Operation Systems with Discretionary Task Completion

Gigi Yuen, Wallace J. Hopp, Seyed M.R. Iravani
Department of Industrial Engineering and Management Sciences
Northwestern University, Evanston, IL

Abstract

Most performance evaluation models in operations management literature have assumed that tasks possess standardized completion criteria. However, in many operation systems, particularly in service and professional work, judgment is frequently required to determine when a task is completed. In thin paper, we show that introducing discretion in task completion adds a fourth variability buffer, quality, to the well known buffers of capacity, inventory and time. To gain insight into the managerial implications of this difference, we model the work of a single worker system with discretionary task completion as a controlled queue. After characterizing the optimal control policy and identifying identifying some practical heuristics, we use this model to examine the differences between discretionary and non-discretionary work. We show that adding capacity may actually increase congestion in systems with discretionary task completion, and information about job types in queue is less useful in systems with discretionary task completion than in systems with non-discretionary task completion.

Keywords: Service; Quality; Variability Buffer; Markov Decision Process;

1 Introduction

Since the 1980's, the American economy has steadily shifted toward service, with 34.2%, 37.1% and 42.7% of the national GDP coming from the service sector[1] in 1987, 1994 and 2001 respectively. The manufacturing sector[2] constituted 25.2%, 22.5% and 20.3% of national GDP in these same years (Bureau of Economic Analysis 2001). This trend is being driven by an increase in the number and size of service-oriented firms. Moreover, even within manufacturing organizations, there is an increasing emphasis on service and professional work. For example, in 1972 semi-skilled operators (e.g., line workers) represented 55% of the General Motors workforce, while non-management professionals (e.g., engineers) represented 5%. By 2001, operators represented only 44%, while professionals had increased to 14% (General Motors 1974 and General Motors 2002).

A large portion of research on operations systems has focused explicitly or implicitly on manufacturing-oriented work systems. Since the early work of Fredrick W. Taylor at the beginning of the 20th

[1] Including sectors of finance, insurance, real estate and services

[2] Including sectors of mining, construction and manufacturing

century, in studies ranging from time and motion studies to the modeling of production flows, industrial engineers and operations management researchers have concentrated on developing better methods for managing manufacturing systems. Considerably less research has been devoted to service systems and almost no industrial engineering work has systematically studied professional work.

An almost universal in past models of manufacturing work systems is that these involve well-defined tasks with *non-discretionary* completion criteria (i.e., determined by objective standards). However, in practice, many operations systems involve less defined tasks with *discretionary* completion criteria (i.e., determined by a worker's subjective standards). In particular, service and professional systems frequently require workers' judgment on when tasks are completed. Discretionary task completion introduces a degree into process times, so a worker can adjust quality of his/her output to manage workload. For our purposes, we label traditional manufacturing work and routine service work (e.g., bank tellers, checkout clerks) as *Non-Discretionary Task Completion (NDTC)*, and professional work and complex service work (e.g., engineers, physicians, financial analysts) as *Discretionary Task Completion (DTC).*

For example, consider the task of installing a car seat into a sedan on an assembly line. The worker has to follow a standardized procedure to secure the seat to the car frame. The output quality has well-defined metrics (e.g., whether the seat is secure) and management can easily specify the completion standard. hence, this task is NDTC. Various service industry tasks, such as transferring money between bank accounts follow similarly well-defined structures, are hence also regarded as NDTC work in our terminology.

In comparison, consider a call center agent who provides computer software support. The agent is typically presented with unstandardized information about the problem, which must be intelligently sorted to diagnose the software issue. Once the problem has been identified, the agent must select a solution approach which may involve trials that could be done during call with the help of the agent or independently by the caller after the call is complete. Hence, the agent has considerable influence over the call duration. Moverover, such an environment, it is nearly impossible to enforce a clear and objective completion standard. For a review of research on call centers, see Gans et al. (2003).

One of the primary challenges in managing manufacturing or service systems is how to better control and mitigate variability. Variability is inherent in virtually all systems, whether they involve DTC or NDTC work. This variability degrades system performance because of a well-known principle of factory physics which states that variability in traditional production systems must be buffered by some combination of capacity, inventory, and time (Hopp and Spearman 2000). In service systems, this reduces to two variability buffers, capacity and time, because services cannot be stored. For example, variability in arrivals to a call center causes queueing delay and hence a time buffer (unless the center is significantly overstaffed, in which case the variability is buffered by capacity instead).

In this paper, we show that discretionary task completion introduces quality as an additional factor for buffering variability in DTC systems. To gain insights into the managerial implications of this newly introduced variability buffer, we examine a single server with discretionary task completion and either one or two classes of tasks. We first characterize the optimal control policy which specifies how long the worker should spend on each task for the single class model. We also introduce two simple threshold heuristics and show that they can be effective under certain conditions. Next, we compare our discretionary completion models with analogous non-discretionary completion models and identify two interesting phenomena that are distinctive to systems with discretionary task completion: (a) in contrast to well-known non-discretionary task completion system behavior, congestion can actually increase when capacity is increased in a discretionary task completion system, and (b) in a multi-class system, the value of task type information about jobs in queue at the time of arrival decreases with the degree of discretion in completion times.

2 Literature Review

Extensive analytical research has been devoted to the design and control of DTC work systems. This work covers a wide range of topics, including work flow design, production scheduling and inventory management (see, e.g., Buzacott and Shanthikumar 1992, Altiok 1996, Hopp and Spearman 2000, Askin and Goldberg 2001, and Halevi 2001.)

Researchers have also conducted empirical and experimental studies to understand the impacts of human behavior on work systems (see Bailey 1998, Banker et al. 2001, and Longenecker et al. 1994.) For example, Schultz et al. (1998) conducted laboratory experiments to investigate how motivation affects worker processing time. A survey and discussion of research at the interface of operations management and human resources management is given in Boudreau et al. (2003).

Many of the principles that have emerged from the above research have implicitly or explicitly focused on systems in which tasks are routine and well defined. Hence, they are applicable to what we term "NDTC" settings. Some researchers have attempted to extend these NDTC insights to DTC environments, ranging from large scale reengineering of a corporation (Hammer and Champy 1993) to small scale process improvement of a product development process (Alder et al. 1995, Krishnan et al. 1997, and Loch and Terwiesch 1996.) Adler et. al. (1995) suggested thinking of "development as a process in which projects move through the knowledge-work equivalent of a job shop," while Loch and Terwiesch (1996) argued that the product development process has many manufacturing-like activities. Although this work has yielded useful results, it is limited by the implicit assumption of NDTC work.

Some recent of modeling work has been done specifically with discretionary task completion in mind. This can be classified into two major streams: (i) models that allow adjusting service rate as dynamic control of queueing systems, and (ii) models that represent the relationship of dynamic pricing and adjustable service rate.

The first stream of research focuses on the characterization and computation of optimal policies and provides limited managerial insights. The two papers most closely related to our work are Stidham and Weber (1989) and George and Harrison (2005), both of which showed the monotonicity of optimal policy which service rate increases with queue length. The two papers allow the service rate to be changed only at job arrival or departure epochs. In contrast, our paper allows service time to be adjusted at any time during processing and hence gives more flexibility to the server.

The second stream of research studies similar systems in which servers determine the optimal combination of pricing and service rate. This includes Debo et al. (2004), and Ata and Shnerson (2005). The work yields interesting results on interactions between service providers and customers but does not explore the implication of discretionary task completion in system design and worker management.

The only paper at which we are aware of that addresses DTC work from the perspective of generating managerial principles is Owen and Jordan (2003). They proposed S-curve model that describes the time dependence of output quality in white-collar work systems. Using simulation, they examined the performance of various scheduling policies in managing jobs with due dates in a single server system.

In this paper, we adopt an optimal control approach as a step toward developing analytic principles of DTC workforce management. The remainder of the paper is organized as follows.

In Section 3, we define a time-dependent quality curve to capture discretionary task completion characteristics of DTC tasks. In Section 4, we study a single-station, single-job class model of a discretionary task completion system, and derive the structure of the optimal control policy. We also conduct a series of numerical studies that examine the efficiency and robustness of alternative heuristic policies. We then investigate the role of discretionary completion times on the effect of an increase in capacity in Section 5. In Section 6, we extend our model to a system with two classes of tasks with different processing time characteristics, and evaluate the value of job type information by comparing the performance of two-job-class systems with combined and separate queues. The paper concludes in Section 7.

3 Discretionary versus Non-discretionary Task Completion

In a manufacturing or routine service environment, output is typically measured as the rate of task completions (e.g., jobs per day or customers per hour). The reason is that the work content of tasks is known (at least in expectation). So if an auto assembly line completes 500 cars in a shift, we know exactly what value has been created (subject to post-sale quality adjustments, which are measured after the fact as warranty repair costs). What ensures this are the non-discretionary standards that define each task needed to build a car. Hence, for a single non-discretionary task, we can describe the (expected) quality or value generated as a function of time with a curve like that in Figure 1.*Left.* This shows a task that requires exactly τ_0 units of time to complete. Processing for less than τ_0 units of time leaves the task unfinished and hence without value to the downstream station (or the customer), while processing more than τ_0 does not add value. This gives us a precise definition of Non-Discretionary Task Completion (NDTC).

However, the completion criteria in some tasks are not so clear. For example, in the previously described customer support example a call can be made short or long at the agent's discretion. Assuming that the agent makes rational use of call time, the expected value to the customer will be an increasing function of time. However, since the total value of information provided by the agent is finite, the value versus time curve must be eventually concave. While many forms are possible, the simplest relationship between processing time and expected value is a concave curve, as depicted in Figure 1.*Right.* Since a curve like this presents the agent with decisions of when to terminate processing, it represents what we mean by Discretionary Task Completion (DTC).

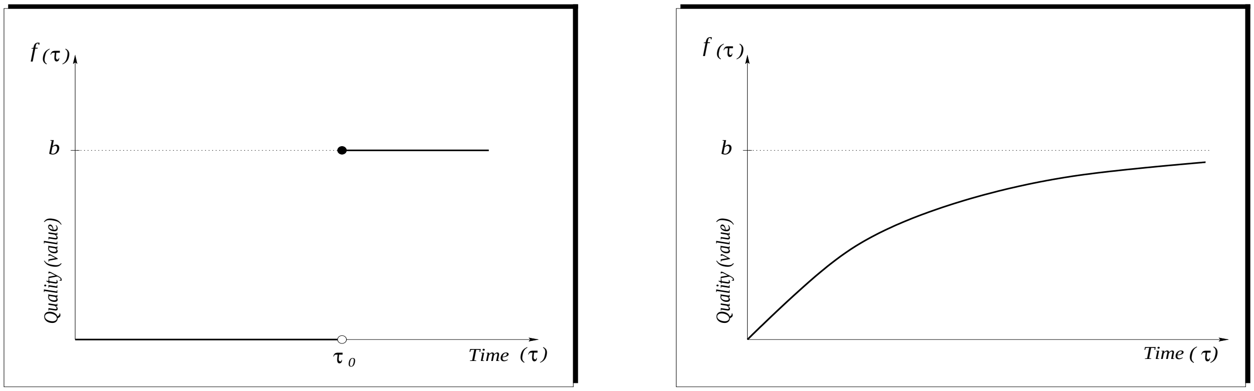

Figure 1: *Left:* Value vs. time for non-discretionary completion tasks, *Right:* Value vs. time for discretionary completion tasks.

A simple way to model the concave curve in Figure 1.*Right* is the exponential function:

$$f(\tau) = b(1 - e^{-a\tau}) \tag{1}$$

The parameter a determines the rate of increase of the value function $f(\tau)$. The larger a is, the faster $f(\tau)$ approaches its upper limit, b. Hence, a characterizes the processing rate, while b represents the highest possible task value.

In practice, it may be difficult to estimate the value-time curve. Nevertheless, in high volume systems such as call centers, it may be feasible to determine the general shape and determine whether a simple family, such as the exponential, is a good approximation. If so, by using data on the proportion of customers that are satisfied within x minutes is sufficient to estimate a, while b can be set to reflect the relative value of different call types. Even if a curve cannot be estimated in practice, we can still draw basic insights into how DTC workers should behave and how DTC work differs from NDTC work by using such a curve as the basis for structural modeling, as we show in the next section.

4 Control Policy of DTC System

To develop a basic model of work with discretionary task completion, we consider a worker who staffs a single station that receives jobs that arrive according to a Poisson process with rate λ. Jobs have the DTC behavior depicted in Figure 1.*Right* and are processed according to a first-come-first-served (FCFS) discipline. A holding cost h per unit time is charged for each task while it is in the system. If the worker spends τ units of time on a task, it generates value $f(\tau)$, where $f(\tau)$ is defined in Equation 1. To guarantee a minimum level of average quality f_{min}, we assume that all tasks are required to be given at least τ_{min} units of processing (where $\tau_{min} \geq 0$). The greater τ_{min} is, the less flexibility the worker has in adjusting the service time of each task.

Our goal is to find a task completion policy that maximizes the value generated per unit time. In a call center, for example, this leads to rules that determine the amount of time an agent should spend on the current caller given knowledge of the number of customers waiting in queue.

4.1 Model Formulation

To develop an optimization model, we discretize the time horizon into equal, non-overlapping infinitesimal intervals $\delta\tau$, where $\delta\tau$ is small enough to ensure the probability of having more than one arrival during $\delta\tau$ is almost zero. Thus, the probability of having one arrival during interval $\delta\tau$ is $\lambda\delta\tau$. We define $t_{min} = \tau_{min}/\delta\tau$. We can then formulate the problem as a Markov Decision Process (MDP) in which:

- *State Space S* includes states (n, t), where n is the number of jobs in the system, and t is the number of time intervals that the job under service has been worked on. Both n and t are non-negative integers.
- *Decision epochs* are the beginning of every period.
- *Action space A* includes actions *Keep*, and *Release*. Action "Keep" requires the worker to continue working on the current job for one more period, while action "Release" requires that the worker stop working on the current job and release it.

Without loss of generality, we assume that the worker can release a job at the beginning of a period, but jobs arrive only at the end of a period. We define $V(n, t)$ as the profit at state (n, t) per transition (with length δt). The optimality equation of the MDP model for $n \geq 1$ and $t \geq t_{min} + 1$ is therefore as follows:

$$\delta\tau\ g + V(n,t) = Max \begin{cases} -nh\delta\tau + (1-\lambda\delta\tau)V(n,t+1) + \lambda\delta\tau V(n+1,t+1) & : \ \text{Keep} \\ -(n-1)h\delta\tau + f(t\delta\tau) + (1-\lambda\delta\tau)V(n-1,\mathbf{1}_n) + \lambda\delta\tau V(n,\mathbf{1}_n) & : \ \text{Release} \end{cases}$$

where g is the optimal average profit per unit time, and $\mathbf{1}_n = 0$ if $n = 1$, and $\mathbf{1}_n = 1$ if $n \geq 2$. For $n \geq 1$, and $t \leq t_{min}$, we have

$$\delta\tau\ g + V(n,t) = -nh\delta\tau + (1-\lambda\delta\tau)V(n,t+1) + \lambda\delta\tau V(n+1,t+1),$$

and for $n = 0$,

$$\delta\tau\ g + V(0,0) = (1-\lambda\delta\tau)V(0,0) + \lambda\delta\tau V(1,0).$$

4.2 Optimal Control Policy

Since task completion time is determined by the worker, utilization in a DTC system is also controlled by the worker. However, because of our assumption that all jobs must receive at least minimal processing, we define $\rho_{min} = \lambda\tau_{min}$. It is clear that the system is only stable when $\rho_{min} < 1$.

The structure of the optimal service policy in a stable system is characterized by Theorems 1, 2 and 3. For ease of presentation, we omit $\delta\tau$ in the value function $f(t\delta\tau)$. Therefore, $f(t)$ represents the value when the job is released after $t\delta\tau$ units of time. The proofs of our analytical results are available in online appendix at http://users.iems.northwestern.edu/∼gigi/research.htm.

Theorem 1 *If $\rho_{min} < 1$, then there exists an optimal stationary policy that maximizes the total average profit per unit time. Furthermore, the gain rate g is constant and the value iteration algorithm converges.*

To further characterize the optimal policy, we classified all systems into two categories based on whether or not $f(t) < ht$ for all t. For systems where this condition holds, it is easy to show that the profit function is negative and decreasing in t and hence we prove that:

Theorem 2: *If $f(t) < ht$ for all t, then the optimal policy is to spend the minimum amount of time (i.e., t_{min}) on each job.*

For systems where $f(t) > ht$ for some t, we begin by establishing the following properties of the value function:

Proposition *If $f(t) \geq ht$ for some t, then the optimality equation has the following properties:*

C1: $V(n+1,t) - V(n,1)$ *is non-increasing in n for $n \geq 1$ and $t \geq t_{min}+1$*

C2: $V(n,t) - V(n,t+1)$ *is non-increasing in n for $n \geq 1$ and $t \geq 1$*

C3: $V(n,t) - V(n-1,t)$ *is non-increasing in n for $n \geq 2$ and $t \geq 1$*

C4a: $V(1,t+1) - V(2,t+1) + V(1,1) - V(0,0) \geq 0$ *for $t \geq t_{min}$*

C4b: $V(2,t+1) - V(3,t+1) + V(2,1) - V(1,0) \geq 0$ *for $t \geq t_{min}$*

C5: $V(n,t+1) - V(n,t) \geq 0$ *for $n \geq 1$ and $t \geq 1$*

C6: $(1-\lambda\delta\tau)[V(n-1,1) - V(n,t+1)] + \lambda\delta\tau[V(n,1) - V(n+1,t+1)] + f(t)$ *is non-decreasing in t for $n \geq 2$ and $t \geq t_{min}+1$*

Since Condition **C1** holds, we know that if it is optimal to keep a job in state (n,t), where $n \geq 2$, then it is also optimal to keep a job in state $(n-1,t)$. For $n < 2$, we can prove using a sample path argument that there exists an upper bound on the amount of service the server should provide.

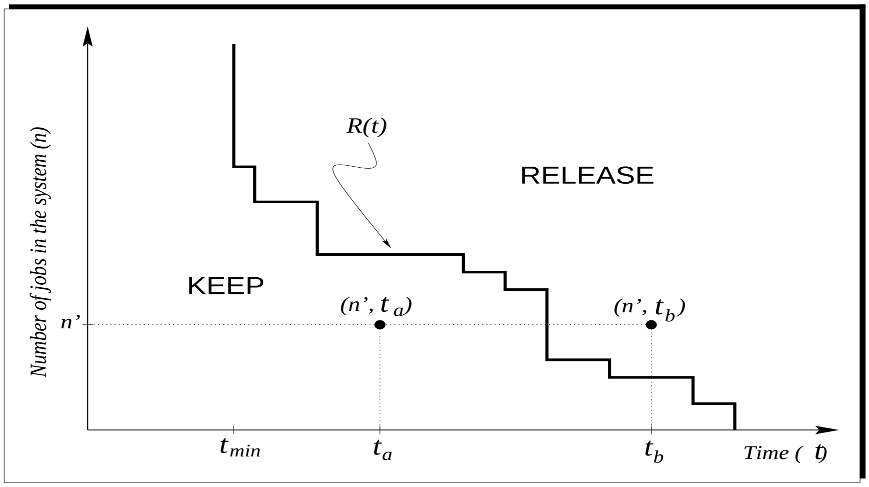

Figure 2: Typical structure of the worker's optimal policy.

Theorem 3 *If $f(t) > ht$ for some t, then there exists a non-increasing threshold function $n = R(t)$ such that it is optimal to keep the job in state (n, t) if $n \leq R(t)$ and release the job if $n > R(t)$.*

Figure 2 shows the typical structure of the optimal service policy. The boundary that separates the "Keep" region from the "Release" region is composed of state-dependent thresholds that determine the optimal decision in different states. For example, in state (n', t_a) where $t_{min} \leq t_a$, the current job has already been processed for t_a units of time. In this state, since the potential value generated by working on the job for one more period outweighs the holding cost of keeping n' jobs in the system, it is optimal to keep the job. On the other hand, in state (n', t_b), since the potential benefit of spending one more period processing the current job does not exceed the holding cost of keeping n' jobs in the system, the optimal decision is to release the current job.

The structure of the optimal control policy demonstrates how DTC systems mitigate congestion by varying the quality of outputs. When the queue is long, the system lowers product quality by processing jobs for a shorter amount of time. This helps the system reduce congestion and keep the holding cost low. In a call center, this implies that during peak demand times, an agent will tend to provide less service (shorter calls) than during low demand times in order to prevent the queue from growing so long that waiting time becomes unacceptable. Hence, depending on the relative costs of waiting versus hurrying, adjusting the quality buffer to avoid excessive time buffering can be an appropriate strategy for a call center.

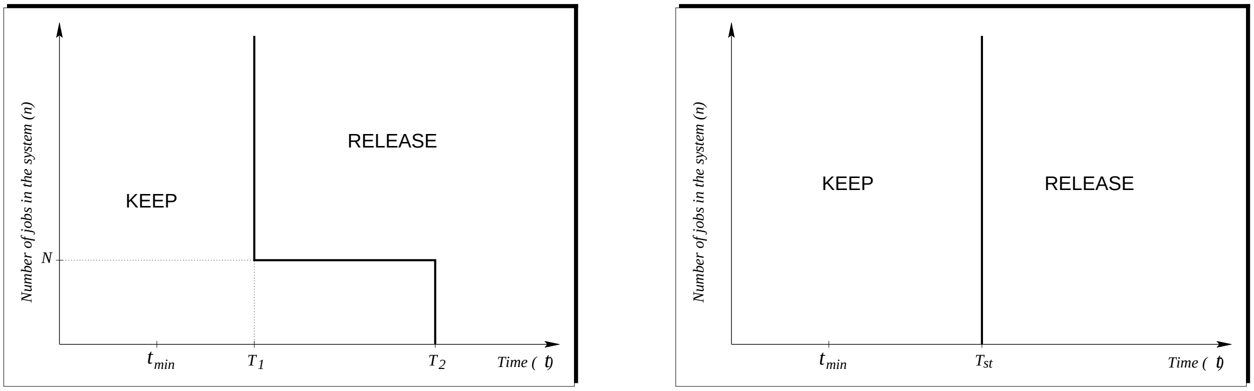

Figure 3: *Left:* The double-threshold (DT) policy, *Right:* The single-threshold (ST) policy.

4.3 Performance of Heuristic Policies

The optimal policy in Figure 2 presents two major difficulties: (i) it has a complex structure which may make implementation impractical, and (ii) jobs are released at different points of time during their processing, resulting in inconsistent quality, which might erode long-term customer satisfaction. These issues motivated us to consider heuristic policies that may be better suited to practice.

The first heuristic is called the *Double-Threshold (DT) policy* and has its structure illustrated in Figure 3 *Left*. The DT policy has three parameters (N, T_1, T_2), where $T_2 \geq T_1 \geq t_{min}$. Under this policy, the server works on the current job for T_2 units of time as long as there are less than or equal to N jobs in the system. Otherwise, the current job is processed for T_1 units of time. For example, in an emergency room operating under the DT policy, physicians can choose to provide either basic treatment or extended treatment depending on the number of patients awaiting for care. If the emergency room is busy, physicians will only provide basic treatment to remove immediate threats to life before moving on to the next patient. If the emergency room is not busy, physicians will provide extended care in which detailed diagnosis and additional treatments are conducted to enhance the patients' long-term well-being.

The second heuristic, called the *Single-Threshold (ST) policy*, is a special case of the DT policy for which $T_{ST} = T_1 = T_2 \geq t_{min}$. Under this policy, all jobs are processed for the same amount of time T_{ST}. Hence, the ST policy corresponds to the policy for NDTC work systems in which processing times are independent of the work backlog. That is, there is discretion over the completion time of a task but we choose not to use it. For example, a physician in general practice might choose to see patients for the specified amount of time regardless of the number of people in the waiting room.

We studied the performance of the DT and ST heuristics relative to that of the optimal policy in various systems in order to describe the conditions under which these heuristics perform well or poorly. We used a value iteration algorithm for each case to obtain the optimal profit, as well as the profit under the DT and the ST heuristics. We define $\tau_{max} = arg_{\tau}\{f(\tau) = 0.99b\}$ and $t_{max} = \tau_{max}\delta\tau$. Therefore, t_{max} is the maximum number of time intervals the worker is allowed to spend on a job, which is also the time required to obtain 99% of the maximum profit (we use 99% instead of 100% because of the asymptotic property of exponential function). The parameters we investigated are:

- *Shape of value function A:* Define A as,
$$A = \frac{F(\tau_{max}) - \tau_{max}(b/2)}{\tau_{max}(b/2)},$$
where $F(x) = \int_0^x f(\tau)\, d\tau$. This measure characterizes the shape of the value function. At $A = 0$, the value-time relationship is linear from $\tau = 0$ to τ_{max}. As A increases, quality increases more rapidly with time. We considered values $A = 0.24$, 0.59 and 0.79, to represent situations ranging from strongly "DTC" to nearly "NDTC" work.

- *Minimum Traffic Intensity* ρ_{min}: The parameter ρ_{min} is a relative measure of the minimum level of system congestion. We considered values from 0.1 to 0.55 in increments of approximately 0.15, depending on A and f_{min}.

- *Holding Cost to Maximum Value Ratio h/b:* This ratio characterizes holding cost relative to (maximum) job value. We considered values 0.01, 0.02 and 0.04. A small ratio (i.e., 0.01) represents situations where the customer dissatisfaction generated from a long wait is small compared to the satisfaction gained from the best possible service.

- *Minimum quality* f_{min}/b: This ratio measures the minimum quality a customer is willing to accept relative to the maximum quality possible. We considered values of 20%, 40% and 60%.

- *Process flexibility φ:* Define $\phi = (\tau_{max} - \tau_{min})/\tau_{max}$. This measure represents the amount of discretion a worker has in choosing processing time. We considered values ranging from 0.6 to 1 in increments of 0.2, depending on A and f_{min}. The larger this value is, the more discretion a worker has in selecting processing time.

For computing purposes, the state space was truncated at $N = 30$ after determining by experiment that increasing N beyond 30 does not have a significant effect on the average profit. We set the unit time $\delta\tau$ such that the probability of more than one arrival during a period of length $\delta\tau$ is less than 0.001. Hence, $\delta\tau = 0.64$ when the arrival rate $\lambda = 0.1$; $\delta\tau = 0.32$ when the arrival rate $\lambda = 0.2$, and so on. The number of time units for the value-time curve was truncated at τ_{max}. For example, when $A = 0.79$ and $\lambda = 0.2$, it is straightforward to compute $\tau_{max} = 6.2$, and hence $M = 6.2/0.32 = 20$. By repeating this process for different values of (N, T_1, T_2) and T_{ST}, we obtained the profit under the *best* DT and ST policies.

Our numerical study of 108 cases showed that the DT policy performs well under most parameter settings. We observed that the DT policy results in an average of only 0.5% less profit per unit time than the optimal policy. The maximum difference we observed was 2%. The reason that the DT policy is very close to optimal under most conditions is because it is able to adjust the quality level to respond to changes in the work backlog. Evidently, even the crude adjustment afforded by only two process time settings is enough to take advantage of quality flexibility.

The ST policy, however, only performs well under a restricted range of parameter settings. Our numerical study showed that the ST policy results in an average of 5% less profit per unit time than the optimal policy, with a maximum percent difference of 18%. We found that the ST policy performs well in: (*i*) systems with high A, which represent work systems that approach their maximum value quickly, (*ii*) systems with small h/b, where holding cost is not significant compared to the potential quality earned, (*iii*) systems with small process flexibility (ϕ), and (*iv*) systems with fairly high minimum traffic intensity (ρ_{min}). One of the key factors that affects the ST policy performance is the shape of the value-time function $f(\tau)$. The closer $f(\tau)$ is to a step-function (e.g., large A and small process flexibility ϕ), the better the ST policy performs. We can summarize the results of these tests in the following observations:

Observation 1 *The DT heuristic performs almost as well as the optimal service policy in single station single-class DTC systems.*

Observation 2 *The ST heuristic performs reasonably well in DTC systems only when the tasks are fairly similar to NDTC tasks.*

The difference in process time control between the heuristic policies and the optimal policy results in different levels of quality variation. Workers under the optimal policy have the highest variation in task processing time, since it can range from t_{min} to t_{max} depending on what is the best release time based on queue length. In contrast, the ST policy imposes absolute control on

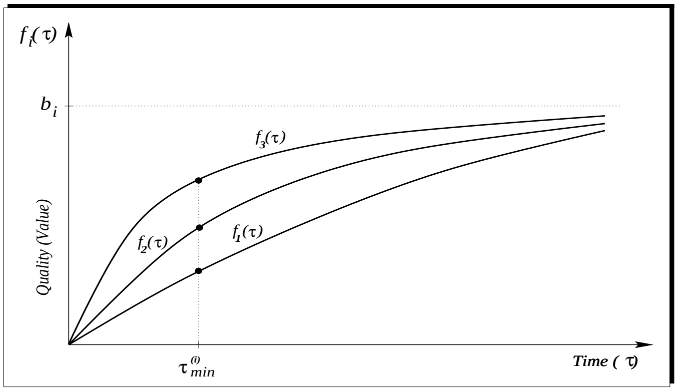

Figure 4: Increase in Capacity, i.e., profit functions $f_1(\tau)$, $f_2(\tau)$, and $f_3(\tau)$.

processing time and results in no quality variation at all. The DT policy has an intermediate level of quality variation. If high variation affects customer perceptions of fairness, and thereby has a long-term effect on profitability, the DT heuristic might actually be preferred to the optimal policy in practice.

5 Capacity in DTC Systems

It is well-known that, increasing system capacity reduces congestion in NDTC environments (Hopp and Spearman 2000, Chapter 9.) One might reasonably expect the same to be true in DTC systems. However, since DTC systems introduce quality as an additional variability buffer, we need to look closely at the factors that affect system congestion in order to understand the role of capacity. In this section, we perform numerical experiments to investigate the question: does a capacity increase in a DTC work system always decrease queue length?

In single-worker NDTC systems, an increase in capacity corresponds to a decrease in the average effective processing time of a job. In DTC systems, since capacity is defined by the value-time curve, an increase in capacity can be measured in various ways.

For our purposes, we define increase in capacity as an upward shift in the value-time curve (see Figure 4). That is, the new curve completely dominates the old one because at every point in time, the value earned from working on a job is greater (or equal to) after a worker's skills improved through training or learning. Formally, $f_1(\tau) \leq f_2(\tau) \leq f_3(\tau)$ for all τ. For an exponential value-time function, an increase in capacity can be represented by an increase in the parameter a.

To investigate the effects of changing capacity, we conducted another set of numerical tests using an exponential value-time function with values of $a = 0.25, 0.5, 1$, which refer to as low, medium and high capacity, respectively. To isolate the effect of capacity, we consider value-time functions $f_i(\tau)$ with the same τ^i_{min} and b_i for $i = 1, 2, 3$. We then used MDP to compute the optimal service policy for single station DTC systems with the various capacities. We used the convergence requirement and truncated state space we described in Section 4. To calculate the average queue length (i.e., congestion), we fed the optimal policy back into the MDP program as the decision structure but set $f(\tau) = 0$ for all t, and $h = 1$. Hence, the average profit obtained from the second value iteration gave the average queue length.

Table 1. The effects of increase in capacity on queue length

Min. Traffic	Holding	Rate		Expected	% Change in E[Queue]	
Intensity	Cost	Parameter	Average	Queue	Compared with	Compared with
ρ_{min}	Ratio h/b	a	Profit/Time	Length	$a = 0.25$	$a = 0.5$
0.3	0.02	0.25	12.6	1.35	–	–
		0.5	19.5	1.49	**11%**	–
		1	25.0	1.26	-6%	-16%
	0.04	0.25	10.3	1.05	–	–
		0.5	16.9	1.11	**6%**	–
		1	22.8	0.96	-8%	-13%
	0.06	0.25	8.3	0.90	–	–
		0.5	14.9	0.94	**4%**	–
		1	21.1	0.78	-13%	-17%
0.4	0.02	0.25	14.2	1.36	–	–
		0.5	23.2	1.68	**24%**	–
		1	31.7	1.54	-14%	-8%
	0.04	0.25	11.8	1.10	–	–
		0.5	20.4	1.23	**12%**	–
		0.4	29.0	1.17	**7%**	-5%
	0.06	0.25	9.7	1.02	–	–
		0.5	18.1	1.05	**3%**	–
		1	26.8	0.98	-4%	-7%
0.5	0.02	0.25	15.3	1.41	–	–
		0.5	26.1	1.73	**22%**	–
		1	37.4	1.85	**31%**	**7%**
	0.04	0.25	12.8	1.18	–	–
		0.5	23.1	1.33	**13%**	–
		1	34.3	1.36	**15%**	**2%**
	0.06	0.25	10.5	1.15	–	–
		0.5	20.7	1.11	-4%	–
		1	31.8	1.12	-3%	**1%**

Table 1 presents some of the results from our experiments that illustrated some surprising behavior. In sharp contrast to the behavior of NDTC work systems, an increase in capacity in a DTC work system sometimes results in an *increase* in average queue length (hence average waiting times). Table 1 shows that this counter-intuitive phenomenon is more prevalent in systems with: (i) high minimum traffic intensity, ρ_{min}, and (ii) relatively low holding cost to value ratio, h/b.

Hence, we can make the following observation:

Observation 3 *In DTC systems, congestion may intensify when capacity increases.*

The underlying reason for this behavior is the presence of quality as a variability buffer in discretionary task completion systems. This makes it possible to take advantage of an increase in capacity to either reduce queueing (and hence holding cost), or increase average quality level (and hence value), or both. The optimal mix of these depends on the relative costs. When the ratio of holding cost to value (h/b) is high, it makes senses to apply the additional capacity to reduce congestion. But when h/b is low, the extra capacity should be used to improve quality. In some cases, it can even make sense to actually increase queueing in response to a capacity increase in order to facilitate an even greater increase in quality.

Finally, we observe by comparing the last two columns of Table 1 that the increase in congestion is more significant when capacity is increased from low to medium ($a = 0.25$ to 0.5) than when it is increased from low to high ($a = 0.25$ to 1.0.) The explanation for this is that, because the value-time function is concave, using capacity to increase value exhibits diminishing returns. Hence, as the capacity increase grows larger, the optimal policy will eventually dedicate some of the additional capacity to reducing queue length.

We also studied the effect of increase in capacity by adding an additional server to a single-server system. Using a new MDP model, we determined the optimal policy and the corresponding queue length. We observed in similar phenomenon, namely the queue length may increase as capacity increases (through additional server).

From a managerial perspective, this result is potentially significant. For example, in a call center, adding capacity (via staff or by means of technology) might be expected to shorten customer waiting times. But our results suggest that if management insists on shorter delays, they may induce a suboptimal solution. Hence, it is important for management to measure and control both quality and responsiveness in setting improvement goals.

6 Value of Information

In this section, we consider the value of customer information and task sequencing issues for DTC systems. To accomplish this, we consider a simple two-class model. Similar to the single-class model, we assume that tasks of types 1 and 2 arrive according to Poisson processes with rates λ_1 and λ_2, respectively. A task of type q ($q = 1, 2$) has an average value function $f_q(\tau)$ and minimum processing time τ^q_{min} which guarantees a level of average quality f^q_{min}.

Many operations systems involve multiple task types. For example, in a call center, agents often handle multiple types of calls. Since the different types of calls require different conversation lengths to satisfy customers, they will have different value functions. The value functions may also vary due to different level of experience an agent has in handling the call types. Similarly, in an emergency room, physicians may classify the cases as life threatening or non-life threatening. The different case types require different procedures, and hence different service times.

We consider our model under the following information scenarios:

- *Complete Information:* The task type is known for each job as soon as it enters the system.
- *Partial Information:* The task type of an entering job is not known and is only revealed when the worker starts processing it.

Note that under complete information the worker knows the number of jobs of *each type* in the queue, while under partial information, he/she only knows the *total* number of jobs in the queue, but not the number of each type or their sequence in the queue.

6.1 Partial Information Case

We first consider a single worker system in which the task type is not revealed until service begins, that is, the partial information case. This situation arises in call centers in which callers do not identify themselves or their needs until they speak to an agent.

We developed a Markov Decision Process (MDP) model to determine an optimal service policy that considers the current state (type of job currently under process and the queue length) and specifies how much more time the worker should spend on the current job in order to maximize the value generated per unit time. As we did in the single-class system in Section 4, we discretize time into equal, non-overlapping infinitesimal intervals $\delta\tau$, where $\delta\tau \rightarrow 0$, and define $t^q_{min} = \tau^q_{min}/\delta\tau$. But now, since the overall arrival rate is $\lambda = \lambda_1 + \lambda_2$, the probability that an arriving job is of type 1 is $\gamma = \lambda_1/\lambda$ and $(1-\gamma)$ is the probability of type 2. We define the minimum utilization $\rho_{min} = \sum_{q=1}^{2} \lambda_q t^q_{min}$.

The decision epochs in the MDP are at the beginning of each period and the available actions are *Keep* and *Release*. Since the mix of job types in the queue is affected by the process times of the jobs, the holding cost cannot be computed as a weighted average of the job holding costs with weights proportional to arrival rates. Hence, we need to define the state of the system to be $(x_1, x_2, x_3, ..., t)$, where $x_j = q$ indicates that the job at the j^{th} location in queue is type q, in order to compute the holding cost when $h_1 \neq h_2$. However, this information should not be used to obtain the optimal action in each state because of our assumption that it is not available to the worker in the partial information case.

We use a linear programming (LP) approach to tackle the problem. We start with the standard LP for the full information MDP (to ensure the proper holding costs), and then add constraints to enforce a FCFS service policy where the optimal policy only depends on (n, t, q). To accomplish this, we define state i in the LP by vector $(x_1, x_2, x_3, \ldots, x_{n_i}, t_i, a_i)$, where x_j ($x_j = 1$ or 2) is the job type at the j^{th} location in the queue, t_i is the amount of time spent on the current job, $a_i \in \{Keep, Release\}$ is the action chosen in state i, and n_i is the total number of jobs in the system in state i. Note that x_1 indicates the type of the task under process. We define:

- y_i as the probability of being in state i at any instant, and $\mathbf{Y}$ as the vector of y_is,
- C_i as the profit of being in state i for duration $\delta\tau$, which is the revenue generated from job processing at release minus the sum of holding costs of type 1 and type 2 jobs in the system per $\delta\tau$.
- p_{ij} as the transition probability from state i to state j, and $\mathbf{P}$ is the square transition probability matrix composed of p_{ij}, and
- z_i as a binary variable which has value 1 when the probability of being in state i is greater than zero, and is zero otherwise.

We formulate the LP to obtain the optimal expected profit in partial information system as follows:

$$Max \qquad \sum_i C_i \; y_i$$

$$s.\,t.: \quad \sum_i y_i = 1 \tag{2}$$

$$\mathbf{Y} = \mathbf{YP} \tag{3}$$

$$y_i \leq z_i \qquad \forall\ i \tag{4}$$

$$y_k \geq \mathcal{M}\ z_i \qquad \forall\ i,\ k, \quad \text{where } n_i = n_k \text{ and } a_i = a_k \tag{5}$$

$$y_i \geq 0 \qquad \text{for all } i \tag{6}$$

where $\mathcal{M}$ is a *very small* positive number with magnitude in the range from 10^{-7} to 10^{-9}. Constraints (2), (3), and (6) are standard MDP constraints (see Hiller and Lieberman 2001). Constraints (4) and (5) are added to force the system to ignore the composition of the queue in optimizing the profit. This is done by using a binary variable z_i for each state i. When y_i is positive, the corresponding action is optimal and Constraint (4) gives z_i the value of one. Then, since $\mathcal{M}$ is a very small positive number, Constraint (5) requires all the states with the same queue length and action to have their y_i to be positive and smaller than one. To satisfy these constraints, all states with the same queue length must use the same action. Despite the fact that information on queue composition is available (to ensure correct holding cost calculation), the system makes decisions as if it is unaware of it and bases its decision only on queue length. As a result, it behaves exactly as if it had only the information available in the partial information case.

6.2 Complete Information Case

We now consider the complete information case, in which the worker knows the number of jobs of each type in queue. In a call center system, this information could come from an automatic filtering process that requires customers to select from two call types (e.g., account billing inquiry and service inquiry.) By routing calls to different queues, the system allows the agent to choose the type of customer to process next. The agent can also use the information on the number and type of customers in queue to determine how long to process each task. Similarly, in an emergency room, case type information could come from a preliminary evaluation by other medical personnel at registration. The patients can then be prioritized accordingly for treatments.

To develop a Markov Decision Process (MDP) model of the single station, two-class DTC work system, we define the following:

- *State Space S* includes states (n_1, n_2, t, q), where n_1 and n_2 are the number of type 1 and type 2 jobs in the system, respectively, t is the number of time intervals the job under service has been worked on, and q is the type of job under service. The variables n_1, n_2 and t are non-negative integers.

- *Decision epochs* are the beginning of each period.

- *Action space A* includes actions *Keep (K)*, *Release and Process Type 1 (R1)*, and *Release and Process Type 2 (R2)*. Action "Keep" requires the worker to continue working on the current job for one more period. Action "Release" requires the worker to stop working on the current job and release it. The worker can then choose between a type 1 and type 2 job for processing if jobs are available. Otherwise, the worker becomes idle.

Letting h_i denote the holding cost per unit time of task type i, we can write the optimality equation of the MDP model for the case where the job in process is type 1 ($q = 1$), and $n_1 \geq 2$, $n_2 \geq 1$ and

$t \geq t_{min} + 1$ as follows:

$$\delta\tau g + V(n_1, n_2, t, 1) = max \begin{cases} \lambda\delta\tau\Big[\gamma V(n_1+1, n_2, t+1, 1) + (1-\gamma)V(n_1, n_2+1, t+1, 1)\Big] & : \textit{(K)} \\ \quad + (1-\lambda\delta\tau)V(n_1, n_2, t+1, 1) - (n_1 h_1 + n_2 h_2)\delta\tau & \\ & \\ \lambda\delta\tau\Big[\gamma V(n_1, n_2, 1, 1) + (1-\gamma)V(n_1-1, n_2+1, 1, 1)\Big] & : \textit{(R1)} \\ \quad + (1-\lambda\delta\tau)V(n_1-1, n_2, 1, 1) - \Big[(n_1-1)h_1 + n_2 h_2\Big]\delta\tau & \\ \quad + f_1(t) & \\ & \\ \lambda\delta\tau[\gamma V(n_1, n_2, 1, 2) + (1-\gamma)V(n_1-1, n_2+1, 1, 2) & : \textit{(R2)} \\ \quad + (1-\lambda\delta\tau)V(n_1-1, n_2, 1, 2) - \Big[(n_1-1)h_1 + n_2 h_2\Big]\delta\tau & \\ \quad + f_1(t) & \end{cases}$$

where g is the optimal average profit per unit time. The optimality equations for the other three cases where $n_1 = 1$, $n_2 \geq 1$ and $t \geq t_{min} + 1$; or $n_1 \geq 1$, $n_2 = 0$ and $t \geq t_{min} + 1$; or $n_1 = 1$, $n_2 = 0$ and $t \geq t_{min} + 1$ can be written in similar fashion. For the cases where $t \leq t_{min}$, optimality equations can also be written in similar fashion by considering only action *Keep*.

For a fair comparison with the capacitated partial information system, we limit the maximum number of jobs in the system to be five, where $n_1 + n_2 \leq 5$. The above MDP enables us to compute the optimal profit for the complete information case.

6.3 Job Type Information in DTC versus NDTC Systems

Now that we have models for both the partial and complete information cases, we can investigate the value of information about job types (VOI) in queue. In addition, we can ask whether the benefit of having this information is higher in DTC or NDTC work systems. We define the value of information as the percent increase in profit that results from using information about queue composition. Letting P be the average profit per unit time, we define the value of information as:

$$\frac{P_{\text{complete info.}} - P_{\text{partial info.}}}{P_{\text{partial info.}}}$$

For the numerical study, we studied 128 cases for each system (DTC and NDTC). We used the convergence requirement and truncated state space used previously in Section 4. Using the same parameters that we defined for the single-class study in Section 4.3, we studied the following parameter settings for each job class $q = 1, 2$:

- *Shape of value function A_q:* We considered values $A_q = 0.24$ and 0.79.
- *Minimum Traffic Intensity ρ_{min}*: We considered values from 0 to 0.8 in increments of approximately 0.2.

- *Holding Cost to Maximum Value Ratio h_q/b_q:* We considered values 0.01 and 0.04.
- *Minimum quality f^q_{min}/b_q*: We considered the values 20% and 60%.
- *Process flexibility ϕ_q:* We considered values ranging from 0.6 to 1 in increments of approximately 0.2.

Specific to the multi-class system, we consider the following parameters that describe the relationship between the different job classes:

- *Probability of a Type 1 Arrival* γ: We set this parameter to $\gamma = 0.5$. Our tests focused on cases with equal arrival probabilities for type 1 and type 2 jobs, in order to emphasize the impact of job type differences (e.g., differences in holding costs and value-time functions).
- *Ratio of Maximum Profits* b_1/b_2: We considered the values 1 and 4. When this ratio is 1, both value-time functions have the same maximum values. Depending on the parameter a, this may lead to $f_1 < f_2$ for some t. However, when the value is 4, we have $f_1 > f_2$ for all t.
- *Ratio of Profit Function Shape Parameter* A_1/A_2: We considered values 0.25, 1 and 4. This ratio describes the relative shapes of the value-time functions. The greater the ratio, the faster the type 1 value-time function increases over time $(df_1(t)/dt)$ relative to the type 2 value-time function $(df_2(t)/dt)$.

Computing VOI in DTC System

First, we studied the value of information about task types in queue for DTC systems. The profit for the complete information case was obtained by solving the MDP in Section 6.2 for DTC capacitate systems. Then, we compared this with the profit for the partial information case by solving the LP in Section 6.1. We implemented the LP for systems with maximum queue length of five using AMPL (Fourer et. al. 1993). We truncated the queue length to a maximum of 5 due to computational limitations. In systems with a large number of states, each state would have a very small probability, particularly for those corresponding to long queues. Due to precision limitations in computers, a state with less than 10^{-5} probability may be rounded to zero and cause its corresponding integer variable z_i to be set to zero. This would destroy the integrity of Constraint (4) and result in an incorrect solution.

To investigate the value of job type information in systems with larger state spaces, we developed an approximation method for the partial information system to obtain upper bounds on the VOI for DTC systems. The approximation method uses the results for the $M/M/1$ queueing system with two classes of customers (see Gross and Harris 1985, equation 3.45) and our MDP to obtain a lower bound on the profit. We do not present the details of the approximation method due to space constraints.

Computing VOI in NDTC System

Second, in order to compare the value of information in DTC and NDTC environments, we specified a step function for the NDTC case (see Figure 1.*Left*) that "matches" the value-time function in the DTC case. We did this by fitting a step function to a given value-time function such that the step function matched the optimal ST policy for that function.

Using these step functions as the value-time functions, the profit in the NDTC environment for the complete information case was found by solving the MDP in Section 6.2. For the partial information case, we modified the LP described in Section 6.1 to restrict attention to policies that process tasks of the same class for the same amount of time (i.e. using the ST policy). Then, for each case, we enumerated all possible combinations of ST policies for the two job classes (i.e. from τ^q_{min} to τ^q_{max}) and chose the best outcome as the profit under partial information.

Our experiments showed that the value of job type information in NDTC systems is at least 2 times greater than thatin DTC systems. For instance, in our exact LP model, the VOI in NDTC systems can be up to 19%, which is 5 times greater than the maximum VOI in DTC systems. In our approximate model, the VOI in NDTC systems can be up to 23%, which is 2.5 times greater than the maximum upper bound on the VOI in DTC systems. Hence, we conclude:

Observation 4 *Job type information is less valuable in a DTC environment than in a NDTC environment.*

The underlying reason for observation 4 is that DTC systems can compensate for a lack of information by controlling queue length through adjustment of quality output (i.e., by changing processing times). When information is not available in NDTC systems and a 'wrong' job is chosen for processing, there is no way to compensate. For example, if a job with a very long processing time is chosen when the queue is long, the system will suffer by keeping many jobs waiting. In contrast, in a DTC system, this job could be released more quickly in order to reduce waiting time of the jobs in queue. The lack of flexibility in NDTC environments makes job type information more valuable as a means for avoiding costly mistakes.

As an example, consider a call center where agents provide technical support for a software product. In such a system, customers are likely to have a wide range of needs, from assistance in learning how to use basic functions to suggestions for improving software performance. But it may be difficult to differentiate the actual needs of the customers without conversing with them. Hence, separating the calls into different queues as they enter the system could be very costly. Fortunately, our results suggest that the value of knowing the call types in queue for these discretionary completion times tasks is small. As a result, the benefit of modifying the system from a partial information setting to a complete information setting is unlikely to warrant the cost.

In contrast, consider call center agents who are responsible for basic account support, including assisting customers to change their account information and answering service inquiries. Each of these call types have standard quality measures since a request is either completely fulfilled or not fulfilled at all. Hence, the value-time curves will have shapes closer to a (NDTC) step-function. Our results suggest that the value of information on call types in queue for these non-discretionary completion time tasks could be large. Fortunately, in such a setting, the call types are more clearly defined and can be easily differentiated using an automatic filtering system (e.g., "press one to change your account information.") Consequently, it seems that technology for classifying customers is more beneficial in precisely the environments where it is most feasible.

7 Conclusions

In this paper, we have presented analytical models that capture one important characteristic of many operation systems - discretionary task completion. Our main insight is that, in work systems with discretionary task completion times, it is attractive to adjust processing time, and hence job quality, in response to system congestion. In factory physics terminology, quality is a variability buffer (along with capacity, inventory and time). The presence of this fourth type of variability buffer causes systems with discretionary completion times to exhibit some very different behaviors from systems with non-discretionary completion times. Most strikingly, in contrast with DTC systems, congestion may actually intensify as we increase capacity in NDTC systems. This occurs when value gained from higher quality overrides the increase in holding cost. The implication from a managerial standpoint is that quality must be incorporated into the evaluation of operational improvements in systems with discretionary completion times.

By considering a two-job-class system, we also showed that the ability of DTC systems to leverage quality flexibility makes their performance less dependent on job-type information than less flexible NDTC systems. This implies that investment in technology to identify job types in queue is most attractive in systems where tasks are fairly similar to NDTC tasks.

The above insights were obtained by examining the optimal service policy for single-station DTC work system. However, the optimal policy is probably too complex to implement in real-world settings, particularly since value-time curves can only be approximated. We showed that the double-threshold heuristic achieves nearly optimal performance, while the single-threshold heuristic performs reasonably well only when tasks are fairly similar to NDTC tasks. Both of these policies would be simple to implement (possibly by trial-and-error) in practice.

The work in this paper represents a first step toward modeling and understanding service and professional work systems with discretionary process times. To develop a more comprehensive set of principles concerning design and management of service and professional work systems, further modeling work is needed.

For instance, it would be valuable to extend the modeling of the value-time relationship from deterministic to stochastic, where the level of quality output for a certain processing time is a random variable. For example, two patients in an emergency room could both have a minor leg injury (i.e., same task type), but have different age and medical histories which affects the effectiveness of the time a physician spends on them. A stochastic quality-time curve would capture another dimension of uncontrolled variability that may be important in DTC systems. With such a model, we could investigate interesting questions such as: how do the different variability buffers work together to mitigate the additional process variability? How does a time-based threshold policy perform differently than a quality-based threshold policy?

It would also be useful to extend the single station model to network models of multi-stage DTC work systems. An example of such network is an emergency health care system in which patients go through several stages, such as: pre-screening by nurses at registration, initial diagnosis by the doctor on duty, receipt of treatment possibly involving counsel by specialists, post-treatment monitoring and finally discharge. These stages could be iterative if complications arise during treatment.

In multi-stage systems, each stage has a distinct purpose and requires a different skill set. As such, they raise questions about interactions and impact of discretionary task completion at the different stages; for example, what is the best way to accommodate variable quality from upstream stages at downstream stages? How do we define the bottleneck in such DTC work systems?

The long-term goal of research into DTC work systems is to gain insights into the best ways to configure and train service and professional workforces. By understanding the interactions between capacity, variability and customer service (as determined by both quality and responsiveness), DTC workers can make better time management decisions, while managers can make better decisions concerning worker cross training, information sharing and collaborative work policies.

References

Adler P., A. Mandelbaum A., V. Nguyen, and E. Schwerer. 1995. From project to process management: an empirically-based framework for analyzing product development time. *Management Science* 41(3). 458-484

Altoik T. 1996. Performance evaluation of manufacturing systems. New York: Springer-Verlag.

Askin R. G., and J. B. Goldberg. 2001. Design and analysis of lean production systems. Hoboken, NJ: Wiley.

Ata, B., and S. Shneorson. 2005. Dynamic Control of an M/M/1 service system with adjustable arrival and service rates. Working Paper.

Bailey D. E. 1998. Comparison of maufacturing performance of three team structures in semiconductor plants. *IEEE Transaction Engineering Management* 45(1):20-32.

Banker R., J. Field, and K. Sinha. 2001. Work-team implementation and trajectories of manufacturing quality: A longitudinal study. *Manufacturing and Services Operations Management* 3(10):25-42.

Boudreau J., W. Hopp, J. McClain, and I. J. Thomas. 2003. On the interface between operations and human resources management. *Manufacturing and Service Operations Management* 5(3):179-202.

Bureau of Economic Analysis, US Department of Commerce. 2001. Gross domestic product by industry. <http://www.bea.doc.gov/bea/dn2/gdpbyind_data.htm>. Accessed 2004 Dec 7.

Buzacott J. A. and J. G. Shanthikumar. 1992. Stochastic models of manufacturing systems. New York: Prentice Hall.

Debo, L. G., L. B. Toktay, and L. N. Van Wassenhove. 2004. Queueing for Expert Services. Working Paper.

Fourer R., M. G. David, and B. W. Kernighan. 1993. AMPL: A modeling language for mathematcial programming. Danvers, MA: Boyd and Fraser Publishing.

Gans N., G. Koole, and A. Mandelbaum. 2003. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management* 5(2):79-141.

General Motors. 1974. GM Public Interest Report, Detroit, MI.

General Motors. 2002. GM Corporate Responsibity Report, Detroit, MI.

Gross D. and C. M. Harris. 1985. *Fundamentals of Queueing Theory*, Second Edition. Wiley: New York.

Halevi G. 2001. Handbook of production management methods. Burlington, MA: Elsevier Butterworth-Heinemann.

Hammer M., and J. Champy. 1993. Reengineering the corporation: A manifesto for business revolution. New York: Haper Business.

Hopp W., and M. Spearman. 2000. Factory physics: Foundations of manufacturing management, 2nd edition. Burr Ridge, IL: McGraw-Hill.

Hillier F. S., and G. J. Lieberman. 2001. Introduction to operations research, 7th edition. New York: McGraw-Hill.

Krishnan, V., S. D. Eppinger, and D. E. Whitney. 1997. A model-based framework to overlap product development activities. *Management Science* 43(4):437-351.

Loch, C. H., and C. Terwiesch. 1999. Accelerating the process of engineering change orders: capactiy and congestion effects. *Journal of Product Innovation Management* 16(2):145-169.

Longenecker C., J. A. Scazzero, and T. C. Stansfield. 1994. Quality improvement through team goal setting, feedback, and problem solving: A field experiment. *International Journal of Quality and Reliability Management* 11(4):45-52.

Owen S. H., and W. C. Jordan. 2003. An extended framework for studying white-collar work. Working paper, GM Research Laboratories, Warren, MI.

Puterman, M. L. 1994. Markov Decision Processes. New York: Wiley.

Sennott, L. I. 1996. The convergence of value iteration in average cost Markov decision chains. *Operations Research Letters* 19:11-16.

Schultz K.L., Juran D.C., Boudreau J.W., McClain J.O., and Thomas L.J. 1998. Modeling and worker motivation in Just In Time production systems. *Management Science* 44(12):595-607.