Modeling and Analysis of Uncertain Time-Critical Tasking Problems (UTCTP)

Donald P. Gaver
Patricia A. Jacobs
Gennady Samorodnitsky

1. *The Problem*

Consider modeling and operational analysis of a generic asymmetrical *service-system situation* in which (a) Red *agents*, such as, military or facility-destructive hostile threats, arrive according to some partially known and possibly changing pattern in time and space; and (b) Reds have effectively limited unknown deadlines, or times of availability for Blue *service*, i.e. detection, classification, and attack in a military setting. Cases of known deadlines are important and somewhat analogous; see Lehoczky (1996, 1997a, 1997b) and Doytchinov *et al*. (2001).

Think of the Reds as presenting tasks to be performed, or to be subjects of service. In a military context Reds may be perceived enemy targets, but in a medical emergency room setting they're arriving casualties. In a call center they are requests for information; see Becker et al (2000). In a Homeland Security (HLS) scenario a Red may be a container ship approaching a port, or a truck approaching a border, either possibly carrying explosives or chemical-biological offensive agents. We consider the Blue problem of processing such Red tasks effectively and efficiently under time constraints and limited information, hence the necessity to control the amount of service given.

Appropriate service effort typically differs between task types; it may not always be completely provided, and may be partial and incomplete, owing to deficiency of time, information or resources. In general, task service is by a Blue force of task-server agents also of various types, possibly varying in number and organization, and at different locations, but which attempt to share information and the service burden. Such complex agent systems are considered elsewhere, using insights provided in this report.

Some General Questions: How to match the Red tasks to currently appropriate Blue servers? How many, and what types of, Blue servers are needed to cope with the range of Red demands? How adequate is Blue service of Red, where "service" here means neutralization of threat (or recovery of endangered isolated personnel, such as a downed pilot or downed aircraft human and other valuable and sensitive cargo), or stabilization of an injury, or identification and proper "decontamination" of a platform carrying dangerous cargo? There are many examples of the generic situation we consider.

The Blue military objective is to successfully service as many tasks as possible rather than to minimize queues, while hostile Reds attempt to avoid "service," at least until they can accomplish their purpose, often to damage Blue. The models presented and analyzed suggest Blue force requirements and capability combinations for confronting specified challenges with acceptable success rates.

To summarize, such service system issues arise ubiquitously in military operations of all kinds, as well as in Homeland Security (HLS) and in military force protection, emergency management situations, and many natural hazard response scenarios, such as after earthquakes or tidal waves. They also occur in call center design and operation,

wherein specialized operators are made available to assist users of new software issues (see Becker et al. (2000)).

The plan of this paper is as follows. In Section 2 we describe a simple version of the basic problem: A single-type Red task arrival stream confronts a single Blue server that can process just one Red task at a time; success probability is related to allocated processing time, but is considered fixed/constant for a given selected processing mode. The Red tasks each have randomly limited availability time for processing, so if that time for a task exceeds any waiting plus service allocation time then service is delivered with a possibly successful outcome. Otherwise, the task is lost (leaks through defenses, or dies while awaiting treatment, etc.).

In Section 3 the investigation is broadened to include several task types. We suggest approximations to the proportion of tasks that are successfully served. We examine Blue defense's decision options so as to achieve maximum success rate, i.e., minimum leakage probability. These are (a) to select for next service the waiting task with greatest chance of survival to be serviced, and (b) to allocate service resources so as to balance time spent on the currently served task against losing tasks waiting. Control Policies are proposed that may then be refined and evaluated by use of heuristic search procedures such as Genetic Algorithms, and by adaptations of Dynamic Programming.

2. ***One Red Task-Type vs. Single Blue Service Agent***

Consider the simplest case of a single Blue service-providing agent (BSA) confronting a random stream of identical loss-susceptible Red service-requiring agents (RSAs).

Model 1

Assume first that members of the RSA stream arrive at (enter the sector of) a single BSA. Rate of approach is λ; after arrival the nth-to-arrive RSA has a loss time L_n: unless served within time L_n, the task vanishes. Optionally, $\{\boldsymbol{L}_n\}$ is a sequence of independent identically distributed random variables, with expectation $E[\boldsymbol{L}_n] = q^{-1}$, but more generality is possible. For convenience, $\boldsymbol{L}_n$ may be exponentially distributed (Markov) with rate q. The BSA assigned service time to each RSA is a constant; this is effectively a setup time; the military version is called target mensuration and deconfliction, and refers to the provision of an estimate of the target/threat location by a human operator using sensor assets plus assurances that friendlies and neutrals are out of range; it should also account for weapon transit time to a target. In medical emergencies it may be initial diagnosis and stabilization of a new patient. We emphasize that in the present case it is immaterial whether a service is known to be completed, successfully or not, before the assigned service completes. Realistically, the BSA has the benefit of follow-up observations, which in military applications is called Battle Damage Assessment (BDA), these are also realistically, uncertain. Note that if follow-up can be conducted immediately, and service repeated if deemed necessary, the above setup replaces a single service attempt by a sequence of assigned services: "Shoot-Look-Shoot", in military jargon.

2.1 ***Fluid or Deterministic Approximation***

Let $R(t)$ be the number of RSAs present in the service region, adopting a simple fluid/deterministic model for the number of RSA tasks in the system at time t; i.e., $\{R(t), t > 0\}$ is just a real-valued function of continuous time, t. Then, we write

$$\frac{dR(t)}{dt} = l - q\,R(t) - \frac{R(t)}{1+R(t)}\frac{1}{t}, \tag{2.1}$$

where t is the constant assigned service time, and the effect of service congestion from server saturation is represented by the *Filipiak approximation* (Filipiak, 1988), the term $R(t)/(1+R(t))$; clearly this term is ~$R(t)$ for $R(t)$ small, and saturates to ~1 as $R(t)$ becomes large, thus reflecting the eventual service rate limitation to $1/t$. If desired, the constant parameters in (2.1) can be made time dependent.

Steady-State Success Rate

Let $p(t)$ denote the success probability of a task that completes service allocation before loss if t units of time are allocated to service. Then the success rate satisfies

$$\frac{dS(t)}{dt} = \frac{R(t)}{1+R(t)}\frac{p(t)}{t} \tag{2.2}$$

We can use (2.1) to obtain an approximation to the steady-state average number of tasks present in the system by setting the derivative equal to zero and solving for $R = R(\infty)$. Put $r = l\,t$, $g = q\,t$,

$$R(\infty) = \frac{2r}{1+g-r+\sqrt{(1+g-r)^2+4gr}} \tag{2.3}$$

Substituting the above estimate in (2.2) and remembering that tasks arrive at rate l, the steady-state success probability is estimated by

$$P_F(l, q, t) = \left(R(\infty)/(1+R(\infty))\right)\frac{p(t)}{t\,l}. \qquad (2.3a)$$

2.2 ***A Self-Thinning Approximation for an M/G/1 Model of Success Rate***

Consider a standard/classical M/G/1 queueing model with a constant service time t where tasks defect after exponential amount of time with the mean q^{-1}. Assume defection during service is not observable. It is shown in Gaver *et al*. (2000) that a good approximation to the long-run probability that successful service is achieved is

$$P(l, q, t) = \frac{2}{1+r+\sqrt{(1+r)^2 - 4rd(q)}}\,p(t) \qquad (2.4)$$

where $d(q) = \left(1-e^{-qt}\right)/qt$ and $p(t)$ is the probability a task given t time units of service is successfully completed. The traffic intensity is $r = l\,t$. An argument for (2.4) appears in Appendix 2. An analytical/mathematical "exact" approach (forward Kolmogorov equation) is detailed in Appendix 1. The expression (2.4) is remarkably convenient and numerically accurate; however, an improvement is also given in Appendix 2.

Suppose task service times have a distribution F_S but service completion and task loss during service are not observable; then an approximation to the probability of successful task completion for a task that enters service is

$$p(t) = \int_0^t e^{-qs} f_S(s)ds \geq e^{-qt} F_S(t) \equiv s(t). \qquad (2.5)$$

Analytical Model for Single-Service Success Probability

An analytically tractable and flexible expression for $F_S(t)$ is the *Fréchet distribution* of extreme value theory (qualitatively appropriate here since it approximates the distribution of maxima); see Resnick (1987):

$$F_S(t) = pe^{-(a/t)^b}, \tag{2.6}$$

where the constant p represents the maximum probability of success, achieved as $t \to \infty$; a is a scale, and b a shape parameter, both positive; if $p<1$, F_S is a defective distribution. The exponent in the exponential function is unity (1) when $a=t$, when the success probability becomes $F_S(t) = pe^{-1} = 0.37\,p$, or about one-third of the maximum possible, and this independently of b. For $t<a$ the exponent increases rapidly as b increases; likewise, it decreases rapidly for $t>a$ with increasing p, representing threshold behavior at $t=a$.

"Optimum" Single-Server Success Probability

Provided a task enters service, i.e., survives wait in queue, a lower bound on the probability that it is successful is given by the RHS of (2.5):

$$s(t) = e^{-qt} F_S(t) = e^{-qt}\, pe^{-(a/t)^b} \tag{2.7}$$

for the present model of (2.6). There is a unique maximizing value of t, namely

$$t_o = \left(\frac{b}{q}\right)^{1/(1+b)} a^{\,b/(1+b)} \tag{2.8}$$

which implies

$$s^{\#}(t_o) = pexp\left[-(1+b)\left(\frac{qa}{b}\right)^{b/(1+b)}\right] \qquad (2.9)$$

Under the policy that all tasks that start service receive t units of service an approximate upper bound to the probability an arriving task will successfully complete service is

$$P(l,\ q,\ t) \geq \frac{2s(t)}{1+l\,t+\sqrt{(1+l\,t)^2-4\frac{l}{q}\left[1-e^{-q\,t}\right]}} \equiv \frac{A(t;q)}{B(t;q)} \qquad (2.10)$$

The maximizing t for the lower bound (2.10) with distribution (2.6) can be found numerically. A first order approximation is (2.8). Note that the maximizing t of (2.8) does not involve l. An improved approximate maximizing t can be obtained by applying one iteration of a Newton procedure to $\frac{d}{dt}\left[A(q;t)/B(q;t)\right]$ around the original approximating t_0. One iteration of Newton's procedure evaluated at t_0 results in

$$t_1 = t_0 - \frac{\frac{d}{dt}\left[A(q;t_0)/B(q;t_0)\right]}{\frac{d^2}{dt^2}\left[A(q;t_0)/B(q;t_0)\right]} \qquad (2.11)$$

The maximizing values of t from (2.8) for $a = \sqrt{-\ln(0.9)}$ and $b = 3$ for various values of q and l are displayed in the table below. Also displayed is the value of t resulting from one Newton iteration and the resulting lower bounds on the approximate probabilities of an arriving task being successfully served. The table presents the approximate probability of successful service estimated for the previously computed t by evaluating (2.4). Also displayed are the results from simulation of the queueing system with Fréchet service times and service truncated at the various computed t. Service completion is not observed and each served task is given t units of service; service is

successful if the service time is less than t. Each simulation consists of 50 replications; each replication is of 6000 tasks.

λ	θ	Maximizing τ from (2.8): t_0 [One iteration of Newton: t_1]	Lower Bound on the Probability of Success using t_0 in (2.10) [Lower Bound on the Proba-bility of Success for the Newton t_1 in (2.10)]	Approximate Probability of Service Success (2.4) for t_0 [Probability of Service Success; Simulation of system] (Std Error)	Approximate Probability of Service Success, (2.4), for Newton t_1 [Probability of Service Success; Simulation of System] (Std Error)
0.1	0.25	0.80 [0.78]	0.72 [0.76]	0.85 [0.85] (0.0008)	0.84 [0.84] (0.0007)
	0.50	0.67 [0.66]	0.63 [0.63]	0.74 [0.74] (0.0008)	0.73 [0.74] (0.0009)
0.5	0.25	0.80 [0.69]	0.72 [0.73]	0.80 [0.81] (0.0007	0.79 [0.79] (0.0007)
	0.50	0.67 [0.61]	0.60 [0.60]	0.70 [0.70] (0.0008)	0.69 [0.68] (0.0009)
1	0.25	0.80 [0.47]	0.64 [0.61]	0.72 [0.74] (0.0009)	0.64 [0.65] (0.001)
	0.50	0.67 [0.50]	0.53 [0.54]	0.62 [0.64] (0.001)	0.58 [0.60] (0.0009)

Table 1

Discussion: The easily computed t_0 performs well in all cases computed. The approximate probabilities of successful task completion agree well with the simulation results for the actual system using the service policy.

2.4 *Towards Discrete Optimal Static Service Responses*

Suppose that a BSA has several service options against RSA available. Service option k is characterized in terms of setup time, t_k, and corresponding success probability $p_k(t_k)$. Under some circumstances it is the practice to take several simultaneous service actions once setup has been performed; under special conditions more than one, e.g., n_k independent service processes may be applied so as to raise the effective success probability, in which case, replacement of $p_k(t_k)$ by $1-(1-p_k(t_k))^{n_k}$ is justified.

If estimates of task arrival rate, l, and loss rate, q, are available then the optimal static response is to select that option, k^*, that maximizes the probability of surviving without loss to reach the server, surviving service without loss, and actually delivering successful service. Therefore, it is necessary to identify

$$k^* = ArgMaxP(l, q, t_k)$$

using, say, (2.4), with $p(a)$ replaced with $p_k(t_k)$. This then identifies the appropriate service option, which need not be that with highest $p_k(t_k)$ value.

Numerical Examples

Here are numerical examples that illustrate the models proposed. Suppose the rate of approach l = 0.95 and the rate of loss q = 1. Suppose there are these service options

k\	t_k	p_k	Success Probabilities			
			Fluid	Self-Thinning	Modif. M/G/1	Cond. Prob. Given Start Service
1	0.5	0.8	0.31	0.42	0.43	0.49
2	0.25	0.7	0.43	0.53	0.53	0.55
3	0.15	0.6	0.45	0.51	0.51	0.52

Table 2

Probabilistic model probabilities agree to about two significant digits; they all concur that $k^* = 2$: ($t = 0.25$, $p = 0.7$), the intermediate case. This choice remains optimal for the probabilistic models even if task arrival rate drops to 0.75, but the probability of overall success increases.

3. *Several (J ³ 1) Red Task Types vs. Single Blue Service Agent*

Next consider a single Blue (BSA) confronting a random stream of *different* loss susceptible RSAs. The arrival process of RSAs of type j is Poisson (l_j) (j is a member of (1, . . . J)) independent of the other task types. Service times for tasks of type j are independent and have a distribution F_j.

Assume the times until loss are independent with those for RSAs of type j having an exponential distribution with mean $1/q_j$.

Appendix 3 displays a system of forward Kolmogorov (Takaçs-Beneš) equations for the limiting task virtual waiting time for this model. Successive substitution/iteration results in a system of equations for the probability a task of type j survives its wait in queue for $j = 1,\ldots,J$. Approximate probabilities that an arriving task will start service are also detailed in Appendix 3.

Examples:

Although it is possible to obtain the probability an arriving task will start service numerically, the calculation can be tedious for more than two task types. The table below displays results of two approximations for the probability an arriving task will start service and the results from a simulation; Bullock (2003). The simulation results are for 50 replications with 6000 tasks of each type per replication. Both service completion and task loss while in service are observable. Task loss in queue is observable.

Task Type	Task Arrival Rate (Lambda)	Task Loss Rate (Theta)	Means of the Exponential Service Times	Approx. Probability Arriving Task Starts Service (A3-8)	Approx. Probability Arriving Task Starts Service Using Filtered Busy Period (A3-17)	Simulation Fraction of Tasks to Start Service (Std error 50 replications)
1	0.10	1	1	0.78	0.81	0.78 (0.0007)
2	0.10	0.5	3	0.83	0.89	0.84 (0.0007)
3	0.10	0.25	5	0.88	0.97	0.89 (0.0006)
1	0.30	1	1	0.47	0.45	0.46 (0.001)
2	0.30	0.5	3	0.55	0.62	0.57 (0.001)
3	0.30	0.25	5	0.65	0.87	0.70 (0.0009)
1	0.75	1	1	0.22	0.13	0.13 (0.001)
2	0.75	0.5	3	0.28	0.25	0.25 (0.001)
3	0.75	0.25	5	0.37	0.60	0.43 (0.001)

Table 3

Discussion: The first approximation appears to do well for lightly loaded systems. The approximation that thins the arrivals during a busy period appears to do better for heavily loaded systems. The minimum of the two approximations gives very reasonable agreement with the simulation results for the cases studied.

3.1 *Myopic Policies for choice of next task type to serve and amount of service to provide*

Suppose that loss during service and the completion of service are not observable. A policy is to serve a customer of type j for a time t_j. If the task's service time is less than

t_j, then the task is served successfully; otherwise it is served unsuccessfully. A successfully served task of type *j*, results in a reward r_j.

The question: Allocate service time $t_j(\bar{n})$ to a type *j* customer as a function of the number of tasks of various types in queue; where $\bar{n} = (n_1, n_2, ..., n_J)$ with n_j being the number of customers of type *j* waiting in queue. Choose the next type of task to serve and the amount of time to give it so as to maximize the long run average reward.

Suppose we allocate $t_j > 0$ units of time to a waiting task of type *j*. The expected reward received is

$$R_E(j, t_j; \bar{n}) = r_j \int_0^{t_j} e^{-q_j y} F_j(dy) \tag{3.1}$$

The expected reward leaving service during the service time is

$$\begin{aligned} & R_L(j, t_j; \bar{n}) \\ & = r_j + r_j (n_j - 1) A_j(t_j; q_j) + \sum_{k \neq j} r_k n_k A_j(t_j; q_k) \\ & + \sum_{k=1}^{J} r_k l_k B_j(t_j; q_k) \end{aligned} \tag{3.2}$$

where

$$A_j(t; q) = \left[1 - e^{-q t}\right] \tag{3.3}$$

$$B_j(t, q) = t - \frac{1}{q}\left[1 - \exp\{-q t\}\right] \tag{3.4}$$

The myopic policy is to select that *j* and t_j which maximizes the proportion of expected reward stream gained during the next service time.

$$\left(j_0(\overline{n}), t_{j_0}(\overline{n}) \right) = \underset{j, t_j}{\operatorname{argmax}} \frac{R_E\left(j, t_j; \overline{n}\right)}{R_L\left(j, t_j; \overline{n}\right)} \tag{3.5}$$

The policy is myopic because it only optimizes the immediate gain. Similar policies can be obtained for cases in which the service completion is observable and/or the loss of a task during service is observable, etc. Discussion and study of non- myopic policies appear elsewhere.

*Exampl*es:

All service times and impatience times are independent and exponentially distributed. There are two task types that arrive according to independent Poisson processes. Task losses while waiting and while being served are observed. Task service completion is also observed. In either case, a new task can begin immediately. The myopic policy determines the next task type to be served and the length of the service time to give it. If a task has not completed service when its allocated service time is over, it departs and no reward is collected.

Below are results from a simulation. Each simulation replication is for 100 time units. The number of replications for cases with larger arrival rates is 100. The number of replications for cases with smaller arrival rates is 500. The FIFO policy serves the first task in queue until the task is served to completion. The smart FIFO serves the first task in queue until time t where t is the value that maximizes (3.5). The myopic policy serves that task which maximizes (3.5).

			Reward Task 1=3 Reward Task 2=1			Reward Task 1=1 Reward Task 2=3		
Arrival Rate Task1 [Task2]	Mean Impatience Time Task 1 [Task 2]	Mean Service Time Task 1 [Task 2]	Percent Reward Received FIFO	Percent Reward Received Smart FIFO	Percent Reward Received Myopic	Percent Reward Received FIFO	Percent Reward Received Smart FIFO	Percent Reward Received Myopic
0.1 [0.2]	1 [2]	1 [1.5]	48.48 (0.45)	46.58 (0.46)	48.35 (0.49)	53.04 (0.44)	48.03 (0.44)	52.29 (0.44)
0.25 [0.50]	1 [2]	1 [1.5]	41.96 (0.47)	39.26 (0.45)	43.7 (0.44)	46.5 (0.49)	42.09 (0.5)	46.95 (0.5)
0.5 [1]	1 [2]	1 [1.5]	32.96 (0.35)	32.51 (0.33)	35.18 (0.34)	38.1 (0.34)	34.51 (0.32)	38.1 (0.33)
1 [2]	1 [2]	1 [1.5]	20.8 (0.32)	21.51 (0.3)	25.51 (0.3)	25.33 (0.36)	22.67 (0.3)	25.92 (0.35)
2 [4]	1 [2]	1 [1.5]	8.92 (0.14)	10.46 (0.17)	17.88 (0.21)	13.3 (0.18)	13.69 (0.19)	14.25 (0.18)
4 [8]	1 [2]	1 [1.5]	3.96 (0.06)	4.31 (0.07)	12.54 (0.14)	6.91 (0.09)	7.02 (0.09)	7.08 (0.1)
8 [16]	1 [2]	1 [1.5]	1.87 (0.03)	1.79 (0.02)	7.47 (0.08)	3.57 (0.04)	3.56 (0.04)	3.62 (0.05)
0.1 [0.2]	5 [4]	1 [1.5]	74.21 (0.41)	70.13 (0.41)	73.8 (0.41)	69.19 (0.4)	64.77 (0.43)	69.53 (0.4)
0.25 [0.50]	5 [4]	1 [1.5]	65.33 (0.5)	62.05 (0.48)	66.29 (0.46)	60.27 (0.48)	51.16 (0.47)	61.26 (0.47)
0.5 [1]	5 [4]	1 [1.5]	48.4 (0.4)	47.9 (0.41)	54.35 (0.41)	44.42 (0.41)	42.24 (0.36)	46.27 (0.39)
1 [2]	5 [4]	1 [1.5]	27.86 (0.38)	28.42 (0.38)	41.45 (0.35)	24.09 (0.35)	24.82 (0.32)	28.65 (0.36)
2 [4]	5 [4]	1 [1.5]	14.66 (0.2)	15.11 (0.21)	27.94 (0.25)	11.73 (0.14)	12.44 (0.16)	14.48 (0.18)
4 [8]	5 [4]	1 [1.5]	8.02 (0.09)	8.52 (0.13)	14.71 (0.15)	5.95 (0.07)	6.02 (0.08)	7.1 (0.08)
8 [16]	5 [4]	1 [1.5]	4.1 (0.06)	4.5 (0.06)	7.46 (0.08)	2.79 (0.04)	2.87 (0.04)	3.54 (0.05)

Table 4

Discussion: Not surprisingly, the three policies perform about the same for lightly loaded systems. For lightly loaded systems FIFO results in larger percentage of rewards because all tasks that start service are given a full service time. For heavier loaded systems, FIFO

and smart FIFO perform about the same. In heavier loaded systems giving priority to a task type becomes more important.

3.2 *A Simulation Study of the Effect of Task Priorities and Service Discipline on Task Completion*

In this section, the effect of different task priorities and service disciplines are explored using simulation.

Tasks arrive according to a Poisson process with rate $l = 3.5$. Tasks are of type 1 with probability 0.5 and of type 2 otherwise. A task of type 1 requires service of length $a_1 = 0.2$. A task of type 2 requires a service of length $a_2 = 0.5$. Thus, $r = l E[\boldsymbol{S}] = 1.23$ where $\boldsymbol{S}$ is the service time of an arriving customer. Each arriving task is lost after an independent random length of time. In this model, losses during service are not observable. Two distributions of task loss times are considered: the uniform and the exponential. . All simulation with the same task loss time distributions, use the same simulated arrival times, task types, and task loss times; the difference between the replications is the service discipline and task priority. Table 5 displays the simulated fractions of tasks completed using different service disciplines and task priorities. Each simulation replication consists of 2000 tasks.

Traffic Intensity=1.23

Distribution of loss times for tasks of type 1	Distribution of loss times for tasks of type 2	Choice of next task to serve	Task type with priority	Frac. of tasks that complete (std. error computed as if observ. are indep.)	Frac. tasks of type 1 that complete (std. error computed as if observ. are indep.)	Frac. tasks of type 2 that complete (std. error computed as if observ. are indep.)
Exponential with mean 1	Exponential with mean 3.33	FCFS	2	0.53 (0.01)	0.32 (0.01)	0.73 (0.01)
		FCFS	1	0.64 (0.01)	0.65 (0.02)	0.63 (0.02)
		LCFS	None	0.60 (0.01)	0.50 (0.02)	0.68 (0.01)
		FCFS	None	0.56 (0.01)	0.42 (0.02)	0.69 (0.01)
Uniform on (0.5, 1.5) (2000 tasks)	Uniform on (2.83, 3.83)	LCFS	1	0.79 (0.01)	0.90 (0.01)	0.68 (0.01)
		FCFS	2	0.58 (0.01)	0.21 (0.01)	0.95 (0.01)
		LCFS	2	0.60 (0.01)	0.30 (0.01)	0.91 (0.01)
		FCFS	1	0.73 (0.01)	0.93 (0.01)	0.51 (0.02)
		LCFS	None	0.71 (0.01)	0.64 (0.02)	0.79 (0.01)
		FCFS	None	0.56 (0.01)	0.23 (0.01)	0.90 (0.01)

Table 5

Discussion:

The approximating filtering model for exponential times to loss results in the probability that a task of type 1 survives the queue is equal to 0.626 and the probability a task of type 2 survives the queue is 0.703. For exponential times to loss, the probability a task 1 that starts service is not lost during service is 0.82. The probability a task of type 2 that starts service is not lost during service is 0.86.

Exponential loss times result in a task's position in queue giving no information on the remaining time until the task is lost. Thus, there is no statistical difference in the fraction of tasks completed successfully between first come first served (FCFS) and last come first served (LCFS). Since the mean loss time for task 1 is less than that for task 2, giving task 1 priority results in a greater fraction of tasks successfully completed. The myopic policy of Section 4 when the reward for successful completion of both tasks is 1 is to give task 1 priority. Notice that with no priorities, the LCFS discipline outperforms the FCFS discipline in spite of exponentially distributed deadlines. This behavior is a consequence of the different mean impatience times: the longer a task stays in the system, the more likely it is to be of type 2.

Uniform loss times result in a task's position in queue giving information on the remaining time until the task is lost. The mean loss time for task 1 is less than that for task 2 and task 1 has a shorter service time than task 2. Thus, the best policy is to give task 1 priority with the service discipline last come first served (LCFS). The phenomena noted for this case also apply for systems with smaller traffic intensities.

4. *Concluding Remarks*

Modeling uncertain time-critical service systems is a difficult but vitally important practical problem. Exact computations are often either impossible or very challenging computationally, especially with multiple customer types. Special challenges are present when deciding on a service policy in order to make the system as efficient as possible.

In this paper we have presented several approximation procedures that are computationally easy and, at least in the examples we have looked at, provide valuable information about the efficiency of the service system under different service options. An important feature of these approximations is that they stay computationally feasible even for many task types and/or heavily loaded systems.

We have also introduced a heuristic myopic service policy that attempts to maximize *locally* the system efficiency. This policy has performed well under scenarios we have considered.

A number of important issues are left for future work. One such issue is improving the myopic policies into (approximately) optimal policies. A possible approach is a dynamic programming-based procedure that is being developed in Samorodnitsky, Gaver and Jacobs (2003). Another untouched issue is that of nonstationarity: what happens if the parameters of the system change with time, and need to be constantly estimated in order to update the service policy and keep the system running efficiently. We hope to address the latter question in the near future.

Acknowledgements

The authors wish to acknowledge useful comments by Gideon Weiss, John Hiles, Glen Takahara and John Lehoczky. John Lehoczky suggested Approximation II in Appendix 2.

References

Baccelli, F., Boyer, P. and Hebuterne, G. "Single-server queues with impatient customers," *Adv. Appl. Prob*. Vol. 16, pp. 887-905.

Becker, K. J., D. P. Gaver, K. D. Glazebrook, P. A. Jacobs, and S. Lawphongpanich, (2000), "Allocation of tasks to specialized processors: a planning approach," *European Journal of Operational Research*, Vol. 126, pp. 80-88.

Boots, N. K. and H. Tijms, (1999), "A multi-server queueing system with impatient customers," *Management Science*, **45(3)**, pp. 444-448.

Boots, N. K. and H. Tijms, (1998), "An M/M/C queue with impatient customers," presented at the First International Workshop on Retrial Queues, Universidad Complutense de Madrid, Madrid, Spain, September 22-24, 1998.

Brown, M. and F. Proschan, (1983), "Imperfect repair," *J. Appl. Prob*. **20**, pp. 851-859.

Bullock, G. L. (2003), "UTCT Simulation Model" A simulation implemented in C++.

Cox, D. R. and W. L. Smith, (1961), *Queues*, Chapman & Hall, London, UK.

Doytchinov, B., J. Lehoczky, and S. Shreve, (2001), "Real-time queues in heavy traffic with earliest-deadline-first queue discipline," *Ann. Applied Prob.* **11**, pp 332-378.

Filipiak, J., (1988), *Modeling and Control of Dynamic Flows in Communication Networks*, Springer-Verlag, Berlin, Germany.

Gaver, D. P. and P. A. Jacobs, (2002), "BATTLESPACE/INFORMATION WAR (BAT/IW): A System-of-Systems Model of a Strike Operation," Naval Postgraduate School Technical Report, NPS-OR-02-005, Monterey, CA.

Gaver, D. P. and Jacobs, P. A., (2000), "Servicing impatient tasks that have uncertain outcomes," Naval Postgraduate School Technical Report, NPS-OR-00-001, Monterey, CA.

Jiang, Z., T. G. Lewis, and J-Y. Colin, (1996), "Scheduling hard real-time constrained periodic tasks on multiple processors," *J. Systems & Software*, **19(11)**, pp. 102-118.

Kleinrock, L., (1976), *Queueing Systems, Vol. I: Theory*, Wiley (Interscience), New York.

Lehoczky, J. P., (1996), "Real-time queueing theory," *Proceedings of the IEEE Real-Time Systems Symposium*, December 1996, pp. 186-195.

Lehoczky, J. P., (1997a), "Using real-time queueing theory to control lateness in real-time systems," *Performance Evaluation Review*, **25(1)**, pp. 158-168.

Lehoczky, J. P., (1997b), "Real-time queueing network theory," *Proceedings of the IEEE Real-Time Systems Symposium*, December 1997, pp. 58-67.

Liu, C. L. and J. W. Layland, (1973), "Scheduling algorithms for multiprogramming in a hard real-time environment," *J. Automatic Computing Machinery*, **20(1)**, pp. 40-61.

Osmundson, J., (2000), "A systems engineering methodology for information systems," *Systems Engineering*, Vol. 3, No. 2, July 2000, pp. 68-81.

Resnick, Sidney I., (1987), *Extreme Values, Regular Variation, and Point Processes*, Springer-Verlag, New York.

Righter, R. (1987) "The stochastic sequential assignment problem with random deadlines," *Probability in the Engineering and the Informational Sciences*, Vol. 1, pp. 189-202

Samorodnitsky, G., D. P. Gaver and P.A. Jacobs, (2003), "Choosing the best customer to serve and ratio control problem," Working paper.

Ward, A. R. and Bambos, N. "On stability of queueing networks with job deadlines," Technical Report, SU NETLAB-2001-08/01, Engineering Library, Stanford University, Stanford, CA, August 25, 2001.

Whitt, W., (1999), "Improving service by informing customers about anticipated delays," *Management Science*, **45(2)**, pp. 192-207.

APPENDIX 1
Modified Takaçs-Beneš Equations with Exponential Refusal/Reneging

Let tasks arrive at a service facility according to a Poisson process with rate l. Service times are independent and identically distributed. Let $W(t)$ be the total virtual work in the system at time t. Each task has a deadline that is exponentially distributed with mean $1/q$: if the waiting time or virtual work present when the task arrives exceeds the deadline the task does not enter the system. This is equivalent to the situation in which tasks whose deadlines have elapsed *when they reach the server* are not served; see e.g. Baccelli *et al* (1984) and Ward *et al.* (2001). We will then use a modification to obtain an approximation for the situation in which a deadline may also elapse *during* service.

A.1 Statistically Identified Deadlines and Service to Completion

We start with sketching an argument for derivation of the steady state probability that an arriving customer will be successfully served. Let the steady state distribution function of $W(t)$ be

$$F_W(x;t;q) = P\{W(t) \le x\}. \tag{A1-1}$$

A standard renewal theoretical argument shows that $F_W(\cdot;q)$ has a right continuous density $p(z;q)$ on $(0,\infty)$. Express this as

$$F_W(x;t;q) = p_0(t;q) + \int_0^x p(z;t;q)\,dz, \tag{A1-1a}$$

where

$$p_0(t;q) = P\{W(t) = 0\}. \tag{A1-1b}$$

Since, given $W(t)$, the task joins the queue with probability $e^{-qW(t)}$, the probability its deadline does not expire while in queue, one can write

$$F_W(x;q) = F_W(x+\Delta t;q)[1-\lambda\Delta t] + \lambda\Delta t\int_0^x \left(1-e^{-q y}+e^{-q y}B(x-y)\right) F_W(dy;q) + o(\Delta t) \tag{A1-2}$$

where B is the distribution function of the positive service time $\boldsymbol{C}$. Dividing by Δt and letting $\Delta t \downarrow 0$ we obtain

$$p(x;q) = \lambda\int_0^x e^{-q y}\left(1-B(x-y)\right) F_W(dy;q). \tag{A1-3}$$

Use Laplace transforms

$$y(s;q) = \int_0^\infty e^{-sx} F_W(dx;q) \text{ and } b^*(s) = \int_0^\infty e^{-sx} B(dx).$$

Then (A1-3) implies

$$y(s;q) = p_0(q) + r\,y(s+q;q)\,d(s) \tag{A1-4}$$

where

$$d(s) = \frac{1-b^*(s)}{sE[\boldsymbol{C}]}. \tag{A1-5}$$

Substituting $s = 0$ yields $y(0; q) = 1 = p_0(q) + r\,y(q; q)$, hence

$$p_0 = 1 - r\,y(q;q) = 1 - \lambda E\left[e^{-qW}\right]E[\boldsymbol{C}]. \tag{A1-6}$$

Iterative solution to the equation (A1-4)

Since

$$y(s;q) = [1 - r\,y(q;q)] + r\,y(s+q;q)\,d(s)$$

it follows that, putting $s = nq$, and defining $y(q) \equiv y(q;q)$ we have

$$y(nq,q) = \left[1 - r\,y(q)\right] + r\,y((n+1)q;q)\,d(nq).$$

An inductive argument gives us

$$y'(q) = A'(0,q) + \ldots + A'(n;q) - r\,y'(q)\left[A'(0,q) + \ldots + A'(n;q)\right] + r\,y'(n+1;q)\,A'(n;q). \tag{A1-6a}$$

where

$$A'(0,q) = 1$$
$$A'(n;q) = r^{n}\,d'(nq)\,d'(n-1;q)\times\ldots\times d'(q). \tag{A1-6b}$$

For $q > 0$, $A(n;q) \to 0$. Thus, the probability that an arriving task joins and survives the *queue* before deadline elapse is

$$y'(q) = \frac{\sum_{k=0}^{\infty} A'(k;q)}{1 + r\sum_{k=0}^{\infty} A'(k;q)}. \tag{A1-7}$$

It is clear that the infinite sum converges faster than exponentially fast for $q > 0$, and that this is true for any r-value.

Further, for *any s*

$$y\,(s;q) = \left[1 - r\,y\,(q)\right]\sum_{k=0}^{\infty} C(k;s) \tag{A1-8}$$

where

$$C'(k;s) = r^{k}\prod_{i=0}^{k-1} d'(s+iq), \quad k \geq 1 \tag{A1-9}$$

and

$$C(0;s) = 1.$$

A.2 Services Subject to Detectable Exponential Deadline

If a task deadline's elapse is detectable during service and the task *is then terminated*, then the distribution of service time, $\boldsymbol{C}$, must be replaced by that of $\boldsymbol{C}_T = \min(\boldsymbol{C}$, deadline), the *allowed service time.* Consequently, the service times *that contribute to the virtual waiting time* are, thanks to the exponential deadline assumption, iid with mean

$$E[\boldsymbol{C}_T] = \frac{1 - E\left[e^{-q\boldsymbol{C}}\right]}{q} \tag{A1-10}$$

and tail-transform now

$$d_T(s;q) = \frac{1 - E\left[e^{-(q+s)\boldsymbol{C}}\right]}{(q+s)E[\boldsymbol{C}_T]} = \frac{d(q+s)}{d(q)}. \tag{A1-11}$$

These replace $E[\boldsymbol{C}]$ in r, and $d(s)$ in the previous solution, (A1-7).

APPENDIX 2
Approximations to Queue Survival for the M/G/1 System with Deadline-Sensitive Delay (The "Self-Thinning" Approximation)

Consider the arrival of tasks with exponentially (mean $1/q$) distributed deadlines. Given the virtual waiting time, $\boldsymbol{W}(t)$, at the time of the arrival of a customer, the probability that the deadline of the customer will not elapse before reaching the server, is simply $e^{-q\boldsymbol{W}(t)}$. Instead of letting the customer join the queue and then defect if the deadline does elapse before entering service, the same outcome is achieved by simply accepting the customer into the queue with probability $e^{-q\boldsymbol{W}(t)}$. Based on that, we propose two approximations to the proportion of customers that are successfully served.

Approximation I

If $\boldsymbol{W}$ has the steady state virtual workload distribution, then the probability that an arriving customer is successfully served is $y(q) = Ee^{-q\boldsymbol{W}}$. We neglect the dependence between the fates of different customers by pretending that the outcomes are decided via a sequence of independent coin tosses with success probability $y(q)$. The resulting system becomes an M/G/1 queue with traffic intensity $ry(q)$. Then the success probability $y(q)$ should approximately satisfy the Pollaczek-Khinchine (P-K) formula for M/G/1 queues:

$$\psi(\theta) \equiv E\left[e^{-\theta W}\right] = \frac{1-\left[\psi(\theta)\,\lambda\right]E[\beta]}{1-\left[\psi(\theta)\,\lambda\right]E[\beta]\left\{\dfrac{1-E\left[e^{-\theta\beta}\right]}{\theta\,E[\beta]}\right\}}. \qquad \text{(A2-1)}$$

The simple formula differs somewhat from the solution (A1-7) of the modified Takaçs-Beneš equation for the same assumed arrival-queue interaction; but is in handy closed form.

The expression (A2-1) is a quadratic in the desired probability, the solution of which is

$$\psi(\theta) = \frac{2}{1+r+\sqrt{(1+r)^2 - 4r\,\delta(\theta)}} \qquad \text{(A2-2)}$$

where $r = \lambda E[\beta]$ as usual, and $\delta(\theta)$ given by (A1-5) is the transform of the service/completion time *tail* or survivor distribution. The approximate probability of successful transit to the server given by this simple expression is unity when $\theta \to 0$ (no degradation, or infinite deadline), as long as $r < 1$; if $\theta \to \infty$ then, since deadlines are now stringent, the only hope of initiating service is to arrive when there is no server activity, i.e., with probability $1/(1 + r)$, and this time any (positive) r-value is permitted. In general, there are no restrictions on r in (A2-2): a long queue generates many rejections, and thus does not ever remain long, or grow indefinitely. Empirically, the simple expressions, (A2-2) and (A2-9) below, supply a lower bound that has been shown numerically to be a good approximation to the exact solution of such a reneging or refusal model. Note that the same logic gives as an approximation for the transform of virtual waiting time of non-refused tasks, $\boldsymbol{W}$, the formula

$$\psi(x;\theta) = \frac{1-\lambda\,\psi(\theta)E[\beta]}{1-\lambda\,\psi(\theta)E[\beta]\left\{\dfrac{1-E\left[e^{-x\beta}\right]}{xE[\beta]}\right\}} \qquad \text{(A2-3)}$$

Approximation II

A refined version of the above accounts for the different experience of a new task that arrives to find the server busy ($\boldsymbol{W} > 0$), as contrasted to one that arrives to find it idle ($\boldsymbol{W} = 0$). Put

$$\gamma_+(q) = E\left[e^{-q\boldsymbol{W}} \,|\, \boldsymbol{W} > 0\right] \tag{A2-3}$$

the marginal long-run rate of task acceptance given that the server is busy. From the Pollaczek-Khinchine formula

$$\begin{aligned} E\left[e^{-s\mathrm{W}} \,|\, \mathrm{W} > 0\right] &= \left(\frac{(1-r)\,r\,\delta(s)}{1-r\,\delta(s)} - 1\right)\frac{1}{r} \\ &= \frac{(1-r)\,\delta(s)}{1-r\,\delta(s)}. \end{aligned} \tag{A2-4}$$

We view r in this expression as the traffic intensity during a busy period. Then the same logic as the one used in derivation of Approximation I above says that the acceptance probability during a busy period $\gamma_+(q)$ should approximately satisfy the above expression with r replaced by $r\,\gamma_+(q)$. This results in the equation

$$\gamma_+(q) = (1 - r\,\gamma_+(q))\frac{\delta(q)}{1 - r\,\gamma_+(q)\,\delta(q)}. \tag{A2-5}$$

In other words, an auxiliary randomization (biased coin flip) adjusts for the imposition of the deadline, as before in Approximation I, but in a somewhat more refined manner. The solution of (A2-5) is

$$\gamma_+(q) = \frac{2\delta(q)}{(1 + r\delta(q)) + \sqrt{(1 + r\delta(q))^2 - 4r\delta(q)^2}}. \tag{A2-6}$$

For such a γ_+-filtered system the expected duration of a busy period, $E[\boldsymbol{B}]$, should satisfy

$$\begin{aligned} E[\boldsymbol{B}] &= E[\boldsymbol{C}] + r\gamma_+(q)\,E[\boldsymbol{B}] \\ &= E[\boldsymbol{C}] \big/ (1 - r\gamma_+(q)). \end{aligned} \tag{A2-7}$$

Consequently, an alternating renewal process argument gives us, as the long-run proportion of time that the server is idle,

$$P\{W=0\} = \frac{l^{-1}}{l^{-1}+E[B]} = \frac{1-r\,y_+(q)}{1+r[1-y_+(q)]}. \qquad \text{(A2-8)}$$

Now the probability that an arriving task is admitted (not refused, and eventually served) is

$$\begin{aligned} y_2(q) &= P\{W=0\} + (1-P\{W=0\})\,y_+(q) \\ &= \frac{1}{1+r[1-y_+(q)]}, \end{aligned} \qquad \text{(A2-9)}$$

which differs from (A2-2) owing to the more refined conditioning imposed.

Approximation II improves somewhat on Approximation I in all cases explored numerically to date.

APPENDIX 3
Solution to a Modification of the Takaçs-Beneš Equation for Multiple Task Types

Consider a generalization of the model in Appendix 1. Tasks from *J* task classes arrive to a service facility according to independent Poisson processes with rate l_j for the j^{th} task class; let $l = \sum_{j=1}^{J} l_j$. Service times are independent, and service times for each task class are identically distributed. Each task of the j^{th} class has an exponentially distributed deadline with the mean $1/q_j$, with the usual independence assumptions. Once again, we start with the case where customers whose deadlines have elapsed when they reach the server are not served, but no defection occurs while in service. Equivalently, a customer whose deadline is shorter than the waiting time at the moment of arrival, does

not enter the system. Arguments analogous to those in Appendix 1 show that the Laplace transform of the steady state virtual waiting time in the system satisfies

$$y\,(s;?) = p_0(?) + \sum_j r_j d_j(s)\, y\,(s+q_j;?) \tag{A3-1}$$

where

$$d_j(s) = \frac{1-b_j^*(s)}{s\,E\left[\boldsymbol{C}_j\right]}, \tag{A3-2}$$

with $r_j = l_j E[\boldsymbol{C}_j]$, and $b_j^*(s) = E[e^{-s\boldsymbol{C}_j}]$. Here $\boldsymbol{C}_j$ is a generic service time of a class j task; put $r = \sum_{j=1}^{J} r_j$. Furthermore,

$$p_0(q) = 1 - \sum_j r_j\, y\left(q_j;q\right). \tag{A3-3}$$

An iterative procedure similar to the one used in Appendix 1 shows that the probability $y(q_j) \equiv y(q_j;q)$ that a task of type j will start service (not be lost while in queue) satisfies, for each $n = 1,2,\ldots$, the equation

$$\begin{aligned} y\,(q_j) &= A^{(j)}(0,q) + \ldots + A^{(j)}(n,q) \\ &\quad - \left[A^{(j)}(0,q) + \ldots + A^{(j)}(n,q)\right] \sum_{i=1}^{J} r_i y\,(q_i) + E_j(n+1;q), \end{aligned} \tag{A3-4}$$

where

$$A^{(j)}(0,q) = 1$$

$$A^{(j)}(n;q) = \sum_{k_1=1}^{J} r_{k_1} d_{k_1}(q_j) \sum_{k_2=1}^{J} r_{k_2} d_{k_2}(q_j + q_{k_1}) \ldots \sum_{k_n=1}^{J} r_{k_n} d_{k_n}(q_j + q_{k_1} + \ldots + q_{k_{n-1}})$$

and

$$E^{(j)}(n;q) = \sum_{k_1=1}^{J} r_{k_1} d_{k_1}(q_j) \ldots \sum_{k_n=1}^{J} r_{k_n} d_{k_n}(q_j + q_{k_1} + \ldots + q_{k_{n-1}})\, y\,(q_j + q_{k_1} + \ldots + q_{k_{n-1}} + q_{k_n})$$

for $n = 1,2,\ldots$. As in Appendix 1, $E^{(j)}(n;q) \to 0$ as $n \to \infty$, and we obtain a system of linear equations for success probabilities

$$y(q_j) = \sum_{k=0}^{\infty} A^{(j)}(k;q) - \sum_{k=0}^{\infty} A^{(j)}(k;q) \sum_{i=1}^{J} r_i y(q_i) \tag{A3-5}$$

for $j = 1,\ldots,J$, which we can solve by replacing the infinite sums by their finite approximations.

A computationally attractive approximation to the solution to the equations (A3-5) is as follows. We start with a self-thinning approximation to the entire aggregation of tasks. Let p be the overall proportion of tasks that start service. Suppose we thin arriving tasks with probability p. Then the transform of the virtual waiting time in queue is

$$E\left[e^{-s\boldsymbol{W}}\right] = \frac{1 - p\sum_{j=1}^{J} l_j E\left[\boldsymbol{C}_j\right]}{1 - \sum_{j=1}^{J} l_j p\left[\frac{1 - E\left[e^{-s\boldsymbol{C}_j}\right]}{s}\right]}. \tag{A3-6}$$

On the other hand

$$\begin{aligned} p &= \sum_{j=1}^{J} \frac{l_j}{\sum_{k=1}^{J} l_k} E\left[e^{-q_j \boldsymbol{W}}\right] \\ &= \sum_{j=1}^{J} \frac{l_j}{\sum_{k=1}^{J} l_k} \left[\frac{1 - pr}{1 - p\sum_{k=1}^{J} r_k d_k\left(q_j\right)}\right] \equiv B(p). \end{aligned} \tag{A3-7}$$

Note, then $B(p)$ is decreasing in p on [0,1/ r] and is always between 0 and 1. Hence, the above equation always has a unique solution $\tilde{p}$ in [0,1]. The approximation for the probability a task of type j starts service is

$$y\left(q_j\right) = E\left[e^{-q_j W}\right] = \frac{1-\tilde{p}r}{1-\tilde{p}\sum_{k=1}^{J} r_k d_k\left(q_j\right)}. \tag{A3-8}$$

A more refined approximation makes use of the fact that the task that arrives when the server is idle always starts service. First of all,

$$E\left[e^{-sW}\right] = P\{W=0\} + \left[1-P\{W=0\}\right]E\left[e^{-sW} \mid W>0\right]. \tag{A3-9}$$

Hence, using the P-K formula for $E\left[e^{-sW}\right]$ results in

$$E\left[e^{-sW} \mid W>0\right] = \frac{1-r}{r}\frac{\sum_{j=1}^{J} r_j d_j(s)}{1-\sum_{j=1}^{J} r_j d_j(s)}. \tag{A3-10}$$

If a proportion p of tasks arriving during a busy period gets to the server, our usual self-thinning approximation results in

$$E\left[e^{-sW} \mid W>0\right] = \frac{1-rp}{1-\sum_{j=1}^{J} r_j d_j(s)}\frac{\sum_{j=1}^{J} r_j d_j(s)}{r} \tag{A3-11}$$

Thus, p has to satisfy the relation

$$p = \sum_{j=1}^{J} \frac{l_j}{l} E\left[e^{-q_j W} \mid W > 0 \right]$$

$$= \sum_{j=1}^{J} \frac{l_j}{l} \frac{1 - r\,p}{1 - \sum_{k=1}^{J} r_k d_k\left(q_j\right)} \frac{\sum_{k=1}^{J} r_k d_k\left(q_j\right)}{r}. \qquad \text{(A3-12)}$$

The same argument as before shows that this equation has a unique solution $\tilde{p}_b$ in $(0, 1 \wedge r^{-1})$.

Let

$$y_+\left(q_j\right) = \sum_{j=1}^{J} \frac{l_j}{l} \frac{1 - \tilde{p}_b\, r}{1 - \sum_{k=1}^{J} r_k d_k\left(q_j\right)} \frac{\sum_{k=1}^{J} r_k d_k\left(q_j\right)}{r} \qquad \text{(A3-13)}$$

for j=1,…, J.

The expected length of a busy period satisfies the approximate relation

$$E[\boldsymbol{B}] = \sum_{j=1}^{J} \frac{l_j}{l} E\left[\boldsymbol{C}_j\right] + \tilde{p}_b\, r\, E[\boldsymbol{B}] \qquad \text{(A3-14)}$$

and so, approximately,

$$E[\boldsymbol{B}] = \frac{\sum_{j=1}^{J} \frac{l_j}{l} E\left[\boldsymbol{C}_j\right]}{1 - \tilde{p}_b\, r}. \qquad \text{(A3-15)}$$

Since

$$P\{\boldsymbol{W} = 0\} = \frac{l^{-1}}{l^{-1} + E[\boldsymbol{B}]} \qquad \text{(A3-16)}$$

our final approximation is

$$y_B(q_j) = P\{W > 0\} y_+(q_j) + P\{W = 0\}$$
$$= \frac{1 - \tilde{p}_b r + r\, y_+(q_j)}{1 + r(1 - \tilde{p}_b)}. \qquad \text{(A3-17)}$$