Attention Allocation for Decision Making Queues[☆]

Vaibhav Srivastava[a], Ruggero Carli[b], Cédric Langbort[c], and Francesco Bullo[a]

[a] *Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, USA,* {vaibhav,bullo}@engineering.ucsb.edu
[b] *Department of Information Engineering, University of Padova , Italy,* carlirug@dei.unipd.it
[c] *Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA,* langbort@illinois.edu

Abstract

We consider the optimal servicing of a queue with sigmoid server performance. There are various systems with sigmoid server performance including systems involving human decision making, visual perception, human-machine communication and advertising response. The tasks arrive at the server according to some stochastic process. Each task has a deadline that is incorporated as a latency penalty. We investigate the trade-off between the reward obtained by processing the current task and the penalty incurred due to the tasks waiting in the queue. We study this optimization problem in a Markov decision process (MDP) framework. We characterize the properties of the optimal policy for the MDP and show that the optimal policy may drop some tasks, that is, may not process a task at all. We determine an approximate solution to the MDP using certainty-equivalent receding horizon optimization framework and determine performance bounds on the proposed receding horizon policy. We also suggest guidelines for the design of such queues.

Keywords: sigmoid utility, human decision making, control of queues, human-robot interaction

1. Introduction

The recent national robotic initiative [2] motivates research and applications emphasizing the interaction of human with symbiotic co-robot partners. Such co-robots will facilitate better interaction between the human partner and the automaton. In complex and information rich environments, one of the key roles for these co-robots is to help the human partner efficiently focus their attention. A particular example of such a setting is the surveillance mission, where the human operator monitors the evidence collected by the autonomous agents [3, 4]. The excessive amount of information available in such systems often results in poor decisions by the human operator [5]. This emphasizes the need for the development of a support system that helps the human operator optimally focus their attention.

Recently, there has been a significant interest in understanding the physics of human decision making [6]. Several mathematical models for human decision making have been proposed [6, 7, 8]. These models suggest that the correctness of the decision of a human operator in a binary decision making scenario evolves as a sigmoid function of the time-duration allocated for the decision. When a human operator has to serve a queue of decision making tasks *in real time,* the tasks (e.g., feeds from a camera network) waiting in the queue lose value continuously. This trade-off between the correctness of the decision and the loss in the value of the pending tasks is of critical importance for the performance of the human operator. In this paper, we address this trade-off, and determine the optimal duration allocation policies for the human operator serving a decision making queue. The sigmoid function also models the quality of human-machine communication [8], the human performance in multiple target search [9], the advertising response function [10], and the expected profit in simultaneous bidding [11]. Therefore, the analysis presented in this paper can also be used to determine optimal human-machine communication policies, optimal search strategies, the optimal advertisement duration allocation, and optimal bidding strategies. In this paper, we generically refer to the server with sigmoid performance as a human operator and the tasks as the decision making tasks.

There has been a significant interest in the study of the performance of a human operator serving a queue. In an early work, Schmidt [12] models the human as a server and numerically studies a queueing model to determine the performance of a human air traffic controller. Recently, Savla et al [13] study human supervisory control for unmanned aerial vehicle operations: they model the system by a simple queuing network with two components in series, the first of which is a spatial queue with vehicles as servers and the second is a conventional queue with human operators as servers. They design joint motion coordination and operator scheduling policies that minimize the expected time needed to classify a target after its appearance. The performance of the human operator based on their utilization history has been incorporated to design maximally stabilizing task release policies for a human-in-the-loop queue in [14, 15]. Bertuccelli et al [16] study the human supervisory control as a queue with re-look tasks. They study the policies in which the operator can put the tasks in an orbiting

[☆] This work has been supported in part by AFOSR MURI Award FA9550-07-1-0528. A preliminary version of this work [1] entitled ”Task release control for decision making queues” was presented at American Control Conference, 2011, San Francisco, CA.

queue for a re-look later. An optimal scheduling problem in the human supervisory control in studied in [17]. Crandall et al [18] study optimal scheduling policy for the operator and discuss if the operator or the automation should be ultimately responsible for selecting the task. Powel et al [19] model mixed team of humans and robots as a multi-server queue and incorporate a human fatigue model to determine the performance of the team. They present a comparative study of the fixed and the rolling work-shifts of the operators.

The optimal control of queueing systems [20] is a classical problem in queueing theory. There has been significant interest in the dynamic control of queues; e.g., see [21] and references therein. In particular, Stidham et al [21] study the optimal servicing policies for an M/G/1 queue of identical tasks. They formulate a semi-Markov decision process, and describe the qualitative features of the solution under certain technical assumptions. In the context of M/M/1 queues, George et al [22] and Adusumilli et al [23] relax some of technical assumptions in [21]. Hernández-Lerma et al [24] determine optimal servicing policies for the identical tasks and some arrival rate. They adapt the optimal policy as the arrival rate is learned. In another related work, Zafer et al [25] study static queue with monomial and exponential utilities. They approximate the problem with a continuous time MDP. In the case of the dynamic queue, they propose a heuristic that solves the static problem at each stage.

In this paper, we study the problem of optimal time-duration allocation in a queue of binary decision making tasks with a human operator. We refer to such queues as *decision making queues*. In contrast to the aforementioned works in queues with human operator, we do not assume that the tasks require a fixed (potentially stochastic) processing time. We consider that each task may be processed for any amount of time, and the performance on the task is known as a function of processing time. Moreover, we assume that tasks come with processing deadlines and incorporate these deadlines as a soft constraint, namely, latency penalty (penalty due to delay in processing of a task). We consider two particular problems. First, we consider a static queue with latency penalty. Here, the human operator has to serve a given number of tasks. The operator incurs a penalty due to the delay in processing of each task. This penalty can be thought of as the loss in value of the task over time. Second, we consider a dynamic queue of the decision making tasks. The tasks arrive according to a stochastic process and the operator incurs a penalty for the delay in processing each task. In both the problems, there is a trade-off between the reward obtained by processing a task and the penalty incurred due to the resulting delay in processing other tasks. We address this particular trade-off. The problem considered in this paper is similar to the problem considered in [21, 22, 23]. The main differences between these works and the problem considered in this paper are: (i) we consider a deterministic service process, and this yields an optimality equation quite different from the optimality equation obtained for Markovian service process; (ii) we consider heterogeneous tasks while the aforementioned works consider identical tasks. These works either propose approximate solution strategies customized to their setup, or rely on standard methods, e.g., value iteration in case of finite action space. In our problem, the heterogeneous nature of tasks significantly increases the dimension of the state space and makes the computation of optimal policies computationally intractable. We resolve this issue by utilizing certainty-equivalent receding horizon framework [26, 27, 28] to approximately compute the solution.

The major contributions of this work are fourfold. First, we determine the optimal duration allocation policy for the static decision making queue with latency penalty. We show that the optimal policy may not process all the tasks in the queue and may drop a few tasks. Second, we pose a Markov decision process (MDP) to determine the optimal allocations for the dynamic decision making queue. We then establish some properties of this MDP. In particular, we show an optimal policy exists and it drops task if queue length is greater than a critical value. Third, we employ certainty-equivalent receding horizon optimization to approximately solve this MDP. We establish performance bounds on the certainty-equivalent receding horizon solution. Fourth and finally, we suggest guidelines for the design of decision making queues. These guidelines suggest the maximum expected arrival rate at which the operator expects a new task to arrive soon after optimally processing the current task.

The remainder of the paper is organized as follows. We present some preliminaries and the problem setup in Section 2. The static queue with latency penalty is considered in Section 3. We pose the optimization problem associated with the dynamic queue with latency penalty and study its properties in Section 4. We present and analyze receding horizon algorithm for dynamic queue with latency penalty in Section 5. Our conclusions are presented in Section 6.

2. Preliminaries and Problem setup

We consider the problem of optimal time duration allocation for a human operator. The decision making tasks arrive at a given rate and are stacked in a queue. A human operator processes these tasks on the *first-come first-serve* basis (see Figure 3.) The human operator receives a unit reward for the correct decision, while there is no penalty for a wrong decision. We assume that the tasks can be parametrized by some variable and the variable takes value in a finite set $\mathcal{D} \subseteq \mathbb{R}$. Let the performance of the operator on a task with parameter $d \in \mathcal{D}$ be a function $f_d : \mathbb{R}_{\geq 0} \to [0, 1)$ of the duration operator allocates to the task. A performance function relevant to the discussion in this paper is the probability of making the correct decision. The evolution of the probability of correct decision by a human operator has been studied in cognitive psychology literature [7, 6]. We now briefly review some human decision making models:

Pew's model: For a two alternative forced choice task, the probability of correct decision D_1 given that hypothesis H_1 is true and time t has been spent to make the decision is:

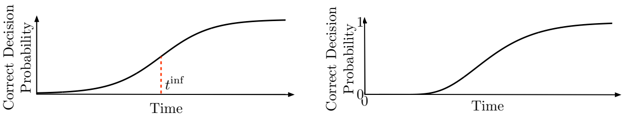
$$\mathbb{P}(D_1|H_1,t) = \frac{p_0}{1+e^{-(at-b)}},$$

where $p_0 \in [0,1]$, $a, b \in \mathbb{R}$ are some parameters specific to the human operator [7].

Drift diffusion model: For a two alternative forced choice task, conditioned on the hypothesis H_1, the evolution of the evidence for decision making is modeled as a drift-diffusion process [6], that is, for a given drift rate $\beta \in \mathbb{R}_{>0}$, and a diffusion rate $\sigma \in \mathbb{R}_{>0}$, the evidence Λ at time t is normally distributed with mean βt and variance $\sigma^2 t$. The decision is made in favor of H_1 if the evidence is greater than a decision threshold $\eta \in \mathbb{R}_{>0}$. Therefore, the conditional probability of the correct decision D_1 given that hypothesis H_1 is true and time t has been spent to make the decision is:

$$\mathbb{P}(D_1|H_1,t) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \int_{\eta}^{+\infty} e^{\frac{-(\Lambda-\beta t)^2}{2\sigma^2 t}} d\Lambda.$$

Log-normal model: The reaction times of a human operator in several missions have been studied in [29] and are shown to follow a log-normal distribution. In this context, a relevant performance function is the probability that the operator reacts within time t. This corresponds to the cumulative distribution function of the log-normal distribution.



(a) Pew's model (b) Drift diffusion model

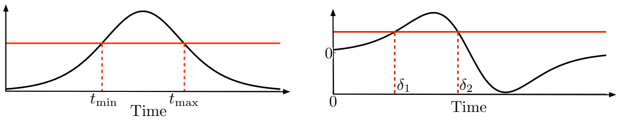
Figure 1: The evolution of the probability of the correct decision under Pew's and drift diffusion model. Both curves look similar and are sigmoid.

All these models suggest that the human performance is well captured by a sigmoid function. A sigmoid function is a doubly differentiable function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined by

$$f(t) = f_{\text{cvx}}(t)\mathcal{I}(t < t^{\text{inf}}) + f_{\text{cnv}}(t)\mathcal{I}(t \geq t^{\text{inf}}),$$

where f_{cvx} and f_{cnv} are monotonically increasing convex and concave functions, respectively, $\mathcal{I}(\cdot)$ is the indicator function and $t^{\text{inf}} \in \mathbb{R}_{>0}$ is the inflection point. The derivative of a sigmoid function is a unimodal function that achieves its maximum at t^{inf}. Further, $f'(0) \geq 0$ and $\lim_{t\to+\infty} f'(t) = 0$. Also, $\lim_{t\to+\infty} f''(t) = 0$. A typical graph of the first and second derivative of a sigmoid function is shown in Figure 2.

We consider two particular problems. First, in Section 3, we consider a static queue with latency penalty, that is, the scenario where the human operator has to perform $N \in \mathbb{N}$ decision making tasks, but each task loses value at a constant rate per unit delay in its processing. Second, in Sections 4 and 5 we consider a dynamic queue of decision making tasks where each task loses value at a constant rate per unit delay in its processing. The loss in the value of a task may occur due to the processing deadline on the task. In other words, the latency penalty is a soft constraint that captures the processing deadline on the task. For such a decision making queue, we are interested in the optimal time-duration allocation to each task. Alternatively, we are interested in the arrival rate that will result in the desired accuracy for each task. We intend to design a decision support system that tells the human operator the optimal time-duration allocation to each task.



(a) (b)

Figure 2: (a) First derivative of the sigmoid function and the penalty rate. A particular value of the derivative may be attained at two different times. The total benefit, that is, the sigmoid reward minus the latency penalty, decreases up to $t_{\min}$, increases from $t_{\min}$ to $t_{\max}$, and then decreases again. (b) Second derivative of the sigmoid function. A particular positive value of the second derivative may be attained at two different times.

Remark 1 (***Soft constraints versus hard constraints*****).** The processing deadlines on the tasks can be incorporated as hard constraints as well, but the resulting optimization problem is combinatorially hard. For instance, if the performance of the human operator is modeled by a step function with the jump at the inflection point and the deadlines are incorporated as hard constraints, then the resulting optimization problem is equivalent to the N-dimensional knapsack problem [30]. The N-dimensional knapsack problem is NP-hard and admits no fully polynomial time approximation algorithm for $N \geq 2$. The standard [30] approximation algorithm for this problem has factor of optimality $N+1$ and hence, for large N, may yield results very far from the optimal. The close connections between the knapsack problems with step functions and sigmoid functions (see [31]) suggest that efficient approximation algorithms may not exist for the problem formulation where processing deadlines are modeled as hard constraints. □

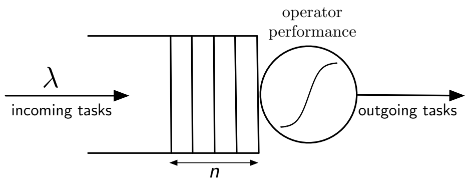


Figure 3: Problem setup. The decision making tasks arrive at a rate λ. These tasks are served by a human operator with sigmoid performance. Each task loses value while waiting in the queue.

3. Static queue with latency penalty

3.1. Problem description

Consider that the human operator has to perform $N \in \mathbb{N}$ decision making tasks in a prescribed order (task labeled "1" should be processed first, etc.) Let the human operator allocate duration t_ℓ to the task $\ell \in \{1, \dots, N\}$. Let the difficulty of the task ℓ

be $d_\ell \in \mathcal{D}$. According to the importance of the task, a weight $w_\ell \in \mathbb{R}_{\geq 0}$ is assigned to the task ℓ. The operator receives a reward $w_\ell f_{d_\ell}(t_\ell)$ for allocating duration t_ℓ to the task ℓ, while they incur a latency penalty c_ℓ per unit time for the delay in its processing. The objective of the human operator is to maximize their average benefit and the associated optimization problem is:

$$\underset{\boldsymbol{t}\in\mathbb{R}^N_{\geq 0}}{\text{maximize}} \quad \frac{1}{N}\sum_{\ell=1}^{N}(w_\ell f_{d_\ell}(t_\ell) - (c_\ell + \cdots + c_N)t_\ell), \tag{1}$$

where $\boldsymbol{t} = \{t_1, \ldots, t_N\}$ is the duration allocation vector.

3.2. Optimal solution

We start by establishing some properties of sigmoid functions. We study the optimization problem involving a sigmoid reward function and a linear latency penalty. In particular, given a sigmoid function f and a penalty rate $c \in \mathbb{R}_{>0}$, we wish to solve the following problem:

$$\underset{t\in\mathbb{R}_{\geq 0}}{\text{maximize}} \quad f(t) - ct. \tag{2}$$

The derivative of a sigmoid function is not a one-to-one mapping and hence, not invertible. We define the pseudo-inverse of the derivative of a sigmoid function f with inflection point t^{inf}, $f^\dagger : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ by

$$f^\dagger(y) = \begin{cases} \max\{t \in \mathbb{R}_{\geq 0} \mid f'(t) = y\}, & \text{if } y \in (0, f'(t^{\text{inf}})], \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Notice that the definition of the pseudo-inverse is consistent with Figure 2(a).

Lemma 1 (***Sigmoid function and linear penalty***)**.** *For the optimization problem* (2)*, the optimal solution t^* is*

$$t^* \in \operatorname{argmax}\{f(\beta) - c\beta \mid \beta \in \{0, f^\dagger(c)\}\}.$$

Proof. The global maximum lies at the point where first derivative is zero or at the boundary of the feasible set. The first derivative of the objective function is $f'(t) - c$. If $f'(t^{\text{inf}}) < c$, then the objective function is a decreasing function of time and the maximum is achieved at $t^* = 0$. Otherwise, a critical point is obtained by setting first derivative to zero. We note that $f'(t) = c$ has at most two roots. If there exist two roots, then the second derivative at the smaller root is positive, while the second derivative at the larger root is negative. Thus, the larger root corresponds to local maximum. Similarly, if there exists only one root, then it corresponds to a local maximum. The global maximum is determined by comparing the local maximum with the value of the objective function at the boundary $t = 0$. This completes the proof. □

Definition 1 (***Critical penalty rate***)**.** For a given sigmoid function f and penalty rate $c \in \mathbb{R}_{>0}$, let the solution of the problem (2) be $t^*_{f,c}$. The critical penalty rate ς_f is defined by

$$\varsigma_f = \sup\{c \in \mathbb{R}_{>0} \mid t^*_{f,c} \in \mathbb{R}_{>0}\}. \tag{4}$$

Note that the critical penalty rate is the slope of the tangent to the sigmoid function f from the origin. □

The optimal solution to problem (2) for different values of penalty rate c is shown in Figure 4. One may notice the optimal solution jumps down to zero at the critical penalty rate. This jump in the optimal allocation gives rise to combinatorial effects in the problems involving multiple sigmoid functions.

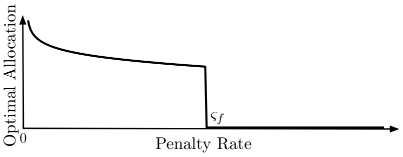


Figure 4: Optimal solution to the problem (2) as a function of linear penalty rate c. The optimal solution $t^* \to +\infty$ as the penalty rate $c \to 0^+$.

We can now analyze optimization problem (1).

Theorem 2 (***Static queue with latency penalty***)**.** *For the optimization problem* (1)*, the optimal allocation to task $\ell \in \{1, \ldots, N\}$ is*

$$t^*_\ell \in \operatorname{argmax}\{w_\ell f_{d_\ell}(\beta) - (c_\ell + \cdots + c_N)\beta \mid \beta \in \{0, f^\dagger_{d_\ell}((c_\ell + \ldots + c_N)/w_\ell)\}\}.$$

Proof. The proof is similar to the proof of Lemma 1. □

Remark 2 (***Comparison with a concave utility***)**.** The optimal duration allocation for the static queue with latency penalty decreases to a critical value with increasing penalty rate, then jumps down to zero. In contrast, if the performance function is concave instead of sigmoid, then the optimal duration allocation decreases continuously to zero with increasing penalty rate. □

3.3. Numerical Illustrations

We now present an example to elucidate on the ideas presented in this section.

Example 1 (***Static queue and heterogeneous tasks***)**.** The human operator has to serve $N = 10$ heterogeneous tasks and receives an expected reward $f_{d_\ell}(t) = 1/(1 + \exp(-a_\ell t + b_\ell))$ for an allocation of duration t secs to task ℓ, where d_ℓ is characterized by the pair (a_ℓ, b_ℓ). The following are the parameters and the weights associated with each task:

$$\begin{aligned}(a_1, \ldots, a_N) &= (1, 2, 1, 3, 2, 4, 1, 5, 3, 6),\\ (b_1, \ldots, b_N) &= (5, 10, 3, 9, 8, 16, 6, 30, 6, 12), \text{ and}\\ (w_1, \ldots, w_N) &= (2, 5, 7, 4, 9, 3, 5, 10, 13, 6).\end{aligned}$$

Let the vector of penalty rates be

$$\boldsymbol{c} = (0.09, 0.21, 0.21, 0.06, 0.03, 0.15, 0.3, 0.09, 0.18, 0.06)$$

per second. The optimal allocations are shown in Figure 3.3. The importance and difficulty level of a task are encoded in the associated weight and the inflection point of the associated sigmoid function, respectively. The optimal allocations depend on the difficulty level, the penalty rate, and the importance of the tasks. For instance, task 6 is a relatively simple but less important task and is dropped. On the contrary, task 8 is a relatively difficult but very important task and is processed. □

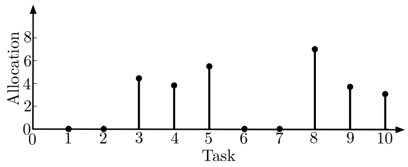


Figure 5: Static queue with latency penalty. The optimal allocations depends of the difficulty level, the penalty rate and the importance of the tasks.

4. Dynamic queue with latency penalty: problem description and properties of optimal solution

In the previous section, we developed policies for static queue with latency penalty. We now consider dynamic queue with latency penalty, that is, the scenario where the tasks arrive according to a stochastic process and wait in a queue to get processed. We assume the tasks lose value while waiting in the queue. The operator's objective is to maximize their infinite horizon reward. In the following we pose the problem as an MDP and study its properties.

4.1. Problem description

We study the optimal policies for the human operator serving a queue of decision making tasks. We now define various components of the problem:

Description of Tasks: We make following assumptions on the decision making tasks: (i) tasks arrive according to Poisson process with rate $\lambda \in \mathbb{R}_{>0}$; (ii) each task is parameterized by a variable $d \in \mathcal{D}$, where $\mathcal{D}$ is a finite set of parameters for the task; (iii) a task with parameter $d \in \mathcal{D}$ is characterized by a triplet of operator's performance function f_d, the latency penalty rate c_d, and the weight w_d assigned to the task; (iii) the parameter associated with each task is sampled from a probability distribution function $p : \mathcal{D} \to [0, 1]$. Let the realized parameter for task $\ell \in \mathbb{N}$ be d_ℓ. Thus, the operator receives a compensation $w_{d_\ell} f_{d_\ell}(t_\ell)$ for a duration allocation t_ℓ to task ℓ, while they incur a latency penalty c_{d_ℓ} per unit time for the delay in its processing. The objective of the operator is to maximize their infinite horizon expected reward. To this end, the support system suggests the optimal time duration that the human operator should allocate to a given task. We assume that such time-duration is suggested at the start of a stage and is not modified during the stage. We now formulate this optimization problem as an MDP, namely, Γ.

Description of MDP Γ*:* Let the stage $\ell \in \mathbb{N}$ of the MDP Γ be initiated with the processing of task ℓ. We now define the elements of the MDP Γ:

(i) *Action and State variables:* We choose the action variable at stage ℓ as the time-duration to be allocated to task ℓ, denoted by $t_\ell \in \mathbb{R}_{\geq 0}$. We choose the state variable at stage ℓ as the vector of parameters $\boldsymbol{d}_\ell \in \mathcal{D}^{n_\ell}$ associated with each task in the queue, where $n_\ell \in \mathbb{N}$ is the queue length at stage ℓ. Note that the definition of the stage and the state variable are consistent under the following assumption:

Assumption 1 (***Non-empty queue***)**.** Without loss of generality, we assume that the queue is never empty. If queue is empty at some stage, then the operator waits for the next task to arrive, and there is no penalty for such waiting time. □

(ii) *Reward Structure:* We define the reward $r : \mathcal{D}^{n_\ell} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ obtained by allocating duration t to the task ℓ by

$$r(\boldsymbol{d}_\ell, t) = w_{d_\ell} f_{d_\ell}(t) - \frac{1}{2}\Big(\sum_{i=\ell}^{\ell+n_\ell-1} c_{d_i} + \sum_{j=\ell}^{\ell+n'_\ell-1} c_{d_j} \Big) t,$$

where $\boldsymbol{d}_\ell \in \mathcal{D}^{n'_\ell}$ is the vector of penalty rates for the tasks in the queue and n'_ℓ is the queue length just before the end of stage ℓ.

Note that the queue length while a task is processed may not be constant, therefore, the latency penalty is computed as the average of the latency penalty for the tasks present at the start of processing the task and the latency penalty for the tasks present at the end of processing the task. Such averaging is consistent with the expected number of arrivals being a linear function of time for Poisson process.

(iii) *Value function:*

The MDP with finite horizon length $N \in \mathbb{N}$ maximizes the value function $V_N : \mathcal{D}^{n_1} \times \mathcal{B}(\{1, \dots, N\} \times \mathcal{D}^\infty, \mathbb{R}_{\geq 0}) \to \mathbb{R}$ defined by

$$V_N(\boldsymbol{d}_1, \boldsymbol{t}^{\text{finite}}) = \sum_{\ell=1}^{N} \mathbb{E}[r(\boldsymbol{d}_\ell, \boldsymbol{t}^{\text{finite}}(\ell, \boldsymbol{d}_\ell))],$$

where $n_1 \in \mathbb{N}$ is the initial queue length, $\mathcal{D}^\infty = \cup_{i \in \mathbb{N}} \mathcal{D}^i$, $\boldsymbol{t}^{\text{finite}}$ is a finite horizon duration-allocation policy, and $\mathcal{B}(\{1, \dots, N\} \times \mathcal{D}^\infty, \mathbb{R}_{\geq 0})$ is the space of bounded below functions defined from $\{1, \dots, N\} \times \mathcal{D}^\infty$ to $\mathbb{R}_{\geq 0}$. $\mathcal{B}(\{1, \dots, N\} \times \mathcal{D}^\infty, \mathbb{R}_{\geq 0})$ represents the space of policies, that is, the duration allocation as a function of stage and state. We will focus on stationary policies, i.e., policies that are independent of stage, and for stationary policies, the policy space reduces to $\mathcal{B}(\mathcal{D}^\infty, \mathbb{R}_{\geq 0})$

Under a stationary policy $\boldsymbol{t}^{\text{stat}}$, the infinite horizon average value function of the MDP $V_{\text{avg}} : \mathcal{D}^{n_1} \times \mathcal{B}(\mathcal{D}^\infty, \mathbb{R}_{\geq 0}) \to \mathbb{R}$ is defined by

$$V_{\text{avg}}(\boldsymbol{d}_1, \boldsymbol{t}^{\text{stat}}) = \lim_{N \to +\infty} \frac{1}{N} V_N(\boldsymbol{d}_1, \boldsymbol{t}^{\text{stat}}).$$

We also define the infinite horizon discounted value function $V_\alpha : \mathcal{D}^{n_1} \times \mathcal{B}(\mathcal{D}^\infty, \mathbb{R}_{\geq 0}) \to \mathbb{R}$ by

$$V_\alpha(\boldsymbol{d}_1, \boldsymbol{t}^{\text{stat}}) = \sum_{\ell=1}^{+\infty} \alpha^{\ell-1} \mathbb{E}[r(\boldsymbol{d}_\ell, \boldsymbol{t}^{\text{stat}}(\boldsymbol{d}_\ell))],$$

where $\alpha \in (0, 1)$ is the discount factor.

4.2. Properties of optimal solution

We now study some properties of the MDP Γ and its solution. Let $V_\alpha^* : \mathcal{D}^{n_1} \to \mathbb{R}_{\geq 0}$ denote the optimal infinite horizon α-discounted value function. We also define $N_{\max} = \lfloor \max\{w_d \varsigma_{f_d} / c_d \mid d \in \mathcal{D}\} \rfloor$.

Lemma 3 (***Properties of MDP*** Γ)**.** *The following statements hold for the MDP Γ and its infinite horizon average value function:*

(i). *there exists a solution to the MDP Γ;*

(ii). *an optimal stationary policy allocates zero duration to the task ℓ if $n_\ell > N_{\max}$;*

Proof. It can be verified that the conditions of Theorem 2.1 in [32] hold for MDP Γ and the optimal discounted value function exists. To prove the existence of a solution to the the average value formulation of the MDP Γ, we note that

$$V_\alpha(\boldsymbol{d}_1, \boldsymbol{t}) = \sum_{\ell=1}^{+\infty} \alpha^{\ell-1}\mathbb{E}[r(\boldsymbol{d}_\ell, t_\ell)] \le \frac{w_{\max} - c_{\min}}{(1-\alpha)},$$

for each $\boldsymbol{d}_1 \in \mathcal{D}^{n_1}$, where n_1 is initial queue length, $w_{\max} = \max\{w_d \mid d \in \mathcal{D}\}$ and $c_{\min} = \min\{c_d \mid d \in \mathcal{D}\}$. Therefore, $V^*_\alpha(\boldsymbol{d}_1) \le (w_{\max} - c_{\min})/(1-\alpha)$. Moreover, $V^*_\alpha(\boldsymbol{d}_1) \ge V_\alpha(\boldsymbol{d}_1, \boldsymbol{0}) = 0$. Hence, $|V^*_\alpha(\boldsymbol{d}_1) - V^*_\alpha(\boldsymbol{d}_0)| \le 2|w_{\max} - c_{\min}|/(1-\alpha)$, for any $\boldsymbol{d}_0 \in \mathcal{D}^{n_0}, n_0 \in \mathbb{N}$. Thus, the conditions of Theorem 5.2 in [32] hold and this establishes the first statement.

We now establish the second statement. We note that for a state associated with queue length $n > N_{\max}$, the reward is non-positive and is zero only if the allocation at that stage is zero. Moreover, for a Poisson arrival process, the probability that the queue length is non-decreasing increases with the allocation at current stage. Thus a positive allocation increases the probability of non-positive reward at future stages. Therefore, a zero duration allocation for $n > N_{\max}$ maximizes the reward at current stage and maximizes the probability of getting positive rewards at future stages. Consequently, the optimal stationary policy allocates zero duration for a queue length greater than $N_{\max}$. □

5. Dynamic queue with latency penalty: receding horizon algorithm

We rely on the certainty-equivalent receding horizon framework [26, 27, 28] to approximately solve the MDP Γ. In the certainty-equivalent approximation, the future uncertainties are replaced with their expected values [26]. For an allocation of duration t_ℓ at stage ℓ, the expected number of arrivals for a Poisson process with rate λ is λt_ℓ. Accordingly, the evolution of the queue length under certainty-equivalent approximation is

$$\bar{n}_{\ell+1} = \max\{1, \bar{n}_\ell - 1 + \lambda t_\ell\},$$

where $\bar{n}_\ell$ represents predicted queue length at stage ℓ under certainty-equivalent approximation, and $\bar{n}_1 = n_1$. The certainty-equivalent approximation also replaces the parameters of tasks that have not yet arrived by their expected values, and accordingly, assigns them the expected performance function $\bar{f} : \mathbb{R}_{\ge 0} \to [0, 1)$, the expected importance $\bar{w}$, and the expected latency penalty $\bar{c}$ defined by

$$\bar{f}(t) = \frac{1}{\bar{w}}\mathbb{E}_p[w_d f_d(t)],$$

$\bar{w} = \mathbb{E}_p[w_d]$, and $\bar{c} = \mathbb{E}_p[c_d]$, respectively, where $\mathbb{E}_p[\cdot]$ represents the expected value with respect to the measure p.

The receding horizon framework solves a finite horizon optimization problem at each iteration. We denote the receding horizon policy that solves a N-horizon certainty-equivalent problem at each stage by N-RH policy. We now study such certainty-equivalent finite horizon optimization problem.

5.1. Certainty-equivalent finite horizon optimization

We now study the finite horizon optimization problem with horizon length N that the certainty-equivalent receding horizon policy solves at each iteration. Given horizon length N, current queue length n_ℓ, the realization of the sigmoid functions $f_1, \dots, f_{n_\ell}$, the associated latency penalties $c_1, \dots, c_{n_\ell}$ and the importance levels $w_1, \dots, w_{n_\ell}$. In certainty-equivalent problem, the true parameters of the tasks are used for the tasks that have already arrived, while the expected values of the parameters are used for the tasks that have not yet arrived. In particular, if current queue length is less than the horizon length, i.e., $n_\ell < N$, then we define the reward associated with task $j \in \{1, \dots, N\}$ by

$$r_j = \begin{cases} r_j^{\text{rlzd}}, & \text{if } 1 \le j \le n_\ell, \\ r_j^{\exp}, & \text{if } n_\ell + 1 \le j \le N, \end{cases} \tag{5}$$

where $r_j^{\text{rlzd}} = w_j f_j(t_j) - (\sum_{i=j}^{n_\ell} c_i + (\bar{n}_j - n_\ell - j + 1)\bar{c})t_j - \bar{c}\lambda t_j^2/2$ is the reward computed using the realized parameters, and $r_j^{\exp} = \bar{w}\bar{f}(t_j) - \bar{c}(\bar{n}_\ell - j + 1)t_j - \bar{c}\lambda t_j^2/2$ is the reward computed using the expected values of the parameters. If the current queue length is greater than the horizon length, i.e., $n_\ell \ge N$, then we define all the reward using realized parameters, i.e., $r_j = r_j^{\text{rlzd}}$, for each $j \in \{1, \dots, N\}$.

$$\begin{aligned} &\underset{\boldsymbol{t} \ge 0}{\text{maximize}} && \frac{1}{N}\sum_{j=1}^{N} r_j \\ &\text{subject to} && \bar{n}_{j+1} = \max\{1, \bar{n}_j - 1 + \lambda t_j\}, \ \bar{n}_1 = n_\ell, \end{aligned} \tag{6}$$

where $\boldsymbol{t} = \{t_1, \dots, t_N\}$ is the duration allocation vector.

The optimization problem (6) is difficult to handle analytically. For the special case in which tasks are identical, we provide a procedure to determine the exact solution to problem (6) in the Appendix of this paper. This procedure also provides insights into the implication of sigmoid performance function on the optimal policy. For the general case, we resort to the discretization of the action and the state space and utilize the backward induction algorithm to approximately solve the dynamic program (6). Let us define maximum allocation to any task $\tau^{\max} = \max\{f_d^\dagger(c_d/w_d) \mid d \in \mathcal{D}\}$. We now state the following results on the efficiency of discretization:

Lemma 4 (***Discretization of state and action space***)**.** *For the optimization problem* (6) *and the discretization of the action and the state space with a uniform grid of width $\epsilon > 0$, the following statements hold:*

(i). *the state space and the action space can be restricted to compact spaces* $[1, N_{\max} + 1]$, *and* $[0, \tau^{\max}]$, *respectively;*

(ii). *the policy obtained through the discretized state and action space is within $O(\epsilon)$ of optimal;*

(iii). *the computational complexity of the solution is $O(N/\epsilon^2)$.*

Proof. It follows from Lemma 1 that for $\bar{n}_j > \max\{w_d \varsigma_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$, $r_j + \bar{c}\lambda t_j^2/2$ achieves its global maximum at $t_j = 0$. Hence, for $\bar{n}_j > \max\{w_d \varsigma_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$, r_j achieves its global maximum at $t_j = 0$. Moreover, the certainty-equivalent queue length at a stage $k > j$ is a non-decreasing function of of the allocation at stage j. Thus, the reward at stage $k > j$ decreases with allocation t_j. Therefore, the optimal policy for the optimization problem (6) allocates zero duration at stage j if $\bar{n}_j > \max\{w_d \varsigma_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$, and subsequently, the queue length decreases by unity at next stage. Thus, any certainty-equivalent queue length greater than $\max\{w_d \varsigma_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$ can be mapped to $N_{\max} + \texttt{frac}$, where `frac` is the fractional part of the certainty-equivalent queue length. Consequently, the state space can be restricted to the compact set $[1, N_{\max} + 1]$. Similarly, for $t_j > \tau^{\max}$, the reward at stage j in optimization problem (6) is a decreasing function of the allocation t_j, and the rewards at stages $k > j$ are decreasing function of allocation t_j. Therefore, the allocation to each task is less than $\tau^{\max}$. This completes the proof of the first statement.

Since the action variable and the state variable in problem (6) belong to compact sets and the reward function and the state evolution function is Lipschitz, it follows from Proposition 2 in [33] that the value function obtained using the discretized action and state space is within $O(\epsilon)$ of the optimal value function. This establishes the second statement.

The third statement is an immediate consequence of the fact that computational complexity of a finite horizon dynamic program is the sum over stages of the product of cardinalities of the state space and the action space in each stage. □

5.2. Performance of receding horizon algorithm

We now derive performance bounds on the receding horizon procedure. First, we determine a global upper bound on the performance of any policy for the MDP Γ. Then, we develop a lower bound on the performance of 1-RH policy. Without loss of generality, we assume that the initial queue length is unity. If the initial queue length is non-unity, then we drop tasks till queue length is unity. Note that this does not affect the infinite horizon average value function. We also assume that the latency penalty is small enough to ensure an optimal non-zero duration allocation if only one task is present in the queue, that is, $c_d \le w_d \varsigma_{f_d}$, for each $d \in \mathcal{D}$. We now introduce some notation. For a given λ and $d \in \mathcal{D}$, define

$$c_d^{\text{crit}} = \min\{c \in \mathbb{R}_{>0} \mid \operatorname{argmax}\{t \in \mathbb{R}_{\ge 0} \mid w_d f_d(t) - ct - \bar{c}\lambda t^2/2\} = 0\}.$$

For a given task $d \in \mathcal{D}$, c_d^{crit} is the minimum penalty rate at which the maximum reward associated with the task is zero. Let $c_d^{\max}$ be the optimal value of the following optimization problem:

$$\begin{aligned}
\underset{x_{id} \in \{0,1\}}{\text{maximize}} \quad & \sum_{i=1}^{N_{\max}-1} \sum_{d \in \mathcal{D}} c_d x_{id} \\
\text{subject to} \quad & \sum_{i=1}^{N_{\max}-1} \sum_{d \in \mathcal{D}} c_d x_{id} < c_d^{\text{crit}} - c_d \\
& \sum_{d \in \mathcal{D}} x_{id} \le 1, \ \text{for each } i \in \{1, \dots, N_{\max} - 1\}.
\end{aligned}$$

For a given current task $d \in \mathcal{D}$, $c_d^{\max}$ is the maximum penalty rate due to all other tasks in the queue at which the 1-RH policy allocates a non-zero duration to current task. Let the associated non-zero allocation be t_d^{crit} defined by $t_d^{\text{crit}} = \operatorname{argmax}\{w_d f_d(t) - (c_d + c_d^{\max})t - \bar{c}\lambda t^2/2 \mid t \in \mathbb{R}_{\ge 0}\}$. Let $\tau_d^{\max} = f_d^{\dagger}(c_d/w_d)$, for each $d \in \mathcal{D}$. We will show that $\tau_d^{\max}$ is the maximum duration an optimal policy may allocate to task d. Let $\boldsymbol{t}^{\text{unit}}$ be the 1-RH policy.

Theorem 5 (***Bounds on performance***)**.** *For the MDP Γ and 1-RH policy the following statements hold:*

(i). *the average value function satisfy the following upper bound*

$$V_{\text{avg}}(\boldsymbol{d}_1, \boldsymbol{t}) \le \max_{d \in \mathcal{D}} \{w_d f_d(\tau_d^{\max}) - c_d \tau_d^{\max}\},$$

for each $n_1 \in \mathbb{N}$ and any policy $\boldsymbol{t}$;

(ii). *the average value function satisfy the following lower bound for 1-RH policy:*

$$V_{\text{avg}}(\boldsymbol{d}_1, \boldsymbol{t}^{\text{unit}}) \ge \min_{d \in \mathcal{D}} \frac{w_d f_d(t_d^{\text{crit}}) - (c_d + c_d^{\max}) t_d^{\text{crit}} - \bar{c}\lambda {t_d^{\text{crit}}}^2/2}{\lambda \tau_{\max} + e^{-\lambda \tau_{\max}}},$$

for each $n_1 \in \mathbb{N}$.

Proof. We start by establishing the first statement. We note that the reward at stage ℓ is $r(\boldsymbol{d}_\ell, t_\ell) \le w_{d_\ell} f_{d_\ell}(t_\ell) - c_{d_\ell} t_\ell$. It follows from Lemma 1 that the maximum value $w_{d_\ell} f_{d_\ell}(t_\ell) - c_{d_\ell} t_\ell$ is achieved at $t_\ell = \tau_{d_\ell}^{\max}$. Therefore, the reward at each stage is upper bounded by $\max\{w_d f_d(\tau_d^{\max}) - c_d \tau_d^{\max} \mid d \in \mathcal{D}\}$ and the first statement follows.

To establish the second statement, we note that if some new tasks arrive at a stage, then the optimal policy processes at least one of these tasks or processes at least one task already in the queue. Therefore, the optimal policy processes at least one task for each set of tasks arrived. The minimum reward obtained after processing task d is $w_d f_d(t_d^{\text{crit}}) - (c_d + c_d^{\max}) t_d^{\text{crit}} - \bar{c}\lambda {t_d^{\text{crit}}}^2/2$. If n_{arr} is the number of arrived tasks, the fraction of tasks processed is $1/n_{\text{arr}}$. Under Assumption 1, the expected number of arrivals at stage ℓ for a Poisson process is $\mathbb{E}[n_{\text{arr}}] = \lambda t_\ell + e^{-\lambda t_\ell} \mathbb{P}(n_\ell = 1)$, where the second term corresponds to the situation when the queue becomes empty after processing task ℓ and the operator waits for new task. Since $1/n_{\text{arr}}$ is a convex function and $\mathbb{E}[n_{\text{arr}}] \le \lambda \tau_{\max} + e^{-\lambda \tau_{\max}}$, it follows from the Jensen's inequality that the expected fraction of tasks processed is greater than $1/(\lambda \tau_{\max} + e^{-\lambda \tau_{\max}})$. This completes the proof of the last statement. □

Similar bounds can be derived for the N-RH policy. Since the future queue length increases with increasing t_1, the total penalty increases in problem (6) with increasing t_1. Thus, with increasing N, the receding horizon policy drops tasks at a smaller critical penalty c_d^{crit}. Consequently, the total penalty on each task decreases and the performance improves.

5.3. Numerical Illustrations

We now elucidate on the concepts discussed in this section with an example.

Example 2 (***RH policy***)**.** Suppose that the human operator has to serve a queue of tasks with Poisson arrival at the rate λ per sec. The set of the tasks is the same as in Example 1 and each task is sampled uniformly from this set. 1-RH and 10-RH policies for a sample evolution of the queue at an arrival rate $\lambda = 0.5$ per second are shown in Figure 6 and 7, respectively. The sequence of tasks arriving is the same for both the policies. The RH policy tends to drop the tasks that are difficult and unimportant. The difficulty of the tasks is characterized by the inflection point of the associated sigmoid functions. The queue length under the 1-RH policy is higher than the 10-RH policy. A comparison of the RH policies is shown in Figure 8. We obtained these performance curves through Monte-Carlo simulations. □

Remark 3 (***Comparison with a concave utility***)**.** With the increasing penalty rate as well as the increasing arrival rate, the time duration allocation decreases to a critical value and then jumps down to zero for the dynamic queue with latency penalty. In contrast, if the performance function is concave instead of sigmoid, then the duration allocation decreases continuously to zero with increasing penalty rate as well as increasing arrival rate. □

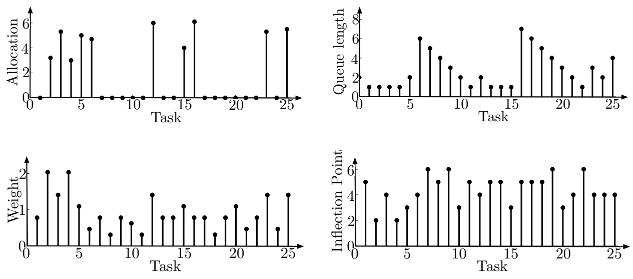


Figure 6: 10-RH policy for a *sample evolution* of the dynamic queue with latency penalty.

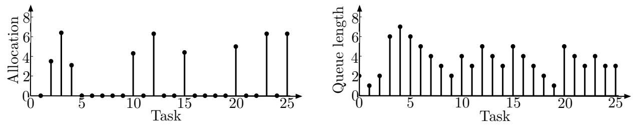


Figure 7: 1-RH policy for a *sample evolution* of the dynamic queue with latency penalty.

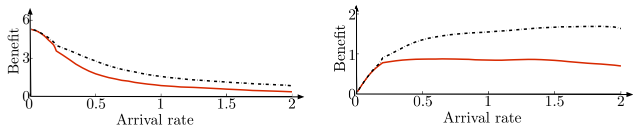


Figure 8: Empirical expected benefit per unit task and per unit time. The dashed-dotted black curve represents the 10-RH policy and the solid red curve represents the 1-RH policy, respectively.

Discussion 6 (***Design of queue***)**.** The performance of the RH policy as a function of the arrival rate is shown in Figure 8. It can be seen that the expected benefit per unit task, that is, the value of the average value function under the RH policy, decreases slowly till a critical arrival rate and then starts decreasing quickly. This critical arrival rate corresponds to the situation where a new task is expected to arrive as soon as the operator finishes processing the current task. The objective of the designer is to achieve a good performance on each task and therefore, the arrival rate should be picked close to this critical arrival rate. If each task is identical and is characterized by $d \in \mathcal{D}$, then it can be verified that the critical arrival rate is $\lambda_d^{\text{crit}} = 1/\tau_d^{\text{crit}}$, where $\tau_d^{\text{crit}} = f_d^{\dagger}(2c_d/w_d)$. In the context of heterogeneous tasks, if each task is sampled from p, then the critical arrival rate is $\sum_{d\in\mathcal{D}} p(d)\lambda_d^{\text{crit}}$. In general, designer may have other performance goals for the operator, and accordingly, may choose higher task arrival rate. □

6. Conclusions

We presented optimal servicing policies for the queues where the performance function of the server is a sigmoid function. First, we considered a queue with no arrival and a latency penalty. It was observed that the optimal policy may drop some tasks. Second, a dynamic queue with latency penalty was considered. We posed the problem in an MDP framework and proposed an approximate solution in the certainty-equivalent receding horizon optimization framework. We derived performance bounds for the proposed solution and suggested guidelines for choosing the expected arrival rate for the queue.

The decision support system designed in this paper assumes that the arrival rate of the tasks as well as the parameters in the performance function are known. An interesting open problem is to come up with policies which perform an online estimation of the arrival rate and the parameters of the performance function and simultaneously determine the optimal allocation policy. Another interesting problem is to incorporate more human factors into the optimal policy, for example, situational awareness, fatigue, etc. The policies designed in this paper rely on first-come first-serve discipline to process tasks. It would be of interest to study problems with other processing disciples, for example, preemptive queues. We focused on open loop optimization of human performance interacting with automata. A significant future direction is to incorporate human feedback and study closed loop policies that are jointly optimal for the human operator as well as the automaton. Some preliminary results in this direction as presented in [34].

7. Appendix: Finite horizon optimization for identical tasks

In this section, we consider the special case of the finite horizon optimization problem (6) in which tasks are identical and propose a procedure to obtain the exact solution. We remark that even if the tasks are heterogeneous, many a times extensive experiments can not be done to determine operator's performance on each task. Under such circumstances, each task is treated as identical and a performance function associated with average data is used for each task. We also note that the optimal human attention allocation policy is needed to counter the information overload situations. The information overload situations correspond to the heavy traffic regime of the queue and we focus on this particular regime. In the following, we denote the sigmoid function and the latency penalty associated with each task by f and c, respectively. Let the inflection point associated with f be t^{inf}. We assume that the weight associated with each task is unity. We note that under the heavy-traffic regime the certainty-equivalent queue length is $\bar{n}_\ell = n_1 - \ell + 1 + \lambda \sum_{j=1}^{\ell-1} t_j$. Substituting the certainty-equivalent queue length into the objective function of the optimization problem (6), we obtain the function $J : \mathbb{R}^N_{\geq 0} \to \mathbb{R}$ defined by

$$J(\boldsymbol{t}) := \frac{1}{N} \sum_{\ell=1}^{N} \Big(f(t_\ell) - c(n_1 - \ell + 1)t_\ell - c\lambda t_\ell \sum_{j=1, j\neq\ell}^{N} t_j - \frac{c\lambda t_\ell^2}{2} \Big),$$

where c is the penalty rate, λ is the arrival rate, and n_1 is the initial queue length. Thus, the optimization problem (6) is equivalent to

$$\underset{\boldsymbol{t}\geq 0}{\text{maximize}} \quad J(\boldsymbol{t}). \tag{7}$$

Assume that the solution to the optimization problem (6) allocates a strictly positive time only to the tasks in the set $\mathcal{T}_{\mathrm{proc}} \subseteq \{1, \dots, N\}$, which we call the *set of processed tasks*. (Accordingly, the policy allocates zero time to the tasks in $\{1, \dots, N\} \setminus \mathcal{T}_{\mathrm{proc}}$). Without loss of generality, assume

$$\mathcal{T}_{\mathrm{proc}} := \{\eta_1, \dots, \eta_m\},$$

where $\eta_1 < \dots < \eta_m$ and $m \leq N$. A duration allocation vector $\boldsymbol{t}$ is said to be consistent with $\mathcal{T}_{\mathrm{proc}}$ if only the tasks in $\mathcal{T}_{\mathrm{proc}}$ are allocated non-zero duration.

Lemma 7 (***Properties of maximum points***)**.** *For the optimization problem* (7)*, and a set of processed tasks* $\mathcal{T}_{\mathrm{proc}}$*, the following statements hold:*

(i). *a global maximum point* $\boldsymbol{t}^*$ *satisfy* $t^*_{\eta_1} \geq t^*_{\eta_2} \geq \dots \geq t^*_{\eta_m}$*;*

(ii). *a local maximum point* $\boldsymbol{t}^\dagger$ *consistent with* $\mathcal{T}_{\mathrm{proc}}$ *satisfies*

$$f'(t^\dagger_{\eta_k}) = c(n_1 - \eta_k + 1) + c\lambda \sum_{i=1}^{m} t^\dagger_{\eta_i}, \textit{ for all } k \in \{1, \dots, m\}; \tag{8}$$

(iii). *the system of equations* (8) *can be reduced to*

$$f'(t^\dagger_{\eta_1}) = \mathcal{P}(t^\dagger_{\eta_1}), \textit{ and } t^\dagger_{\eta_k} = f^\dagger(f'(t^\dagger_{\eta_1}) - c(\eta_k - \eta_1)),$$

for each $k \in \{2, \dots, m\}$*, where* $\mathcal{P} : \mathbb{R}_{>0} \to \mathbb{R} \cup \{+\infty\}$ *is defined by*

$$\mathcal{P}(t) = \begin{cases} p(t), & \textit{if } f'(t) \geq c(\eta_m - \eta_1), \\ +\infty, & \textit{otherwise,} \end{cases}$$

where $p(t) = c(n_1 - \eta_1 + 1 + \lambda t + \lambda \sum_{k=2}^{m} f^\dagger(f'(t) - c(\eta_k - \eta_1)))$*;*

(iv). *a local maximum point* $\boldsymbol{t}^\dagger$ *consistent with* $\mathcal{T}_{\mathrm{proc}}$ *satisfies*

$$f''(t_{\eta_k}) \leq c\lambda, \textit{ for all } k \in \{1, \dots, m\}.$$

Proof. We start by proving the first statement. Assume $t^*_{\eta_j} < t^*_{\eta_k}$ and define the allocation vector $\bar{\boldsymbol{t}}$ consistent with $\mathcal{T}_{\mathrm{proc}}$ by

$$\bar{t}_{\eta_i} = \begin{cases} t^*_{\eta_i}, & \text{if } i \in \{1, \dots, m\} \setminus \{j, k\}, \\ t^*_{\eta_j}, & \text{if } i = k, \\ t^*_{\eta_k}, & \text{if } i = j. \end{cases}$$

It is easy to see that

$$J(\boldsymbol{t}^*) - J(\bar{\boldsymbol{t}}) = (\eta_j - \eta_k)(t^*_{\eta_j} - t^*_{\eta_k}) < 0.$$

This inequality contradicts the assumption that $\boldsymbol{t}^*$ is a global maximum of J.

To prove the second statement, note that a local maximum is achieved at the boundary of the feasible region or at the set where the Jacobian of J is zero. At the boundary of the feasible region $\mathbb{R}^N_{\geq 0}$, some of the allocations are zero. Given the m non-zero allocations, the Jacobian of the function J projected on the space spanned by the non-zero allocations must be zero. The expressions in the theorem are obtained by setting the Jacobian to zero.

To prove the third statement, we subtract the expression in equation (8) for $k = j$ from the expression for $k = 1$ to get

$$f'(t_{\eta_j}) = f'(t_{\eta_1}) - c(\eta_j - \eta_1). \tag{9}$$

There exists a solution of equation (9) if and only if $f'(t_{\eta_1}) \geq c(\eta_j - \eta_1)$. If $f'(t_{\eta_1}) < c(\eta_j - \eta_1) + f'(0)$, then there exists only one solution. Otherwise, there exist two solutions. It can be seen that if there exist two solutions $t^\pm_j$, with $t^-_j < t^+_j$, then $t^-_j < t_{\eta_1} < t^+_j$. From the first statement, it follows that only possible allocation is t^+_j. Notice that $t^+_j = f^\dagger(f'(t_{\eta_1}) - c(\eta_j - \eta_1))$. This choice yields feasible time allocation to each task $\eta_j, j \in \{2, \dots, m\}$ parametrized by the time allocation to the task η_1. A typical allocation is shown in Figure 9(a). We further note that the effective penalty rate for the task η_1 is $c(n_1 - \eta_1 + 1) + c\lambda \sum_{j=1}^{m} t_{\eta_j}$. Using the expression of $t_{\eta_j}, j \in \{2, \dots, m\}$, parametrized by t_{η_1}, we obtain the expression for $\mathcal{P}$.

To prove the last statement, we observe that the Hessian of the function J is

$$\frac{\partial^2 J}{\partial \boldsymbol{t}^2} = \mathrm{diag}(f''(t_{\eta_1}), \dots, f''(t_{\eta_m})) - c\lambda \mathbf{1}_m \mathbf{1}_m^T,$$

where $\mathrm{diag}(\cdot)$ represents a diagonal matrix with the argument as diagonal entries. For a local maximum to exist at non-zero duration allocations $\{t_{\eta_1}, \dots, t_{\eta_m}\}$, the Hessian must be negative semidefinite. A necessary condition for Hessian to be negative semidefinite is that diagonal entries are non-positive. □

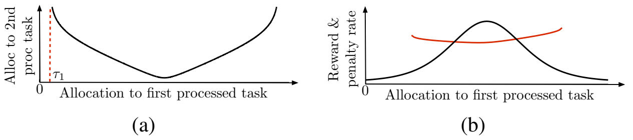
We refer to the function $\mathcal{P}$ as the *effective penalty rate* for the first processed task. A typical graph of $\mathcal{P}$ is shown in Figure 9(b). Given $\mathcal{T}_{\text{proc}}$, a feasible allocation to the task η_1 is such that $f'(t_{\eta_1}) - c(\eta_j - \eta_1) > 0$, for each $j \in \{2, \dots, m\}$. For a given $\mathcal{T}_{\text{proc}}$, we define the minimum feasible duration allocated to task η_1 (see Figure 9(a)) by

$$\tau_1 := \begin{cases} \min\{t \in \mathbb{R}_{\geq 0} \mid f'(t) = c(\eta_m - \eta_1)\}, & \text{if } f'(t^{\text{inf}}) \geq c(\eta_m - \eta_1), \\ 0, & \text{otherwise.} \end{cases}$$

Let $f''_{\max}$ be the maximum value of f''. We now define the points at which the function $f'' - c\lambda$ changes its sign (see Figure 2(b)):

$$\delta_1 := \begin{cases} \min\{t \in \mathbb{R}_{\geq 0} \mid f''(t) = c\lambda\}, & \text{if } c\lambda \in [f''(0), f''_{\max}], \\ 0, & \text{otherwise,} \end{cases}$$

$$\delta_2 := \begin{cases} \max\{t \in \mathbb{R}_{\geq 0} \mid f''(t) = c\lambda\}, & \text{if } c\lambda \leq f''_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$



(a)

(b)

Figure 9: (a) Feasible allocations to the second processed task parametrized by the allocation to the first processed task. (b) The penalty rate and the sigmoid derivative as a function of the allocation to the first task.

Theorem 8 (***Finite horizon optimization***)**.** *Given the optimization problem* (7)*, and a set of processed tasks* $\mathcal{T}_{\text{proc}}$*. The following statements are equivalent:*

(i). *there exists a local maximum point consistent with* $\mathcal{T}_{\text{proc}}$*;*

(ii). *one of the following conditions hold*

$$f'(\delta_2) \geq \mathcal{P}(\delta_2), \quad \textit{or} \tag{10}$$

$$f'(\tau_1) \leq \mathcal{P}(\tau_1), \;\; f'(\delta_1) \geq \mathcal{P}(\delta_1), \; \textit{and } \delta_1 \geq \tau_1. \tag{11}$$

Proof. A critical allocation to task η_1 is located at the intersection of the graph of the reward rate $f'(t_{\eta_1})$ and the effective penalty rate $\mathcal{P}(t_{\eta_1})$. From Lemma 7, a necessary condition for the existence of a local maximum at a critical point is $f''(t_{\eta_1}) \leq c\lambda$, which holds for $t_{\eta_1} \in (0, \delta_1] \cup [\delta_2, \infty)$. It can be seen that if condition (10) holds, then the function $f'(t_{\eta_1})$ and the effective penalty function $\mathcal{P}(t_{\eta_1})$ intersect in the region $[\delta_2, \infty[$. Similarly, condition (11) ensures the intersection of the graph of the reward function $f'(t_{\eta_1})$ with the effective penalty function $\mathcal{P}(t_{\eta_1})$ in the region $(0, \delta_1]$. □

For a given horizon length N, the potential sets of processed tasks are the elements of the set $\{0, +\}^N$. The feasibility of a given set of processed tasks can be determined using Theorem 8. For each feasible set of processed tasks, the optimal allocations can be determined using the third statement in Lemma 7, and the value for each potential set of processed task can be compared to determine the optimal solution of optimization problem (6) with identical tasks.

[1] V. Srivastava, R. Carli, F. Bullo, and C. Langbort. Task release control for decision making queues. In *American Control Conference*, pages 1855–1860, San Francisco, CA, USA, June 2011.

[2] E. Guizzo. Obama commanding robot revolution announces major robotics initiative. IEEE Spectrum, June 2011.

[3] W. M. Bulkeley. Chicago's camera network is everywhere. The Wall Street Journal, November 17, 2009.

[4] C. Drew. Military taps social networking skills. The New York Times, June 7, 2010.

[5] T. Shanker and M. Richtel. In new military, data overload can be deadly. The New York Times, January 16, 2011.

[6] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen. The physics of optimal decision making: A formal analysis of performance in two-alternative forced choice tasks. *Psychological Review*, 113(4):700–765, 2006.

[7] R. W. Pew. The speed-accuracy operating characteristic. *Acta Psychologica*, 30:16–26, 1969.

[8] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, 3 edition, 2000.

[9] S. K. Hong and C. G. Drury. Sensitivity and validity of visual search models for multiple targets. *Theoretical Issues in Ergonomics Science*, 3(1):85–110, 2002.

[10] D. Vakratsas, F. M. Feinberg, F. M. Bass, and G. Kalyanaram. The shape of advertising response functions revisited: A model of dynamic probabilistic thresholds. *Marketing Science*, 23(1):109–119, 2004.

[11] M. H. Rothkopf. Bidding in simultaneous auctions with a constraint on exposure. *Operations Research*, 25(4):620–629, 1977.

[12] D. K. Schmidt. A queuing analysis of the air traffic controller's work load. *IEEE Transactions on Systems, Man & Cybernetics*, 8(6):492–498, 1978.

[13] K. Savla, T. Temple, and E. Frazzoli. Human-in-the-loop vehicle routing policies for dynamic environments. In *IEEE Conf. on Decision and Control*, pages 1145–1150, Cancún, México, December 2008.

[14] K. Savla and E. Frazzoli. A dynamical queue approach to intelligent task management for human operators. *Proceedings of the IEEE*, 100(3):672–686, 2012.

[15] K. Savla and E. Frazzoli. Maximally stabilizing task release control policy for a dynamical queue. *IEEE Transactions on Automatic Control*, 55(11):2655–2660, 2010.

[16] L. F. Bertuccelli, N. Pellegrino, and M. L. Cummings. Choice modeling of relook tasks for UAV search missions. In *American Control Conference*, pages 2410–2415, Baltimore, MD, USA, June 2010.

[17] L. F. Bertuccelli, N. W. M. Beckers, and M. L. Cummings. Developing operator models for UAV search scheduling. In *AIAA Conf. on Guidance, Navigation and Control*, Toronto, Canada, August 2010.

[18] J. W. Crandall, M. L. Cummings, M. Della Penna, and P. M. A. de Jong. Computing the effects of operator attention allocation in human control of multiple robots. *IEEE Transactions on Systems, Man & Cybernetics. Part A: Systems & Humans*, 41(3):385–397, 2011.

[19] N. D. Powel and K. A. Morgansen. Multiserver queueing for supervisory control of autonomous vehicles. In *American Control Conference*, pages 3179–3185, Montréal, Canada, June 2012.

[20] L. I. Sennott. *Stochastic Dynamic Programming and the Control of Queueing Systems*. Wiley, 1999.

[21] S. Stidham Jr. and R. R. Weber. Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Operations Research*, 37(4):611–625, 1989.

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

[23] K. M. Adusumilli and J. J. Hasenbein. Dynamic admission and service rate control of a queue. *Queueing Systems*, 66(2):131–154, 2010.

[24] O. Hernández-Lerma and S. I. Marcus. Adaptive control of service in queueing systems. *Systems & Control Letters*, 3(5):283–289, 1983.

[25] M. Zafer and E. Modiano. Optimal rate control for delay-constrained data transmission over a wireless channel. *IEEE Transactions on Information Theory*, 54(9):4020–4039, 2008.

[26] D. Bertsekas. Dynamic programming and suboptimal control: A survey from ADP to MPC. *European Journal of Control*, 11(4-5):310–334, 2005.

[27] H. S. Chang and S. I. Marcus. Approximate receding horizon approach for Markov decision processes: Average reward case. *Journal of Mathematical Analysis and Applications*, 286(2):636–651, 2003.

[28] J. Mattingley, Y. Wang, and S. Boyd. Receding horizon control: Automatic generation of high-speed solvers. *IEEE Control Systems Magazine*, 31(3):52–65, 2011.

[29] D. N. Southern. Human-guided management of collaborating unmanned vehicles in degraded communication environments. Master's thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2010.

[30] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[31] V. Srivastava and F. Bullo. Hybrid combinatorial optimization: Sample problems and algorithms. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 7212–7217, Orlando, FL, USA, December 2011.

[32] A. Arapostathis, V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus. Discrete-time controlled Markov processes with average cost criterion: A survey. *SIAM Journal on Control and Optimization*, 31(2):282–344, 1993.

[33] D. P. Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(6):415–419, 1975.

[34] V. Srivastava, A. Surana, and F. Bullo. Adaptive attention allocation in human-robot systems. In *American Control Conference*, pages 2767–2774, Montréal, Canada, June 2012.