This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2015.2454771, IEEE Transactions on Automatic Control

A cμ Rule for Two-Tiered Parallel Servers

Soroush Saghafian, Michael H. Veatch

.

***Abstract*—The cμ rule is known to be optimal in many queueing systems with memoryless service and inter-arrival times. We establish a cμ rule for a parallel flexible server system with two tiers. Multiple job classes arrive and wait in separate queues. The first tier contains classes that can only be served by one server each, and the second tier contains a class that can be served by all servers. The cμ preferences give priority to the first tier. Sequencing decisions are made to minimize linear holding costs.**

***Index Terms*—parallel servers, queueing control, greedy policies.**

I. Introduction

While most queueing network control problems have complex optimal policies consisting of switching curves or surfaces in the state space, the multiclass queue often has a simple optimal policy: the $c\mu$ rule. This rule greedily removes customers from the system. We establish a cμ rule for systems with parallel servers and certain cμ preferences. Multiple job classes arrive according to Poisson processes and wait in separate queues. Heterogeneous servers each can serve a subset of the classes at different rates, with exponential service times. Sequencing decisions are made to minimize linear holding costs.

For a system with two "tiers" of customer classes, one tier containing classes that can only be served by one server each and the second tier containing a class that can be served by all servers, we prove that if the $c\mu$ rule gives priority to the first tier then it is optimal under fairly general conditions. Our proofs (i) adapt the stochastic coupling time-interchange method of [1] to multiple servers and (ii) use operator-based dynamic programming arguments.

Parallel server models are useful for describing the set of job classes that each machine can process in a factory or the cross training of workers to perform some subset of the tasks, e.g., serve certain classes of customers in a telephone call center. One application of our two-tiered model is an intenational customer support center with calls in English, Spanish, and French (see Figure 1). Server 1 speaks only English, while servers 2 and 3 are bilingual. This application fits well with the three scenarios for which we prove $c\mu$ rules. Class-independent service rates (Theorem 2) represent fully bilingual servers, equally fluent in both languages. Specialists (Theorem 3) is the scenario where server 1 is faster in English than servers 2 or 3. Collaborative service (Theorem 4), where more than one server can serve a job, is rare in support centers, but is more common in manufacturing. Two tiers frequently occur in service systems when tier 1 contains a low-skill task, which all workers can perform, and tier 2 contains skilled tasks. Distributed computing systems are another example of collaborative service. If most processors handle local as well as networked jobs, a two-tiered structure applies.

The parallel server control problem has been studied by [2], [3], [4] and [5] and applied to prioritizing patients in a hospital in [6]. A $c\mu$ rule for the "N" system with two servers and two customer classes is proven in [7] for a model that allows upgrades. Optimality of the $c\mu$ rule for the "W" system (three classes and two servers) is established in [8] for a model that allows server failures. Our two-tiered structure is more complex than either of these systems. The $c\mu$ rule has also been studied for queues with abandonments in [9] and flexible servers in a general network in [10]. The last paper uses throughput maximization, which is an easier task than minimizing holding costs. Heavy traffic optimality of a generalized $c\mu$ rule for the multiclass queue is discussed in [11] and [12] and extended to networks in [13]. Unlike their work, our result does not require any heavy traffic assumption.

II. A Two-Tiered Flexible Server System

Jobs arrive in class $j = 1, \ldots, N$ according to independent Poisson processes with rates λ_j and are served by separate servers at rate μ_{jj}. Servers $j = 2, \ldots, N$ can also serve class 1 at rate μ_{j1}. Service times are independent and exponentially distributed. Service is preemptive. Let $x_j(t)$ be the number of class j jobs in the system at time t. The control action is $u_{ij} = 1$ if server i serves class j and 0 otherwise. We will also write u_i for the class served by server i, e.g., $u = (2, 0)$ if server 1 is serving class 2 and server 2 is idle. Admissible policies $u = \{u(t), t \geq 0\}$ are nonanticipating, assign a server to at most one class at a time, and only assign as many servers to a class as there are jobs. Let $\mathcal{U}(x)$ be the set of admissible actions in state x.

Soroush Saghafian is with Harvard Kennedy School, Harvard University, Cambridge, MA 02138, Soroush_Saghafian@hks.harvard.edu.

Michael H. Veatch is with Department of Mathematics, Gordon College, Wenham, MA 01984, mike.veatch@gordon.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2015.2454771, IEEE Transactions on Automatic Control

An optimal policy minimizes expected discounted holding cost

$$J_u(x) = E_{x,u} \int_0^\infty c^{\mathrm{T}} x(t) e^{-\alpha t} dt,$$

where $\alpha > 0$ is the discount rate, $c > 0$ is the vector of holding costs and $E_{x,u}$ denotes expectation with respect to the initial state $x(0) = x$ and policy u. In this context, we can restrict our attention to stationary Markov policies and replace $u(t)$ with $u(x(t))$. We say that the system is *stabilizable* if there exists an admissible policy u under which $\{x(t), t \geq 0\}$ has a *finite mean* equilibrium distribution (or equivalently, if the long-run average holding cost under u is finite).

Consider the uniformized, discrete-time Markov chain and rescale time so that the potential event rate is $\Lambda + \sum_{j=1}^N \mu_{j\max} = 1$, where $\Lambda = \sum_{j=1}^N \lambda_j$ and $\mu_{j\max} = \max\{\mu_{j1}, \mu_{jj}\}$. Henceforth $x(t)$ and $u(t)$, $t = 0, 1, 2, \ldots$ will denote the state and control at period t. The uniformized process is equivalent in that the finite-horizon cost

$$J_u(x, T) = E_{x,u} \sum_{t=0}^{T-1} \beta^t c^{\mathrm{T}} x(t)$$

has limit $J_u(x)$ as $T \to \infty$. Here $\beta = (1+\alpha)^{-1}$ and u refers to a finite horizon policy $u(x,t)$, $t = 0, \ldots, T-2$. Also let $J_*(x,T)$ and $J_*(x)$ denote the optimal costs and $u^*(x,t;T)$ and $u^*(x)$ the corresponding optimal policies. Our proofs of optimal policy characteristics are for all $T < \infty$. Thus, they also hold for an infinite horizon. Since they are optimal for all $\beta < 1$, they are also optimal for the average cost criterion.

Let $D_j x = x - e_j$, where e_j is the unit vector with jth component equal to one. Define the functional operators T_a, T_u, and T_* on any function J defined on the state space as follows:

$$T_a J(x) = \sum_{j=1}^N \lambda_j J(x + e_j) \tag{1}$$

$$\begin{aligned} T_u J(x) &= \sum_{i=1}^N \sum_{j \in \{1,i\}} u_{ij} \mu_{ij} J(D_j x) \\ &+ (1 - \Lambda - \sum_{i=1}^N \sum_{j \in \{1,i\}} u_{ij} \mu_{ij}) J(x) \\ &= (1-\Lambda) J(x) - \sum_{i=1}^N \sum_{j \in \{1,i\}} u_{ij} \mu_{ji} \Delta_j J(D_j x) \end{aligned} \tag{2}$$

$$T_* J(x) = \min_{u \in \mathcal{U}(x)} T_u J(x) \tag{3}$$

where $\Delta_j J(x) = J(x + e_j) - J(x)$. Combining these, $\mathbf{T} = c^{\mathrm{T}} x + \beta (T_a + T_*)$ is the dynamic programming operator, with

$$J_*(x, T+1) = \mathbf{T} J_*(x, T). \tag{4}$$

Three proofs will use stochastic coupling arguments. To construct coupled processes, let $W(t)$, $t = 0, 1, 2, \ldots$ be independent uniform $[0,1]$ random variables. The transition at time t is determined by $W(t)$ and $u(t)$, say $\phi(W(t), u(t))$. Let $x'(\cdot)$ be a process that uses policy u' and transitions $\phi(W(t), u'(t))$, $t = 0, 1, 2, \ldots$. We say that x and x' are coupled because they have the same potential transitions, determined by $W(t)$. We start by showing that all optimal policies are nonidling.

Lemma 1. *All optimal policies are nonidling: no server is idle if there is a job that it can serve.*

Proof. Use induction on T. For $T = 1$, nonidling is optimal because greedy policies are nonidling. Suppose all optimal policies $u^*(x,t;T)$ are nonidling and consider the time horizon $T+1$. Let $u = u^*(x,t;T+1)$. By the inductive hypothesis, u is nonidling for $t \geq 1$. Suppose u idles server i that could serve class j at $t = 0$ in state x. In the coupled process x', use the nonidling action at time 0, i.e., $u_i'(0) = j$. If server i completes service, the next states differ: $x(1) - x'(1) = e_j$ with probability μ_{ij}. The processes then use the same action $u'(t) = u(t)$, $t \geq 1$ whenever admissible. When this action is not admissible (because $x_j'(t) = 0$ or, if $j = 1$, all class 1 jobs are served by other servers), idle server i; the processes will merge if server i completes service, i.e., $x(s) = x'(s)$ with probability one, $s \geq t$. Let τ be the minimum of $T-1$ and the time at which the processes merge; if they never separate, set $\tau = 0$. The cost difference is $J_u(x, T+1) - J_{u'}(x, T+1) = E_x \sum_{t=1}^\tau \beta^t \mu_{ij} c_j > 0$, contradicting the optimality of u. □

Using the static allocation LP and Theorem 1 of [8], the following stability result is easily obtained.

Theorem 1 (Stability)**.** *For $j = 1, \ldots, N$, let $\rho_j = \lambda_j / \mu_{jj}$, $a_j = \mu_{j1}/\mu_{11}$, and define $\bar{\rho} = \max_{j=2,\ldots,N} \rho_j$. (i) The two-tiered system is stabilizable if (a) $\bar{\rho} < 1$, and (b) $\rho_1 - \sum_{j=2}^N a_j (1 - \rho_j) < 1$. (ii) The two-tiered system is not stabilizable if either (a) $\bar{\rho} > 1$, or (b) $\rho_1 - \sum_{j=2}^N a_j (1 - \rho_j) > 1$.*

III. Optimality of the $c\mu$ Rule

We assume the system is stabilizable and that

$$c_j \mu_{jj} \geq c_1 \mu_{j1}, \quad j = 2, \ldots N. \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2015.2454771, IEEE Transactions on Automatic Control

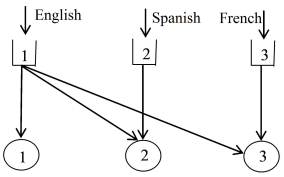

Fig. 1: A two-tiered network with three servers and classes.

Then the $c\mu$ rule is: If $x_j > 0$, server j serves class j; otherwise, it is available to serve class 1, $j = 2, \ldots, N$. Available servers with largest μ_{j1} have priority for class 1.

Our main result is that, under (5) and some additional conditions, the $c\mu$ rule is optimal. First, consider class-independent service rates:

$$\mu_{jj} = \mu_{j1}, \quad j = 2, \ldots N. \tag{6}$$

Using (6), (5) reduces to $c_j \geq c_1, \quad j = 2, \ldots N$.

Theorem 2 (Class-Independent Service Rates)**.** *If for the two-tiered system (5) and (6) hold, then the $c\mu$ rule is optimal.*

Proof. Use stochastic coupling and induction on T. For $T = 1$, the $c\mu$ rule is optimal because it is greedy. Suppose the $c\mu$ rule is optimal for some T and consider the horizon $T + 1$. Let u be an optimal policy for the process $x(\cdot)$. By the inductive hypothesis, u can be chosen to follow the $c\mu$ rule for $t = 1, \ldots, T$. Suppose server $j > 1$ does not follow the $c\mu$ rule at $t = 0$, i.e., $u_j(0) = 1$, $x_j \geq 1$, and $c_j\mu_{jj} > c_1\mu_{j1}$. In the coupled process $x'(\cdot)$, use the dedicated action at time 0, i.e., $u'_j(0) = j$. If server j completes service, the next states differ: $x(1) - x'(1) = e_j - e_1$ with probability μ_{jj}. The processes then use the same action $u'(t) = u(t)$, $t \geq 1$ whenever admissible. When this action is not admissible (because $x'_j(t) = 0$), server j serves class 1: $u'_j(t) = 1$. The processes will merge if server j completes service. Let τ be the minimum of $T - 1$ and the time at which the processes merge; if they never separate, set $\tau = 0$. The cost difference is $J_u(x, T+1) - J_{u'}(x, T+1) = E_x \sum_{t=1}^{\tau} \beta^t \mu_{jj}(c_j - c_1) > 0$. $\square$

We have shown that u is not optimal and the induction is complete.

Now consider the condition that server 1 is a *specialist* for class 1:

$$\mu_{j1} \leq \mu_{11} \quad j = 2, \ldots N. \tag{7}$$

Theorem 3 (Specialist)**.** *If for the two-tiered system (5) and (7) hold, then the $c\mu$ rule is optimal.*

Proof. Consider a specific server $j \neq 1$. Since nonidling is optimal by Lemma 1, server j follows the $c\mu$ rule in states where $x_1 = 0$ or $x_j = 0$. That leaves states $\mathcal{A}_j = \{x \in Z_+^n : x_1 \geq 1\,,\, x_j \geq 1\}$. Consider state $x \in \mathcal{A}_j$. If $x_1 \geq N$, then the min in (3) decomposes into a separate min for each server, and server j follows the $c\mu$ rule at time T if

$$\mu_{jj}\Delta_j J(D_j x) \geq \mu_{j1}\Delta_1 J(D_1 x) \tag{8}$$

holds for $J(\cdot) = J_*(\cdot, T)$. Condition (8) is also sufficient for server j to follow the $c\mu$ rule in states with $x_1 < N$, for if the action $u_j = j$ is preferable without considering the impact on other servers, then it is optimal. Let $\mathcal{J}$ be the set of functions defined on the state space that satisfy property (8) for all $x \in \mathcal{A}_j$ and all $j = 2, \ldots N$. Since $J_*(\cdot, 0) = 0$, $J_*(\cdot, 0) \in \mathcal{J}$. Using induction on T, Lemma 2 (see below) and (4), we have $J_*(\cdot, T) \in \mathcal{J}$ for all T. $\square$

Lemma 2 (Preservation)**.** *Consider a two-tiered system that satisfies (5) and (7). If $J \in \mathcal{J}$, then $\mathbf{T}J \in \mathcal{J}$.*

Proof. First note that $c^{\mathrm{T}}x$ and T_a preserve property (8) because they are nonnegative linear combinations and shifts. It remains to show that T_* preserves (8). Assume $J \in \mathcal{J}$ and consider a specific server $j \neq 1$. Let $\mathcal{B}(x) = \{i : x_i = 0\}$, $\mathcal{N}(x) = \{i : x_i \geq 1\}$, and $\mathcal{B}_1(x) = \{i : u_i(x) = 1\}$ (the servers that are serving class 1). Since $J \in \mathcal{J}$, servers with nonempty queues (other than server 1) serve their own class, and (3) can be written

$$\begin{aligned} T_*J(x) &= (1-\Lambda)J(x) - \sum_{i \in \mathcal{B}_1(x)} \mu_{i1}\Delta_1 J(D_1 x) \\ &\quad - \sum_{i \in \mathcal{N}(x)\setminus\{1\}} \mu_{ii}\Delta_i J(D_i x). \end{aligned} \tag{9}$$

We let $x \in \mathcal{A}_j$ and consider five cases: (1) $x_1 = 1$ and $x_j \geq 1$, (2) $x_1 \geq 2$, $x_j \geq 2$, and $\mathcal{B}(x) = \emptyset$, (3) $x_1 \geq 2$, $x_j \geq 2$, and $\mathcal{B}(x) \neq \emptyset$, (4) $x_1 \geq 2$, $x_j = 1$, and $\mathcal{B}(x) = \emptyset$, and (5) $x_1 \geq 2$, $x_j = 1$, and $\mathcal{B}(x) \neq \emptyset$.

Case 1 ($x_1 = 1$ and $x_j \geq 1$): By (7), $\mathcal{B}_1(x) = \mathcal{B}_1(x - e_j) = \{1\}$, while $\mathcal{B}_1(x - e_1) = \emptyset$. Note that $\mathcal{N}(x - e_1) = \mathcal{N}(x)\setminus\{1\}$. Then

$$\begin{aligned} \Delta_1 T_* J(D_1 x) &= T_*J(x) - T_*J(x - e_1) \\ &= (1-\Lambda)\Delta_1 J(D_1 x) - \mu_{11}\Delta_1 J(D_1 x) \\ &\quad - \sum_{i \in \mathcal{N}(x)\setminus\{1\}} \mu_{ii}\Delta_1\Delta_i J(D_i D_1 x) \\ &= (1-\Lambda)\Delta_1 J(D_1 x) - \mu_{11}\Delta_1 J(D_1 x) \\ &\quad - \sum_{i \in \mathcal{N}(x)\setminus\{1\}} \mu_{ii}\Delta_i\Delta_1 J(D_i D_1 x) \\ &= (1-\Lambda)\Delta_1 J(D_1 x) - \mu_{11}\Delta_1 J(D_1 x) \\ &\quad - \sum_{i \in \mathcal{N}(x)\setminus\{1\}} \mu_{ii}[\Delta_1 J(D_1 x) - \Delta_1 J(D_i D_1 x)] \end{aligned}$$

$$\begin{aligned} \Delta_j T_* J(D_j x) &= T_*J(x) - T_*J(x - e_j) \\ &\geq (1-\Lambda)\Delta_j J(D_j x) \\ &\quad - \sum_{i \in \mathcal{N}(x)} \mu_{ii}[\Delta_j J(D_j x) - \Delta_j J(D_i D_j x)], \end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2015.2454771, IEEE Transactions on Automatic Control

where the inequality for $\Delta_j T_* J(D_1 x)$ is obtained by considering the case where $x_j > 1$, and hence by (perhaps infeasibly) assigning server j to class j at state $x - e_j$. Combining the above,

$$\begin{aligned}&\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x) \geq \\ &(1 - \Lambda - \sum_{i\in\mathcal{N}(x)} \mu_{ii})[\mu_{jj}\Delta_j J(D_j x) - \mu_{j1}\Delta_1 J(D_1 x)] \\ &+\mu_{jj}\mu_{11}\Delta_j J(D_j D_1 x) \\ &+ \sum_{i\in\mathcal{N}(x)} \mu_{ii}[\mu_{jj}\Delta_j J(D_i D_j x) - \mu_{j1}\Delta_1 J(D_i D_1 x)] \geq 0,\end{aligned}$$

where the last inequality follows from all lines in the previous expression being non-negative because $J \in \mathcal{J}$. Note that $D_i D_j x = D_j(D_i x)$ and $D_i D_1 x = D_1(D_i x)$. Also, $\Delta_j J(\cdot) \geq 0$ since idling is not optimal and that $J \in \mathcal{J}$.

Case 2 ($x_1 \geq 2$, $x_j \geq 2$, and $\mathcal{B}(x) = \emptyset$): By (5), all servers serve their own class in states x, $x - e_1$, and $x - e_j$. Again differencing (9),

$$\begin{aligned}\Delta_1 T_* J(D_1 x) &= (1-\Lambda)\Delta_1 J(D_1 x) \\ &- \sum_{i=1}^{N} \mu_{ii}[\Delta_1 J(D_1 x) - \Delta_1 J(D_i D_1 x)] \quad (10) \\ \Delta_j T_* J(D_j x) &= (1-\Lambda)\Delta_j J(D_j x) \\ &- \sum_{i=1}^{N} \mu_{ii}[\Delta_j J(D_j x) - \Delta_j J(D_i D_j x)].\end{aligned}$$

Thus,

$$\begin{aligned}&\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x) \\ &= (1 - \Lambda - \sum_{i=1}^{N} \mu_{ii})[\mu_{jj}\Delta_j J(D_j x) - \mu_{j1}\Delta_1 J(D_1 x)] \\ &+ \sum_{i=1}^{N} \mu_{ii}[\mu_{jj}\Delta_j J(D_i D_j x) - \mu_{j1}\Delta_1 J(D_i D_1 x)] \geq 0,\end{aligned}$$

where the inequality follows from both lines in the previous expression being non-negative because $J \in \mathcal{J}$.

Case 3 ($x_1 \geq 2$, $x_j \geq 2$, and $\mathcal{B}(x) \neq \emptyset$): By (8) and (7), server 1 and some servers in $\mathcal{B}(x)$ serve class 1, while the others may idle. Note that $\mathcal{B}_1(x - e_j) = \mathcal{B}_1(x)$. Moreover, $\mathcal{B}_1(x - e_1) = \mathcal{B}_1(x)$ if $x_1 > |\mathcal{B}_1(x)|$, and $\mathcal{B}_1(x - e_1) \subset \mathcal{B}_1(x)$ otherwise. Thus, a lower bound for $\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x)$ can be obtained by assuming $\mathcal{B}_1(x - e_1) = \mathcal{B}_1(x)$. Note that this provides an upper bound for $\mu_{j1}\Delta_1 T_* J(D_1 x)$, and hence, a lower bound for $\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x)$. Doing so we have:

$$\begin{aligned}&\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x) \geq \\ &(1 - \Lambda - \sum_{i\in\mathcal{B}_1(x)} \mu_{i1} \\ &\qquad - \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii})[\mu_{jj}\Delta_j J(D_j x) - \mu_{j1}\Delta_1 J(D_1 x)] \\ &+ \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\mu_{jj}\Delta_j J(D_1 D_j x) - \mu_{j1}\Delta_1 J(D_1 D_1 x)] \\ &+ \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii}[\mu_{jj}\Delta_j J(D_i D_j x) - \mu_{j1}\Delta_1 J(D_i D_1 x)] \geq 0,\end{aligned}$$

where the last inequality follows from each line in the previous inequality being non-negative because $J \in \mathcal{J}$.

Case 4 ($x_1 \geq 2$, $x_j = 1$, and $\mathcal{B}(x) = \emptyset$): In states x and $x - e_1$, all servers serve their own class as in Case 2, so (10) applies. However, $\mathcal{B}_1(x - e_j) = \{1, j\}$. Differencing (9),

$$\begin{aligned}\Delta_j T_* J(D_j x) &= (1-\Lambda)\Delta_j J(D_j x) \\ &- \mu_{jj}\Delta_j J(D_j x) + \mu_{j1}\Delta_1 J(D_j D_1 x) \\ &- \sum_{i:i\neq j} \mu_{ii}[\Delta_j J(D_j x) - \Delta_j J(D_i D_j x)].\end{aligned}$$

Hence, we have:

$$\begin{aligned}&\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x) = \\ &(1 - \Lambda - \sum_{i=1}^{N} \mu_{ii})[\mu_{jj}\Delta_j J(D_j x) - \mu_{j1}\Delta_1 J(D_1 x)] \\ &+ \sum_{i:i\neq j} \mu_{ii}[\mu_{jj}\Delta_j J(D_i D_j x) - \mu_{j1}\Delta_1 J(D_i D_1 x)] \geq 0,\end{aligned}$$

where the last inequality follows from each line in the previous equality being non-negative because $J \in \mathcal{J}$.

Case 5 ($x_1 \geq 2$, $x_j = 1$, and $\mathcal{B}(x) \neq \emptyset$): If $x_1 > |B(x)|+1$, then all servers with empty queues serve class 1 in states x, $x - e_1$, and $x - e_j$; otherwise, some servers idle. We consider these two subcases separately.

Subcase 5.1 ($x_1 > |\mathcal{B}(x)| + 1$): There is no idling in states x, $x - e_1$, or $x - e_j$ and $\mathcal{B}_1(x)$=$\mathcal{B}_1(x - e_1) = \mathcal{B}(x) \cup \{1\}$, while $\mathcal{B}_1(x - e_j) = \mathcal{B}(x) \cup \{1, j\}$. Using (9) in these states,

$$\begin{aligned}\Delta_1 T_* J(D_1 x) &= (1-\Lambda)\Delta_1 J(D_1 x) \\ &- \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\Delta_1 J(D_1 x) - \Delta_1 J(D_1 D_1 x)] \\ &- \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii}[\Delta_1 J(D_1 x) - \Delta_1 J(D_i D_1 x)] \\ \Delta_j T_* J(D_j x) &= (1-\Lambda)\Delta_j J(D_j x) - \mu_{jj}\Delta_j J(D_j x) + \mu_{j1}\Delta_1 J(D_j D_1 x) \\ &- \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\Delta_j J(D_j x) - \Delta_j J(D_j D_1 x)] \\ &- \sum_{i\in\mathcal{N}(x)\setminus\{1,j\}} \mu_{ii}[\Delta_j J(D_j x) - \Delta_j J(D_i D_j x)].\end{aligned}$$

Therefore, we have:

$$\begin{aligned}&\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x) \\ &= (1 - \Lambda - \sum_{i\in\mathcal{B}_1(x)} \mu_{i1} \\ &\qquad - \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii})[\mu_{jj}\Delta_j J(D_j x) - \mu_{j1}\Delta_1 J(D_1 x)] \\ &+ \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\mu_{jj}\Delta_j J(D_1 D_j x) - \mu_{j1}\Delta_1 J(D_1 D_1 x)] \\ &+ \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii}[\mu_{jj}\Delta_j J(D_i D_j x) - \mu_{j1}\Delta_1 J(D_i D_1 x)] \geq 0,\end{aligned}$$

where the last inequality follows from each line in the previous equality being non-negative (since $J \in \mathcal{J}$).

Subcase 5.2 ($x_1 \leq |\mathcal{B}(x)|+1$): Let $k \in \mathcal{B}_1(x)$ be the server that becomes idle in state $x - e_1$, i.e., $\mu_{k1} = \min_{i\in\mathcal{B}_1(x)} \mu_{i1}$. First, assume $\mu_{k1} \geq \mu_{j1}$, so that server j becomes idle in state

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2015.2454771, IEEE Transactions on Automatic Control

$x - e_j$. From (9), we have:

$$\begin{aligned}
\Delta_1 T_* J(D_1 x) &= (1-\Lambda)\Delta_1 J(D_1 x) - \mu_{k1}\Delta_1 J(D_1 x) \\
&- \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\Delta_1 J(D_1 x) - \Delta_1 J(D_1 D_1 x)] \\
&- \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii}[\Delta_1 J(D_1 x) - \Delta_1 J(D_i D_1 x)] \\
\Delta_j T_* J(D_j x) &= (1-\Lambda)\Delta_j J(D_j x) - \mu_{jj}\Delta_j J(D_j x) \\
&- \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\Delta_1 J(D_1 x) - \Delta_1 J(D_1 D_1 x)] \\
&- \sum_{i\in\mathcal{N}(x)\setminus\{1,j\}} \mu_{ii}[\Delta_j J(D_j x) - \Delta_j J(D_i D_j x)].
\end{aligned}$$

Therefore, we have:

$$\begin{aligned}
&\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x) \\
&= (1-\Lambda - \sum_{i\in\mathcal{B}_1(x)} \mu_{i1} \\
&\qquad - \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii})[\mu_{jj}\Delta_j J(D_j x) - \mu_{j1}\Delta_1 J(D_1 x)] \\
&+\mu_{j1}\mu_{k1}\Delta_1 J(D_1 x) - \mu_{jj}\mu_{jj} J(D_j x) \\
&+ \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\mu_{jj}\Delta_j J(D_1 D_j x) - \mu_{j1}\Delta_1 J(D_1 D_1 x)] \\
&+ \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii}[\mu_{jj}\Delta_j J(D_i D_j x) - \mu_{j1}\Delta_1 J(D_i D_1 x)] \\
&\geq (1-\Lambda - \sum_{i\in\mathcal{B}_1(x)} \mu_{i1} \\
&\qquad - \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii})[\mu_{jj}\Delta_j J(D_j x) - \mu_{j1}\Delta_1 J(D_1 x)] \\
&+ \sum_{i\in\mathcal{B}_1(x)} \mu_{i1}[\mu_{jj}\Delta_j J(D_1 D_j x) - \mu_{j1}\Delta_1 J(D_1 D_1 x)] \\
&+ \sum_{i\in\mathcal{N}(x)\setminus\{1\}} \mu_{ii}[\mu_{jj}\Delta_j J(D_i D_j x) - \mu_{j1}\Delta_1 J(D_i D_1 x)] \geq 0,
\end{aligned}$$

where the first inequality follows from the assumption that $\mu_{k1} \geq \mu_{j1}$, and the second inequality follows from each line in the previous inequality being non-negative because $J \in \mathcal{J}$.

Finally, if $\mu_{k1} < \mu_{j1}$, $\mathcal{B}_1(x - e_j) = (\mathcal{B}_1(x)\setminus\{k\}) \cup \{j\} = \mathcal{B}_1(x - e_1) \cup \{j\}$. Following a similar line of proof to the previous case, $\mu_{jj}\Delta_j T_* J(D_j x) - \mu_{j1}\Delta_1 T_* J(D_1 x) \geq 0$ in this case as well, and hence the proof is complete. □

Our final result establishes optimality of the $c\mu$ rule when we change the model by allowing servers to collaborate on jobs. Service rates are additive when collaborating. For example, if $N = 3$ and $x = (1, 0, 0)$, the action $u_{11} = u_{21} = u_{31} = 1$ serves class 1 with rate $\mu_{11} + \mu_{21} + \mu_{31}$.

Theorem 4 (Collaboration)**.** *If for the two-tiered system collaboration is allowed and (5) holds, then the $c\mu$ rule is optimal.*

Proof First we prove the theorem for $N = 3$ servers. Suppose the $c\mu$ rule is optimal for some T and consider the horizon $T+1$. Let u be an optimal policy. By the inductive hypothesis, u can be chosen to follow the $c\mu$ rule for $t = 1, \ldots, T$. Suppose u does not follow the $c\mu$ rule at $t = 0$. Since the system structure is symmetric in servers 2 and 3, we may assume that server 2 does not follow the $c\mu$ rule, i.e., $u_{21}(0) = 1$, $x_2 \geq 1$, and $c_2\mu_{22} > c_1\mu_{21}$. Use stochastic coupling and time interchange. Let $\tilde{x}$ be the process with policy $\tilde{u}$, $\tilde{x}(0) = x$ and $\tilde{W}(t) = W(t)$, except that times 0 and 1 are interchanged for $\tilde{W}$. Let $\mathcal{E}(t)$ and $\tilde{\mathcal{E}}(t)$ be the *event* at time t in x and $\tilde{x}$, which is either A_j (class j arrival), S_{ij} (server i completes a class j service), or $\emptyset$ (self transition). Also, let S_j denote a class j service completion by any server. We will construct $\tilde{u}$ so that in each case (i) the processes merge, i.e., $\tilde{x}(t) = x(t)$ with probability 1, $t \geq 2$, or (ii) the expected cost for $t = 2, \ldots, T$ is the same for x and $\tilde{x}$. Consider six cases for the initial state.

Case 1. $x \geq (2, 1, 2)$. Since $u(1)$ is known (dedicated), interchange $t = 0, 1$ for $\tilde{u}$. Note that $\tilde{u}$ is feasible because $\tilde{x}(1) \geq (1, 0, 1)$. Since $\widetilde{W}$ and $\tilde{u}$ use the same time interchange, the process merges.

Case 2. $x_1 = 1$, $x_2 \geq 1$, $x_3 \geq 2$. The action $u(1)$ is not known at $t = 0$, so we cannot use the time interchange to construct $\tilde{u}$. Set $\tilde{u}_{ii}(0) = 1$ (dedicated). If $\widetilde{\mathcal{E}}(0) = S_1$, then server 1 is starved and server 2 idles, $\tilde{u}_{11}(1) = \tilde{u}_{2j}(1) = 0$; otherwise, $\tilde{u}(1) = u(0)$ (server 2 helps).

Observe that both systems have the same number of S_1 events by time 1 (either 0 or 1). The same is true of S_2 events because $u_{22}(0) = \tilde{u}_{22}(1) = 0$. Also, both policies serve class 3. Hence, the processes merge.

Case 3. $x_1 \geq 2$, $x_2 \geq 1$, $x_3 = 1$. Use $\tilde{u}$ from case 1 for servers 1 and 2. For server 3, $\tilde{u}$ uses the same *policy* as u: $\tilde{u}_{33}(0) = 1$; if $\widetilde{\mathcal{E}}(0) = S_3$, then $\widetilde{u}_{31}(1) = 1$, otherwise $\widetilde{u}_{33}(1) = 1$.

Both systems have the same number of S_{11}, S_{21}, and S_2 events by time 1 because $\widetilde{u}$ uses time interchange for servers 1 and 2. Also, the number of S_3 events by time 1 is the same in both systems. To see this, observe that if $\mathcal{E}(0) = S_3$, then either $\widetilde{\mathcal{E}}(0) = S_3$ or $\widetilde{\mathcal{E}}(1) = S_3$. However, the number of S_{31} events by time 1 can differ in the two systems. The events $\mathcal{E}(0) = S_3$, $\mathcal{E}(1) = S_{31}$, $\widetilde{\mathcal{E}}(0) = \emptyset$, and $\widetilde{\mathcal{E}}(1) = S_3$ occur with probability $\mu_3 \max\{\mu_{31} - \mu_{33}, 0\}$ and result in $x_1(2) - \widetilde{x}_1(2) = -1$. The symmetric events, $\widetilde{\mathcal{E}}(0) = S_3$, $\widetilde{\mathcal{E}}(1) = S_{31}$, $\mathcal{E}(0) = \emptyset$, and $\mathcal{E}(1) = S_3$ occur with the same probability and result in $x_1(2) - \widetilde{x}_1(2) = 1$. All other events result in $x_1(2) = \widetilde{x}_1(2)$. Therefore, the distribution of $x_1(2) - \widetilde{x}_1(2)$ is symmetric about 0 and the expected cost for $t = 2, \ldots, T$ is the same under u and $\tilde{u}$.

Case 4. $x_1 = 1$, $x_2 \geq 1$, $x_3 = 1$. Use $\tilde{u}$ from case 2 for servers 1 and 2 and from case 3 for server 3. Combining

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2015.2454771, IEEE Transactions on Automatic Control

the arguments from these cases, only buffer 1 may differ at $t = 2$ and again the distributions of $x(2)$ and $\widetilde{x}(2)$ are identical, making the expected cost for $t = 2, \ldots, T$ the same for both.

Case 5. $x_1 \geq 2$, $x_2 \geq 1$, $x_3 = 0$. Use $\tilde{u}$ from case 1 for servers 1 and 2. For server 3, $\tilde{u}$ uses the same *policy* as u: $\tilde{u}_{31}(0) = 1$; if $\widetilde{\mathcal{E}}(0) = A_3$, then $\widetilde{u}_{33}(1) = 1$, otherwise $\widetilde{u}_{31}(1) = 1$. The number of S_{31} and S_3 events by time 1 can differ in the two systems, so only $x_2(2) = \widetilde{x}_2(2)$ with probability 1. Again both systems have the same policy for server 3 and the distributions of $x(2)$ and $\widetilde{x}(2)$ are identical, making the expected cost for $t = 2, \ldots, T$ equal.

Case 6. $x_1 = 1$, $x_2 \geq 1$, $x_3 = 0$. Use $\tilde{u}$ from case 2 for servers 1 and 2 and from case 5 for server 3. The argument from case 5 applies.

In all cases, $u_{21}(0) = 1$ while $\widetilde{u}_{22}(0) = 1$, giving a relative expected cost of $J_u(x, T+1) - J_{\tilde{u}}(x, T+1) = E_x \beta c^{\mathrm{T}}[x(1) - \widetilde{x}(1)] = \beta(c_2\mu_{22} - c_1\mu_{21}) > 0$. The last equality uses the fact that μ_{ij} is the probability of event S_{ij} when class j is being served by server i. We have shown that u is not optimal and the induction is complete.

The proof for N servers is similar with the following changes. Note that servers $2, \ldots, N$ have the same structure and again assume server 2 does not follow the $c\mu$ rule. Cases 1 and 2 are unchanged, with $x_3, \ldots, x_N \geq 2$. Cases 3 and 5 are replaced by $x_1 \geq 2$, $x_2 \geq 1$, and all possible partitions of classes $3, \ldots, N$ into three sets, with queue lengths of 0, 1, or greater than 1. Classes with queue lengths of 0 are treated as in Case 5, classes with queue lengths of 1 are treated as in Case 3, and classes with larger queue lengths can be ignored as in Case 1. Cases 4 and 6, with $x_1 = 1$ and $x_2 \geq 1$, are replaced in a similar manner. The details are omitted. □

IV. Other Networks

The "W" network is a special case of the two-tiered network with three classes but without server 1. When (5) holds, the $c\mu$ rule is optimal for this network; see [8]. Intuitively, the $c\mu$ rule is optimal for the two-tiered network because it maximizes the amount of work available to other servers. We conjecture that this principle applies whenever the greedy action does not by choice take work away from another server. Specifically, if (i) only server i can serve class i and (ii) $c_i\mu_{ii} \geq c_j\mu_{ij}$, $j \neq i$ then we conjecture that it is optimal for server i to give priority to class i. For example, in the "floater" network in Figure 2, if $c_3\mu_{33} \geq c_2\mu_{32}$ and $c_3\mu_{33} \geq c_1\mu_{12}$, then we expect server 3 to give priority to class 3. It would be of interest to prove this conjecture.

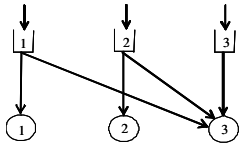

Fig. 2: The "floater" network.

References

[1] C. Buyukkoc, P. Varaiya, and J. Walrand, "The $c\mu$-rule revisited," *Adv. in Appl. Probab.*, vol. 17, pp. 237–238, 1985.

[2] J. Harrison, "Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete-review policies," *Ann. Appl. Probab.*, vol. 8, no. 3, pp. 822–848, 1998.

[3] S. Bell and R. Williams, "Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: Asymptotic optimality of a threshold policy," *Ann. Appl. Probab.*, vol. 11, no. 3, pp. 608–649, 2001.

[4] J. Harrison and M. López, "Heavy traffic resource pooling in parallel-server systems," *Queueing Systems*, vol. 33, no. 4, pp. 339–368, 1999.

[5] O. Akgun, R. Righter, and R. Wolff, "Partial flexibility in routeing and scheduling," *Adv. in Appl. Probab.*, vol. 45, pp. 673–691, 2013.

[6] S. Saghafian, W. Hopp, M. Van Oyen, J. Desmond, and S. Kronick, "Complexity-Augmented triage: A tool for improving patient safety and operational efficiency," *Manufacturing and Service Oper. Mang.*, vol. 16, pp. 329–345, 2014.

[7] D. Down and M. Lewis, "The N-network model with upgrades," *Probability in the Engineering and Informational Sciences*, vol. 24, no. 2, p. 171, 2010.

[8] S. Saghafian, M. Van Oyen, and B. Kolfal, "The "W" network and the dynamic control of unreliable flexible servers," *IIE Transactions*, vol. 43, no. 12, pp. 893–907, 2011.

[9] D. Down, G. Koole, and M. Lewis, "Dynamic control of a single-server system with abandonments," *Queueing Systems*, vol. 67, no. 1, pp. 63–90, 2011.

[10] S. Andradóttir, H. Ayhan, and D. G. Down, "Dynamic server allocation for queueing networks with flexible servers," *Operations Research*, vol. 51, no. 6, pp. 952–968, 2003.

[11] J. V. Mieghem, "Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule," *Ann. Appl. Probab.*, vol. 5, no. 3, pp. 809–833, 1995.

[12] A. Mandelbaum and A. L. Stolyar, "Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule," *Operations Research*, vol. 52, no. 6, pp. 836–855, 2004.

[13] J. Dai and W. Lin, "Maximum pressure policies in stochastic processing networks," *Operations Research*, vol. 53, no. 2, pp. 197–218, 2005.