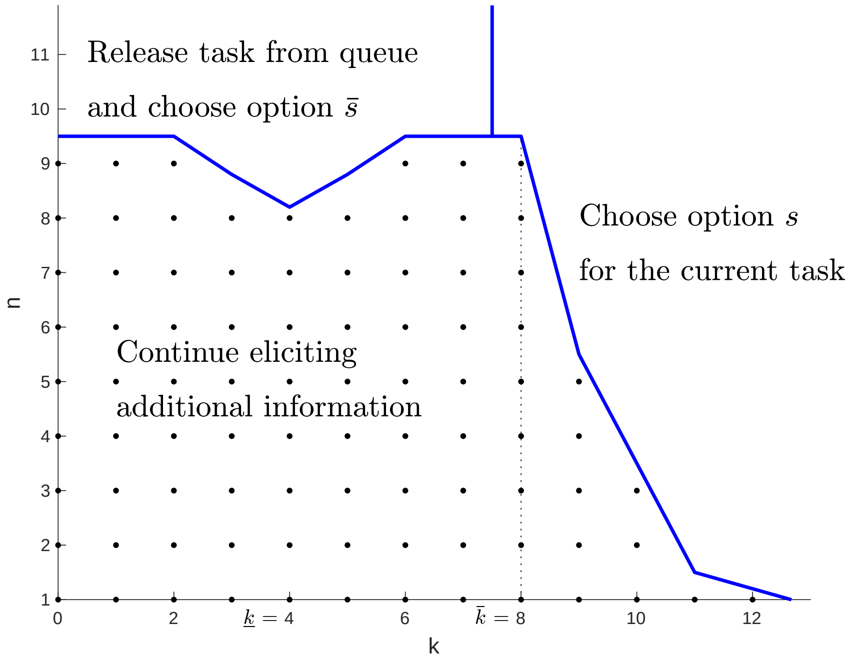

Figure 1 Illustration of the Optimal Policy for $p_0 < \theta$.

tasks the DM allows (weakly) decreases as the search progresses in both the beginning and end of the search, but actually (weakly) *increases* in the middle of it. She does this first by making a priori decisions to release some accumulated tasks, and later by aborting the current search and deciding based on her updated belief.

The exact structure of the optimal policy depends on the DM's prior belief about the better option. In fact, the highly non-monotonic property of $\hat{n}(k)$ only occurs when the search tends to fasify the DM's prior belief ($p_0 < \theta$). When the search tend to confirm the DM's prior belief, however, the maximum number of accumulated tasks always decreases as the search progresses. We formalize this result below.

Corollary 1. *We have $\bar{k} = 0$ if and only if $p_0 \geq \theta$. In particular, threshold $\hat{n}(k)$ is non-increasing in k for all k if $p_0 \geq \theta$ and releasing a task from the queue without search is never (strictly) optimal.*

Recall that for confirming search tasks ($p_0 \geq \theta$), the DM prefers option s a priori and becomes increasingly confident about this choice as the search progresses. In other words, the decision problem never becomes more ambivalent with inconclusive observations. By the same token, therefore, Corollary 1 reveals that the decision ambivalence effect mentioned in Section 3 is the main driver for the non-monotonicity of threshold $\hat{n}(k)$ in Theorem 1. In particular, the effect explains why the DM sometimes allows the workload to increase in the middle of a search. And this effect only exists for falsifying tasks ($p_0 < \theta$).

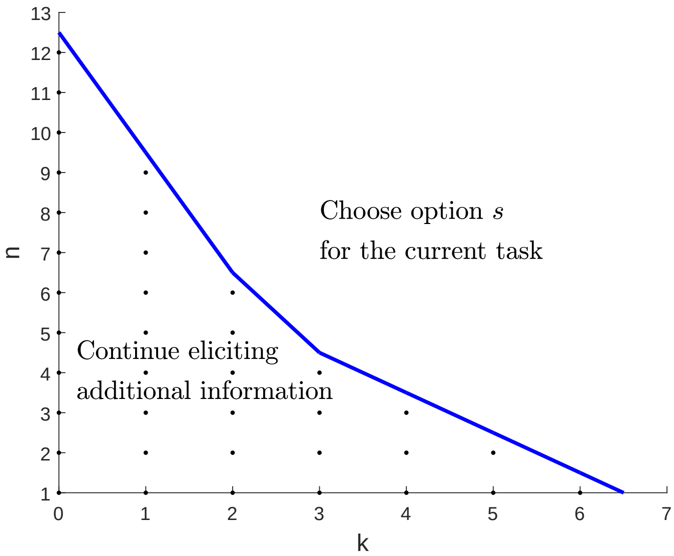

Figure 2 Illustration of the Optimal Policy for $p_0 > \theta$.

Figure 1 depicts the optimal policy for a case with $p_0 < \theta$,[4] where a search tends to falsify the DM's prior belief that option $\bar{s}$ is better for any new task. The two dimensional state space is divided into three regions, each indicating which action is optimal. The dots mark the states where the optimal decision is to continue searching (or, the reccurent states, except state $(0,0)$). As indicated by Theorem 1, the frontier that delimits this region is first non-increasing, then non-decreasing and finally non-decreasing again in k.[5]

By contrast, Figure 2 depicts a case where $p_0 > \theta$.[6] In this context, a search tends to confirm the DM's prior belief. As stated in Corollary 1, the DM never releases a task from the queue, but instead may abort the current search. The state space is divided into two regions in this case, and the frontier that delimits the region where the DM pursues the search is now always non-increasing.

6. Extensions and Robustness

In this section we explore the robustness of the finding and insights stemming from Theorem 1. To do this, we extend our main model in several directions. In Section 6.1, we first consider tasks with heterogeneous prior probabilities and the possibility of delegating tasks (generating a fixed releasing value). Section 6.2 analyzes the impact of incurring a set-up cost when starting a new task, while 6.3 considers a case where observation takes the form of imperfect and noisy signals.

[4] In this example, $p_0 = 0.09$, $v = 32$, $\bar{v} = 15$, $c = 10$, $\bar{c} = 14$, $\alpha = 0.1$, $\beta = 0.48$, $\lambda = 1$, $\mu = 20$, $\gamma = 0.05$, and $w(n) = 2.25 \times n$. In this case $\theta = 0.4 > p_0$, $k_\theta = 4$, and $\bar{k} = 8$.

[5] In Figure 1, the maximum of $\hat{n}(k)$ when $k < \bar{k}$ happens to be equal to the maximum of $\hat{n}(k)$ when $k > \bar{k}$. This is not necessary the case for other sets of parameters.

[6] In this example, $p_0 = 0.5$, $v = 30$, $\bar{v} = 15$, $c = 10$, $\bar{c} = 20$, $\alpha = 0.2$, $\beta = 0.5$, $\lambda = 1$, $\mu = 20$, $\gamma = 0.05$, and $w(n) = 12 \times n$. In this case $\theta = 0.46 < p_0$, and $k_\theta = \bar{k} = 0$.

The possibility of collecting observations for multiple tasks in parallel, by switching among them, is explored in Section 6.4. Finally, Section 6.5 studies a set-up where tasks do not involved a decision, in the sense that early termination always corresponds to a failure.

6.1. Decision tasks with heterogeneous priors and releasing value

In this section, we generalize the model in two ways. First, we assume that all tasks appear homogenous ex-ante, but have heterogenous priors that the DM can quickly learn. This introduces some heterogeneity among tasks. Second, we consider a general value associated to releasing without search an accumulated task. This, for instance, enables us to account for the possibility of delegating an accumulated task to others.

Specifically, we assume that the prior probability of a task follows cumulative probability distribution F. The DM learns the exact value of p_0 just before collecting the first observation on the task. Learning p_0 is very fast compared to collecting an observation, and we assume for simplicity that this step is instantaneous. Because prior p_0 is now heterogenous, the number of inconclusive signals k is not enough to represent the state of the system anymore. Instead, over time the state space needs to directly track the DM's posterior probability, which we refer to as p.

In addition, we assume that the value for releasing a decision task from the queue is a constant R. In our base model, we assumed that $R = R_0 \equiv r_0 \vee \bar{r}_0$. In this section, we do not restrict the value of R, which can be any real number. Cases where $R \neq R_0$ may correspond to situations in health care, for instance, where accumulated diagnostic tasks (such as interpretation of medical imagery) are outsourced to better or worst physicians (with $R < R_0$ and $R > R_0$, respectively).

Given posterior probability p at any point in time, we define the values for committing to s and $\bar{s}$ as $r_p \equiv pv - (1-p)\bar{c}$ and $\bar{r}_p \equiv (1-p)\bar{v} - pc$, respectively. Further, we define the probability of receiving a conclusive signal as $c_p \equiv \alpha p + \beta(1-p)$, and the posterior probability of the better option being s following an inconclusive signal as $p_+ \equiv p(1-\alpha)/[p(1-\alpha) + (1-\beta)(1-p)]$.

The corresponding state space is $(n, p) \in \{1, 2, \ldots\} \times [0, 1] \cup (0, 0)$, and the Bellman equations become

$$J(n,p) = \max\Big\{\max\big\{r_p \vee \bar{r}_p + \mathbb{E}[J(n-1,p_0)],\ R + J(n-1,p)\big\},\ h_J^g(n,p)\Big\}, \quad \text{for } n > 1, \tag{7}$$

$$J(1,p) = \max\Big\{r_p \vee \bar{r}_p + \mathbb{E}[J(0,p_0)],\ \ h_J^g(1,p)\Big\}, \tag{8}$$

$$J(0,p) = \lambda \mathbb{E}[J(1,p_0)]/(\lambda+\gamma), \tag{9}$$

where the value for collecting an additional observation, $h_J^g(n,k)$, is equal to

$$h_J^g(n,p) = -\,w(n) + \mu\,[\alpha p v + \beta(1-p)\bar{v}] + \lambda J(n+1,p) + \mu c_p \mathbb{E}[J(n-1,p_0)] + \mu\,(1-c_p)\,J\big(n,p_+\big). \tag{10}$$

The following result characterizes the optimal policy of problem (7)-(10).

THEOREM 2. *A threshold $\hat{n}(p) \geq 0$ for any $p \in [0,1]$, and two thresholds $\bar{p} \geq \underline{p}$ with $\bar{p} \geq \theta$ uniquely exist, such that in state (n,p),*

- *if $n < \hat{n}(p)$, the DM elicits additional information for the current task;*
- *if $n \geq \hat{n}(p)$ and $p < \bar{p}$, the DM releases a waiting task and chooses option $\bar{s}$ without search,*
- *if $n \geq \hat{n}(p)$ and $p \geq \bar{p}$, the DM aborts the current search and chooses option s.*

Further, threshold $\hat{n}(p)$ is monotonically non-increasing in p for $p < \underline{p}$ or $p \geq \bar{p}$, but monotonically non-decreasing for $\underline{p} \leq p < \bar{p}$.

In short, Theorem 2 shows that the optimal policy for this generalized model has a structure similar to the one described in Theorem 1, where the non-monotonicity of $\hat{n}(p)$ in p replaces the non-monotonicity $\hat{n}(k)$ in k.

6.2. Setup Cost

In this section, we analyze a generalization of our base model in which the DM incurs a fixed setup cost when initializing the search process on a new task. In particular, there is a cost $\kappa \leq \min\{r, \bar{r}\}$ when starting the search on a task. Therefore, we need to distinguish between when search has started for a task or not yet. Since we use $k \geq 0$ to represent the number of (inconclusive) searches that has been conducted on a task, we extend the state space and use $k = -1$ to capture states in which no search has be started on any task in the system. The corresponding Bellman equations are

$$J(n,-1) = \max\Big\{r_0 \vee \bar{r}_0 + J(n-1,-1),\ -\kappa + J(n,0)\Big\}, \tag{11}$$

$$J(n,k) = \max\Big\{\max\big\{r_k \vee \bar{r}_k + J(n-1,-1),\ r_0 \vee \bar{r}_0 + J(n-1,k)\big\},\ h_J^f(n,k)\Big\}, \quad \text{for } n > 1, k \geq 0, \tag{12}$$

$$J(1,k) = \max\Big\{r_k \vee \bar{r}_k + J(0,-1),\ h_J^f(1,k)\Big\}, \tag{13}$$

$$J(0,-1) = \lambda J(1,-1)/(\lambda+\gamma), \tag{14}$$

in which h_J^f is defined as,

$$h_J^f(n,k) = -w(n) + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v}\right] + \lambda J(n+1,k) + \mu c_k J(n-1,-1) + \mu\left(1-c_k\right) J\big(n,k+1\big). \tag{15}$$

The maximization in (11) captures the decision on whether the DM should start searching a task or dropping it. It is worth noting that if $\kappa = 0$, the Bellman equations (11)-(15) reduce to (2)-(5). The following theorem demonstrates that the structure of the optimal control policy remains the same if we increase κ from zero to positive.

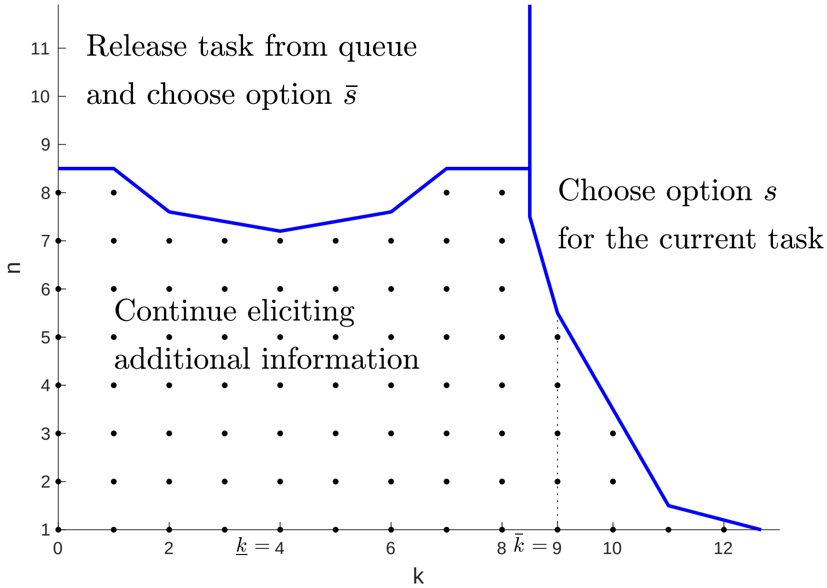

Figure 3 Illustration of the Optimal Policy for $p_0 < \theta$.

THEOREM 3. *At state* $(n, -1)$ *for any* $n > 0$*, it is optimal to start searching a task. For any* $k \geq 0$*, the optimal control policy is described by a threshold* $\hat{n}(k)$ *that has the same structure as described in Theorem 1.*

Figure 3 demonstrates the optimal policy structure, using the same model parameters as in Figure 1, except with $\kappa = 0.45$. As we can see from the figure, the structure of the optimal threshold remains the same as in Figure 1. In the presence of a set-up cost, however, releasing an accumulated task from the queue becomes more attractive to the DM, and hence, the number of states for which this decision is optimal increases. Correspondingly, as the value of κ increases from zero to a positive number, thresholds $\hat{n}(k)$ go down for all k and the threshold $\bar{k}$ shifts to the right compared with Figure 1.

6.3. Noisy Signals

In our base model, each observation is assumed to be either conclusive or inconclusive. In this section, we consider instead observations that return noisy signals indicating the true state of the world with uncertainty. In our set-up, the state of the world corresponds to the true type of a task.

Specifically, we assume that each task is of one of two types, s and $\bar{s}$. Each observation returns a signal that is either positive (+) or negative (−). A positive signal and a negative signal are more indicative that the true type of the task as s and $\bar{s}$, respectively, in the sense that $P(+|s) = P(-|\bar{s}) = \beta > 0.5$, and $P(-|s) = P(+|\bar{s}) = 1 - \beta < 0.5$. (In other words, for simplicity we consider identical and symmetric signals.)

In this set-up, the number of observations k is not sufficient anymore to describe the state of the system. Instead, we consider the *difference* between the numbers of positive and negative signals that have been collected thus far, which we denote as d. The larger the difference d, the more the DM believes that the task is of type s. We define p_d as the DM's subjective probability that the task is of type s, with $p_d = p_0$ at the beginning of a search process. As new signals arrive, p_d, is updated according to Bayes' Rule, such that

$$p_{d-1} = \frac{(1-\beta)p_d}{(1-\beta)p_d + \beta(1-p_d)} \;<\; p_d \;<\; p_{d+1} = \frac{\beta p_d}{\beta p_d + (1-\beta)(1-p_d)} \tag{16}$$

and thus probability p_d is increasing in d.

At any point in type, the DM can abort the search and choose either type s or $\bar{s}$. Similar to equation (1), the expected value from choosing between options s and $\bar{s}$ given difference d are $r_d \equiv p_d v - (1-p_d)\bar{c}$, and $\bar{r}_d \equiv (1-p_d)\bar{v} - p_d c$, respectively. Further, we define $d_\theta = \min\{d : p_d \leq \theta\}$, which corresponds to the threshold k_θ in our base model.

The Bellman equations for this problem are,

$$J(n,d) = \max\Big\{ \max\big\{ r_d \vee \bar{r}_d + J(n-1,0), r_0 \vee \bar{r}_0 + J(n-1,d)\big\}, h_J^d(n,d)\Big\}, \text{ for } n \geq 1, \tag{17}$$

$$J(1,d) = \max\Big\{ r_d \vee \bar{r}_d + J(0,0), \quad h_J^d(1,d)\Big\}, \tag{18}$$

$$J(0,0) = \lambda J(1,0)/(\lambda+\gamma), \tag{19}$$

where term h_J^d is defined as

$$\begin{aligned} h_J^d(n,d) = & -w(n) + \lambda J(n+1,d) + \mu\left[\beta p_d + (1-\beta)(1-p_d)\right] J(n,d+1) \\ & + \mu\left[(1-\beta)p_d + \beta(1-p_d)\right] J\big(n,d-1\big). \end{aligned} \tag{20}$$

The next result characterizes the optimal policy solving equations (17)-(20),

THEOREM 4. *A threshold $\hat{n}(d) \geq 0$, and three thresholds $\bar{d} \geq \check{d} \geq \underline{d}$, with $\bar{d} \geq d_\theta \geq \underline{d}$, uniquely exist, such that in state (n,d),*

- *if $n < \hat{n}(d)$, the DM elicits additional information for the current task;*
- *if $n \geq \hat{n}(d)$, on the other hand,*
  - *if $d \leq \underline{d}$, the DM aborts the current search and chooses type $\bar{s}$,*
  - *if $\underline{d} < d < \bar{d}$ the DM releases a waiting task and chooses type $\bar{s}$ without search,*
  - *if $d \geq \bar{d}$, the DM aborts the current search and chooses type s.*

Further, threshold $\hat{n}(d)$ is monotonically non-decreasing in d for $d \leq \underline{d}$ or $d \in [\check{d}, \bar{d})$, but monotonically non-increasing in d for $d \in (\underline{d}, \check{d}]$ or $d \geq \bar{d}$.

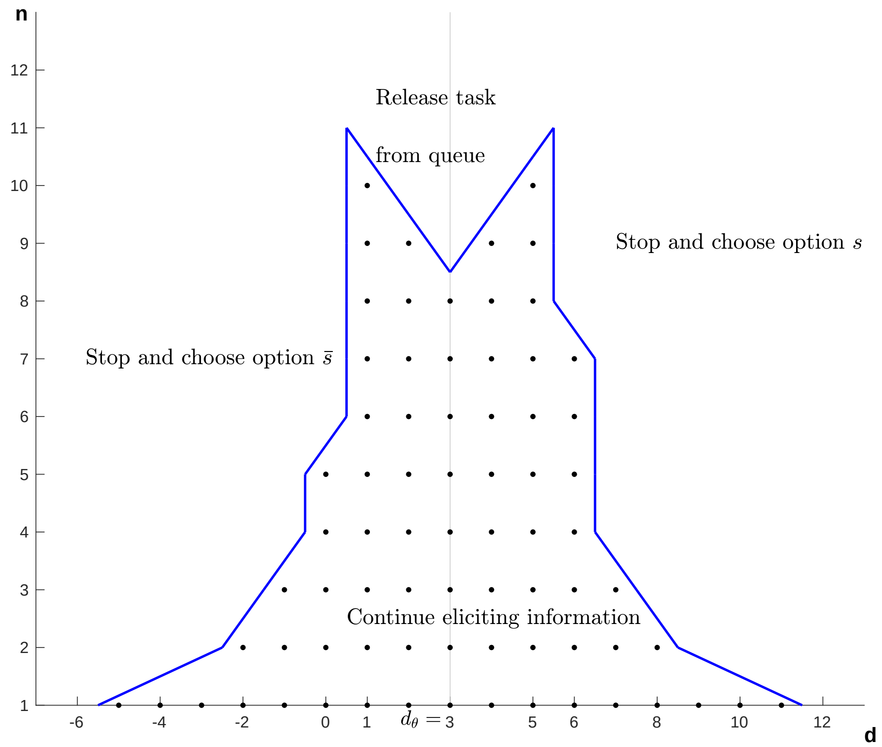

Figure 4 Illustration of the Optimal Policy for $p_0 > \theta$.

Figure 4 depicts the optimal control policy described in Theorem 4.[7] It is worth noting that similar to Figure 4, the threshold $\hat{n}(d)$ takes a downward dip in an interval in d when releasing a task from the queue is optimal for $n > \hat{n}(d)$.

6.4. Collecting observations on multiple tasks

In our base model, the search process consists in collecting observations on a single task. When switching tasks is feasible and does not incur high set-up times or costs, the DM may put a search on hold to work on another one. That is, the search process consists in collecting observations for multiple tasks in parallel by switching among them, while updating the corresponding posterior probabilities accordingly. The DM then needs to dynamically determine which task to search next.

The corresponding optimal sequencing of decision tasks is extremely hard to determine, especially when tasks can accumulate. The dynamic programming formulation requires a state space of infinite dimension because it needs to track the number of collected observations for each task on hold. Even if we restrict the number of tasks that can be put on hold, the structure of the optimal policy is in general too intricate to yield interesting insights.

[7] In this example, $p_0 = 0.25$, $v = 1600$, $\bar{v} = 2000$, $c = 0$, $\bar{c} = 0$, $\beta = 0.6$, $\lambda = 0.9$, $\mu = 20$, $\gamma = 0.012$, and $w(n) = 1 \times n$. In this case $d_\theta = 3$, $\underline{d} = -6$ and $\bar{d} = 12$.

Nonetheless, in this section we numerically explore the optimal policy in a set-up in which the DM can collect observations on at most two tasks in parallel (by switching between them), as opposed to only one task in the base case model. In this case, the state space is a three dimensional vector (n, k_1, k_2), in which n is the total number of decision tasks in the system and k_i the numbers of inconclusive searches that have been run thus far on task $i = 1, 2$. This enables us to visualize the optimal policy in a three-dimensional space.

When putting a task on hold is possible, the DM needs to choose at any point in time among the following alternatives: (i) terminate the search on task 1 and decide between options s and $\bar{s}$, (ii) terminate the search on task 2 and decide between options s and $\bar{s}$, (iii) release without search some accumulated tasks and decide between options s and $\bar{s}$ for each one of them (iv) continue the search to acquire an additional observation on task 1, (v) continue the search to acquire an additional observation on task 2. The corresponding Bellman equations (after uniformization) are[8]

$$\begin{aligned} J(n,k_1,k_2) &= \max\Big\{ \max\left\{ r_{k_1} \vee \bar{r}_{k_1} + J(n-1,0,k_2),\ r_{k_2} \vee \bar{r}_{k_2} + J(n-1,k_1,0)\right\},\ r_0 \vee \bar{r}_0 + J(n-1,k_1,k_2) \\ &\qquad h^1_J(n,k_1,k_2),\ h^2_J(n,k_1,k_2)\Big\},\ \text{for } n \geq 3, k_1, k_2, \quad (21) \\ J(2,k_1,k_2) &= \max\Big\{ \max\left\{ r_{k_1} \vee \bar{r}_{k_1} + J(1,0,k_2),\ r_{k_2} \vee \bar{r}_{k_2} + J(1,k_1,0)\right\}, \\ &\qquad h^1_J(2,k_1,k_2),\ h^2_J(2,k_1,k_2)\Big\},\ \text{for } k_1, k_2, \quad (22) \\ J(1,k,0) &= \max\Big\{ r_k \vee \bar{r}_k + J(0,0,0),\ h^1_J(1,k,0)\Big\}, \quad (23) \\ J(1,0,k) &= \max\Big\{ r_k \vee \bar{r}_k + J(0,0,0),\ h^2_J(1,0,k)\Big\}, \quad (24) \\ J(0,0,0) &= \lambda J(1,0,0)/(\lambda+\gamma), \quad (25) \end{aligned}$$

where the functions $h^1_J(n,k_1,k_2)$ and $h^2_J(n,k_1,k_2)$ are defined for $n \geq 2$ as

$$\begin{aligned} h^1_J(n,k_1,k_2) =& -w(n) + \mu\left[\alpha p_{k_1} v + \beta(1-p_{k_1})\bar{v}\right] + \lambda J(n+1,k_1,k_2) + \mu c_{k_1} J(n-1,0,k_2) \\ &+ \mu\left(1-c_{k_1}\right) J\big(n, k_1+1, k_2\big), \quad \text{and} \quad (26) \\ h^2_J(n,k_1,k_2) =& -w(n) + \mu\left[\alpha p_{k_2} v + \beta(1-p_{k_2})\bar{v}\right] + \lambda J(n+1,k_1,k_2) + \mu c_{k_2} J(n-1,k_1,0) \\ &+ \mu\left(1-c_{k_2}\right) J\big(n, k_1, k_2+1\big). \quad (27) \end{aligned}$$

For $n = 1$, functions $h^1_J(1,k,0)$ and $h^2_J(1,0,k)$ are defined as

$$h^1_J(1,k,0) = -w(1) + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v} + c_k J(0,0,0) + (1-c_k)\, J\big(1,k+1,0\big)\right] + \lambda J(2,k,0), \quad (28)$$

$$h^2_J(1,0,k) = -w(1) + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v} + c_k J(0,0,0) + (1-c_k)\, J\big(n,0,k+1\big)\right] + \lambda J(2,0,k). \quad (29)$$

[8] As the Bellman equations illustrate, the dimension of the state space grows with the number of tasks that are allowed to be put on hold. For a system where the DM can simultaneously collect observations on a maximum of K tasks, the optimal decision on which task to search next corresponds to a mapping defined on a K-dimensional space, which is hard to describe and even compute when $K > 2$.

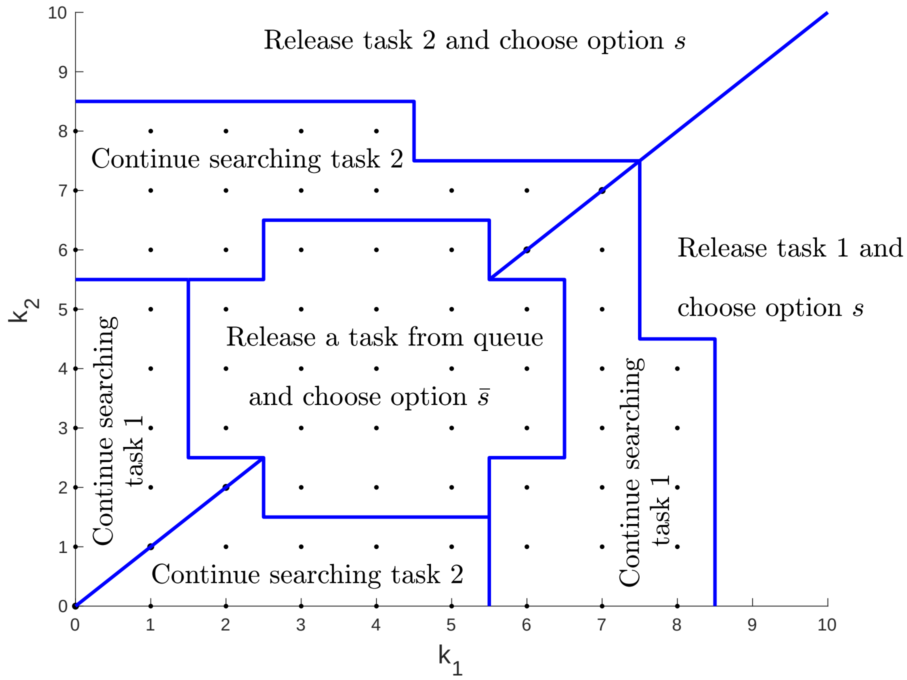

Figure 5 Illustration of the Optimal Policy for $n = 9$.

We solve equations (21)-(29) using the value iteration algorithm (Bertsekas 2007). Figure 5 depicts an example of optimal policy in the cross section $(n = 9, k_1, k_2)$, while Figure 6 depicts the same policy in the cross section $(n, k_1, k_2 = 4)$. The problem parameters for this policy are the same as the ones used to generate Figure 1. The only difference between both cases is that, the DM can actively search two tasks in Figures 5 and 6, as opposed to only one in Figure 1.

As Figure 5 illustrates, the optimal policy splits the state space in different decision regions. Coordinates (k_1, k_2) of each point in this graph represent the numbers of inconclusive observations that have been collected for tasks 1 and 2, respectively. Because tasks are homogenous, the figure is symmetric around the 45° line. When k_i is sufficiently large, the DM terminates task i and chooses option s, for $i = 1, 2$ (choices (iv) and (v), respectively). This corresponds to the last case of Theorem 1, where $k \geq \bar{k}$ and $n \geq \hat{n}(k)$. When (k_1, k_2) belongs to the central dodecagon, the DM releases a waiting task and chooses option $\bar{s}$ without search (choice (iii)). This corresponds to the second case of Theorem 1, where $k < \bar{k}$ and $n \geq \hat{n}(k)$. Otherwise, the DM collects an additional observation for either Task 1 or Task 2 (choices (i) and (ii), respcectively), which generalizes the first case of Theorem 1, where $n < \hat{n}(k)$.

Figure 6 further illustrates the link between this set-up and our base model. Specifically, the region below the non-monotone upper threshold on n in Figure 6 corresponds to the decision of continuing the search (either on task 1 or 2). The threshold is first non-increasing, then non-decreasing and finally non-decreasing in k_1 (while keeping $k_2 = 4$). In addition, the DM alleviates

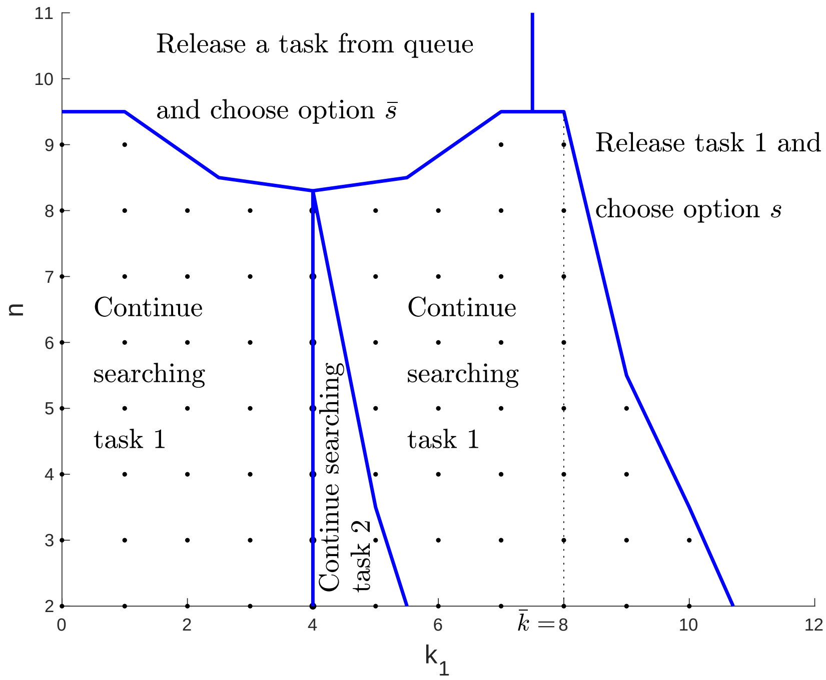

Figure 6 Illustration of the Optimal Policy for $k_2 = 4$.

the workload by releasing a task from the queue if k_1 is below a threshold (denoted as $\bar{k}$ in the figure) and by terminating task 1 otherwise. This corresponds exactly to the general structure depicted by Figure 1 for our base model. Taken together, these results indicate that our main insights continue to hold in this set-up, when the search process is now understood as collecting observations on two tasks, as opposed to just one.

Figures 5 and 6 also shed lights on the optimal switching decisions between the two tasks, a problem that does not exist for Figure 1. To see this, consider the decision regions corresponding to continue searching either task 1 or task 2. Figure 5 shows that for low values of k_1 and k_2, the DM seeks to balance the number of collected observations between the two tasks and is indifferent between them when $k_1 = k_2$. For instance, in state $(k_1 = 1, k_2 = 0)$, the optimal policy prescribes to search task 2, in which case the state moves to $(k_1 = 1, k_2 = 1)$ on the diagonal (if the signal is inconclusive and no new task arrives). For high values of either k_1 or k_2, however, the DM focuses on finishing the task with the higher number of collected observations. For instance, in state $(k_1 = 7, k_2 = 1)$, it is optimal to search task 1 so that the state moves further away from the diagonal.

Figure 6 confirms these findings. When $k_1 < k_2 = 4$, it is optimal to search task 1 so that k_1 approaches k_2. When $k_1 = 4 = k_2$, the DM is indifferent between the two tasks. When $k_1 > 4$, however, the DM should search task 2 so that k_2 now approaches k_1, as long as k_1 is not too large.

This happens when $k_1 < 6$ and $n < 4$. Otherwise, the DM should again focus the search on task 1 to finish it off.

6.5. Tasks without Decisions

Thus far, we have considered situations where the DM can stop the current search and decide to proceed with option s even without being certain that this choice is the correct one. In this case, the DM enjoys v only if this decision turns out to be the right one, which happens with probability p_k. As a result, the early termination value r_k depends on the search length k. In this section, we consider a set-up akin to McCardle et al. (2017), where the the DM only enjoys the benefit of a success v if the search yields a conclusive and positive signal. The DM always chooses option $\bar{s}$ otherwise, and, therefore, the value of early termination is always zero. In this sense, a task does not involve a choice of options.

This set-up corresponds to our base model with $\bar{v} = c = \bar{c} = 0$ and $\beta = 0$, and where the DM's option set is restricted to $\bar{s}$ as long as she aborts the current search without a positive signal. That is, the only conclusive signal is $+$, which indicates success of the project, and yields a value v to the DM. At any point in time, the DM has to decide whether to abandon a project under search, or from the queue, which yields value 0 to the DM, instead of r_k.

Note that because the DM no longer receives value r_k when the project is abandoned after k inconclusive searches, this model is not a special case of our base model (2)-(4). In this case, $c_k = \alpha p_k$, and the corresponding Bellman equation becomes the following.

$$J(n,k) = \max\Big\{\max\big\{J(n-1,0),\ J(n-1,k)\big\},\ h_J^r(n,k)\Big\},\ \text{for } n \geq 1, k, \tag{30}$$

$$J(1,k) = \max\Big\{J(0,0),\ h_J^r(1,k)\Big\}, \tag{31}$$

$$J(0,0) = \lambda J(1,0)/(\lambda+\gamma), \tag{32}$$

where the function $h_J^r(n,k)$ is defined as

$$h_J^r(n,k) = -w(n) + \lambda J(n+1,k) + \mu\alpha p_k\left[v + J(n-1,0)\right] + \mu\left(1-\alpha p_k\right) J\big(n,k+1\big). \tag{33}$$

Without accumulation, the difference between this set-up and our base model does not fundamentally matter, because the general structure of the optimal policy – a fixed threshold on the search time – is the same in both set-ups. With accumulation, however, this difference influences how the DM should alleviate the workload, and thus may affect the structure of the optimal policy characterized in the following Theorem 5.

THEOREM 5. *A non-increasing threshold $\hat{n}(k) \geq 0$ and a threshold $\hat{k}$ exist, such that it is optimal to continue searching a task if $n < \hat{n}(k)$. If $n \geq \hat{n}(k)$, on the other hand, the DM aborts the current search when $k > \hat{k}$, and releases a waiting task when $k \leq \hat{k}$.*

Theorem 5 states that it is never optimal to alleviate the workload by releasing an accumulated task without search. Note that this structure only arises in our base model when $p_0 > \theta$.

7. Conclusion

This paper explores how time pressure in the form of task accumulation might affect the information gathering process of a decision maker. Under this form of pressure, the search costs are endogenous, but can be mitigated by releasing some of the accumulated tasks. In this case, the DM immediately chooses the option she prefers a priori, without a search. This points to the importance of the DM's prior belief when deciding under accumulated pressure.

Our most striking finding is perhaps that the highest workload the DM is willing to sustain may *increase* in the middle of a search process. This contrasts with the beginning and end of the search, where the maximum workload needs to decrease as more information is gathered. This also runs counter to the more intuitive argument that the decreasing marginal value of working on a task should induce the DM to decrease the search cost, and thus the maximum workload, as a result (Hopp et al. 2007).

This finding, however, only emerges in a context of falsifying search tasks, in which a search tends to disprove the DM's prior belief about which option is better. In this case, the decision's ambivalence (as defined in Section 3) increases in the beginning of an inconclusive search. This effect compounds with an inconclusiveness effect, by which the more a search is inconclusive, the more likely it will remain so in the future. Overall, therefore, the need to increase the workload in the middle of a search can be traced back to the ambivalence of a decision task.

We believe that this insight is in fact general. Specifically, a task in our paper corresponds to a basic search problem inspired by the literature (McCardle et al. 2017). This set-up captures essential features of an information gathering problem, but more complex decision tasks exist (DeGroot 1970). Our results suggest that for these richer (albeit more challenging) problems, the DM may have an incentive to increase the maximum workload as the search progresses, as long as the information gathering process may increase the decision's ambivalence. In this case, the DM should regulate the workload by releasing some accumulated tasks when the decision's ambivalence is sufficiently high, and by completing the current task otherwise.

Nonetheless, our basic search problem and its related literature assume a DM without any cognitive limitation in terms of speed or capacity. An interesting problem, therefore, is whether our insights change with a bounded rational DM. Recent models of limited attention provide a promising direction to address this question. For instance, Che and Mierendorff (2017) study a setting similar to ours without congestion, except that the DM needs to choose what type of signals to obtain. In this spirit, an extension of our work could account for the DM's limited attention towards the number of accumulated tasks (see, for instance, Canyakmaz and Boyaci 2018 for a recent application of rational inattention in a queuing setting).

Another important assumption of our main model is that the DM holds the same prior belief for all tasks. This seems reasonable, since determining which prior applies to a new task arguably requires gathering some information on the task, which, in effect, boils down to our current set-up. We nonetheless consider an extension with heterogenous priors that are learned immediately. This provides a promising starting point to explore more sophisticated situations with heterogenous tasks. For instance, our current set-up can be generalized to multiple streams of tasks with different priors, each accumulating in a separate queue. This raises the intriguing questions of which task should be processed first, and how the different priors might affect the shape of the optimal policy.

Finally, from a technical perspective, we fully characterize the structure of the optimal policy for a partially observable Markov decision process. To that end, we propose a proof by induction on the number of tasks in the system. This may provide a useful approach to tackle other decision problems under time pressure in the form of accumulation.

References

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

Arrow, K., D. Blackwell, and M. Girshick (1949). Bayes and minimax solutions of sequential decision problems. *Econometrica 17*(3/4), 213–244.

Bertsekas, D. (2005). *Dynamic Programming and Optimal Control*, Volume I. Athena Scientific.

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

Bouns, G. (2003). Queueing models with admission and termination control-monotonicity and threshold results. Ph.D. thesis, Technische Universiteit Eindhoven.

Canyakmaz, C. and T. Boyaci (2018). Opaque queues: Service systems with rationally inattentive customers.

Che, Y.-K. and K. Mierendorff (2017). Optimal sequential decision with limited attention. *working paper*.

de Vericourt, F. and Y. Zhoug (2005). Managing response time in a call-routing problem with service failure. *Operations Research 53*(6), 968–981.

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

George, J. and J. Harrison (2001). Dynamic control of a queue with adjustable service rate. *Operations Research 49*(5), 720–731.

Gerdtz, M. and T. Bucknall (2001). Triage nurses' clinical decision making: An observational study of urgency assessment. *Journal of Advanced Nursing 35*(4), 550–561.

Girotra, K., C. Terwiesch, and U. K. (2007). Valuing r&d projects in a portfolio: Evidence from the pharmaceutical industry. *Management Science 53*(9), 1452–1466.

Hopp, W., S. Iravani, and G. Yuen (2007). Operations systems with discretionary task completion. *Management Science 53*(1), 61–77.

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

Loch, C. and C. Terwiesch (1999). Accelerating the process of engineering change orders: Capacity and congestion effects. *Journal of Product Innovation Management 16*(2), 145–159.

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

McCardle, K. F., I. Tsetlin, and R. L. Winkler (2017). When to abandon a research project and search for a new one. *Operations Research*.

Savage, L. (1972). *The Foundations of Statistics*. Dover Publications.

Smith, J. and C. Ulu (2012). Technology adoption with uncertain future costs and quality. *Operations Research 60*(2), 262–274.

Smith, J. and C. Ulu (2017). Risk aversion, information acquisition, and technology adoption. *Operations Research 65 (4) 1011-1028.*(4), 262–274.

Travers, D. (1999). Triage: How long does it take? how long should it take? *Journal of Emergency Nursing 25*(3), 238–240.

Ulu, C. and J. Smith (2009). Uncertainty, information acquisition and technology adoption. *Operations Research 57*(3), 740–752.

Appendix. Proofs

We start by presenting the proofs for Lemmas 1–5 and Propositions 3–4, which establish certain properties of the optimal value function corresponding to Bellman equations (2) – (4). Then, we use those properties to show that the structure of the optimal policy is as stated in Theorem 1. Next, we provide the proof for Proposition 2, because a more general version of Lemma 4 is proved in this appendix that encompasses the no congestion model as a special case. The proofs for Theorems 2–5 follow similar steps to Theorem 1 and are presented at the end of the appendix. Throughout the appendix and in order to simplify the derivations, we impose the following assumption, which is without loss of generality.

Assumption 1. *Without loss of generality, we assume hereafter that* $\big(r_0 \vee \bar{r}_0\big) \geq 0$ *and* $R \geq 0$.

Proof of Lemma 1

The monotonicity of p_k, c_k, r_k and $\bar{r}_k$ is established as follows:

$$\beta > \alpha \Leftrightarrow (1-\beta)(1-p_k) < (1-\alpha)(1-p_k) \Leftrightarrow (1-\alpha)p_k + (1-\beta)(1-p_k) < (1-\alpha)$$
$$\Leftrightarrow \frac{(1-\alpha)}{(1-\alpha)p_k + (1-\beta)(1-p_k)} > 1 \Leftrightarrow \frac{(1-\alpha)p_k}{(1-\alpha)p_k + (1-\beta)(1-p_k)} > p_k \Leftrightarrow p_{k+1} > p_k \,.$$

$$p_{k+1} > p_k \Leftrightarrow (\alpha-\beta)p_{k+1} < (\alpha-\beta)p_k \Leftrightarrow (\alpha-\beta)p_{k+1} + \beta < (\alpha-\beta)p_k + \beta \Leftrightarrow c_{k+1} < c_k \,.$$
$$p_{k+1} > p_k \Leftrightarrow p_{k+1}(v+\bar{c}) - \bar{c} > p_k(v+\bar{c}) - \bar{c} \Leftrightarrow r_{k+1} > r_k \,.$$
$$p_{k+1} > p_k \Leftrightarrow -p_{k+1}(\bar{v}+c) + \bar{v} < -p_k(\bar{v}+c) + \bar{v} \Leftrightarrow \bar{r}_{k+1} < \bar{r}_k \,.$$

■

Proof of Lemma 2

Since we have a maximization problem, it suffices to show that all instantaneous costs are bounded from above. To verify this, note that $-w(n)$, r_k, $\bar{r}_k$, v and $\bar{v}$ are all bounded from above for all n and k. Thus, the Negativity Assumption holds and the result immediately follows (Proposition 3.1.6 of Bertsekas (2007)).

■

Proof of Lemma 3

Denote $a^*(n,k)$ as the optimal action when the system is in state (n,k). Also, introduce policy π, which is different from the optimal policy in the following manner: Instead of taking action $a^*(n,k)$ when the system is in state (n,k), policy π takes action $a^*(n+1,k)$.

Consider two systems, referred to as systems I and II, which are in states $(n+1,k)$ and (n,k), respectively. Suppose system I follows the optimal policy whereas system II follows policy π. Further, assume that the two systems evolve according to *the exact same sample path*, which we denote by z.[9] We are interested in the evolution of the two systems until the number of tasks in system I

[9] A sample path represents a realization of all random variables, including arrivals, service times, and test results.

reaches n for the first time.[10] Define T^z to denote the time needed to reach such an outcome under sample path z. There are two possible scenarios that may cause the number of tasks in system I to reach n:

(i) System I dismisses a task from the queue at time $t = T^z$. Let $\mathcal{Z}_1$ represent the set of all sample paths corresponding to this scenario. In this case, at time $t = T^z$, system I is in state $(n+1, k^z)$ for some $k^z \geq 0$ so that $a^*(n+1, k^z)$ prescribes dismissing a task from the queue. At this point, system II (which is in state (n, k^z)) stops following policy π, and switches to the optimal policy. Therefore, both systems end up at state (n, k^z), and both follow the optimal policy afterwards.

Now, let Π_i^z represent the total payoff that system $i \in \{I, II\}$ collects under sample path z (including waiting cost, rewards for correct decisions, and penalties for incorrect identifications) during $t \in [0, T^z)$. Note that for any sample path z, Π_I^z and Π_{II}^z are equivalent except system II always carries one less task for all $t \in [0, T^z)$. This implies that $\Pi_{II}^z \geq \Pi_I^z$, and the equality holds only if $T^z = 0$. Thus, for all $z \in \mathcal{Z}_1$, we have

$$\Pi_{II}^z + e^{-\gamma T^z} J^*(n, k^z) \geq \Pi_1^z + e^{-\gamma T^z} J^*(n, k^z)$$

$$\Leftrightarrow$$

$$\Pi_{II}^z + e^{-\gamma T^z}\big(r_0 \vee \bar{r}_0\big) + e^{-\gamma T^z} J^*(n, k^z) \geq \Pi_1^z + e^{-\gamma T^z}\big(r_0 \vee \bar{r}_0\big) + e^{-\gamma T^z} J^*(n, k^z)\,.$$

Furthermore, $J^*(n,k) \geq J^*(n-1,k) + \big(r_0 \vee \bar{r}_0\big) \geq J^*(n-1,k)$, which implies $J^*(n,k) - n\big(r_0 \vee \bar{r}_0\big) \nearrow n$. In particular, $J^*(n,0) - n\big(r_0 \vee \bar{r}_0\big) \nearrow n$ and hence, we can deduce

$$\underbrace{\Pi_{II}^z + e^{-\gamma T^z} J^*(n, k^z)}_{\text{System II's total payoff under path } z} - e^{-\gamma T^z} J^*(n-1, 0) \geq \underbrace{\Pi_1^z + e^{-\gamma T^z}\big(r_0 \vee \bar{r}_0\big) + e^{-\gamma T^z} J^*(n, k^z)}_{\text{System I's total payoff under path } z} - e^{-\gamma T^z} J^*(n, 0)\,. \tag{34}$$

(ii) System I releases the task in service by announcing it as type s or $\bar{s}$. Let $\mathcal{Z}_2$ represent the set of all sample paths corresponding to this scenario. In this case, after time T^z is elapsed, the two systems are in states $(n, 0)$ and $(n-1, 0)$, respectively. From this point onward, system II stops following policy π, and switches to the optimal policy. Let Π_i^z represent the total payoff that system i collects under sample path z during $t \in [0, T^z]$. As before, we know that $\Pi_{II}^z \geq \Pi_I^z$ since system I carries one more task for all $t \in [0, T^z]$. Thus, for all $z \in \mathcal{Z}_2$, we can write

$$\Pi_{II}^z \geq \Pi_I^z$$

$$\Leftrightarrow$$

[10] Note that before reaching this outcome, the number of tasks in system I may go up and down multiple times due to task arrivals and task dismissals/completions.

$$\underbrace{\Pi^z_{II} + e^{-\gamma T^z} J^*(n-1,0)}_{\text{System II's total payoff under path } z} \quad -e^{-\gamma T^z} J^*(n-1,0) \geq$$
$$\underbrace{\Pi^z_{I} + e^{-\gamma T^z} J^*(n,0)}_{\text{System I's total payoff under path } z} \quad -e^{-\gamma T^z} J^*(n,0)\,. \tag{35}$$

Putting (34) and (35) together, and assuming that $f(.)$ is the probability density function for the sample path realizations, we have

$$\begin{aligned}
&\int_{z\in\mathcal{Z}_1} \left[\Pi^z_{II} + e^{-\gamma T^z} J^*(n,k^z) - e^{-\gamma T^z} J^*(n-1,0)\right] f(z)dz \\
&+ \int_{z\in\mathcal{Z}_2} \left[\Pi^z_{II} + e^{-\gamma T^z} J^*(n-1,0) - e^{-\gamma T^z} J^*(n-1,0)\right] f(z)dz \;\geq \\
&\int_{z\in\mathcal{Z}_1} \left[\Pi^z_{I} + e^{-\gamma T^z}\left(r_0 \vee \bar{r}_0\right) + e^{-\gamma T^z} J^*(n,k^z) - e^{-\gamma T^z} J^*(n,0)\right] f(z)dz \\
&+ \int_{z\in\mathcal{Z}_2} \left[\Pi^z_{I} + e^{-\gamma T^z} J^*(n,0) - e^{-\gamma T^z} J^*(n,0)\right] f(z)dz \\
\Rightarrow& \\
&\int_{z\in\mathcal{Z}_1} \left[\Pi^z_{II} + e^{-\gamma T^z} J^*(n,k^z)\right] f(z)dz + \int_{z\in\mathcal{Z}_2} \left[\Pi^z_{II} + e^{-\gamma T^z} J^*(n-1,0)\right] f(z)dz \\
&- \mathbb{E}_z\left[e^{-\gamma T^z}\right] J^*(n-1,0) \;\geq \\
&\int_{z\in\mathcal{Z}_1} \left[\Pi^z_{I} + e^{-\gamma T^z}\left(r_0 \vee \bar{r}_0\right) + e^{-\gamma T^z} J^*(n,k^z)\right] f(z)dz + \int_{z\in\mathcal{Z}_2} \left[\Pi^z_{I} + e^{-\gamma T^z} J^*(n,0)\right] f(z)dz \\
&- \mathbb{E}_z\left[e^{-\gamma T^z}\right] J^*(n,0)\,.
\end{aligned}$$

Note that the first line in the above inequality provides a lower bound for $J^*(n,k)$ since it represents the total payoff collected during $t\in[0,\infty)$, under all possible sample paths, for a system starting from state (n,k) and following a suboptimal policy. On the other hand, the third line is equal to $J^*(n+1,k)$ since it represents the total payoff collected during $t\in[0,\infty)$, under all possible sample paths, for a system starting from state $(n+1,k)$ and following the optimal policy. It follows that,

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

Finally, we add the above inequality to the following one

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

which gives us,

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

■

Next we proceed to prove a more general version of Lemma 4, where there is an upper bound, N, on the number of tasks allowed in the system. Therefore, our main model corresponds to the special case where N approaches infinity, while the model in Section 4 corresponds to $N=1$.

First, define operators T and Δ, respectively, as

$$\begin{aligned} TJ(n,k) &= -w(n) + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v}\right] + \mu\left[\alpha p_k + \beta(1-p_k)\right] J^*(n-1,0) \\ &\quad + \mu\left[(1-\alpha)p_k + (1-\beta)(1-p_k)\right] J\big(n,k+1\big) + \lambda J(n+1,k) \quad \text{for } 1 \le n < N, \\ TJ(N,k) &= -w(N) + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v}\right] + \mu\left[\alpha p_k + \beta(1-p_k)\right] J^*(N-1,0) \\ &\quad + \mu\left[(1-\alpha)p_k + (1-\beta)(1-p_k)\right] J\big(N,k+1\big) + \lambda J(N,k). \end{aligned}$$

$$\begin{aligned} \Delta J(n,k) &= \max\Big\{TJ(n,k), r_k \vee \bar{r}_k + J^*(n-1,0),\ r_0 \vee \bar{r}_0 + J(n-1,k)\Big\}, \text{ for } n \ge 2, k \ge 0 \\ \Delta J(1,k) &= \max\Big\{TJ(1,k), r_k \vee \bar{r}_k + J^*(0,0)\Big\}, \text{ for } k \ge 0 \\ \Delta J(0,0) &= \lambda J(1,0)/(\lambda+\gamma)\,. \end{aligned}$$

Note that the only difference between the definition TJ and h_J defined in (5) is that we use $J^*(n-1,0)$ in place of $J(n-1,0)$ here. We need the following result before proving Lemma 4.

LEMMA 6. *The optimal value function $J^*(.,.)$ corresponding to the Bellman equations* (2)-(4) *uniquely exists and satisfies $\Delta J^*(.,.) = J^*(.,.)$. Further, $J^*(.,.)$ can be obtained by the value iteration algorithm starting from any arbitrary function $J_0(.,.)$, i.e.,*

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

Proof: Since we have a maximization problem, it suffices to show that all instantaneous costs are bounded from above. Then, the Negativity Assumption holds and the result immediately follows (Proposition 3.1.6 of Bertsekas (2007)).

First, it is obvious that $-w(n)$, v, ,$\bar{v}$, r_k and $\bar{r}_k$ are bounded from above for all n and k. Thus, we only need to establish an upper bound for $J^*(n-1,0)$, which immediately follows from discounting. In particular,

$$J^*(0,0) \le \mathbb{E}\left[e^{-\gamma\tau}\right]\Big(\max\{v,\bar{v}\} + J^*(0,0)\Big)\,,$$

where τ is a random variable representing the time until the next arrival, and follows exponential distribution with rate λ. Thus,

$$J^*(0,0) \le \frac{\lambda}{\lambda+\gamma}\Big(\max\{v,\bar{v}\} + J^*(0,0)\Big) \quad \Leftrightarrow \quad J^*(0,0) \le \frac{\lambda}{\gamma}\max\{v,\bar{v}\} \quad \Rightarrow$$

$$J^*(n-1,0) \le J^*(N,0) \le N\max\{v,\bar{v}\} + J^*(0,0) \le \left[N + \frac{\lambda}{\gamma}\right]\max\{v,\bar{v}\}\,.$$

■

Proof of Lemma 4

We prove that operator Δ simultaneously propagates the following two properties. This, together with its convergence established in Lemma 6, completes the proof:

(i) $J(n,k) \le J^*(n-1,0) + p_k v + (1-p_k)\bar{v}$.

(ii) $J(n,k) - r_k \searrow k$.

Suppose function $J(.,.)$ satisfies properties (i) and (ii). We start by showing the propagation of property (i). For this, we show that function $\Delta J(\cdot,\cdot)$ satisfies the property as long as function $J(\cdot,\cdot)$ does. Note that, for $n \ge 2$,

$$J(n-1,k) + \big(r_0 \vee \bar{r}_0\big) \underbrace{\le}_{\text{property (i)}} p_k v + (1-p_k)\bar{v} + J^*(n-2,0) + \big(r_0 \vee \bar{r}_0\big) \le p_k v + (1-p_k)\bar{v} + J^*(n-1,0)\,.$$

Further, $r_k \le p_k v$, and $\bar{r}_k \le (1-p_k)\bar{v}$, and hence,

$$r_k \vee \bar{r}_k + J^*(n-1,0) \le p_k v + (1-p_k)\bar{v} + J^*(n-1,0)\,.$$

Therefore, to show $\Delta J(.,.)$ satisfies property (i), it suffices to show that $TJ(n,k) \le p_k v + (1-p_k)\bar{v} + J^*(n-1,0)$. For this,

$$\begin{aligned}
TJ(n,k) \quad = \quad & -w(n) + \lambda J\big(\min\{n+1,N\},k\big) + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v}\right] \\
& +\mu\left[(1-\alpha)p_k + (1-\beta)(1-p_k)\right] J\big(n,k+1\big) + \mu\left[\alpha p_k + \beta(1-p_k)\right] J^*(n-1,0) \\
\underbrace{\le}_{\text{property (i)}} \quad & \\
& -w(n) + \lambda\left[J^*(n,0) + p_k v + \beta(1-p_k)\bar{v}\right] + \mu\left[\alpha p_k v + \beta(1-p_k)\bar{v}\right] \\
& +\mu\left[(1-\alpha)p_k + (1-\beta)(1-p_k)\right] \Big(J^*\big(n-1,0\big) + p_{k+1}v + \beta(1-p_{k+1})\bar{v}\Big) \\
& +\mu\left[\alpha p_k + \beta(1-p_k)\right] J^*(n-1,0) \\
= \quad & \\
& -w(n) + \lambda J^*(n,0) + (\lambda+\mu)\left[p_k v + (1-p_k)\bar{v}\right] + \mu J^*(n-1,0). \qquad (36)
\end{aligned}$$

From Lemma 3, we know that $J^*(n,0) - J^*(n-1,0) \le J^*(1,0) - J^*(0,0)$. Thus,

$$\begin{aligned}
& \lambda\Big(J^*(n,0) - J^*(n-1,0)\Big) \le \lambda\Big(J^*(1,0) - J^*(0,0)\Big) = \gamma J^*(0,0) \le \gamma J^*(n-1,0) \\
\Leftrightarrow & \\
& \lambda J^*(n,0) \le (\lambda+\gamma) J^*(n-1,0)\,.
\end{aligned}$$

The above inequality, together with (36), leads to

$$\begin{aligned}
TJ(n,k) & \le -w(n) + (\lambda+\gamma)J^*(n-1,0) + (\lambda+\mu)\left[p_k v + (1-p_k)\bar{v}\right] + \mu J^*(n-1,0) \qquad (37) \\
& \le J^*(n-1,0) + p_k v + (1-p_k)\bar{v}\,.
\end{aligned}$$

We now prove the propagation of property (ii). From property (i), we have

$$-p_k v - (1-p_k)\bar{v} \le J^*(n-1,0) - J\big(n,k+1\big)$$

$$\Rightarrow$$

$$-p_k(\beta-\alpha)v - \beta\bar{v} \le (\beta-\alpha)\left[J^*(n-1,0) - J\big(n,k+1\big)\right]$$

$$\Leftrightarrow$$

$$-p_k(1-p_k)(\beta-\alpha)v - (1-p_k)\beta\bar{v} \le (\beta-\alpha)(1-p_k)\left[J^*(n-1,0) - J\big(n,k+1\big)\right]$$

$$\Rightarrow$$

$$-(p_{k+1}-p_k)(1-\alpha)v - (1-p_k)\beta\bar{v} \le (\beta-\alpha)(1-p_k)\left[J^*(n-1,0) - J\big(n,k+1\big)\right]$$

$$\Leftrightarrow$$

$$(1-p_k)\alpha v - \alpha(1-p_{k+1})v - (1-p_k)\beta\bar{v} \le (\beta-\alpha)(1-p_k)\left[J^*(n-1,0) - J\big(n,k+1\big)\right] + (p_{k+1}-p_k)v$$

$$\Rightarrow$$

$$(1-p_k)\alpha v - \alpha(1-p_{k+1})(v+\bar{c}) - (1-p_k)\beta\bar{v}$$

$$\le$$

$$(\beta-\alpha)(1-p_k)\left[J^*(n-1,0) - J\big(n,k+1\big)\right] + (p_{k+1}-p_k)(v+\bar{c})$$

$$\Leftrightarrow$$

$$(1-p_k)\alpha v - (1-p_k)(v+\bar{c}) - (1-p_k)\beta\bar{v}$$

$$\le$$

$$(\beta-\alpha)(1-p_k)\left[J^*(n-1,0) - J\big(n,k+1\big)\right] - (1-\alpha)(1-p_{k+1})(v+\bar{c})$$

$$\Leftrightarrow$$

$$(p_{k+1}-p_k)\alpha v - (p_{k+1}-p_k)(v+\bar{c}) - (p_{k+1}-p_k)\beta\bar{v}$$

$$\le$$

$$(\beta-\alpha)\big(p_{k+1}-p_k\big)\left[J^*(n-1,0) - J\big(n,k+1\big)\right] - (\beta-\alpha)p_{k+1}(1-p_{k+1})(v+\bar{c})$$

$$\Leftrightarrow$$

$$\alpha p_{k+1}v + \beta(1-p_{k+1})\bar{v} + (\beta-\alpha)p_{k+1}(1-p_{k+1})(v+\bar{c}) - r_{k+1}$$

$$\le$$

$$\alpha p_k v + \beta(1-p_k)\bar{v} + (\beta-\alpha)\big(p_{k+1}-p_k\big)\left[J^*(n-1,0) - J\big(n,k+1\big)\right] - r_k$$

$$\Leftrightarrow$$

$$\alpha p_{k+1}v + \beta(1-p_{k+1})\bar{v} + [(1-\alpha)p_{k+1} + (1-\beta)(1-p_{k+1})]\big(p_{k+2}-p_{k+1}\big)(v+\bar{c}) - r_{k+1}$$

$$\leq$$

$$\alpha p_k v + \beta(1-p_k)\bar{v} + (\beta-\alpha)\big(p_{k+1}-p_k\big)\left[J^*(n-1,0) - J\big(n,k+1\big)\right] - r_k$$

$$\Leftrightarrow$$

$$\alpha p_{k+1} v + \beta(1-p_{k+1})\bar{v} + [(1-\alpha)p_{k+1} + (1-\beta)(1-p_{k+1})]\left(r_{k+2} - r_{k+1}\right) - r_{k+1}$$

$$\leq$$

$$\alpha p_k v + \beta(1-p_k)\bar{v} + (\beta-\alpha)\big(p_{k+1}-p_k\big)\left[J^*(n-1,0) - J\big(n,k+1\big)\right] - r_k$$

$$\Leftrightarrow$$

$$\alpha p_{k+1} v + \beta(1-p_{k+1})\bar{v} + [(1-\alpha)p_{k+1} + (1-\beta)(1-p_{k+1})]\left(r_{k+2} - r_{k+1}\right)$$

$$+ [\alpha p_{k+1} + \beta(1-p_{k+1})]\, J^*(n-1,0) - r_{k+1}$$

$$\leq$$

$$\alpha p_k v + \beta(1-p_k)\bar{v} - (\beta-\alpha)\big(p_{k+1}-p_k\big) J\big(n,k+1\big) + [\alpha p_k + \beta(1-p_k)]\, J^*(n-1,0) - r_k$$

$$\Leftrightarrow$$

$$\alpha p_{k+1} v + \beta(1-p_{k+1})\bar{v} + [(1-\alpha)p_{k+1} + (1-\beta)(1-p_{k+1})]\Big(J\big(n,k+1\big) - r_{k+1} + r_{k+2}\Big)$$

$$+ [\alpha p_{k+1} + \beta(1-p_{k+1})]\, J^*(n-1,0) - r_{k+1}$$

$$\leq$$

$$\alpha p_k v + \beta(1-p_k)\bar{v} + [(1-\alpha)p_k + (1-\beta)(1-p_k)]\, J\big(n,k+1\big)$$

$$+ [\alpha p_k + \beta(1-p_k)]\, J^*(n-1,0) - r_k$$

$$\underbrace{\Rightarrow}_{\text{property (ii)}}$$

$$\alpha p_{k+1} v + \beta(1-p_{k+1})\bar{v} + [(1-\alpha)p_{k+1} + (1-\beta)(1-p_{k+1})]\, J\big(n,k+2\big)$$

$$+ [\alpha p_{k+1} + \beta(1-p_{k+1})]\, J^*(n-1,0) - r_{k+1}$$

$$\leq$$

$$\alpha p_k v + \beta(1-p_k)\bar{v} + [(1-\alpha)p_k + (1-\beta)(1-p_k)]\, J\big(n,k+1\big)$$

$$+ [\alpha p_k + \beta(1-p_k)]\, J^*(n-1,0) - r_k$$

$$\underbrace{\Rightarrow}_{\text{property (ii)}}$$

$$\lambda J\big(\min\{n+1,N\},k+1\big) + \mu\,[\alpha p_{k+1} v + \beta(1-p_{k+1})\bar{v}]$$

$$+\mu\,[(1-\alpha)p_{k+1} + (1-\beta)(1-p_{k+1})]\, J\big(n,k+2\big)$$

$$+\mu\,[\alpha p_{k+1} + \beta(1-p_{k+1})]\, J^*(n-1,0) - (\lambda+\mu) r_{k+1}$$

$$\leq$$

$$\lambda J\big(\min\{n+1,N\},k\big)+\mu\,[\alpha p_k v+\beta(1-p_k)\bar{v}]$$

$$+\mu\,[(1-\alpha)p_k+(1-\beta)(1-p_k)]\,J\big(n,k+1\big)+\mu\,[\alpha p_k+\beta(1-p_k)]\,J^*(n-1,0)-(\lambda+\mu)r_k$$

$$\Rightarrow$$

$$TJ(n,k+1)-r_{k+1}\leq TJ(n,k)-r_k\,.$$

Finally, we know from property (ii) that $J(n-1,k+1)+\big(r_0\vee\bar{r}_0\big)-r_{k+1}\leq J(n-1,k)+\big(r_0\vee\bar{r}_0\big)-r_k$. Further, $\big(r_{k+1}\vee\bar{r}_{k+1}\big)-r_{k+1}<\big(r_k\vee\bar{r}_k\big)-r_k$, and hence, $J^*(n-1,0)+\big(r_{k+1}\vee\bar{r}_{k+1}\big)-r_{k+1}<J^*(n-1,0)+\big(r_k\vee\bar{r}_k\big)-r_k$.

Putting all together, the three terms in $\Delta(\cdot,\cdot)$ satisfy property (ii) and thus,

$$\Delta J(n,k+1)-r_{k+1}\leq\Delta J(n,k)-r_k\,.$$

The proof for $J^*(n,k)-\bar{r}_k\nearrow k$ follows similar steps, and therefore is omitted.

■

Proof of Lemma 5

Denote $a^*(n,k)$ as the optimal action when the system is in state (n,k). Further, define policies $\overline{\pi}$ and $\underline{\pi}$ as follows:

- Policy $\overline{\pi}$ takes the action $a^*(n+1,k)$ when the system is in state (n,k), and
- Policy $\underline{\pi}$ takes the action $a^*(n-1,k)$ when the system is in state (n,k).

Consider four systems, referred to as systems I, II, III, and IV, that are in states $(n+1,k)$, $(n-1,k)$, (n,k), and (n,k), respectively. Suppose the first and the second systems are following the optimal policy (starting from states $(n+1,k)$ and $(n-1,k)$, respectively), whereas the third and the fourth systems are following policies $\overline{\pi}$ and $\underline{\pi}$, respectively (both starting from state (n,k)). In addition, assume that the four systems evolve according to *the exact same sample path*, which we denote by z. This implies that arrivals and test completions occur at the same time for all systems. We are interested in the evolution of the four systems until either: (i) the first system dismisses a task from the queue, or (ii) the second system reaches state $(n-2,0)$, whichever that happens first. Let T^z denote the time needed to reach such an outcome under sample path z. Thus, at time $t=T^z$, one of the following two scenarios has happened:

(i) System I dismisses a task from the queue at time $t=T^z$. Let $\mathcal{Z}_1$ represent the set of all sample paths corresponding to this scenario. In this case, at time $t=T^z$, system I is in state (n^z+1,k^z) for some $n^z\geq 2$ and $k^z\geq 0$, so that $a^*(n^z+1,k^z)$ prescribes dismissing a task from the queue. On the other hand, systems II and IV are also at states (n'^z-1,k'^z) and (n'^z,k'^z), respectively, for some $n'^z\geq n$ and $k'^z\geq 0$. At this point, systems III and IV (which are in states (n^z,k^z) and (n'^z,k'^z), respectively) stop following policies $\overline{\pi}$ and $\underline{\pi}$, and switch to the optimal policy.

Now, let Π_i^z represent the total payoff that system i collects under sample path z during $t \in [0, T^z)$. Note that for any sample path z, payoffs Π_I^z and Π_{III}^z are equivalent except system I always carries one more task for all $t \in [0, T^z)$. Similarly, payoffs Π_{II}^z and Π_{IV}^z are equivalent except system II always carries one less task during $t \in [0, T^z)$. Putting these two together, we can deduce from the convexity of the waiting cost function $w(.)$ that $\Pi_I^z + \Pi_{II}^z \le \Pi_{III}^z + \Pi_{IV}^z$. Thus, for all $z \in \mathcal{Z}_1$, we can write

$$\underbrace{\Pi_I^z + e^{-\gamma T^z}\left(r_0 \vee \bar{r}_0\right) + e^{-\gamma T^z} J^*(n^z, k^z)}_{\text{System I's total payoff under path } z} + \underbrace{\Pi_{II}^z + e^{-\gamma T^z} J^*(n'^z - 1, k'^z)}_{\text{System II's total payoff under path } z} \le$$
$$\underbrace{\Pi_{III}^z + e^{-\gamma T^z} J^*(n^z, k^z)}_{\text{System III's total payoff under path } z} + \underbrace{\Pi_{IV}^z + e^{-\gamma T^z} J^*(n'^z, k'^z)}_{\text{System IV's total payoff under path } z}. \tag{38}$$

(ii) System II reaches state $(n-2, 0)$ by releasing the task in service and announcing it as type s or $\bar{s}$. Let $\mathcal{Z}_2$ represent the set of all sample paths corresponding to this scenario. In this case, after time T^z is elapsed, the four systems are in states (n^z, k^z), $(n-2, 0)$, $(n^z - 1, k^z)$, and $(n-1, 0)$, respectively, for some $n^z \ge n$ and $k^z \ge 0$. From this point onward, systems III and IV stop following policies $\overline{\pi}$ and $\underline{\pi}$, respectively, and switch to the optimal policy.

Let Π_i^z represent the total payoff that system i collects under sample path z during $t \in [0, T^z]$. Similar to scenario (i) described above, we know that $\Pi_I^z + \Pi_{II}^z \le \Pi_{III}^z + \Pi_{IV}^z$. Thus, for all $z \in \mathcal{Z}_2$, we can write

$$\Pi_I^z + \Pi_{II}^z \le \Pi_{III}^z + \Pi_{IV}^z$$
$$\Rightarrow$$
$$\underbrace{\Pi_I^z + e^{-\gamma T^z} J^*(n^z, k^z)}_{\text{System I's total payoff under path } z} + \underbrace{\Pi_{II}^z + e^{-\gamma T^z} J^*(n-2, 0)}_{\text{System II's total payoff under path } z} \le$$
$$\underbrace{\Pi_{III}^z + e^{-\gamma T^z} J^*(n^z - 1, k^z)}_{\text{System III's total payoff under path } z} + \underbrace{\Pi_{IV}^z + e^{-\gamma T^z} J^*(n-1, 0)}_{\text{System IV's total payoff under path } z}, \tag{39}$$

where the inequality follows from $J^*(n^z, k^z) - J^*(n-1, 0) \le J^*(n^z - 1, k^z) - J^*(n-2, 0)$ as induced by Lemma 3.

Putting (38) and (39) together, and assuming that $f(.)$ is the probability density function for the sample path realizations, we have

$$\int_{z \in \mathcal{Z}_1} \left[\Pi_I^z + e^{-\gamma T^z}\left(r_0 \vee \bar{r}_0\right) + e^{-\gamma T^z} J^*(n^z, k^z) + \Pi_{II}^z + e^{-\gamma T^z} J^*(n'^z - 1, k'^z)\right] f(z) dz$$
$$+ \int_{z \in \mathcal{Z}_2} \left[\Pi_I^z + e^{-\gamma T^z} J^*(n^z, k^z) + \Pi_{II}^z + e^{-\gamma T^z} J^*(n-2, 0)\right] f(z) dz \le$$
$$\int_{z \in \mathcal{Z}_1} \left[\Pi_{III}^z + e^{-\gamma T^z} J^*(n^z, k^z) + \Pi_{IV}^z + e^{-\gamma T^z} J^*(n'^z, k'^z)\right] f(z) dz$$

$$+\int_{z\in\mathcal{Z}_2}\Big[\Pi^z_{III}+e^{-\gamma T^z}J^*(n^z-1,k^z)+\Pi^z_{IV}+e^{-\gamma T^z}J^*(n-1,0)\Big]\,f(z)dz$$

$$\Rightarrow$$

$$\int_{z\in\mathcal{Z}_1}\Big[\Pi^z_I+e^{-\gamma T^z}\big(r_0\vee\bar{r}_0\big)+e^{-\gamma T^z}J^*(n^z,k^z)\Big]\,f(z)dz+\int_{z\in\mathcal{Z}_2}\Big[\Pi^z_I+e^{-\gamma T^z}J^*(n^z,k^z)\Big]\,f(z)dz$$
$$+\int_{z\in\mathcal{Z}_1}\Big[\Pi^z_{II}+e^{-\gamma T^z}J^*(n'^z-1,k'^z)\Big]\,f(z)dz+\int_{z\in\mathcal{Z}_2}\Big[\Pi^z_{II}+e^{-\gamma T^z}J^*(n-2,0)\Big]\,f(z)dz\le$$
$$\int_{z\in\mathcal{Z}_1}\Big[\Pi^z_{III}+e^{-\gamma T^z}J^*(n^z,k^z)\Big]\,f(z)dz+\int_{z\in\mathcal{Z}_2}\Big[\Pi^z_{III}+e^{-\gamma T^z}J^*(n^z-1,k^z)\Big]\,f(z)dz$$
$$+\int_{z\in\mathcal{Z}_1}\Big[\Pi^z_{IV}+e^{-\gamma T^z}J^*(n'^z,k'^z)\Big]\,f(z)dz+\int_{z\in\mathcal{Z}_2}\Big[\Pi^z_{IV}+e^{-\gamma T^z}J^*(n-1,0)\Big]\,f(z)dz\,.$$

Note that the first line in the above inequality is equal to $J^*(n+1,k)$ since it represents the total payoff collected during $t\in[0,\infty)$, under all possible sample paths, for a system starting from state $(n+1,k)$ and following the optimal policy. By the same logic, the second line in the inequality equals $J^*(n-1,k)$. On the other hand, the third and the fourth lines provide two different lower bounds for $J^*(n,k)$ since they both represent the total payoff collected during $t\in[0,\infty)$, under all possible sample paths, for a system starting from state (n,k) and following two suboptimal policies. It follows that,

$$J^*(n+1,k)+J^*(n-1,k)\le 2J^*(n,k)\,.$$

■

Proof of Proposition 3

Lemma 3 implies that if it is optimal to stop searching a focal task in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Similarly, Lemma 5 implies that if it is optimal to dismiss a task from the queue in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Putting these two results together, we conclude that if it is optimal to continue searching in state (n,k), the same decision must also be optimal in state $(n-1,k)$. Consequently, threshold $\hat{n}(k)$ exists and is defined as the smallest value of n for which continuing the search at state (n,k) is not optimal.

■

Proof of Proposition 4

First, we denote $\phi=v\vee\bar{v}$, and argue that for large enough n, set $\mathcal{K}(n)$ is connected. We have

$$\begin{aligned}h_{J^*}(n,k)&=-w(n)+\mu\left[\alpha p_k v+\beta(1-p_k)\bar{v}\right]+\lambda J^*(n+1,k)+\mu c_k J^*(n-1,0)+\mu\left(1-c_k\right)J^*\big(n,k+1\big)\\&\le -w(n)+\mu\phi+2\lambda\phi+J^*(n-1,0)+\mu c_k J^*(n-1,0)+\mu(1-c_k)\phi+\mu(1-c_k)J^*(n-1,0)\\&\le -w(n)+2(\lambda+\mu)\phi+(\lambda+\mu)J^*(n-1,0)\,.\end{aligned}$$

It follows that

$$h_{J^*}(n,k) - [J^*(n-1,0) + r_k \vee \bar{r}_k] \leq [-w(n) + 2(\lambda+\mu)\phi - \gamma J^*(n-1,0)] - r_k \vee \bar{r}_k \,.$$

Since $r_0 \vee \bar{r}_0 \geq 0$, we have $J^*(n-1,0) \geq 0$. Also, $w(n)$ is weakly convex in n and approaches infinity with n. Therefore, the right hand side of the above inequality becomes strictly negative for sufficiently large values of n, and hence, the continuation decision cannot be optimal. Let $\bar{n}$ represent the smallest value of n for which $J^*(n,k) > h_{J^*}(n,k)$ for all k. Next, suppose $k_1 \in \mathcal{K}(n)$ and $k_2 \in \mathcal{K}(n)$ for $k_1 < k_2$ and $n \geq \bar{n}$. We want to show that $k \in \mathcal{K}(n)$ for all $k_1 < k < k_2$, and hence $\mathcal{K}(n)$ is connected. Otherwise, if $k \notin \mathcal{K}(n)$, we must have $J^*(n,k) = J^*(n-1,0) + r_k$ or $J^*(n,k) = J^*(n-1,0) + \bar{r}_k$, which further implies $J^*(n,k_1) = J^*(n-1,0) + r_{k_1}$ or $J^*(n,k_2) = J^*(n-1,0) + \bar{r}_{k_2}$, respectively, following Lemma 4. This, however, contradicts either $k_1 \in \mathcal{K}(n)$ or $k_2 \in \mathcal{K}(n)$. Therefore, we must have $k \in \mathcal{K}(n)$.

Now consider values n, k_1 and k_2 such that $\mathcal{K}(n+1)$ is connected, $k_1 < k_2$ and $k_1 \in \mathcal{K}(n)$ and $k_2 \in \mathcal{K}(n)$. Define $\tilde{n}(k)+1$ as the smallest value of n for which releasing a task from the queue is optimal when the system is at state (n,k), and note that the existence of this threshold is implied by Lemma 5. We have $\tilde{n}(k_1) < n$, $J^*(n,k_1) = J^*\left(\tilde{n}(k_1),k_1\right) + (n-\tilde{n}(k_1))(r_0 \vee \bar{r}_0)$, and $J^*\left(\tilde{n}(k_1),k_1\right) > J^*\left(\tilde{n}(k_1)-1,k_1\right) + (r_0 \vee \bar{r}_0)$. Furthermore, Lemma 3 implies that $J^*\left(\tilde{n}(k_1),k_1\right) > J^*\left(\tilde{n}(k_1)-1,0\right) + \max\{r_{k_1},\bar{r}_{k_1}\}$. The Bellman equation hence gives us the first equality in the following expression:

$$h_{J^*}\Big(\tilde{n}(k_1),k_1\Big) + (n-\tilde{n}(k_1))(r_0 \vee \bar{r}_0) = J^*\Big(\tilde{n}(k_1),k_1\Big) + (n-\tilde{n}(k_1))(r_0 \vee \bar{r}_0) = J^*(n,k_1) > h_{J^*}\Big(n,k_1\Big),$$

where the last inequality follows from $k_1 \in \mathcal{K}(n)$. Therefore, we have

$$\begin{aligned} w(n) - w\left(\tilde{n}(k_1)\right) >& \mu\left[(1-\alpha)p_{k_1} + (1-\beta)(1-p_{k_1})\right]\Big(J^*(n,k_1+1) - J^*\Big(\tilde{n}(k_1),k_1+1\Big)\Big) \\ &+ \mu\left[\alpha p_{k_1} + \beta(1-p_{k_1})\right]\Big(J^*(n-1,0) - J^*(\tilde{n}(k_1)-1,0)\Big) \\ \geq& \mu\left[\alpha p_{k_1} + \beta(1-p_{k_1})\right]\Big(J^*(n-1,0) - J^*(\tilde{n}(k_1)-1,0)\Big) \\ >& \mu\left[\alpha p_k + \beta(1-p_k)\right]\Big(J^*(n-1,0) - J^*(\tilde{n}(k_1)-1,0)\Big),\ \forall k > k_1, \end{aligned} \tag{40}$$

where the second inequality is implied by the monotonicity of $J^*(.,.)$ in n, and the last inequality is implied by Lemma 1.

Define function $f(n,k) = -w(n) + \mu\left[\alpha p_k + \beta(1-p_k)\right] J^*(n-1,0)$, and note that $f(n,k)$ is concave in n (following the convexity of $w(n)$ and Lemma 5). Further, inequality (40) implies $f(n,k) < f(\tilde{n}(k_1),k)$. The concavity of $f(n,k)$ in n and the fact that $n > \tilde{n}(k_1)$ thus lead to $f(n,k) < f(n-1,k)$ for any $k > k_1$.

In particular, consider $k_2 > k_1$ and $k_2 \in \mathcal{K}(n)$. Because $k_1,\ k_2 \in \mathcal{K}(n)$, Lemma 5 gives us $k_1,\ k_2 \in \mathcal{K}(n+1)$. The induction hypothesis implies that the set $\mathcal{K}(n+1)$ is connected and consequently $k_2 - 1 \in \mathcal{K}(n+1)$. Therefore, $J^*(n+1, k_2-1) = J^*(n, k_2-1)$ and $J^*(n, k_2) = J^*(n-1, k_2)$. Now we argue that $k_2 - 1 \in \mathcal{K}(n)$. In fact,

$$\begin{aligned} h_{J^*}(n, k_2-1) =& f(n,k_2) + \mu\left[(1-\alpha)p_{k_2-1} + (1-\beta)(1-p_{k_2-1})\right] J^*(n,k_2) + \lambda J^*(n+1,k_2-1) \\ <& f(n-1,k_2) + \mu\left[(1-\alpha)p_{k_2-1} + (1-\beta)(1-p_{k_2-1})\right] J^*(n-1,k_2) + \lambda J^*(n,k_2-1) \\ =& h_{J^*}(n-1,k_2-1) \le J^*(n-1,k_2-1) \le J^*(n,k_2-1). \end{aligned}$$

Therefore,

$$J^*(n, k_2-1) > h_{J^*}(n, k_2-1). \tag{41}$$

We also have

$$J^*(n,k_2-1) - r_{k_2-1} \ge J^*(n,k_2) - r_{k_2} > J^*(n-1,0),$$

where the first inequality follows Lemma 4, and the second from $k_2 \in \mathcal{K}(n)$. Hence, we have

$$J^*(n,k_2-1) > r_{k_2-1} + J^*(n-1,0). \tag{42}$$

Following Lemma 4 and the fact that $k_1 \in \mathcal{K}(n)$, we have

$$0 < J^*(n,k_1) - \bar{r}_{k_1} - J^*(n-1,0) \le J^*(n,k_2-1) - \bar{r}_{k_2-1} - J^*(n-1,0).$$

Therefore, we have

$$J^*(n,k_2-1) > \bar{r}_{k_2-1} + J^*(n-1,0). \tag{43}$$

Combining (41), (42) and (43), we know $k_2 - 1 \in \mathcal{K}(n)$. Following the same logic, we know that $k \in \mathcal{K}(n)$ for any $k \in \{k_1, \ldots, k_2\}$.

The monotonicity of $\check{k}(n)$ and $\hat{k}(n)$ follows directly from the definition of threshold $\tilde{n}(k)$ implied by Lemma 5.

■

Proof of Proposition 1

Suppose $\alpha = \beta$. It follows from the definitions of c_k and p_k that $c_k = \alpha$ and $p_k = p_0$ for all $k \ge 0$. As a result, the DM's belief does not change following an inconclusive observation and hence, the DM does not need to keep track of the number of inconclusive observations collected thus far. The state space therefore reduces to one-dimensional and only includes the number of tasks in the system, n. The Bellman equations for this setting is given by,

$$\begin{aligned} &J(n) = \max\left\{r_0 \vee \bar{r}_0 + J(n-1), -w(n) + \mu\alpha\left[p_k v + (1-p_k)\bar{v} + J(n-1)\right] + \lambda J(n+1) + \mu(1-\alpha)J(n)\right\}, \\ &J(0) = \lambda J(1)/(\lambda+\gamma). \end{aligned}$$

The existence of the optimal value function $J^*(.)$ that satisfies the above Bellman equations follows from Lemma 2. Then, we use the result of Lemma 3 to conclude that $J^*(n) - J^*(n-1) \searrow n$. Consequently, if it is optimal to stop the search at state n, the same decision must also be optimal at state $(n+1)$, thereby establishing the threshold $\hat{n}$.

■

Proof of Proposition 2

Consider the proof of Lemma 4 provided above, and notice that the proof is presented for a general case where the total number of tasks in the system is bounded from above by N. In particular, consider the special case of $N = 1$. Since the optimality equation in this case reduces to (6), we can conclude that the optimal value function $J_s^*(\cdot)$ corresponding to (6) also satisfies $J_s^*(k) - r_k \searrow k$ and $J_s^*(k) - \bar{r}_k \nearrow k$.

Now, if it is optimal not to start the search at all, announcing type $\bar{s}$ is optimal since $\bar{r}_0 > r_0$. Otherwise, if it is optimal to start the search in state 0, it will never be optimal to stop the search to announce type $\bar{s}$. To show this, we use contradiction. Specifically, suppose it is optimal to stop and announce type $\bar{s}$ in state k. Then, $J_s^*(k) - \bar{r}_k \nearrow k$ implies that the same decision must also be optimal in state $(k-1)$. Repeating the same argument leads to the optimality of announcing $\bar{s}$ in state 0, which contradicts the premise that the search should ever start.

Finally, the existence of threshold $\bar{k}$ follows immediately from $J_s^*(k) - r_k \searrow k$ (since it establishes that if it is optimal to announce the task as type s in state k, the same decision must also be optimal in state $(k+1)$).

■

Proof of Corollary 1

If $p_0 > \theta$, we have $r_0 > \bar{r}_0$. Thus, dismissing a task from the queue in state (n, k) generates payoff $J^*(n-1, k) + r_0$. However, according to Lemma 4, this value is always dominated by $J^*(n-1, 0) + r_k$ for all $k \geq 0$. As a result, dismissing a task from the queue without search is never *strictly* optimal in state (n, k). Subsequently, from the definition of $\bar{k}$, it follows that $\bar{k} = 0$. The monotonicity of $\hat{n}(k)$ is then implied by Theorem 1.

On the other hand, if $\bar{k} = 0$, dismissing a task from the queue is never strictly optimal (by definition of $\bar{k}$). As a result, for all values of n that are sufficiently large, $J^*(n-1, k) + r_0 \vee \bar{r}_0$ is always dominated by $J^*(n-1, 0) + r_k$. In particular, for $k = 0$ and n large enough, $J^*(n-1, 0) + r_0 \vee \bar{r}_0 \leq J^*(n-1, 0) + r_0$. It follows then that $r_0 \vee \bar{r}_0 \leq r_0$, or equivalently, $\bar{r}_0 \leq r_0$.

■

To proceed with the proof of Theorem 2, we consider the model of Section 6.1, and present the following lemmas first.

LEMMA 7. *The DM's belief increases following an inconclusive observation so that $p_+ > p$, and probability c_p decreases in p. Further, reward r_p increases and reward $\bar{r}_p$ decreases in p.*

Proof: The result follows from the following inequalities:

$$\beta > \alpha \Leftrightarrow (1-\beta)(1-p) < (1-\alpha)(1-p) \Leftrightarrow (1-\alpha)p + (1-\beta)(1-p) < (1-\alpha)$$
$$\Leftrightarrow \frac{(1-\alpha)}{(1-\alpha)p + (1-\beta)(1-p)} > 1 \Leftrightarrow \frac{(1-\alpha)p}{(1-\alpha)p + (1-\beta)(1-p)} > p \Leftrightarrow p_+ > p\,.$$

$$\beta > \alpha \Leftrightarrow (\alpha-\beta)p + \beta \searrow p \Rightarrow c_p \searrow p\,.$$

$$r_p = p(v+\bar{c}) - \bar{c} \Rightarrow r_p \nearrow p\,.$$

$$\bar{r}(p) = -p(\bar{v}+c) + \bar{v} \Rightarrow \bar{r}_p \searrow p\,.$$

■

LEMMA 8. *A function $J^*(\cdot,\cdot)$ exists such that $J^*(\cdot,\cdot)$ satisfies the Bellman equations* (7)-(9) *and $J^*(n,p)$ is the optimal value function given initial state (n,p).*

Proof: The result follows from Proposition 3.1.6 of Bertsekas (2007), because ours is a maximization problem where single period rewards $-w(n)$, v, $\bar{v}$, R, r_p and $\bar{r}_p$ are bounded from above for all values of n and p.

■

LEMMA 9. *For any fixed p, $J^*(n,p) - \mathbb{E}[J^*(n-1,p_0)]$ is non-increasing in n.*

Proof: The proof follows steps similar to the proof of Lemma 3. The main difference is that the two systems start from states $(n+1,p)$ and (n,p), respectively, and the definition of sample path z now also includes the realization of the initial belief p_0 for different tasks (in addition to the realization of arrival times, test completion times, and test results). Further, all terms $r_0 \vee \bar{r}_0$ in the proof of Lemma 3 are now simply replaced with R. More specifically, since $J^*(n,p_0) \geq J^*(n-1,p_0) + R$ for all values of p_0, it follows that $\mathbb{E}[J^*(n,p_0)] \geq \mathbb{E}[J^*(n-1,p_0)] + R$. Then, equation (34) is updated to

$$\underbrace{\Pi_{II}^z + e^{-\gamma T^z} J^*(n,p^z)}_{\text{System II's total payoff under path } z} \quad - e^{-\gamma T^z}\mathbb{E}[J^*(n-1,p_0)] \geq$$
$$\underbrace{\Pi_1^z + e^{-\gamma T^z} R + e^{-\gamma T^z} J^*(n,p^z)}_{\text{System I's total payoff under path } z} - e^{-\gamma T^z}\mathbb{E}[J^*(n,p_0)]\,.$$

Similarly, equation (35) is now updated to

$$\underbrace{\Pi_{II}^{z} + e^{-\gamma T^{z}}\mathbb{E}\left[J^{*}(n-1,p_{0})\right]}_{\text{System II's total payoff under path } z} - e^{-\gamma T^{z}}\mathbb{E}\left[J^{*}(n-1,p_{0})\right] \geq$$
$$\underbrace{\Pi_{I}^{z} + e^{-\gamma T^{z}}\mathbb{E}\left[J^{*}(n,p_{0})\right]}_{\text{System I's total payoff under path } z} - e^{-\gamma T^{z}}\mathbb{E}\left[J^{*}(n,p_{0})\right] .$$

The rest of the proof is exactly similar to the proof of Lemma 3, and we get

$$J^{*}(n,p) - \mathbb{E}\left[J^{*}(n-1,p_{0})\right] \geq J^{*}(n+1,p) - \mathbb{E}\left[J^{*}(n,p_{0})\right] .$$

■

LEMMA 10. *For any fixed n, $J^{*}(n,p) - r_{p}$ is non-increasing in p, and $J^{*}(n,p) - \bar{r}_{p}$ is non-decreasing in p.*

Proof: First, define operators T and Δ, respectively, as

$$\begin{aligned} TJ(n,p) = & -w(n) + \mu\left[\alpha p v + \beta(1-p)\bar{v}\right] + \mu\left[\alpha p + \beta(1-p)\right]\mathbb{E}\left[J^{*}(n-1,p_{0})\right] \\ & + \mu\left[(1-\alpha)p + (1-\beta)(1-p)\right]J\big(n,p_{+}\big) + \lambda J(n+1,p) . \end{aligned}$$

$$\begin{aligned} \Delta J(n,p) &= \max\Big\{TJ(n,p), r_{p} \vee \bar{r}_{p} + \mathbb{E}\left[J^{*}(n-1,p_{0})\right],\ R + J(n-1,p)\Big\}, \text{ for } n \geq 2, p \in [0,1] \\ \Delta J(1,p) &= \max\Big\{TJ(1,p), r_{p} \vee \bar{r}_{p} + \mathbb{E}\left[J^{*}(0,p_{0})\right]\Big\}, \text{ for } p \in [0,1] \\ \Delta J(0,p) &= \lambda \mathbb{E}\left[J(1,p_{0})\right]/(\lambda+\gamma) \text{ for } p \in [0,1] . \end{aligned}$$

Note that the only difference between the definition TJ and h_{J}^{g} defined in (10) is that we use $\mathbb{E}\left[J^{*}(n-1,p_{0})\right]$ in place of $\mathbb{E}\left[J(n-1,p_{0})\right]$ here.

Now, the existence and uniqueness of the optimal value function $J^{*}(.,.)$ corresponding to the above Bellman equations can be proved using steps exactly similar to the proof of Lemma 6. Further, $J^{*}(.,.)$ can be obtained by the value iteration algorithm starting from any arbitrary $J_{0}(.,.)$ so that

$$\lim_{k\to\infty} \Delta^{(k)} J_{0}(.,.) = J^{*}(.,.) .$$

Next, we show that operator Δ simultaneously propagates the following two properties. This, together with its convergence, completes the proof:

(i) $J(n,p) \leq \mathbb{E}\left[J^{*}(n-1,p_{0})\right] + pv + (1-p)\bar{v}$.

(ii) $J(n,p) - r_{p} \searrow p$.

Suppose function $J(.,.)$ satisfies properties (i) and (ii). We start by showing the propagation of property (i). For this, we show that function $\Delta J(\cdot,\cdot)$ satisfies the property as long as function $J(\cdot,\cdot)$ does. Note that, $J^*(n-2,p_0)+R \le J^*(n-1,p_0)$ for all possible values of p_0, and hence, $\mathbb{E}\left[J^*(n-2,p_0)\right]+R \le \mathbb{E}\left[J^*(n-1,p_0)\right]$. For $n \ge 2$,

$$J(n-1,p)+R \underbrace{\le}_{\text{property (i)}} pv+(1-p)\bar{v}+\mathbb{E}\left[J^*(n-2,p_0)\right]+R \le pv+(1-p)\bar{v}+\mathbb{E}\left[J^*(n-1,p_0)\right] .$$

Further, $r_p \le pv$, and $\bar{r}_p \le (1-p)\bar{v}$, and hence,

$$r_p \vee \bar{r}_p + \mathbb{E}\left[J^*(n-1,p_0)\right] \le pv+(1-p)\bar{v}+\mathbb{E}\left[J^*(n-1,p_0)\right] .$$

Therefore, to show $\Delta J(.,.)$ satisfies property (i), it suffices to show that $TJ(n,p) \le pv+(1-p)\bar{v}+\mathbb{E}\left[J^*(n-1,p_0)\right]$. For this, property (i) and simple algebra gives us

$$TJ(n,p) \le -w(n)+\lambda\mathbb{E}\left[J^*(n,p_0)\right]+(\lambda+\mu)\left[pv+(1-p)\bar{v}\right]+\mu\mathbb{E}\left[J^*(n-1,p_0)\right] . \tag{44}$$

From Lemma 9, we know that $J^*(n,p_0)-\mathbb{E}\left[J^*(n-1,p_0)\right] \le J^*(1,p_0)-J^*(0,p_0)$ for all possible values of p_0, and hence $\mathbb{E}\left[J^*(n,p_0)\right]-\mathbb{E}\left[J^*(n-1,p_0)\right] \le \mathbb{E}\left[J^*(1,p_0)\right]-J^*(0,p_0)$. Thus,

$$\lambda\Big(\mathbb{E}\left[J^*(n,p_0)\right]-\mathbb{E}\left[J^*(n-1,p_0)\right]\Big) \le \lambda\Big(\mathbb{E}\left[J^*(1,p_0)\right]-J^*(0,p_0)\Big)=\gamma J^*(0,p_0) \le \gamma\mathbb{E}\left[J^*(n-1,p_0)\right]$$

$\Leftrightarrow$

$$\lambda\mathbb{E}\left[J^*(n,p_0)\right] \le (\lambda+\gamma)\mathbb{E}\left[J^*(n-1,p_0)\right] .$$

The above inequality, together with (44), completes the proof for property (i).

We now prove the propagation of property (ii). From property (i), we have

$$-pv-(1-p)\bar{v} \le \mathbb{E}\left[J^*(n-1,p_0)\right]-J\big(n,p_+\big) .$$

The rest of the proof follows steps exactly similar to the proof of Lemma 4, and leads to

$$TJ(n,p_+)-r_{p_+} \le TJ(n,p)-r_p .$$

Finally, we know from property (ii) that $J(n-1,p_+)+R-r_{p_+} \le J(n-1,p)+R-r_p$. Further, $\big(r_{p_+} \vee \bar{r}_{p_+}\big)-r_{p_+} < \big(r_p \vee \bar{r}_p\big)-r_p$, and hence, $\mathbb{E}\left[J^*(n-1,p_0)\right]+\big(r_{p_+} \vee \bar{r}_{p_+}\big)-r_{p_+} < \mathbb{E}\left[J^*(n-1,p_0)\right]+\big(r_p \vee \bar{r}_p\big)-r_p$.

Putting all together, the three terms in $\Delta(\cdot,\cdot)$ satisfy property (ii) and thus,

$$\Delta J(n,p_+)-r_{p_+} \le \Delta J(n,p)-r_p .$$

The proof for $J^*(n,p)-\bar{r}_p \nearrow p$ follows similar steps, and therefore is omitted.

■

LEMMA 11. *For any fixed p, the optimal value function $J^*(n,p)$ is concave in n with*

$$J^*(n+1,p)+J^*(n-1,p)\leq 2J^*(n,p)\,.$$

Proof: The proof follows steps similar to the proof of Lemma 5. The main difference is that the four systems start from states $(n+1,p)$, $(n-1,p)$, (n,p) and (n,p), respectively, and the definition of sample path z now also includes the realization of the initial belief p_0 for different tasks (in addition to the realization of arrival times, test completion times, and test results). Further, all terms $r_0\vee\bar{r}_0$ in the proof of Lemma 5 are now simply replaced with R. More specifically, equation (38) is updated to

$$\underbrace{\Pi_I^z+e^{-\gamma T^z}R+e^{-\gamma T^z}J^*(n^z,p^z)}_{\text{System I's total payoff under path } z}+\underbrace{\Pi_{II}^z+e^{-\gamma T^z}J^*(n'^z-1,p^z)}_{\text{System II's total payoff under path } z}\leq$$
$$\underbrace{\Pi_{III}^z+e^{-\gamma T^z}J^*(n^z,p^z)}_{\text{System III's total payoff under path } z}+\underbrace{\Pi_{IV}^z+e^{-\gamma T^z}J^*(n'^z,p^z)}_{\text{System IV's total payoff under path } z}.$$

Similarly, equation (39) is now updated to

$$\underbrace{\Pi_I^z+e^{-\gamma T^z}J^*(n^z,p^z)}_{\text{System I's total payoff under path } z}+\underbrace{\Pi_{II}^z+e^{-\gamma T^z}\mathbb{E}\left[J^*(n-2,p_0)\right]}_{\text{System II's total payoff under path } z}\leq$$
$$\underbrace{\Pi_{III}^z+e^{-\gamma T^z}J^*(n^z-1,p^z)}_{\text{System III's total payoff under path } z}+\underbrace{\Pi_{IV}^z+e^{-\gamma T^z}\mathbb{E}\left[J^*(n-1,p_0)\right]}_{\text{System IV's total payoff under path } z}.$$

The rest of the proof is exactly similar to the proof of Lemma 5, and we get

$$J^*(n+1,p)+J^*(n-1,p)\leq 2J^*(n,p)\,.$$

■

LEMMA 12. *Define set $\mathcal{P}(n)$ as,*

$$\mathcal{P}(n)=\Big\{p\in[0,1]:\ J^*(n,p)>\mathbb{E}\left[J^*(n-1,p_0)\right]+r_p\vee\bar{r}_p \text{ and } J^*(n,p)>h^g_{J^*}(n,p)\Big\}\,.$$

Two thresholds $\check{p}(n)$ and $\hat{p}(n)$ exist such that $p\in\mathcal{P}(n)$ if and only if $\check{p}(n)<p<\hat{p}(n)$. Furthermore, $\check{p}(n)$ is non-increasing and $\hat{p}(n)$ non-decreasing in n.

Proof: The proof proceeds by induction on the workload level n, and follows steps similar to the proof of Proposition 4. The main difference is that $J^*(\cdot,0)$ and $r_0\vee\bar{r}_0$ in the proof of Proposition 4 should be replaced with $\mathbb{E}\left[J^*(\cdot,p_0)\right]$ and R, respectively.

■

Proof of Theorem 2

Define $\bar{p}$ as the smallest value of p for which releasing the task in search and choosing option s is optimal at any workload level n. Lemma 9 implies that if it is optimal to stop searching a focal task in state (n,p), the same decision must also be optimal in state $(n+1,p)$. Similarly, Lemma 11 implies that if it is optimal to dismiss a task from the queue in state (n,p), the same decision must also be optimal in state $(n+1,p)$. Putting these two results together, we conclude that if it is optimal to continue searching in state (n,p), the same decision must also be optimal in state $(n-1,p)$. Consequently, threshold $\hat{n}(p)$ exists and is defined as the smallest value of n for which continuing the search at state (n,p) is not optimal.

Next, Lemmas 9 and 11 ensure that for any $n \geq \hat{n}(p)$, the optimality between releasing a waiting task and aborting the current search is determined by the same threshold $\bar{p}$. Further, $\bar{p} \geq \theta$ follows from the fact that $r_{\bar{p}}$ needs to be no less than $\bar{r}_{\bar{p}}$ for all $p < \theta$.

Finally, Lemma 10 implies that stopping to choose option $\bar{s}$ is never optimal. Therefore, for $p < \bar{p}$ and $n \geq \hat{n}(p)$, dismissing a task from the queue is the only decision that can be optimal.

The existence of threshold $\underline{p}$ immediately follows from Lemma 12, which also implies that $\hat{n}(p)$ is non-increasing for $p < \underline{p}$, and non-decreasing for $\underline{p} \leq p < \bar{p}$. The monotonicity of $\hat{n}(p)$ for $p \geq \bar{p}$ follows from Lemma 10, because if it is optimal to release the current task with option s when the system is at state (n,p), the same decision must also be optimal in state (n,p_+).

■

To proceed with the proof of Theorem 3, we consider the model in Section 6.2, and present the following lemmas first.

LEMMA 13. *Consider the following Bellman equations*

$$J(n,k) = \max\Big\{\max\big\{r_k \vee \bar{r}_k + J(n-1,0) - \kappa,\ r_0 \vee \bar{r}_0 + J(n-1,k)\big\},\ h_J^\kappa(n,k)\Big\},\quad \text{for } n > 1, k \geq 0, \tag{45}$$

$$J(1,k) = \max\Big\{r_k \vee \bar{r}_k + J(0,0),\ h_J^\kappa(1,k)\Big\}, \tag{46}$$

$$J(0,0) = \lambda\,[J(1,0) - \kappa]\,/(\lambda+\gamma), \tag{47}$$

where the value for collecting an additional observation, $h_J^\kappa(n,k)$, is equal to

$$h_J^\kappa(n,k) = -\,w(n) + \mu\,[\alpha p_k v + \beta(1-p_k)\bar{v}] + \lambda J(n+1,k) + \mu c_k\,[J(n-1,0) - \kappa \mathbb{I}_{n>1}] + \mu\,(1-c_k)\,J\big(n,k+1\big). \tag{48}$$

For any function $J(n,k)$ with $n \geq 0$ and $k \geq 0$ that satisfy (45)-(48)*, define*

$$\hat{J}(n,k) = J(n,k), \text{ for } n \geq 1, k \geq 0 \quad (49)$$

$$\hat{J}(n,-1) = J(n,0) - \kappa, \quad (50)$$

$$\hat{J}(0,-1) = J(0,0). \quad (51)$$

Then the function $\hat{J}$ solves the Bellman equations (11)-(15)*.*

Proof: First, in order to establish (11), we have

$$\hat{J}(n,-1) = J(n,0) - \kappa = \hat{J}(n,0) - \kappa$$

where the first equality follows from (50) and the second from (49), and

$$J(n,0) - \kappa \geq r_0 \vee \bar{r}_0 + J(n-1,0) - \kappa = r_0 \vee \bar{r}_0 + \hat{J}(n-1,0) - \kappa,$$

following (45) with $k = 0$ and (49).

Next, it is easy to verify that $h^f_{\hat{J}}(n,k) = h^{\kappa}_J(n,k)$ following (49)-(47). Then, again, (49)-(47) imply that

$$\begin{aligned}
\hat{J}(n,k) &= J(n,k) \\
&= \max\Big\{ \max\big\{ r_k \vee \bar{r}_k + J(n-1,0) - \kappa,\ r_0 \vee \bar{r}_0 + J(n-1,k) \big\}, h^{\kappa}_J(n,k) \Big\} \\
&= \max\Big\{ \max\big\{ r_k \vee \bar{r}_k + \hat{J}(n-1,-1),\ r_0 \vee \bar{r}_0 + \hat{J}(n-1,k) \big\},\ h^f_{\hat{J}}(n,k) \Big\},
\end{aligned}$$

which establishes (12).

Equations (13) and (14) again follow (49)-(47) directly.

■

Following Lemma 13, it suffices to establish the result for the optimal value function corresponding to the Bellman equations (45)-(48).

Lemma 14. *A function $J^*(\cdot,\cdot)$ exists such that $J^*(\cdot,\cdot)$ satisfies the Bellman equations* (45)-(48) *and $J^*(n,k)$ is the optimal value function given initial state (n,k).*

Proof: The result follows from Proposition 3.1.6 of Bertsekas (2007), because ours is a maximization problem where single period rewards $-w(n)$, v, $\bar{v}$, κ, r_k and $\bar{r}_k$ are bounded from above for all values of n and k.

■

Lemma 15. *For any fixed k,*

(i) $J^*(n,k) - J^*(n-1,0)$ is non-increasing in n, and

(ii) $J^*(n,k)$ *is concave in* n *so that* $J^*(n+1,k)+J^*(n-1,k)\le 2J^*(n,k)$.

Proof: The proof for the first part follows steps exactly similar to the proof of Lemma 3, with the only difference that the setup cost κ is also incurred in both systems when the search on a new task starts. Similarly, the proof for the second part follows steps exactly similar to the proof of Lemma 5, with the only difference that the setup cost κ is also incurred in all four systems when the search on a new task starts.

■

LEMMA 16. *For any fixed* n, $J^*(n,k)-r_k$ *is non-increasing in* k, *and* $J^*(n,k)-\bar{r}_k$ *is non-decreasing in* k.

Proof: First, define operators T and Δ, respectively, as

$$\begin{aligned} TJ(n,k) &= -w(n)+\mu\left[\alpha p_k v+\beta(1-p_k)\bar{v}\right]+\mu\left[\alpha p_k+\beta(1-p_k)\right]\left[J^*(n-1,p_0)-\kappa\mathbb{I}_{n>1}\right] \\ &\quad +\mu\left[(1-\alpha)p_k+(1-\beta)(1-p_k)\right]J\big(n,k+1\big)+\lambda J(n+1,k)\ . \end{aligned}$$

$$\begin{aligned} \Delta J(n,k) &= \max\Big\{TJ(n,k), r_k\vee\bar{r}_k+J^*(n-1,0)-\kappa,\ r_0\vee\bar{r}_0+J(n-1,k)\Big\},\ \text{for } n\ge 2, k\ge 0 \\ \Delta J(1,k) &= \max\Big\{TJ(1,k), r_k\vee\bar{r}_k+J^*(0,0)\Big\},\ \text{for } k\ge 0 \\ \Delta J(0,0) &= \lambda\left[J(1,0)-\kappa\right]/(\lambda+\gamma)\,. \end{aligned}$$

Now, the existence and uniqueness of the optimal value function $J^*(.,.)$ corresponding to the above Bellman equations can be proved using steps exactly similar to the proof of Lemma 6. Further, $J^*(.,.)$ can be obtained by the value iteration algorithm starting from any arbitrary $J_0(.,.)$ so that

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

Next, we show that operator Δ simultaneously propagates the following two properties. This, together with its convergence, completes the proof:

(i) $J(n,k)\le J^*(n-1,p_0)+p_k v+(1-p_k)\bar{v}-\kappa$.

(ii) $J(n,k)-r_k\searrow k$.

Suppose function $J(.,.)$ satisfies properties (i) and (ii). We start by showing the propagation of property (i). For this, we show that function $\Delta J(\cdot,\cdot)$ satisfies the property as long as function $J(\cdot,\cdot)$ does. The rest of the proof follows steps exactly similar to the proof of Lemma 4 and we derive the following inequality in parallel to (37):

$$\begin{aligned} TJ(n,k) &\le -w(n)+(\lambda+\gamma)J^*(n-1,0)+(\lambda+\mu)\left[p_k v+(1-p_k)\bar{v}-\kappa\right]+\mu J^*(n-1,0) \\ &\le J^*(n-1,0)+p_k v+(1-p_k)\bar{v}-\kappa\,, \end{aligned}$$

where the last inequality follows from the fact that $\kappa \le \min\{v, \bar{v}\}$. This completes the proof for the propagation of property (i). The proof for propagation of property (ii) is similar to the proof of Lemma 4, and the details are ommited.

■

LEMMA 17. *Define set* $\mathcal{K}(n)$ *as,*

$$\mathcal{K}(n) = \left\{ k \ge 0 : \; J^*(n,k) > J^*(n-1,0) + r_k \vee \bar{r}_k - \kappa \text{ and } J^*(n,k) > h^{\kappa}_{J^*}(n,k) \right\}.$$

Two thresholds $\check{k}(n)$ *and* $\hat{k}(n)$ *exist such that* $k \in \mathcal{K}(n)$ *if and only if* $\check{k}(n) < k < \hat{k}(n)$*. Furthermore,* $\check{k}(n)$ *is non-increasing and* $\hat{k}(n)$ *non-decreasing in* n*.*

Proof: The proof proceeds by induction on the workload level n, and follows steps similar to the proof of Proposition 4.

■

Proof of Theorem 3

Define $\bar{k}$ as the smallest value of k for which releasing the task in search and choosing option s is optimal at any workload level n. The first part of Lemma 15 implies that if it is optimal to stop searching a focal task in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Similarly, the second part of the same lemma implies that if it is optimal to dismiss a task from the queue in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Putting these two results together, we conclude that if it is optimal to continue searching in state (n,k), the same decision must also be optimal in state $(n-1,k)$. Consequently, threshold $\hat{n}(k)$ exists and is defined as the smallest value of n for which continuing the search at state (n,k) is not optimal.

Next, Lemmas 15 ensures that for any $n \ge \hat{n}(k)$, the optimality between releasing a waiting task and aborting the current search is determined by the same threshold $\bar{k}$. Further, $\bar{k} \ge k_\theta$ follows from the fact that $r_{\bar{k}}$ needs to be no less than $\bar{r}_{\bar{k}}$ for all $k < k_\theta$.

Finally, Lemma 16 implies that stopping to choose option $\bar{s}$ is never optimal. Therefore, for $k < \bar{k}$ and $n \ge \hat{n}(k)$, dismissing a task from the queue is the only decision that can be optimal.

The existence of threshold $\underline{k}$ immediately follows from Lemma 17, which also implies that $\hat{n}(k)$ is non-increasing for $k < \underline{k}$, and non-decreasing for $\underline{k} \le k < \bar{k}$. The monotonicity of $\hat{n}(k)$ for $k \ge \bar{k}$ follows from Lemma 16, because if it is optimal to release the current task with option s when the system is at state (n,k), the same decision must also be optimal in state $(n,k+1)$.

■

To proceed with the proof of Theorem 4, we consider the model in Section 6.3, and present the following lemmas first.

LEMMA 18. *A function $J^*(\cdot,\cdot)$ exists such that $J^*(\cdot,\cdot)$ satisfies the Bellman equations* (17)-(19) *and $J^*(n,d)$ is the optimal value function given initial state (n,d).*

Proof: The result follows from Proposition 3.1.6 of Bertsekas (2007), because ours is a maximization problem where single period rewards $-w(n)$, r_d and $\bar{r}_d$ are bounded from above for all values of n and d.

■

LEMMA 19. *For any fixed d,*

(i) $J^*(n,d) - J^*(n-1,0)$ is non-increasing in n, and

(ii) $J^*(n,d)$ is concave in n so that $J^*(n+1,d) + J^*(n-1,d) \leq 2J^*(n,d)$.

Proof: The proof for the first part follows steps exactly similar to the proof of Lemma 3, with the only difference that the two systems start from states $(n+1,d)$ and (n,d), respectively. Then, equation (34) is updated to

$$\underbrace{\Pi^z_{II} + e^{-\gamma T^z} J^*(n,d^z)}_{\text{System II's total payoff under path } z} \quad -e^{-\gamma T^z} J^*(n-1,0) \geq$$
$$\underbrace{\Pi^z_1 + e^{-\gamma T^z}(r_0 \vee \bar{r}_0) + e^{-\gamma T^z} J^*(n,d^z)}_{\text{System I's total payoff under path } z} - e^{-\gamma T^z} J^*(n,0)\,.$$

Similarly, equation (35) is now updated to

$$\underbrace{\Pi^z_{II} + e^{-\gamma T^z} J^*(n-1,0)}_{\text{System II's total payoff under path } z} \quad -e^{-\gamma T^z} J^*(n-1,0) \geq$$
$$\underbrace{\Pi^z_I + e^{-\gamma T^z} J^*(n,0)}_{\text{System I's total payoff under path } z} \quad -e^{-\gamma T^z} J^*(n,0)\,.$$

The rest of the proof is exactly similar to the proof of Lemma 3, and we get

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

For the second part, the proof follows steps similar to the proof of Lemma 5. The main difference is that the four systems start from states $(n+1,d)$, $(n-1,d)$, (n,d) and (n,d), respectively. Then, equation (38) is updated to

$$\underbrace{\Pi^z_I + e^{-\gamma T^z}(r_0 \vee \bar{r}_0) + e^{-\gamma T^z} J^*(n^z,d^z)}_{\text{System I's total payoff under path } z} + \underbrace{\Pi^z_{II} + e^{-\gamma T^z} J^*(n'^z - 1,d^z)}_{\text{System II's total payoff under path } z} \leq$$
$$\underbrace{\Pi^z_{III} + e^{-\gamma T^z} J^*(n^z,d^z)}_{\text{System III's total payoff under path } z} + \underbrace{\Pi^z_{IV} + e^{-\gamma T^z} J^*(n'^z,d^z)}_{\text{System IV's total payoff under path } z}\,.$$

Similarly, equation (39) is now updated to

$$\underbrace{\Pi_I^z + e^{-\gamma T^z} J^*(n^z, d^z)}_{\text{System I's total payoff under path } z} + \underbrace{\Pi_{II}^z + e^{-\gamma T^z} J^*(n-2, 0)}_{\text{System II's total payoff under path } z} \leq$$
$$\underbrace{\Pi_{III}^z + e^{-\gamma T^z} J^*(n^z - 1, d^z)}_{\text{System III's total payoff under path } z} + \underbrace{\Pi_{IV}^z + e^{-\gamma T^z} J^*(n-1, 0)}_{\text{System IV's total payoff under path } z} .$$

The rest of the proof is exactly similar to the proof of Lemma 5, and we get

$$J^*(n+1, d) + J^*(n-1, d) \leq 2J^*(n, d) .$$

■

LEMMA 20. *For the optimal value function, $J^*(.,.)$, and any fixed n we have*

$$J^*(n, d) - \bar{r}_d \nearrow d \text{ and } J^*(n, d) - r_d \searrow d .$$

Proof: First, define operators T and Δ, respectively, as

$$\begin{aligned} TJ(n,d) &= -w(n) + \lambda J(n+1, d) + \mu\left[(1-\beta)p_d + \beta(1-p_d)\right] J^*(n, d-1) \\ &\quad + \mu\left[\beta p_d + (1-\beta)(1-p_d)\right] J(n, d+1) \ . \end{aligned}$$

$$\begin{aligned} \Delta J(n,d) &= \max\Big\{TJ(n,d), r_d \vee \bar{r}_d + J^*(n-1, 0),\ r_0 \vee \bar{r}_0 + J(n-1, d)\Big\}, \text{ for } n \geq 2 \\ \Delta J(1,d) &= \max\Big\{TJ(1,d), r_d \vee \bar{r}_d + J^*(0,0)\Big\}, \\ \Delta J(0,0) &= \lambda J(1,0)/(\lambda+\gamma) \ . \end{aligned}$$

Now, the existence and uniqueness of the optimal value function $J^*(.,.)$ corresponding to the above Bellman equations can be proved using steps exactly similar to the proof of Lemma 18. Further, $J^*(.,.)$ can be obtained by the value iteration algorithm starting from any arbitrary $J_0(.,.)$ so that

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

Next, we show that operator Δ simultaneously propagates the following two properties. This, together with its convergence, completes the proof:

(i) $J(n,d) - \bar{r}_d \nearrow d$.

(ii) $J(n,d) - r_d \searrow d$.

To show the propagation of property (i), define $p^+ = \beta p + (1-\beta)(1-p)$ and $p^- = (1-\beta)p + \beta(1-p)$ for $p \in [0,1]$. We start from the following equality (simple algebra shows that this equality holds):

$$\bar{r}_d - p_d^+ \bar{r}_{d+1} - p_d^- \bar{r}_{d-1} = \bar{r}_{d-1} - p_{d-1}^+ \bar{r}_d - p_{d-1}^- \bar{r}_{d-2} \ .$$

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

$$\left(p_d^+ - p_{d-1}^+\right)\left(\bar{r}_{d-1} - \bar{r}_d\right) = p_d^+\left(\bar{r}_{d+1} - \bar{r}_d\right) - \bar{r}_d + \bar{r}_{d-1} + (1-p_{d-1}^+)\left(\bar{r}_{d-1} - \bar{r}_{d-2}\right).$$

From property (i) we have

$$\left(p_d^+ - p_{d-1}^+\right)\left(J(n,d-1) - J(n,d)\right) \le p_d^+\left(\bar{r}_{d+1} - \bar{r}_d\right) - \bar{r}_d + \bar{r}_{d-1} + (1-p_{d-1}^+)\left(\bar{r}_{d-1} - \bar{r}_{d-2}\right).$$

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

$$\begin{aligned} p_{d-1}^+ J(n,d) + (1-p_{d-1}^+)\left(J(n,d-1) - \bar{r}_{d-1} + \bar{r}_{d-2}\right) - \bar{r}_{d-1} \le \\ p_d^+\left(J(n,d) + \bar{r}_{d+1} - \bar{r}_d\right) + (1-p_d^+)J(n,d-1) - \bar{r}_d\,. \end{aligned}$$

Applying property (i) again results in

$$p_{d-1}^+ J(n,d) + (1-p_{d-1}^+)J(n,d-2) - \bar{r}_{d-1} \le p_d^+ J(n,d+1) + (1-p_d^+)J(n,d-1) - \bar{r}_d\,.$$

Multiplying both sides by μ and another use of property (i) entails,

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

We next add the inequality $-\gamma\bar{r}_{d-1} \le -\gamma\bar{r}_d$ to the above inequality to get

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

Another use of property (i) and noting that the maximum of a set of increasing functions is still increasing implies that,

$$\Delta J(n,d-1) - \bar{r}_{d-1} \le \Delta J(n,d) - \bar{r}_d$$

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

To show the propagation of property (ii), we start from the following equality:

$$r_d - p_d^+ r_{d+1} - p_d^- r_{d-1} = r_{d-1} - p_{d-1}^+ r_d - p_{d-1}^- r_{d-2}\,.$$

Following steps very similar to those in the proof for property (i) above, we deduce that,

$$\Delta J(n,d-1) - r_{d-1} \ge \Delta J(n,d) - r_d$$

$$\Rightarrow \Delta J(n,d) - r_d \searrow d\,.$$

■

LEMMA 21. *Define set $\mathcal{D}(n)$ as,*

$$\mathcal{D}(n) = \left\{ d: \; J^*(n,d) > J^*(n-1,0) + r_d \vee \bar{r}_d \text{ and } J^*(n,d) > h^d_{J^*}(n,d) \right\}.$$

Two thresholds $\check{d}(n)$ and $\hat{d}(n)$ exist such that $d \in \mathcal{D}(n)$ if and only if $\check{d}(n) < d < \hat{d}(n)$. Furthermore, $\check{d}(n)$ is non-increasing and $\hat{d}(n)$ non-decreasing in n.

Proof: First, we denote $\phi = v \vee \bar{v}$, and argue that for large enough n, set $\mathcal{D}(n)$ is connected. We have

$$\begin{aligned} h^d_{J^*}(n,d) =& -w(n) + \lambda J(n+1,d) + \mu\left[(1-\beta)p_d + \beta(1-p_d)\right] J^*(n,d-1) \\ &+ \mu\left[\beta p_d + (1-\beta)(1-p_d)\right] J(n,d+1) \\ \le& -w(n) + 2\lambda\phi + J^*(n-1,0) + \mu\left[(1-\beta)p_d + \beta(1-p_d)\right](J^*(n-1,0)+\phi) \\ &\mu\left[\beta p_d + (1-\beta)(1-p_d)\right](J(n-1,0)+\phi) \\ \le& -w(n) + 2(\lambda+\mu)\phi + (\lambda+\mu)J^*(n-1,0)\,. \end{aligned}$$

It follows that

$$h^d_{J^*}(n,d) - \left[J^*(n-1,0) + r_d \vee \bar{r}_d\right] \le \left[-w(n) + 2(\lambda+\mu)\phi - \gamma J^*(n-1,0)\right] - r_d \vee \bar{r}_d\,.$$

Since $r_0 \vee \bar{r}_0 \ge 0$, we have $J^*(n-1,0) \ge 0$. Also, $w(n)$ is weakly convex in n and approaches infinity with n. Therefore, the right hand side of the above inequality becomes strictly negative for sufficiently large values of n, and hence, the continuation decision cannot be optimal. Let $\bar{n}$ represent the smallest value of n for which $J^*(n,d) > h^d_{J^*}(n,d)$ for all d. Next, suppose $d_1 \in \mathcal{D}(n)$ and $d_2 \in \mathcal{D}(n)$ for $d_1 < d_2$ and $n \ge \bar{n}$. We want to show that $d \in \mathcal{D}(n)$ for all $d_1 < d < d_2$, and hence $\mathcal{D}(n)$ is connected. Otherwise, if $d \notin \mathcal{D}(n)$, we must have $J^*(n,d) = J^*(n-1,0) + r_d$ or $J^*(n,d) = J^*(n-1,0) + \bar{r}_d$, which further implies $J^*(n,d_1) = J^*(n-1,0) + r_{d_1}$ or $J^*(n,d_2) = J^*(n-1,0) + \bar{r}_{d_2}$, respectively, following Lemma 20. This, however, contradicts either $d_1 \in \mathcal{D}(n)$ or $d_2 \in \mathcal{D}(n)$. Therefore, we must have $d \in \mathcal{D}(n)$.

Now consider values n, d_1 and d_2 such that $\mathcal{D}(n+1)$ is connected, $d_1 < d_2$ and $d_1 \in \mathcal{D}(n)$ and $d_2 \in \mathcal{D}(n)$. First, note $J^*(n,d) > J^*(n-1,0) + r_d \vee \bar{r}_d$ for all $d \in [d_1, d_2]$, because otherwise Lemma 20 is violated. Therefore, it suffices to show that the optimal decision in state (n,d) with $d \in [d_1,d_2]$ in not to continue the search. To show this, we use contradiction. In particular, suppose the system is in state (n,d) and the optimal decision is to continue the search. Now, consider all possible sample paths that can unfold over time. We will either:

(i) have a new arrival before entering either states (n, d_1) or (n, d_2) in which case an accumulated task must be released from the queue (since $\mathcal{D}(n+1)$ is connected), or

(ii) enter one of the states (n, d_1) or (n, d_2) in which case an accumulated task must be released from the queue (since $d_1, d_2 \in \mathcal{D}(n)$) .

Hence, cases (i) and (ii) together imply that under any sample path, an accumulated task must be released from the queue before the termination of search on the current task. The DM, however, would be better off to release an accumulated task from the queue at the outset and thereby incurring a lower waiting cost. This contradicts the optimality of the continuation decision in state (n, d). The optimal decision in this state, therefore, must be to release a task from the queue, which establishes the result that $\mathcal{D}(n)$ is conncected.

The monotonicity of $\check{d}(n)$ and $\hat{d}(n)$ follows directly from the definition of threshold $\tilde{n}(d)$ implied by Lemma 19.

■

Proof of Theorem 4

Define $\bar{d}$ as the smallest value of d for which releasing the task in search and choosing option s is optimal at any workload level n. Similarly, define $\underline{d}$ as the largest value of d for which releasing the task in search and choosing option $\bar{s}$ is optimal at any workload level n. The first part of Lemma 19 implies that if it is optimal to stop searching a focal task in state (n, d), the same decision must also be optimal in state $(n+1, d)$. Similarly, the second part of the same lemma implies that if it is optimal to dismiss a task from the queue in state (n, d), the same decision must also be optimal in state $(n+1, d)$. Putting these two results together, we conclude that if it is optimal to continue searching in state (n, d), the same decision must also be optimal in state $(n-1, d)$. Consequently, threshold $\hat{n}(d)$ exists and is defined as the smallest value of n for which continuing the search at state (n, d) is not optimal.

Next, Lemmas 19 ensures that for any $n \geq \hat{n}(d)$, the optimality between releasing a waiting task and aborting the current search is determined by the same thresholds $\bar{d}$ and $\underline{d}$. Further, $\bar{d} \geq d_\theta$ follows from the fact that $r_{\bar{d}}$ needs to be no less than $\bar{r}_{\bar{d}}$ for all $d < d_\theta$. Similarly, $\underline{d} \leq d_\theta$ follows from the fact that $r_{\underline{d}}$ needs to be no more than $\bar{r}_{\underline{d}}$ for all $d > d_\theta$.

From Lemma 20, if $J^*(n, d) - \bar{r}_d = J^*(n-1, 0)$ then $J^*(n, d-1) - \bar{r}_{d-1} = J^*(n-1, 0)$. 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 $(n, d-1)$. Similarly, Lemma 20 implies that if it is optimal to stop the search at state (n, d) and identify the current task in favor of s, the same action is also optimal at state $(n, d+1)$. Therefore, for $\underline{d} < d < \bar{d}$ and $n \geq \hat{n}(d)$, dismissing a task from the queue is the only decision that can be optimal.

Finally, Lemma 21, implies that $\hat{n}(d)$ is non-increasing for $d \in (\underline{d}, \check{d}]$ and non-decreasing for $d \in [\check{d}, \bar{d})$. Similarly, Lemma 20 implies the monotonicity of $\hat{n}(d)$ for $d \geq \bar{d}$ and $d \leq \underline{d}$.

■

To proceed with the proof of Theorem 5, we consider the model of Section 6.5 and present the following lemmas first.

LEMMA 22. *A function* $J^*(\cdot,\cdot)$ *exists such that* $J^*(\cdot,\cdot)$ *satisfies the Bellman equations* (30)-(32) *and* $J^*(n,k)$ *is the optimal value function given initial state* (n,k)*.*

Proof: The result follows from Proposition 3.1.6 of Bertsekas (2007), because ours is a maximization problem where single period rewards $-w(n)$ and v are bounded from above for all values of n and k.

■

LEMMA 23. *For any fixed* k*,*

(i) $J^*(n,k) - J^*(n-1,0)$ *is non-increasing in* n*, and*

(ii) $J^*(n,k)$ *is concave in* n *so that* $J^*(n+1,k) + J^*(n-1,k) \le 2J^*(n,k)$*.*

Proof: The first part of this lemma is a special case of Lemma 3, and the second part is a special case of Lemma 5. Therefore, the proof immediately follows.

■

Proof of Theorem 5

Define $\hat{k}$ as the smallest value of k for which releasing the task in search is optimal at any workload level n. The first part of Lemma 23 implies that if it is optimal to stop searching a focal task in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Similarly, the second part of the same lemma implies that if it is optimal to dismiss a task from the queue in state (n,k), the same decision must also be optimal in state $(n+1,k)$. Putting these two results together, we conclude that if it is optimal to continue searching in state (n,k), the same decision must also be optimal in state $(n-1,k)$. Consequently, threshold $\hat{n}(k)$ exists and is defined as the smallest value of n for which continuing the search at state (n,k) is not optimal.

Next, Lemmas 23 ensures that for any $n \ge \hat{n}(k)$, the optimality between releasing a waiting task and aborting the current search is determined by the same threshold $\hat{k}$.

■