Flexible Server Allocation and Customer Routing Policies for Two Parallel Queues when Service Rates are not Additive

Hyun-soo Ahn
Operations and Management Science, Ross School of Business
University of Michigan
701 Tappan Street, Ann Arbor, Michigan 48109-1234
hsahn@umich.edu

Mark E. Lewis
School of Operations Research and Information Engineering
Cornell University
226 Rhodes Hall, Ithaca, NY 14853
mark.lewis@cornell.edu

Revised February 9, 2012

Abstract

We consider the question of how routing and allocation can be coordinated to meet the challenge of demand variability in a parallel queueing system serving two types of customers. A decision-maker decides whether to keep customers at the station at which they arrived or to reroute them to the other station. At the same time, the decision-maker has two servers and must decide where to allocate their effort. We analyze this joint decision-making scenario, but add two important twists. First, we allow the combined service rate (when the servers work at the same station) to be *super-additive* or *sub-additive*. This captures positive or negative externalities that arise during collaboration. Second, routing costs are allowed to be strictly positive. We seek an optimal control policy under the discounted or long-run average cost criteria.

Our results show that in the super-additive case jobs should never be routed away from the lower cost queue. When jobs are rerouted from the higher cost queue to the low cost queue the optimal control is monotone in the respective queue lengths. Moreover, we show that the optimal allocation is a non-idling priority rule based on the holding costs. In the sub-additive case we find that the optimal policy need not exhibit such a simple structure. In fact, the optimal allocation need not prioritize one station (it may split the servers), and the optimal routing need not be monotone in the number of customers in each queue. We characterize the optimal policy for a few canonical cases, and discuss why intuitive policies need not be optimal in the general case. An extensive numerical study examines the benefit of dynamically controlling both routing and resource allocation; we discuss when using one of

the two levers – dynamic routing and dynamic allocation – is sufficient and when using both controls is warranted.

1 Introduction

In this paper we consider a fundamental challenge in service systems; how to cope with demand variability with limited resources. Historically, one approach to handle this difficulty is workload balancing through routing arriving customers to different stations. More recently, the use of a flexible workforce has gained considerable attention. We consider the question of how routing and allocation can be coordinated to meet the challenge of demand variability in a parallel queueing system serving two types of customers. A decision-maker decides whether to keep customers at the station at which they arrived or to reroute them to the other station. At the same time, the decision-maker has two servers and must decide where to allocate their effort. There are several areas in practice where such scenarios appear. In order to balance the workload among nurses, hospitals route patients to beds in different specialty units or floors (e.g., a urology patient may be hospitalized in a bed on the cardiology floor). Many of these hospitals cross-train nurses to cover patients in multiple floors and/or patients who need different sub-specialties. These hospitals can use both routing and resource allocation to reduce congestion. At many call centers, it is common practice to route customers to various servers (queues) while the (cross-trained) servers decide the sequencing of incoming calls. Flexible manufacturing systems can increase productivity via cross-trained workers while the routing question is answered at the time the order is placed.

We are interested in analyzing this joint decision-making scenario, but add two important twists. First, when routing is an option, the cost of routing is often assumed to be zero. The general setting we consider allows for positive routing costs. Second, in most of the agile workforce literature, when workers collaborate the service rate is assumed to be additive. We allow the combined service rate (when the servers work at the same station) to be *super-additive* or *sub-additive*. In other words, the service rate when two servers are pooled can be greater than (i.e., super-additive) or less than (sub-additive) the sum of individual service rates. This captures synergistic or disruptive effects when servers collaborate and allows us to examine how the efficiency of pooling affects the optimal use of a flexible workforce.

Our results show that in the super-additive case, jobs should only be rerouted from a higher cost queue to a lower cost queue and that the optimal routing policy is monotone in the queue lengths. The optimal allocation policy is a non-idling priority rule that always pools both servers based on the holding costs. In the sub-additive case, however, the optimal policy can be quite complex and difficult to characterize analytically. For example, the optimal allocation policy

need not prioritize a single queue or need not maximize service rates by splitting the servers. In particular, the rate of service is not necessarily even monotone in the queue lengths. Likewise, the routing policy may not be monotone in queue lengths. In general, none of the structure found in the super-additive case carries over to the sub-additive case. We identify several special cases under which always splitting the servers (except to avoid idling) is optimal. Through extensive numerical study, we examine which of the two controls – dynamic allocation and dynamic routing – is more effective and when it is most beneficial to use both at the same time.

Routing to parallel queues dates back at least to the work of Kingman [14] where the steady state behavior of *join the shortest queue* (JSQ) as a routing policy is studied. Winston [19] shows that JSQ maximizes the (discounted) total number of services completed for the case of exponential inter-arrivals and service times. The case for more general service and arrival processes is discussed in Weber [18]. When customers are of multiple classes (defined by both the arrival and service rates), Winston [20] shows customers with longer service times should be routed to a fastest free server. Routing and in fact, reassignment after arrival with zero fixed costs (also known as jockeying) is considered in Winston [21]. Winston's work was extended to the case of batch arrivals and finite buffers by Hordijk and Koole [12]. When the goal is to minimize the number of customers in the system Derman et al. [5] shows that assigning arriving customers to servers with the largest service rate is optimal. This is generalized to the *Shortest Queue, Faster Server* policy by Hordijk and Koole [13] which formalizes the intuitive concept that an arriving customer should prefer a faster server **and** shorter queue. For a more complete literature review on the routing problem, the interested reader is pointed to the work of Armony [2] where the problem is studied in the context of call centers. The asymptotic optimality of the *fastest server first* policy is shown.

Of course the scheduling literature is equally vast so we will not attempt a complete review. Early work on the analysis and optimization of multi-class *non-preemptive* queues was done in [8] and [9]. Scheduling of servers, with multiple queues quite often follows a priority rule called the c-μ rule. In essence, the decision-maker assigns servers to the station in which cost can be reduced from the system the fastest (see [4]). There are several twists on this original problem like the analysis of a system with one dedicated and one flexible server (the N-network), see [7]. Asymptotic analysis of the N-network was considered by Harrison [10] and Harrison and Lopez [11] and the proof of the structure of the asymptotic optimal policy is shown in Bell and Williams [3]. The (light traffic) N-network with upgrades is considered in Down and Lewis [6]. The closest related paper to the current is that of Andradóttir et al. [1], where the authors consider allocation of synergistic servers in a tandem queueing system, but with the goal of maximizing throughput.

Note that in each case the optimal deployment of a flexible workforce or the optimal routing

policy is considered in isolation. When optimizing routing and a flexible workforce concurrently, a number of questions arise.

- How should one coordinate the routing policy and the flexible workforce?
- How does the pooling efficiency change the optimal deployment of the workforce and routing policy?
- How much benefit can a system realize if we can jointly deploy a flexible workforce and dynamic routing? Under what conditions is deploying a partial control – either dynamic routing or dynamic resource allocation – sufficient to capture most of the benefit as compared to the optimal joint control?

Along with discerning the structure of the optimal control policy, we seek insights into these questions.

2 Preliminaries and Model Description

Consider a system with two parallel stations (queues). Customer arrivals to station k follow independent Poisson processes with rates $\lambda_k > 0$ for $k = 1, 2$. When a job arrives to station k, a decision-maker can either accept the job at that station or can (instantaneously) reroute the job to the other station at a fixed cost r. Along with the (re-)routing costs previously mentioned, the system is charged holding costs h_k per job per unit time at queue k,$k = 1, 2$.[1] Any customer that is accepted or rerouted joins the requisite queue at that station (if one exists) and will be served at that station; there is no jockeying after the routing decision. There are two fully-trained flexible servers that work at (possibly) different rates. Specifically, when server ℓ is working alone at either station, its processing rate is μ_ℓ, $\ell = 1, 2$. When both servers combine their efforts to work at the same station (i.e., pooled), the two servers collaborate on a single job with the rate μ_c. See Figure 1. If $\mu_c \geq \mu_1 + \mu_2$, collaboration is super-additive and represents the case that servers working together increases the overall efficiency of service. This includes the case of additive rates. When $\mu_c < \mu_1 + \mu_2$, collaboration is sub-additive and represents the case where servers working together decreases efficiency possibly due to a shared processing resource. The first case occurs in manufacturing when orders can be filled more quickly if two (and sometimes more) workers complete tasks together. Oddly enough, it also occurs in academia when researchers

[1] We allow $h_1 \neq h_2$ to model cases where the cost of buffer space at one queue is different from the cost at a different queue. For instance, a bed in a cardiology floor is equipped with specialized machines and monitors, thus incurs a higher opportunity cost than a bed in a general medicine floor.

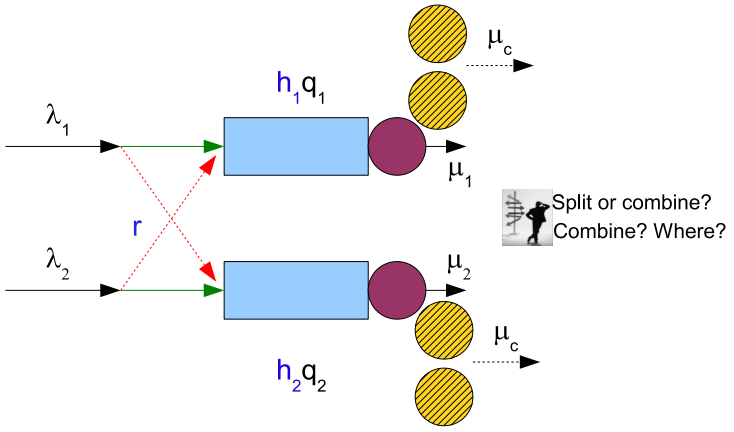


Figure 1: Dynamic Routing and Allocation

collaborate to solve an interesting problem. The second case occurs when servers need to share limited production resources (space, tools, fixtures) during collaboration. It also occurs quite frequently during the "training" of one of the servers. We assume that $\mu_c \geq \max\{\mu_1, \mu_2\}$ so that the service rate when servers collaborate is higher than the rate of an individual server. All job sizes are assumed to be exponential with rate 1.

The decision-maker chooses a policy that (i) routes arriving customers and (ii) allocates the servers to stations in order to minimize the expected costs. More formally, let $\mathbb{X} := \{(i,j)|i,j \in \mathbb{Z}^+\}$ denote the state space where i (j) represents the number of customers currently at station 1 (2) – including any job in service. Let Π be the set of non-anticipating Markovian policies. For a given $\pi \in \Pi$, $Q_k^\pi(s)$ denotes the number of customers at station k at time s under π (again including the one potentially in service). The routing action employed when there is an arrival to station k at time s is denoted by $a_k^\pi(s-)$, where $a_k^\pi(s-) = 2(1)$ implies that we are routing a job that is arriving to queue 1 (2) to queue 2 (1). For a fixed time $t \geq 0$, the finite horizon $\alpha-$discounted expected cost under π starting in state (i,j) is written

$$v_{\alpha,t}^\pi(i,j) = E_{(i,j)}^\pi\Big(\int_0^t e^{-\alpha s}[h_1 Q_1^\pi(s) + h_2 Q_2^\pi(s)]ds$$
$$+ r\Big[\sum_{n=0}^{N_1(t)} e^{-\alpha\sigma_{n,1}} 1_{\{a_1^\pi(\sigma_{n,1}-)=2\}} + \sum_{n=0}^{N_2(t)} e^{-\alpha\sigma_{n,2}} 1_{\{a_2^\pi(\sigma_{n,2}-)=1\}}\Big]\Big),$$

where $\{N_k(s), s \geq 0\}$ are (independent) Poisson processes of rates λ_k and $\sigma_{n,k}$ represents the n^{th} arrival time to station $k = 1, 2$. Likewise, the infinite horizon expected discounted cost and the long-run average cost rate under π are defined $v_\alpha^\pi(i,j) = \lim_{t\to\infty} v_{\alpha,t}^\pi(i,j)$ and $\rho^\pi(i,j) = \limsup_{t\to\infty} \frac{v_{0,t}^\pi(i,j)}{t}$, respectively. In each case the optimal values are given by $w(i,j) = \inf_{\pi\in\Pi} w^\pi(i,j)$ for $w = v_{\alpha,t}, v_\alpha$ or ρ.

In this paper, we consider both the continuous time problem described above and its equivalent uniformized, (cf. [15], [17]) discrete-time analogue. Suppose at each decision epoch a policy prescribes the sequence of (state dependent) decisions that specify allocation and routing, $\pi = \{d_0, d_1, \ldots\}$. Let $\Lambda := \lambda_1 + \lambda_2 + \max[\mu_c, \mu_1 + \mu_2]$ be the uniformization constant. Without loss of generality, assume $\Lambda = 1$ and let $\delta = \frac{\Lambda}{\Lambda+\alpha}$. For initial queue lengths $x = (i, j)$, define

$$v^{\pi}_{N,\delta}(x) := \mathbb{E}^{\pi}_{x} \sum_{n=0}^{N-1} [\delta^n C(X_n, d_n(X_n))], \tag{2.1}$$

$$v^{\pi}_{\delta}(x) := \lim_{N\to\infty} v^{\pi}_{N,\delta}(x), \tag{2.2}$$

$$p^{\pi}(x) := \limsup_{N\to\infty} \frac{1}{N} v^{\pi}_{N,1}(x), \tag{2.3}$$

where $C((i, j), a)$ represents the (expected) cost of using action a in state (i, j). Again, the optimal values in this discrete-time problem are given by $w(i, j) = \inf_{\pi\in\Pi} w^{\pi}(i, j)$ for $w = v_{\delta,n}, v_{\delta}$ or p. It is well-known that the optimal values in the infinite horizon discrete-time problems are the same as those in the continuous-time formulations up to a multiplicative constant ([15]). Moreover, the optimal control policies coincide. The discrete-time finite horizon formulation is used as a means to obtain the structure of optimal policies in each of the infinite horizon cases. Suppose f is a real-valued function on the state space and define the following mapping H_{δ}

$$\begin{aligned} H_{\delta} f(i, j) = {} & \delta\big(\lambda_1 \min[f(i + 1, j), f(i, j + 1) + r] + \lambda_2 \min[f(i + 1, j) + r, f(i, j + 1)]\big) \\ & + \delta \min \begin{bmatrix} \mu_1 f((i-1)^+, j) + \mu_2 f(i, (j-1)^+) + (\Lambda - \lambda_1 - \lambda_2 - \mu_1 - \mu_2) f(i, j), \\ \mu_c f((i-1)^+, j) + (\Lambda - \lambda_1 - \lambda_2 - \mu_c) f(i, j), \\ \mu_c f(i, (j-1)^+) + (\Lambda - \lambda_1 - \lambda_2 - \mu_c) f(i, j). \end{bmatrix} \end{aligned} \tag{2.4}$$

Note that each term in the minimums for (2.4) correspond to the choice of routing an arriving customer or allocating the servers when the value of being in each state is measured by f. Let $v_{0,\delta} = 0$. The (discrete-time) finite and infinite horizon discounted expected cost optimality equations (FHOEs and DCOEs, respectively) and the average cost optimality equation (ACOE) can be written

$$v_{n+1,\delta}(i, j) = ih_1 + jh_2 + H_{\delta} v_{n,\delta}(i, j), \tag{2.5}$$

$$v_{\delta}(i, j) = ih_1 + jh_2 + H_{\delta} v_{\delta}(i, j), \tag{2.6}$$

$$g + y(i, j) = ih_1 + jh_2 + H_1 y(i, j). \tag{2.7}$$

That is to say that the optimal values for the discrete-time finite and infinite horizon discounted cost problems satisfy (2.5) and (2.6), respectively. If there exists (g, y) that satisfies (2.7), then g

is the optimal average cost and y is called a relative value function.

Remark: In the optimality equations above, we assume that each server has a *primary assignment* when it works alone. In practice, it is common that cross-trained workers are pre-assigned to a particular station they will work when they are not helping other stations: For instance, a nurse in the cardiology ward will treat cardiology patients unless she/he is helping patients in another unit. This assumption is not restrictive and does not influence the key trade-off in allocation decisions between pooling and splitting. In fact, the results of this section and those of Section 3 hold with or without this assumption. Unfortunately, general results are elusive in the sub-additive case. For consistency we maintain this assumption throughout the remainder of the paper.

To conclude this section, we show that there is no loss of optimality if we restrict our attention to non-idling policies. Suppose $D_1 f(i,j) := f(i+1,j) - f(i,j)$ and $D_2 f(i,j) := f(i,j+1) - f(i,j)$. The proof of the following simple, but useful result is omitted for brevity.

Lemma 2.1. *For all $i, j, n \geq 0$ we have*

1. *$D_1 v_{n,\delta}(i,j) \geq 0$ and $D_2 v_{n,\delta}(i,j) \geq 0$,*
2. *$D_1 v_{\delta}(i,j) \geq 0$ and $D_2 v_{\delta}(i,j) \geq 0$.*

That is to say that $v_{n,\delta}(i,j)$ and $v_{\delta}(i,j)$ are non-decreasing in the number of customers in either station.

Proposition 2.2. *Under either the finite or infinite horizon discounted cost criterion, there exists an optimal non-idling policy. Consequently, if station 1 (station 2) is empty, then it is optimal for servers to collaborate at station 2 (station 1).*

Proof. See Appendix. ∎

The following theorem provides sufficient conditions to extend the results in this paper to the average cost case. We remark that the condition $\lambda_1 + \lambda_2 < \max\{\mu_c, \mu_1 + \mu_2\}$ guarantees the existence of a stationary policy that has finite average cost; which implies the existence of a stationary distribution under said policy. All proofs can be found in the Appendix.

Theorem 2.3. *Suppose $\lambda_1 + \lambda_2 < \max\{\mu_c, \mu_1 + \mu_2\}$. There is a finite constant $g := \lim_{\delta \uparrow 1}(1-\delta) v_\delta(i,j)$ such that the following hold*

1. *$y_\delta(i,j) := v_\delta(i,j) - v_\delta(0,0)$ converges along a subsequence to a function y on $\mathbb{X}$ such that (g,y) satisfy the average cost optimality equations.*

2. *Any limit point of a sequence of discounted cost optimal policies (as $\delta \uparrow 1$) is average cost optimal.*

The following proposition is an immediate consequence of Theorem 2.3.

Proposition 2.4. *Suppose $\lambda_1+\lambda_2 < \max\{\mu_c, \mu_1+\mu_2\}$. The results of Lemma 2.1 and Proposition 2.2 hold under the average cost criterion with v_δ replaced with y (a relative value function in the ACOE).*

3 Optimal Control Under Super-additive Collaboration

Assume for now that $\mu_c \geq \mu_1 + \mu_2$; the servers working collectively are faster than working separately. Examples of this scenario include lifting or moving heavy objects, brainstorming for ideas, and completing a project. It also includes the case where service can be divided into two separate operations and done in parallel; attaching fixtures to cars, taking measurements of different parts or accessing more than a single file or database. The main structural results of this section are captured in the following two theorems. The first characterizes an optimal allocation while the second discusses optimal routing.

Theorem 3.1. *(Allocation) Suppose $\mu_c \geq \mu_1 + \mu_2$ and that $h_1 \geq h_2$. In the discounted cost case or the average cost case (under the hypothesis of Proposition 2.4), there exists an optimal policy such that both servers collaborate in station 1 whenever $i \geq 1$. Otherwise, servers collaborate in station 2.*

Theorem 3.2. *(Routing) Suppose $\mu_c \geq \mu_1 + \mu_2$ and that $h_1 \geq h_2$. In the discounted cost case or the average cost case (under the hypothesis of Proposition 2.4), there exists an optimal policy such that*

1. *Jobs arriving to station 2 will never be routed to station 1.*

2. *If it is optimal to route a job arriving at station 1 to station 2 in state (i, j), it is also optimal to route a job arriving at station 1 to station 2 in states $(i + 1, j)$ and $(i, j - 1)$.*

There are of course symmetric results for $h_2 \geq h_1$. In the special case where holding costs are the same in both stations, the ability to route is unnecessary. This is summarized in the following corollary that follows directly from the first result of Theorem 3.2.

Corollary 3.3. *Suppose $\mu_c \geq \mu_1 + \mu_2$ and that $h_1 = h_2$. In the discounted cost case or the average cost case (under the hypothesis of Proposition 2.4), there exist an optimal policy that does not route any arriving customers to a different station.*

3.1 Server allocation in the super-additive case

The proof of Theorem 3.1 is broken into several parts. The first is the following proposition.

Proposition 3.4. *In the discounted cost case or the average cost case (under the hypothesis of Proposition 2.4), it is optimal for servers to collaborate.*

An immediate consequence of Proposition 3.4 is that the system works as if there is one server working at rate μ_c and the question becomes where to place this *superserver*. This enables us to simplify the optimality equations (2.5)–(2.7) as follows:

$$v_{n+1,\delta}(i,j) = ih_1 + jh_2 + H_\delta^s v_{n,\delta}(i,j) \tag{3.1}$$

$$v_\delta(i,j) = ih_1 + jh_2 + H_\delta^s v_\delta(i,j), \tag{3.2}$$

$$g + y(i,j) = ih_1 + jh_2 + H_1^s y(i,j) \tag{3.3}$$

where for any real valued function f on $\mathbb{X}$

$$H_\delta^s f(i,j) := \delta\Big(\lambda_1 \min[f(i+1,j), f(i,j+1)+r] + \lambda_2 \min[f(i+1,j)+r, f(i,j+1)]$$
$$+ \mu_c \min\left[f((i-1)^+,j), f(i,(j-1)^+)\right]\Big). \tag{3.4}$$

Suppose for function f on $\mathbb{X}$ that $\Delta f(i,j) := f(i+1,j) - f(i,j+1)$. It should be clear from the optimality equations that it is optimal to allocate the superserver to station 1 (2) in state (i,j) for $i,j \geq 1$ at time $n+1$ when $\Delta v_{n,\delta}(i-1,j-1) \geq \ (\leq)\ 0$. Theorem 3.1 follows immediately from the following result.

Lemma 3.5. *Suppose $\mu_c \geq \mu_1 + \mu_2$ and that $h_1 \geq h_2$. The following hold for $i,j,n \geq 0$*

1. *$\Delta v_{n,\delta}(i,j) \geq 0$.*
2. *The previous results hold replacing $v_{n,\delta}(i,j)$ with $v_\delta(i,j)$ or $y(i,j)$ (under the hypothesis of Proposition 2.4).*

3.2 Routing in the super-additive case

Notice that it is optimal to reroute a customer arriving to station 1 (from queue 1 to queue 2) if $\Delta v_{n,\delta}(i,j) \geq r$ and reroute a customer arriving to station 2 if $\Delta v_{n,\delta}(i,j) \leq -r$. Thus, if routing a customer arriving to station 1 is optimal in state (i,j), a sufficient condition under which it is also optimal to reroute a customer arriving to station 1 in state $(i+1,j)$ is $D_1 \Delta v_{n,\delta}(i,j) = \Delta v_{n,\delta}(i+1,j) - \Delta v_{n,\delta}(i,j) \geq 0$. Similarly, $D_2 \Delta v_{n,\delta}(i,j-1) = \Delta v_{n,\delta}(i,j) - \Delta v_{n,\delta}(i,j-1) \leq 0$

is a sufficient condition for routing to be optimal in state $(i, j-1)$. The next lemma shows that this is the case for the super-additive case. Theorem 3.2 follows from Lemmas 3.5 and 3.6 (see Appendix).

Lemma 3.6. *Suppose $\mu_c \geq \mu_1 + \mu_2$ and that $h_1 \geq h_2$. The following hold for $i, j, n \geq 0$*

1. *$D_1 \Delta v_{n,\delta}(i,j) \geq 0$ and $D_2 \Delta v_{n,\delta}(i,j) \leq 0$ (diagonal dominance).*
2. *The previous results hold replacing $v_{n,\delta}(i,j)$ with $v_\delta(i,j)$ or $y(i,j)$ (under the hypothesis of Proposition 2.4).*

Notice that in the super-additive case, the optimal allocation is a priority policy that prefers serving at the queue with the higher holding cost. Similar logic holds for the routing policy. First, it is never optimal to route a job arriving at a lower-cost queue to a higher-cost queue. The optimal routing policy describing when to route from the high cost queue (queue 1) to the low cost queue (queue 2) can be characterized by a monotone switching curve in the following sense: the more customers there are in station 1, the more a decision-maker prefers moving customers from station 1 to 2. Similarly, the fewer customers in station 2, the more a decision maker prefers moving customers from 1 to 2. This completely characterizes the joint optimal control in the super-additive case.

We would also like to point out that the previous results hold for a more general holding cost function than linear. Suppose in state (i, j) that the holding cost rate is $h(i, j)$ and that the DCOE and ACOE have solutions when this cost rate function replaces the current one. Assuming $h(i, j)$ is non-decreasing, such that $\Delta h(i, j) \geq 0$, $D_1 \Delta h(i, j) \geq 0$ and $D_2 \Delta h(i, j) \geq 0$ the results of Lemmas 3.5 and 3.6 hold, and therefore so do the results of Theorems 3.1 and 3.2.

3.3 Numerical Study

In previous sections, we characterized the structure of optimal policies for the super-additive case. The optimal policy calls for pooling both servers at the station with a higher holding cost and routing more arriving jobs to the queue with lower holding cost as the number of jobs at a more expensive station increases. The optimal policy dynamically adjusts routing and allocation decisions to balance arriving customers (or at least the cost of holding them) throughout the system. Recall from Section 1, we would like to understand (i) under what conditions implementing one dimensional dynamic control (routing or allocation) performs well as compared to the joint optimal control policy and (ii) when the system needs to use both control levers to reduce the cost.

We compare the average cost under the optimal policy to the average costs under four different policies. In the *join the shortest queue policy* (JSQ), an arriving job always joins the shortest

queue. No collaboration is allowed unless one of the queues is empty. In the *pooling at the more expensive queue* policy (Pooling), servers are always pooled at a more expensive non-empty station, but no job is rerouted. When the holding costs are equal, the policy splits the server because jobs at both queues are equally important. In the *routing-only policy*, we allow the system to dynamically route incoming jobs based on the current state, but servers are always split (except to avoid idling). In the *allocation-only policy*, the system dynamically decides how to allocate two servers, but no incoming jobs are rerouted.

In order to complete the numerical analysis of each policy we vary several parameters and use *value iteration* to approximate the optimal values and policies. The set of instances we used is intended to cover many different topologies of the network (both symmetric and asymmetric instances). The parameter values are:

$$(\lambda_1, \lambda_2, h_1, h_2) \in \{(1,1,1,1), (1,1,3,1), (1.5,0.5,1,1), (1.5,0.5,3,1)\};$$

$$(\mu_1, \mu_2) \in \{(1.6,1.6), (2.1,1.1), (1.1,2.1)\}; r \in \{0.5, 5, 100\};$$

$$\mu_c \in \{3.2, 3.6, 4.0\} \text{ (represents pooling efficiency in the super-additive case)}$$

In total, we compare the performance of the five policies – Optimal, JSQ, Pooling, Routing-only, and Allocation-only – for 108 problem instances. The percentage sub-optimality of the four policies is presented in Table 1. We categorize 108 instances using two anchors, r and μ_c, both of which are critical in determining efficiencies of routing and pooling. For instance, routing is inexpensive to implement when r is small, but it is expensive when r is large. For given μ_1 and μ_2, increasing μ_c implies that pooling two servers becomes more efficient.

There are many instances where both JSQ and pooling policies perform poorly, however the pooling policy outperforms JSQ policy in almost all instances. This is particularly true when the holding costs are asymmetric ($h_1 > h_2$ or $h_2 > h_1$). We observe that, as the routing cost increases, JSQ performs worse and the difference in the sub-optimality gap between JSQ and the pooling policy increases. It is interesting to note that the performance of JSQ and the pooling policy degrades as μ_c increases. Putting this into context for JSQ, an increase in μ_c increases the benefit of utilizing a flexible workforce. Note that the optimal policy has the added benefit of dynamically utilizing the flexible workforce in addition to routing to balance the workload. JSQ only tries to do so via crude routing. Thus, when pooling becomes highly efficient, JSQ policy cannot take advantage of a faster service rate enabled by pooled servers. We also note that the performance of the pooling policy degrades as μ_c increases. Further examination shows the pooling policy performs poorly in two situations: (1) when the routing cost is low and/or (2) holding costs are the same for both queues and the pooled service rate is much higher than the sum of individual rates. When r is low, much of workload balancing can be achieved through

	μ_c	JSQ	Pool	Routing only	Allocation only
$r = 0.5$	$\mu_c = 3.2$	36.9%	15.2%	0.00%	15.2%
	$\mu_c = 3.6$	63.6%	20.0%	6.4%	12.3%
	$\mu_c = 4$	84.92%	23.02%	9.3%	9.2%
	average ($r = 0.5$)	61.8%	19.4%	5.2 %	12.4%
$r = 5$	$\mu_c = 3.2$	119.1%	0.3%	12.4%	0.3%
	$\mu_c = 3.6$	170.4%	7.8%	26.7%	0.1%
	$\mu_c = 4$	216.1%	13.8%	37.5%	0.0%
	average ($r = 5$)	80.1%	10.7%	25.6%	0.1%
$r = 100$	$\mu_c = 3.2$	2223.9%	0.0%	13.8%	0.0%
	$\mu_c = 3.6$	2793.0%	7.7%	27.8%	0.0%
	$\mu_c = 4$	3301.6%	13.8%	38.8%	0.0%
	average ($r = 100$)	2772.8%	7.2%	26.8%	0.0%
Total	Average	1001.1%	11.3%	19.2%	4.1%

Table 1: Super-additive case: JSQ, Pool at the expensive queue, Routing only, and Allocation only.

routing, thus pooling at an expensive queue is not beneficial. When the holding costs are the same, no pooling actually occurs under the pooling policy unless one queue is empty. However, our results have already shown that pooling is optimal in the super-additive case. Thus, as μ_c increases, the sub-optimality gap increases.

Similar results can be found when comparing the routing-only and allocation-only polices. Dynamic server allocation alone performs well (in many cases near optimal) at medium and high routing costs. However, when holding costs are low, the allocation-only policy can perform poorly; the average sub-optimality gap when $r = 0.5$ is 12.4%. The performance of the allocation-only policy improves as μ_c increases, making pooling more attractive. On the other hand, the performance of the routing-only policy degrades in μ_c since the policy does not exploit efficient pooling (except to avoid idling).

Table 2 further examines these cases. We observe that when the holding costs are symmetric the routing-only policy performs poorly because routing only balances the load (at a cost) and does not take advantage of the super-additivity. In this case, the allocation-only policy performs close to optimal. However, when the holding costs are asymmetric, the allocation-only performs poorly, but the dynamic routing policy performs close to optimal. This is due to the fact that most jobs are routed to the queue with lower holding cost leaving the queue with higher cost almost empty. Consequently, most of the time both servers are pooled at the lower cost queue and the routing only policy can recoup most of the benefit seen in the optimal policy.

In summary, when the routing cost is medium or high, controlling the allocation of servers

can reap most of the gain obtained from the optimal policy. Even when the routing cost is low, dynamic allocation performs well when the queue is symmetric. When holding costs are not symmetric, however, dynamic routing can be much more effective than dynamic allocation because it routes jobs to a lower cost queue (and leaves the other queue almost empty). This makes both servers work in the low cost queue to avoid idling. Our numerical results also show that when the system can implement only a one dimensional control (either routing or allocation), making the wrong choice can have dire consequences.

4 Optimal Control Under Sub-additive Collaboration

In this section we consider the sub-additive collaboration case. Examples of this case include when servers have to share a limited resource or when there are disruptive influences that slow down the service rate (distractions for people). One important point we would like to make is that results that completely characterize the optimal control like those in the super-additive case do not hold in the sub-additive case. This is due to the fact that collaboration can reduce congestion at a particular station (since $\mu_c \geq \max\{\mu_1, \mu_2\}$) but it slows the rate at which the entire system processes jobs (since $\mu_1 + \mu_2 \geq \mu_c$). Hence, depending on which of the two forces is dominant, the optimal policy can be complicated and unintuitive. Consider the following example.

Example 4.1. *Suppose* $\lambda_1 = 4.0$*;* $\lambda_2 = 5.5$ *(arrival rates);* $\mu_1 = 8.0$*;* $\mu_2 = 7.0$*,* $\mu_c = 14.025$*; (the combined service rate is 93.5% of the sum);* $r = 3$ *(customer routing cost);* $h_1 = 10$*;* $h_2 = 8$ *(holding costs).*

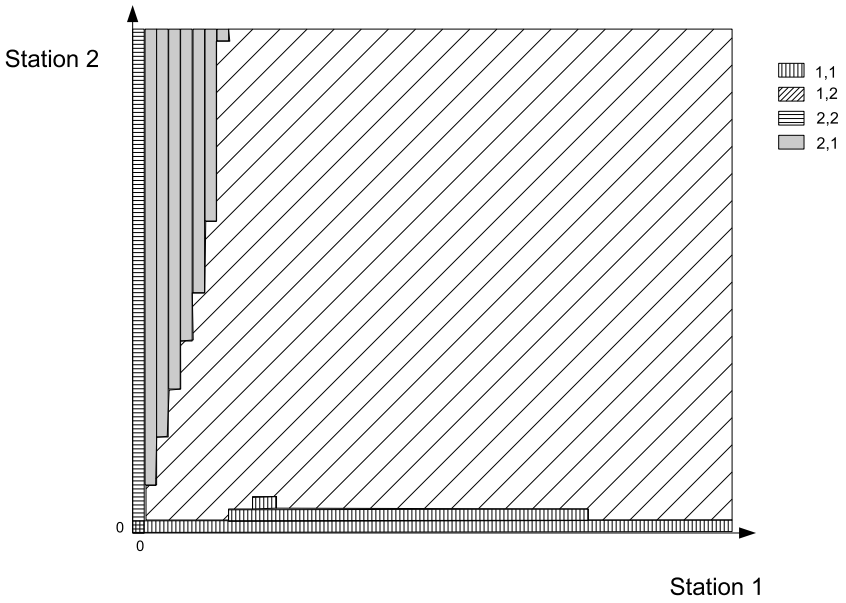
Figure 2 shows the optimal allocation and routing actions for Example 4.1. In Figure 2(a), (m, n) corresponds to where the servers should be allocated. For example $(1, 2)$ means that server 1 is allocated to station 1 while server 2 is allocated to station 2. Similarly, $(2, 1)$ corresponds to the opposite allocation. In Figure 2(b), $(1, 2)$ corresponds to no rerouting while $(1, 1)$ implies that any arriving customer (either to station 1 or 2) should be routed to station 1.

Note first that when station 2 is almost empty (say, $j = 2$) and station 1 has a light load, the servers should be split. When the load on station 1 increases to a medium load both servers are allocated to station 1. As the load increases further we again split the allocation. When considered in conjunction with the routing policy, we observe that as the number of customers at station 1 increases, the optimal policy eventually routes all customers to station 2. That is to say that routing is used to alleviate the congestion (instead of allocation).

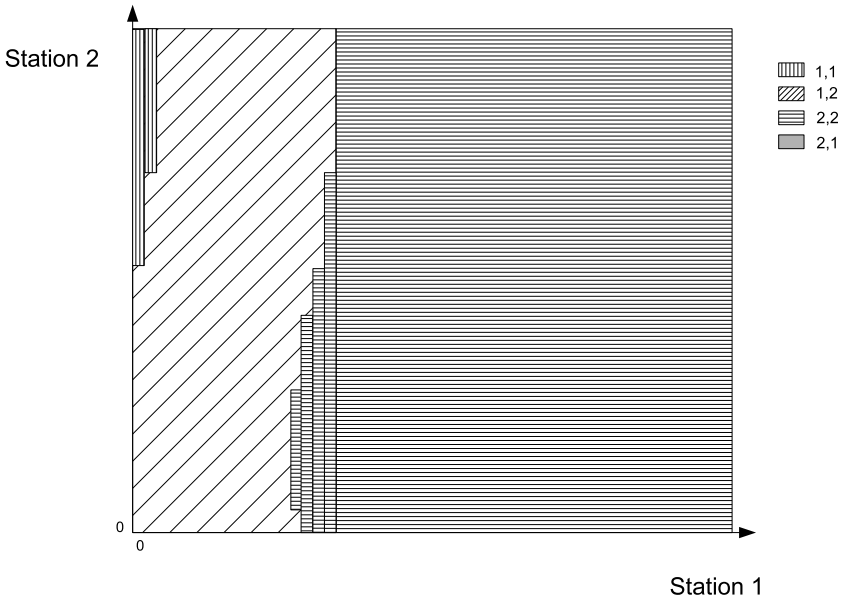
As the above example shows the fact that the optimal policy is not necessarily monotone in queue lengths makes characterizing its the structure for a general system daunting. Balancing the load by intermingling the two control mechanisms is a complex problem. On the other hand,

λ_1	λ_2	μ_1	μ_2	μ_c	h_1	h_2	Routing only	Allocation only
1	1	1.6	1.6	3.6	1	1	15.35 %	0.00 %
1	1	2.1	1.1	3.6	1	1	15.23 %	0.00 %
1	1	1.1	2.1	3.6	1	1	15.23 %	0.00 %
1	1	1.6	1.6	4	1	1	22.11 %	0.00 %
1	1	2.1	1.1	4	1	1	21.94 %	0.00 %
1	1	1.1	2.1	4	1	1	21.94 %	0.00 %
1	1	1.6	1.6	3.6	3	1	0.00 %	15.38 %
1	1	2.1	1.1	3.6	3	1	0.00 %	15.38 %
1	1	1.1	2.1	3.6	3	1	0.00 %	15.38 %
1	1	1.6	1.6	4	3	1	0.00 %	11.11%
1	1	2.1	1.1	4	3	1	0.00 %	11.11%
1	1	1.1	2.1	4	3	1	0.00 %	11.11%
0.5	1.5	1.6	1.6	3.2	1	3	0.00 %	41.99 %
0.5	1.5	2.1	1.1	3.2	1	3	0.00 %	41.99 %
0.5	1.5	1.1	2.1	3.2	1	3	0.00 %	41.99 %
0.5	1.5	1.6	1.6	3.6	1	3	0.00 %	33.93 %
0.5	1.5	2.1	1.1	3.6	1	3	0.00 %	33.93 %
0.5	1.5	1.1	2.1	3.6	1	3	0.00 %	33.93 %
0.5	1.5	1.6	1.6	4	1	3	0.00 %	25.71 %
0.5	1.5	2.1	1.1	4	1	3	0.00 %	25.71 %
0.5	1.5	1.1	2.1	4	1	3	0.00 %	25.71 %
0.5	1.5	1.6	1.6	3.6	1	1	10.31 %	0.00 %
0.5	1.5	2.1	1.1	3.6	1	1	9.35 %	0.00 %
0.5	1.5	1.1	2.1	3.6	1	1	11.2 %	0.00 %
0.5	1.5	1.6	1.6	4	1	1	15.22 %	0.00 %
0.5	1.5	2.1	1.1	4	1	1	14.11 %	0.00 %
0.5	1.5	1.1	2.1	4	1	1	15.92 %	0.00 %

Table 2: Performance of routing-only and allocation-only policies when $r = 0.5$ and $\mu_c = 3.6$ or 4.0



(a) Optimal Allocation



(b) Optimal Routing

Figure 2: Optimal Routing and Allocation for Example 4.1

our intuition says that at one extreme, as r approaches ∞, the decision-maker should rely only on dynamic allocation to balance the workload. When the load is *balanced* (perhaps weighted by the holding costs) the decision-maker should split the servers as pooling is less efficient than splitting two servers. If the loads are significantly unbalanced, the decision-maker would allocate both servers to the longer queue until the workload balance is restored. At the other extreme,

consider a situation where routing is (almost) free. The decision-maker need not consider the cost of balancing the load when making routing decisions. Dynamic routing reduces the frequency at which these imbalances occur. However, a complete characterization of the optimal policy is difficult as the optimal policy depends on the problem parameters as well as the current state. Unlike the super-additive case where pooling is always more efficient, pooling is not necessarily more efficient in all states. In short, it can reduce the number of jobs at a more congested (or expensive) queue faster, at the cost of reducing the rate at which jobs can leave the system when the two servers are split; $\mu_c < \mu_1 + \mu_2$.

We begin our analysis of sub-additive systems by considering the special cases when $r = \infty$ or 0. The results of Sections 4.1 and 4.2 study allocation in these scenarios. Theorem 4.2 contains a condition under which it is optimal to split the servers in the case where no routing is allowed. In Theorem 4.4 we present a sufficient condition where splitting is optimal under the assumption that the routing policy is monotone in the queue lengths. We then consider the case with positive routing costs. We are able to characterize the optimal policy when the system parameters are symmetric; the arrival, service rates (for split servers) and holding costs are the same for both stations. This simplifies the problem enough to make two important observations which follow one basic tenet; allocation and routing can be decoupled in the symmetric system. No matter what the routing policy is we should always split the servers except to avoid idling. Furthermore, no matter what the allocation policy, we should only route from longer queues to shorter queues.

4.1 Allocation without routing

In this section we assume that the decision-maker is unable to route. This assumption allows us to simplify the optimality equations slightly:

$$\begin{aligned} H_\delta f(i,j) = {} & \delta(\lambda_1 f(i+1,j) + \lambda_2 f(i,j+1) \\ & + \delta \min \begin{bmatrix} \mu_1 f((i-1)^+,j) + \mu_2 f(i,(j-1)^+) + (\Lambda - \lambda_1 - \lambda_2 - \mu_1 - \mu_2) f(i,j), \\ \mu_c f((i-1)^+,j) + (\Lambda - \lambda_1 - \lambda_2 - \mu_c) f(i,j), \\ \mu_c f(i,(j-1)^+) + (\Lambda - \lambda_1 - \lambda_2 - \mu_c) f(i,j). \end{bmatrix} \end{aligned} \tag{4.1}$$

The following theorem relates this problem to classical scheduling theory.

Theorem 4.2. *Suppose $\mu_1 h_1 + \mu_2 h_2 \geq \mu_c \max\{h_1, h_2\}$. The following inequalities hold for all $i, j \geq 1$:*

***N**(1)* $(\mu_c - \mu_2) D_2 v_{n,\delta}(i, j-1) - \mu_1 D_1 v_{n,\delta}(i-1, j) \leq 0.$

***N**(2)* $(\mu_c - \mu_1) D_1 v_{n,\delta}(i-1, j) - \mu_2 D_2 v_{n,\delta}(i, j-1) \leq 0.$

1. *For the infinite horizon problem, Properties **N**(1)- **N**(2) hold with $v_\delta(i,j)$ or $y(i,j)$ (under the assumptions of Proposition 2.4) replacing $v_{n,\delta}(i,j)$.*

2. There exists an optimal policy that splits the servers except to avoid idling.

Note that the condition in Theorem 4.2 is similar to the c-μ rule for parallel processing networks. The condition ($\mu_1 h_1 + \mu_2 h_2 \geq \mu_c \max\{h_1, h_2\}$) implies that the rate at which cost is reduced is faster when the servers are split. On the other hand, we note that the proof does not work when the condition is reversed. In essence, we explicitly used the assumption that $\mu_1 + \mu_2 \geq \mu_c$ in the proof. Not only can we reduce costs at a faster rate by splitting servers, but we can also alleviate congestion faster. This leaves the question of allocation when $\mu_c \max\{h_1, h_2\} \geq \mu_1 h_1 + \mu_2 h_2$ and sub-additive service rates open. In this case, whether it is better to reduce cost at a high cost station or reduce congestion at both depends on the current state as well as problem parameters.

4.2 Monotone routing policies with $r = 0$

In this section we consider a system where jobs can be routed free of charge. The main result is that if we restrict attention to monotone routing policies the results of the previous section (i.e., always split the servers) continue to hold.

Definition 4.3. *We say that a policy is a* ***monotone routing policy*** *if the following hold*

1. *If an arriving customer is routed to station 1 in state $(i, j) \in \mathbb{X}$, it is routed to station 1 in state $(i, j + 1)$.*

2. *If an arriving customer is routed to station 2 in state $(i, j) \in \mathbb{X}$, it is routed to station 2 in state $(i + 1, j)$.*

Let $\tilde{v}_\delta$ represent the optimal infinite horizon $\delta-$discounted cost for the routing and allocation problem with the restriction that we only consider monotone routing policies. Similarly, define $\tilde{v}_{n,\delta}$ to be the value iterates of this problem and $\tilde{y}(i, j)$ as the relative value function. Note that now the actions available in state (i, j) depend on the action chosen in states $(i - 1, j)$ and $(i, j - 1)$. That is, if the decision-maker chooses to route an arriving customer to station 1 in state $(i, j - 1)$ (s)he must necessarily route arriving customers to station 1 in state (i, j). Similarly a decision-maker that routes to station 2 in state $(i - 1, j)$, must route to station 2 in state (i, j). The following is the main result of this section.

Theorem 4.4. *Suppose $\mu_1 h_1 + \mu_2 h_2 \geq \mu_c \max\{h_1, h_2\}$. The results of Theorem 4.2 hold with $v_{n\delta}$ replaced with $\tilde{v}_{n,\delta}$, v_δ replaced with $\tilde{v}_\delta$ and y replaced with $\tilde{y}$.*

4.3 The case with strictly positive routing cost, $r > 0$

In this case we partially characterize the structure of the optimal policy for the important case of a symmetric system; $h_1 = h_2 = h$, $\lambda_1 = \lambda_2$ and $\mu_1 = \mu_2$. The optimality equations (2.5)–(2.7) can be rewritten:

$$v_n(i,j) = (i+j)h + H^m v_{n-1}(i,j) \tag{4.2}$$

$$v_\delta(i,j) = (i+j)h + H^m v_\delta(i,j) \tag{4.3}$$

$$g + y(i,j) = (i+j)h + H^m y(i,j) \tag{4.4}$$

where, for any real valued function f on $\mathbb{X}$,

$$H^m f(i,j) = \lambda \min[f(i+1,j), f(i,j+1) + r] + \lambda \min[f(i+1,j) + r, f(i,j+1)]$$

$$+ \min \begin{bmatrix} \mu f((i-1)^+, j) + \mu f(i,(j-1)^+), \\ \mu_c f((i-1)^+, j) + (2\mu - \mu_c) f(i,j), \\ \mu_c f(i,(j-1)^+) + (2\mu - \mu_c) f(i,j) \end{bmatrix} \tag{4.5}$$

The first result states that the allocation and routing decisions can be *decoupled*.

Theorem 4.5. *(Allocation) Suppose $h_1 = h_2 = h$, $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$. For any fixed t (finite or infinite), there exists an optimal policy such that both servers collaborate only when one station is empty. Similarly for the average cost case under the assumptions of Proposition 2.4.*

We note that given Figure 2 the results in Theorem 4.5 are not completely unexpected. When $\mu_1 = \mu_2$ and $h_1 = h_2$ neither station is prioritized and a hypothesis analogous to that of Theorem 4.2 holds.

Theorem 4.6. *(Routing) Suppose $h_1 = h_2 = h$, $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$. In the discounted cost case or the average cost case (under the hypothesis of Proposition 2.4), there exists an optimal policy such that jobs arriving to a shorter station will never be routed to a longer or equal-length station.*

Theorem 4.5 captures the allocation portion of the control. It implies that the optimality equations (4.2)–(4.4) can be simplified by replacing H^m with H^{m*} defined as

$$H^{m*} f(i,j) := \lambda \min[f(i+1,j), f(i,j+1) + r] + \lambda \min[f(i+1,j) + r, f(i,j+1)]$$

$$+ \begin{cases} \mu f(i-1,j) + \mu f(i,j-1) & \text{for } i \geq 1, j \geq 1 \\ \mu_c f((i-1)^+, j) + (2\mu - \mu_c) f(i,j) & \text{for } i \geq 0, j = 0, \\ \mu_c f(i,(j-1)^+) + (2\mu - \mu_c) f(i,j) & \text{for } i = 0, j \geq 0. \end{cases} \tag{4.6}$$

To prove Theorem 4.6 we divide the state space along the ray where $i = j$ and consider which actions should not be used. With the symmetry assumption, note that $v_{n,\delta}(i,j)$ is symmetric; that is $v_{n,\delta}(i,j) = v_{n,\delta}(j,i)$. Similarly for $v_\delta(i,j)$ and $y(i,j)$. The proof of Theorem 4.6 now follows directly by letting $n \to \infty$ to obtain the discounted cost case and along a subsequence of discount factors to get the average cost case.

Proposition 4.7. *Suppose $h_1 = h_2 = h$, $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$. The following inequalities hold:*

1. *For $j \geq i$, $\Delta v_{n,\delta}(i,j) \leq r$.*
2. *For $i \geq j$, $\Delta v_{n,\delta}(i,j) \geq -r$*

We note that in each case – no routing, zero cost with monotone routing, and symmetry – the proofs explicitly use the assumptions. This coupled with Example 4.1 lead us away from general results like those in the super-additive case. On the other hand, the numerical study of the next section leads us to the conclusion that it is quite often the case that only one lever is necessary. The problem is discerning which lever to use is not always simple and getting it wrong can have dire consequences.

4.4 Numerical Study

In previous sections, we show that characterizing the complete structure of the optimal policy for the sub-additive case is difficult. Although we were able to characterize the optimal policy (e.g., condition for always splitting) for several special cases, the optimal policy in general dynamically adjusts routing and allocation decisions to balance arriving customers (or at least the cost of holding them) throughout the system. In contrast to the super-additive case, the optimal policy can switch between pooling both servers and splitting servers depending on queue lengths and problem parameters. In addition, except for a few special cases, the routing and allocation decisions are not separable; thus implementing joint routing and allocation decisions can be quite difficult. As in the super-additive case we seek (i) conditions under which implementing one dimensional control (routing or allocation) dynamically performs almost as well as the optimal policy and (ii) when the system should use both control levers to reduce the cost.

As before, we compare the average cost under the optimal policy to the costs under four different policies: namely, the JSQ, the pooling at the expensive queue, the routing-only, and the allocation-only policies. In order to give a complete picture of performance of these policies, we

	μ_c	JSQ	Pool	Routing only	Allocation only
$r = 0.5$	$\mu_c = 3.1$	31.6%	17.5%	0.0%	15.1%
	$\mu_c = 2.8$	20.9%	229.9%	0.0%	62.4%
	$\mu_c = 2.5$	17.0%	491.7%	0.0%	220.7%
	average ($r = 0.5$)	23.2%	246.4%	0.0%	99.4%
$r = 30$	$\mu_c = 3.1$	297.6%	2.0%	22.2%	0.0%
	$\mu_c = 2.8$	187.2%	111.3%	4.8%	4.5%
	$\mu_c = 2.5$	132.8%	213.8%	0.6%	53.0%
	average ($r = 100$)	205.9%	109.0%	9.2%	19.2%
$r = 150$	$\mu_c = 3.1$	1452.8%	2.0%	27.0%	0.0%
	$\mu_c = 2.8$	1012.2%	102.0%	14.6%	0.0%
	$\mu_c = 2.5$	631.1%	135.9%	3.8%	4.4%
	average ($r = 5$)	1041.9%	69.0%	15.4%	2.2%
Total	Average	423.7%	141.5%	8.2%	40.3%

Table 3: Sub-additive case: performance of JSQ, Pool, Heuristic policies.

λ_1	λ_2	μ_1	μ_2	μ_c	h_1	h_2	r	Routing only	Allocation only
0.8	2	1.6	1.6	2.8	1	3	30	10.7%	13.6%
0.8	2	2.1	1.1	2.8	1	3	30	20.5%	15.0%
0.8	2	1.6	1.6	2.5	1	3	150	6.8%	18.7%
0.8	2	2.1	1.1	2.5	1	3	150	11.5%	7.6%

Table 4: Instances where neither routing-only nor allocation-only performs well.

vary several parameters as below:

$$(\lambda_1, \lambda_2, h_1, h_2) \in \{(1.4, 1.4, 1, 1), (1.4, 1.4, 3, 1), (0.8, 2, 1, 1), (0.8, 2, 1, 3)\};$$

$$(\mu_1, \mu_2) \in \{(1.6, 1.6), (2.1, 1.1), (1.1, 2.1)\}; \quad r \in \{0.5, 30, 150\};$$

$$\mu_c \in \{3.1, 2.8, 2.5\} \text{ (sub-additive)} .$$

The average percentage sub-optimality of the four policies for the 108 problem instances are presented in Table 3.

In most cases, both JSQ and the pooling policy perform poorly. Surprisingly, JSQ performs poorly even when the routing cost is small. A further examination shows that the sub-optimality gap is particularly large when holding costs are asymmetric. This is because routing to a shorter queue is not necessarily optimal when holding costs are asymmetric. The performance of the pooling policy degrades substantially as pooling becomes less efficient (μ_c decreases). Since the service rate of a pooled server is sub-additive, pooling helps reduce the workload at the more

expensive queue, but the overall service rate decreases ($\mu_c < \mu_1 + \mu_2$).

The policy that dynamically allocates the servers performs close to optimal at medium and high routing costs, but it does not perform well when the routing cost is very low and/or pooling is inefficient (μ_c is low). On the other hand, the allocation-only policy performs almost as well as the optimal policy when (i) μ_c is high, (ii) r is high, and (iii) the workload for each queue is stable when the servers are split (i.e., $\lambda_1 < \mu_1$ and $\lambda_2 < \mu_2$). In fact, in 44 out of 108 instances (41% of the cases), the suboptimality gap is 0.0%. In other words, the allocation-only policy can capture most of the benefits of the optimal policy when the pooling is efficient (close to $\mu_1 + \mu_2$), routing is costly, and the loads are (on average) balanced ($\lambda_i < \mu_i, i = 1, 2$). On the other hand, when the routing cost is low or the service rate of a pooled server, μ_c, is low, the policy that dynamically routes arriving jobs performs close to optimal. In fact, we observe that, when keeping everything else constant the performance of the routing-only policy improves as μ_c decreases. This stands to reason, since as μ_c decreases, the optimal joint control is less apt to use server pooling; routing is the only level used in this case.

Table 4 presents instances where neither allocation-only nor routing-only performs well. This was different from the super-additive case in which at least one of the two policies performs close to the optimal policy in all 108 cases. All of these cases have unbalanced loads ($\lambda_i > \mu_i$ for some i), medium or high routing costs, and inefficient pooling. In other words, the system must route some of the jobs (since pooling is not efficient and loads are not balanced) and, at the same time, occasionally pool (since the routing cost is not low). Consequently, neither routing-only nor allocation-only can do well in these cases. It is interesting to note that all three conditions seem to be necessary for both policies to deviate from the optimal. For instance, if we increase μ_c to 3.1, the dynamic-allocation policy performs close to the optimal policy (gap = 0.0%).

5 Conclusions

In this paper we consider joint routing and allocation policies in a two station parallel queueing network. Our model includes routing costs and super-additive or sub-additive collaboration service rates. In the super-additive case, the optimal policy is completely characterized and follows intuition. It calls for clearing the higher cost station first, never routing to the higher cost queue and routing more customers to the lower-cost queue as the number of higher cost customers grows (or the number of low cost customers decreases). In the sub-additive case there is an inherent trade-off between serving faster (by splitting the servers) and reducing costs at a high cost station. This trade-off, combined with the fact that the routing can balance the workload as well, complicates the structure of optimal policy. Consequently, proving analytical results becomes very difficult, and, in many cases, the optimal policy is non-monotone. Instead of trying

to show results along the same lines as the super-additive case, we examine several special cases of the model to characterize the optimal policy. In both the super-additive and the sub-additive cases, we conduct a numerical study to examine which of the two controlling levers – dynamic routing and dynamic allocation – is more beneficial. In the super-additive case, the allocation-only policy performs well except when the routing cost is low and holding costs are not symmetric. In these cases, the routing-only policy performs very well. In the sub-additive case, the allocation-only policy performs well when routing is too costly, loads are on average balanced, and pooling is somewhat inefficient. The routing-only policy performs well when the routing cost is low. However, when the routing cost is medium or high, loads are not balanced ($\lambda_i > \mu_i$ for some i), and pooling is inefficient, neither allocation-only nor routing-only performs well compared to the policy that uses both.

This research leaves the door open for work in several directions. Since our model considers parallel stations, all work that comes to the system receives service at a single station and then leaves. Systems with stations in series, reentrant lines, those with abandonments, etc. are still open to be considered in the sub-additive case. There are also more exotic topologies that involve questions of routing and allocation that can be useful in practice.

6 Acknowledgments

The authors would like to thank Matthew Potoff for several helpful discussions on earlier versions of this model. The work of the second author was supported by the National Science Foundation under Grant Nos. CMMI-0900460 and CMMI-0826255. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Andradóttir, H. Ayhan, and D. G. Down. Queueing systems with synergistic servers. Preprint. 3

[2] M. Armony. Dynamic routing in large-scale service systems with heteroge- neous servers. *Queueing Systems: Theory and Applications*, 51(3):287–329, 2005. 3

[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. *Annals of Applied Probability*, 11(3):608–649, 2001. 3

[4] C. Buyukkoc, P. Varaiya, and J. Walrand. The $c\mu$-rule revisited. *Advances in Applied Probability*, 17(1):237–238, 1985. 3

[5] C. Derman, G. Lieberman, and S. Ross. On the optimal assignment of servers and a repairman. *Journal of Applied Probability*, 17(2):577–581, 1980. 3

[6] D. G. Down and M. E. Lewis. The n-network model with upgrades. *Probability and the Engineering and Informational Sciences*, 2, 2010. to appear. 3

[7] L. Green. Queueing system with general-use and limited-use servers. *Operations Research*, 33(1):168–182, 1985. 3

[8] J. Harrison. Dynamic scheduling of a multiclass queue: Discount optimality. *Operations Research*, 23(2):370–382, 1975. 3

[9] J. Harrison. A priority queue with discounted linear costs. *Operations Research*, 23(2):260–269, 1975. 3

[10] J. Harrison. Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete-review policies. *The Annals of Applied Probability*, 8(3):822–848, August 1998. 3

[11] J. Harrison and M. López. Heavy traffic resource pooling in parallel-server systems. *Queueing Systems: Theory and Applications*, 33(4):339–368, 1999. 3

[12] A. Hordijk and G. Koole. On the optimality of the generalized shortest queue policy. *Probability in the Engineering and Informational Sciences*, 4(4):477–487, 1990. 3

[13] A. Hordijk and G. Koole. On the assignment of customers to parallel queues. *Probability in the Engineering and Informational Sciences*, 6:495–511, 1992. 3

[14] J. Kingman. Two similar queues in parallel. *The Annals of Mathematical Statistics*, 32(4):1314–1323, December 1961. 3

[15] S. Lippman. Applying a new device in the optimization of exponential queueing system. *Operations Research*, 23(4):687–710, 1975. 6

[16] L. I. Sennott. *Stochastic Dynamic Programming and the Control of Queueing Systems*. Wiley Series in Probability and Statistics. John Wiley and Sons, New York, New York, 1999. 24

[17] R. Serfozo. An equivalence between continuous and discrete time Markov decision processes. *Operations Research*, 27(3):616–620, 1978. 6

[18] R. R. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, 15(2):406–413, June 1978. 3

[19] W. Winston. Assignment of customers to servers in a heterogeneous queue- ing system with switching. *Operations Research*, 25(3):469–483, 1977. 3

[20] W. Winston. Optimal dynamic rules for assigning customers to servers in a heterogeneous queueing system. *Naval Research Logistics Quarterly*, 24(2):293–300, 1977. 3

[21] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14:181–189, 1977. 3

A Proofs from Section 2

Proof of Proposition 2.2. We prove the result for the finite horizon case. The argument for the infinite horizon case is identical except that $v_\delta(\cdot)$ replaces $v_{n,\delta}(\cdot)$. Fix $n \geq 0$ and consider time $n+1$. Notice that idling both servers is not optimal unless the system is empty. From Lemma 2.1 and the fact that $\mu_c \geq \max\{\mu_1, \mu_2\}$, we have

$$\mu_c[v_{n,\delta}((i-1)^+, j) - v_{n,\delta}(i,j)] \leq \max\{\mu_1,\mu_2\}[v_{n,\delta}((i-1)^+, j) - v_{n,\delta}(i,j)], \text{ and}$$

$$\mu_c[v_{n,\delta}(i,(j-1)^+) - v_{n,\delta}(i,j)] \leq \max\{\mu_1,\mu_2\}[v_{n,\delta}(i,(j-1)^+) - v_{n,\delta}(i,j)].$$

That is, it is better for the servers to collaborate than for either to work while the other idles. ∎

Proof of Theorem 2.3. We need only show the existence of a (randomized) policy that yields finite average cost and either an irreducible Markov chain or a Markov chain with a single recurrent class where the transient states are absorbed in finite expected time. The result is then an application of Theorems 7.2.3 and 7.5.6 of Sennott [16].

Suppose first that $\mu_c \geq \mu_1 + \mu_2$. Define the stationary policy that assigns both servers to station 2 except to avoid idling, but routes all customers arriving to station 2 to station 1. It should be clear that any state with a positive number of customers in station 2 is transient, while in the long-run station 1 acts as an $M/M/1$ queue. The average queue length of station 1 is thus, $L_1 := \frac{\lambda_1+\lambda_2}{\mu_c-\lambda_1+\lambda_2}$ and the average cost rate is simply $L_1 h_1 + \lambda_2 r < \infty$.

Next consider the case that $\mu_c < \mu_1 + \mu_2$. If $\lambda_1 < \mu_1$ and $\lambda_2 < \mu_2$, then the policy that never routes and always splits the servers is enough to stabilize both stations (they both act as independent $M/M/1$ queues). Suppose $\lambda_1 \geq \mu_1$ and $\lambda_2 < \mu_2$; the symmetric inequalities are analogous. Define $p \leq 1$ such $p\lambda_1 = \mu_1$. Choose $\epsilon > 0$ such that $\lambda_2 + \lambda_1(1 - p + \epsilon) < \mu_2$. The assumption that $\lambda_1 + \lambda_2 < \mu_1 + \mu_2$ guarantees such an ϵ exists. Consider the randomized policy that never routes customers from station 2 to 1, and routes each arriving customer to station 1 to station 2 with probability $1 - p + \epsilon$. The servers remain split always. Since the arrival processes are Poisson, each station acts as independent $M/M/1$ queues. The arrival process to station 1 has rate $\lambda_1(p - \epsilon) < \mu_1$ and station 2 has arrival process $\lambda_2 + \lambda_1(1 - p + \epsilon) < \mu_2$. Each of these queues has finite average queue lengths and therefore finite average costs. Since the state space is irreducible the result follows. ∎

A.1 Proofs from Section 3

Proof of Lemma 3.4. We provide the proof for the finite horizon case. The proof for the infinite horizon cases are identical by replacing $v_{n,\delta}(\cdot)$ with $v_\delta(\cdot)$ or $y(\cdot)$. The case where $i = 0$ or $j = 0$ is proved in Proposition 2.2. Thus, we focus on the case where $i, j \geq 1$. Lemma 2.1 and the assumption that $\mu_c \geq \mu_1 + \mu_2$ imply

$$\begin{aligned}
&\mu_1 v_{n,\delta}(i-1,j) + \mu_2 v_{n,\delta}(i,j-1) - (\mu_1+\mu_2) v_{n,\delta}(i,j) \\
&\quad \geq (\mu_1+\mu_2) \min[v_{n,\delta}(i-1,j) - v_{n,\delta}(i,j), v_{n,\delta}(i,j-1) - v_{n,\delta}(i,j)] \\
&\quad \geq \mu_c \min[v_{n,\delta}(i-1,j) - v_{n,\delta}(i,j), v_{n,\delta}(i,j-1) - v_{n,\delta}(i,j)].
\end{aligned}$$

Analogously, we have

$$\mu_2 v_{n,\delta}(i-1,j) + \mu_1 v_{n,\delta}(i,j-1) - (\mu_1+\mu_2) v_{n,\delta}(i,j) \geq \mu_c \min[v_{n,\delta}(i-1,j) - v_{n,\delta}(i,j), v_{n,\delta}(i,j-1) - v_{n,\delta}(i,j)].$$

The result now follows from the optimality equations (2.5). ∎

Proof of Lemma 3.5. We prove the first result by induction. The second result then holds by letting n approach ∞ in the discounted cost case and by considering the convergence of $v_{\delta_n}(i,j) - v_{\delta_n}(0,0)$ along a subsequence for the average cost case. The first result holds trivially

for $n = 0$. Assume that it holds at n and consider time $n+1$. A little algebra in the optimality equations yields (suppressing the discount factor)

$$\begin{aligned}\Delta v_{n+1,\delta}(i,j) = h_1 - h_2 &+ \lambda_1\Big[\min\{\Delta v_{n,\delta}(i+1,j), r\} - \min\{\Delta v_{n,\delta}(i,j+1), r\} + \Delta v_{n,\delta}(i,j+1)\Big] \\ &+ \lambda_2\Big[\min\{-\Delta v_{n,\delta}(i+1,j), r\} - \min\{-\Delta v_{n,\delta}(i,j+1), r\} + \Delta v_{n,\delta}(i+1,j)\Big] \\ &+ \mu_c\Big[\min\{v_{n,\delta}(i,j), v_{n,\delta}(i+1,(j-1)^+)\} - \min\{v_{n,\delta}((i-1)^+, j+1), v_{n,\delta}(i,j)\}\Big].\end{aligned}$$

Applying the inductive hypothesis, $\Delta v_{n,\delta}(i,j) \geq 0$, we have

$$\begin{aligned}\Delta v_{n+1,\delta}(i,j) = h_1 - h_2 &+ \lambda_1\Big[\min\{\Delta v_{n,\delta}(i+1,j), r\} - \min\{\Delta v_{n,\delta}(i,j+1), r\} + \Delta v_{n,\delta}(i,j+1)\Big] \\ &+ \lambda_2 \Delta v_{n,\delta}(i,j+1) + \mu_c \begin{bmatrix} \Delta v_{n,\delta}(i-1,j) & i \geq 1 \\ 0 & i = 0. \end{bmatrix}\end{aligned}$$

The fact that $\Delta v_{n,\delta}(i,j+1) - \min\{\Delta v_{n,\delta}(i,j+1), r\} \geq 0$ and the inductive hypothesis yields that all of the terms are non-negative. The result follows. ∎

Proof of Theorem 3.1. The proof of Theorem 3.1 follows almost immediately from Lemma 3.5. We prove the result for the finite horizon case. Fix $n \geq 0$ and consider time $n+1$. Suppose $i, j \geq 1$. Lemma 3.5 yields $v_{n,\delta}(i-1,j) - v_{n,\delta}(i,j-1) = -\Delta v_{n,\delta}(i-1,j-1) \leq 0$. Hence, collaborating at station 1 is optimal at time $n+1$. If $i \geq 1, j = 0$, the result follows from Proposition 2.2. Similarly, for the case with $i = 0$ and $j \geq 1$. The infinite horizon cases are analogous. ∎

Proof of Lemma 3.6. We prove the first $D_1 \Delta v_{n,\delta}(i,j) \geq 0$ by induction. The fact that $D_2 \Delta v_{n,\delta}(i,j) \leq 0$ follows by symmetry and the second result then holds by taking limits as $n \to \infty$ in the discounted cost case and by considering the convergence of $v_{\delta_n}(i,j) - v_{\delta_n}(0,0)$ along a subsequence in the average cost case. A little algebra yields (suppressing the dependence on the discount factor)

$$\begin{aligned}D_1 \Delta v_{n+1,\delta}(i,j) = \lambda_1 &\begin{bmatrix} \min\{\Delta v_{n,\delta}(i+2,j), r\} - \min\{\Delta v_{n,\delta}(i+1,j+1), r\} + \Delta v_{n,\delta}(i+1,j+1) \\ - \min\{\Delta v_{n,\delta}(i+1,j), r\} + \min\{\Delta v_{n,\delta}(i,j+1), r\} - \Delta v_{n,\delta}(i,j+1) \end{bmatrix} \\ &+ \lambda_2 \left[\Delta v_{n,\delta}(i+1,j+1) - \Delta v_{n,\delta}(i,j+1)\right] + \mu_c \left[\Delta v_{n,\delta}(i,j) - 1_{\{i \geq 1\}} \Delta v_{n,\delta}(i-1,j)\right].\end{aligned}$$

The terms with coefficient λ_2 and μ_c are non-negative by the inductive hypothesis when $i \geq 1$ and by the inductive hypothesis and Lemma 3.5 when $i = 0$. Consider now the terms with

coefficient λ_1. Notice that whichever action we choose for the minimums corresponding to states $(i+1, j+1)$ and $(i+1, j)$ only stand to decrease the quantity. Suppose that it is optimal to accept arriving customers to station 1 in states $(i+2, j)$ and $(i, j+1)$. By choosing to accept arriving customers in states $(i+1, j+1)$ and $(i+1, j)$ a lower bound on the term with coefficient λ_1 is $\Delta v_{n,\delta}(i+2, j) - \Delta v_{n,\delta}(i+1, j) \geq 0$, where the inequality holds by the inductive hypothesis. Similarly, if it is optimal to reroute customers arriving to station 1 in states $(i+2, j)$ and $(i, j+1)$, then a lower bound for the term with coefficient λ_1 is $\Delta v_{n,\delta}(i+1, j+1) - \Delta v_{n,\delta}(i, j+1) \geq 0$.

Suppose now that it is optimal to reroute a customer in state $(i+2, j)$ while it is optimal to accept a customer in state $(i, j+1)$. By accepting in $(i+1, j+1)$ and rerouting $(i+1, j)$ a lower bound is 0. Finally, if it is optimal to accept a customer in state $(i+2, j)$ and it is optimal to reroute a customer in state $(i, j+1)$, then choosing to reroute in $(i+1, j+1)$ and accept in $(i+1, j)$ yields a lower bound of $\Delta v_{n,\delta}(i+2, j) - \Delta v_{n,\delta}(i+1, j) + \Delta v_{n,\delta}(i+1, j+1) - \Delta v_{n,\delta}(i, j+1) \geq 0$, where the inequality comes from the inductive hypothesis. The proof is complete. ∎

Proof of Theorem 3.2. To show that the first statement holds, suppose that the current state is (i, j). It is optimal to route a customer arriving at station 2 to station 1 at time $n+1$ if $\Delta v_{n,\delta}(i, j)) \leq -r \leq 0$. However, from Lemma 3.5 we have $\Delta v_{n,\delta}(i, j) \geq 0$ for all $i, j \geq 1$. Hence, it is not optimal to route a customer from station 2 to station 1.

Suppose at time $n+1$ that it is optimal to route a job arriving to station 1 when the current state is (i, j). From (3.1), this implies $\Delta v_{n,\delta}(i, j) \geq r$. The first inequality in Statement 1 of Lemma 3.6 yields

$$D_1 \Delta v_{n,\delta}(i, j) = v_{n,\delta}(i+2, j) - v_{n,\delta}(i+1, j) - v_{n,\delta}(i+1, j+1) + v_{n,\delta}(i, j+1) \geq 0.$$

Combining this with the fact that $\Delta v_{n,\delta}(i, j) \geq r$ yields

$$v_{n,\delta}(i+2, j) - v_{n,\delta}(i+1, j) \geq v_{n,\delta}(i+1, j) - v_{n,\delta}(i, j+1) \geq r.$$

This implies that it is also optimal to route a customer arriving at station 1 to station 2 in state $(i+1, j)$. A similar argument holds for $(i, j-1)$ using the second inequality of Statement 1 of Lemma 3.6. ∎

A.2 Proofs from Section 4

Proof of Theorem 4.2. It should be clear that if inequalities **N**(1)- **N**(2) hold for all n, then they hold in the limit and both Statements 1 and 2 hold. We show this by induction. Note that the inequalities **N**(1)- **N**(2) hold trivially for $n = 0$. Assume that they hold for n and consider

$n+1$. The assumption that they hold for n implies that it is optimal to split service for $i, j \geq 1$. Suppose $i, j \geq 2$ and consider,

$$\begin{aligned} v_{n+1}(i,j) - v_{n+1}(i,j-1) = h_2 &+ \lambda_1[v_n(i+1,j) - v_n(i+1,j-1)] + \lambda_2[v_n(i,j+1) - v_n(i,j)] \\ &+ \mu_1[v_{n+1}(i-1,j) - v_{n+1}(i-1,j-1)] \\ &+ \mu_2[v_{n+1}(i,j-1) - v_{n+1}(i,j-2)]. \end{aligned} \tag{A.1}$$

Similarly,

$$\begin{aligned} v_{n+1}(i,j) - v_{n+1}(i-1,j) = h_1 &+ \lambda_1[v_n(i+1,j) - v_n(i,j)] + \lambda_2[v_n(i,j+1) - v_n(i-1,j+1)] \\ &+ \mu_1[v_{n+1}(i-1,j) - v_{n+1}(i-2,j)] \\ &+ \mu_2[v_{n+1}(i,j-1) - v_{n+1}(i-1,j-1)]. \end{aligned} \tag{A.2}$$

Multiply (A.1) by $\mu_c - \mu_2$ and (A.2) by μ_1. Using the inductive hypothesis and the assumption on the costs yields that the difference in each term is non-positive.

Suppose now that $i = 1, j \geq 2$. The argument for the arrival terms remains unchanged. Consider the terms with coefficients related to service.

$$\begin{aligned} &(\mu_c - \mu_2)[\mu_1[v_n(0,j) - v_n(0,j-1)] + \mu_2[v_n(1,j-1) - v_n(1,j-2)]] \\ &\quad - \mu_1[\mu_1 v_n(0,j) + \mu_2 v_n(1,j-1) - \mu_c v_n(0,j-1) - (\mu_1 + \mu_2 - \mu_c)v_n(0,j)] \\ &= (\mu_c - \mu_2 - \mu_1)\mu_1[v_n(0,j) - v_n(0,j-1)] + (\mu_c - \mu_2)\mu_2[v_n(1,j-1) - v_n(1,j-2)]] \\ &\quad - \mu_1[\mu_1 v_n(0,j-1) + \mu_2 v_n(1,j-1) - \mu_c v_n(0,j-1) - (\mu_1 + \mu_2 - \mu_c)v_n(0,j)] \\ &= (\mu_c - \mu_2)\mu_2[v_n(1,j-1) - v_n(1,j-2)] - \mu_1\mu_2[v_n(1,j-1) - v_n(0,j-1)] \leq 0, \end{aligned}$$

where the inequality follows from the inductive hypothesis $\mathbf{N}(1)$ (applied at $(1, j-1)$).

Suppose $i \geq 2$ and $j = 1$ and consider

$$\begin{aligned} &(\mu_c - \mu_2)[\mu_1 v_n(i-1,1) + \mu_2 v_n(i,0) - \mu_c v_n(i-1,0) - (\mu_1 + \mu_2 - \mu_c)v_n(i,0)] \\ &\quad - \mu_1[\mu_1[v_n(i-1,1) - v_n(i-2,1)] + \mu_2[v_n(i,0) - v_n(i-1,0)] \\ &\leq (\mu_c - \mu_2)[\mu_1 v_n(i-1,1) + \mu_2 v_n(i,0) - \mu_c v_n(i-1,0) - (\mu_1 + \mu_2 - \mu_c)v_n(i,0)] \\ &\quad - \mu_1[\mu_1 v_n(i-1,1) + \mu_2 v_n(i,0) - \mu_c v_n(i-1,0) - (\mu_1 + \mu_2 - \mu_c)v_n(i-1,1)] \\ &= (\mu_c - \mu_2)[\mu_1[v_n(i-1,1) - v_n(i,0)] + \mu_c[v_n(i,0) - v_n(i-1,0)]] \end{aligned}$$

$$
\begin{aligned}
&\quad - \mu_1[\mu_2[v_n(i,0) - v_n(i-1,1)] + \mu_c[v_n(i-1,1) - v_n(i-1,0)]] \\
&= \mu_c\mu_1[v_n(i-1,1) - v_n(i,0)] + (\mu_c - \mu_2)\mu_c[v_n(i,0) - v_n(i-1,0)]] \\
&\quad - \mu_c\mu_1[v_n(i-1,1) - v_n(i-1,0)] \\
&= (\mu_c - \mu_1 - \mu_2)\mu_c[v_n(i,0) - v_n(i-1,0)] \leq 0,
\end{aligned}
$$

where the first inequality is due to the fact that we used the (potentially) sub-optimal action of serving at station 2 in state $(i-1,1)$ and the last inequality follows from the assumption that $\mu_c \leq \mu_1 + \mu_2$. The case with $i = j = 1$ is the same as this case except that the first inequality above is an equality. This concludes the proof of $\mathbf{N}(1)$.

Consider now $\mathbf{N}(2)$. The arrival terms $i, j \geq 1$ and the service terms for $i, j \geq 2$ hold in the same way as in $\mathbf{N}(1)$. Consider the service terms for $i \geq 2, j = 1$.

$$
\begin{aligned}
&(\mu_c - \mu_1)[\mu_1[v_n(i-1,1) - v_n(i-2,1)] + \mu_2[v_n(i,0) - v_n(i-1,0)]] \\
&\quad - \mu_2[\mu_1 v_n(i-1,1) + \mu_2 v_n(i,0) - \mu_c v_n(i-1,0) - (\mu_1 + \mu_2 - \mu_c)v_n(i,0)] \\
&= (\mu_c - \mu_1 - \mu_2)[\mu_2[v_n(i,0) - v_n(i-1,0)] + (\mu_c - \mu_1)\mu_1[v_n(i-1,1) - v_n(i-2,1)] \\
&\quad - \mu_2[\mu_1 v_n(i-1,1) + \mu_2 v_n(i-1,0) - \mu_c v_n(i-1,0) - (\mu_1 + \mu_2 - \mu_c)v_n(i,0)] \\
&= (\mu_c - \mu_1)\mu_1[v_n(i-1,1) - v_n(i-2,1)] - \mu_1\mu_2[v_n(i-1,1) - v_n(i-1,0)] \leq 0
\end{aligned}
$$

where the inequality follows from the inductive hypothesis. Let $i = 1, j \geq 2$.

$$
\begin{aligned}
&(\mu_c - \mu_1)[\mu_1 v_n(0,j) + \mu_2 v_n(1,j-1) - \mu_c v_n(0,j-1) - (\mu_1 + \mu_2 - \mu_c)v_n(0,j)] \\
&\quad - \mu_2[[\mu_1[v_n(0,j) - v_n(0,j-1)] + \mu_2[v_n(1,j-1) - v_n(1,j-2)]] \\
&\leq (\mu_c - \mu_1)[\mu_1 v_n(0,j) + \mu_2 v_n(1,j-1) - \mu_c v_n(0,j-1) - (\mu_1 + \mu_2 - \mu_c)v_n(0,j)] \\
&\quad - \mu_2[\mu_1 v_n(0,j) + \mu_2 v_n(1,j-1) - \mu_c v_n(0,j-1) - (\mu_1 + \mu_2 - \mu_c)v_n(1,j-1)] \\
&= (\mu_c - \mu_1)[\mu_c v_n(0,j-1) - (\mu_2 - \mu_c)v_n(0,j)] \\
&\quad - \mu_2[\mu_1 v_n(0,j) - \mu_c v_n(0,j-1)] \\
&= (\mu_c - \mu_1 - \mu_2)[v_n(0,j) - v_n(0,j-1) \leq 0,
\end{aligned}
$$

where the first inequality follows since we have used the sub-optimal action of assigning both servers to station 1 in state $(1, j-1)$. Note that the same argument here holds if $i = j = 1$ except that the first inequality is an equality. This concludes the proof of $\mathbf{N}(2)$. ∎

Proof of Theorem 4.4. Consider $\mathbf{N}(1)$. Clearly, $\mathbf{N}(1)$ holds for $n = 0$. Assume it holds for n. Since the system dynamics when it comes to service are exactly the same as without routing, the inequality on these terms holds in precisely the same manner as Theorem 4.2. Consider now $(\mu_c - \mu_2)[\tilde{v}_{n+1,\delta}(i,j) - \tilde{v}_{n+1,\delta}(i,j-1)] - \mu_1[\tilde{v}_{n+1,\delta}(i,j) - \tilde{v}_{n+1,\delta}(i-1,j)]$. If the optimal monotone routing action is the same in all 4 starting states, there is an increase in the number of customers at the appropriate station. The induction hypothesis yields the result. For example, if it is optimal to route to station 2 in state $(i-1,j)$, then the decision-maker also routes to station 2 in state (i,j) and thus routes to station 2 in state $(i,j-1)$. The terms associated with arrivals are

$$\begin{aligned}&[\lambda_1+\lambda_2]\big[(\mu_c-\mu_2)[\tilde{v}_{n,\delta}(i,j+1)-\tilde{v}_{n,\delta}(i,j)]-\mu_1[\tilde{v}_{n,\delta}(i,j+1)-\tilde{v}_{n,\delta}(i-1,j+1)]\big]\\&=[\lambda_1+\lambda_2][(\mu_c-\mu_2)[D_2\tilde{v}_{n,\delta}(i,j)]-\mu_1[D_1\tilde{v}_{n,\delta}(i-1,j+1)]]\le 0,\end{aligned}$$

where the inequality follows from the inductive hypothesis applied at state $(i,j+1)$.

Suppose it is optimal to route to station 1 in state $(i-1,j)$ and station 2 in state (i,j). The fact that the policy is monotone implies that the customer would be routed to station 2 in state $(i,j-1)$. The terms associated the arrivals are $[\lambda_1+\lambda_2](\mu_c-\mu_1-\mu_2)[\tilde{v}_{n,\delta}(i,j+1)-\tilde{v}_{n,\delta}(i,j)]\le 0$, where the inequality holds by the monotonicity of $\tilde{v}_n$ and the assumption that $\mu_c \le \mu_1+\mu_2$.

If it is optimal to route to station 1 in states $(i-1,j)$ and (i,j), but to station 2 in state $(i,j-1)$, the arrival terms are

$$\begin{aligned}&[\lambda_1+\lambda_2]\big[(\mu_c-\mu_2)[\tilde{v}_{n,\delta}(i+1,j)-\tilde{v}_{n,\delta}(i,j)]-\mu_1[\tilde{v}_{n,\delta}(i+1,j)-\tilde{v}_{n,\delta}(i,j)]\big]\\&=[\lambda_1+\lambda_2][(\mu_c-\mu_1-\mu_2)[D_1\tilde{v}_{n,\delta}(i,j)]\le 0.\end{aligned}$$

Lastly, as alluded to the case where each arrival is routed to station 1 follows by the inductive hypothesis at $(i+1,j)$ in the same way as when they are routed to station 2.

Consider now $\mathbf{N}(2)$ and again only the arrival terms of $(\mu_c-\mu_1)[\tilde{v}_{n+1,\delta}(i,j)-\tilde{v}_{n+1,\delta}(i-1,j)]-\mu_2[\tilde{v}_{n+1,\delta}(i,j)-\tilde{v}_{n,\delta}(i,j-1)]$. To stay within the class of monotone policies, routing to station 1 in state $(i,j-1)$ implies we route to station 1 in states (i,j) and $(i-1,j)$. The arrival terms are $(\lambda_1+\lambda_2)(\mu_c-\mu_1)[\tilde{v}_{n,\delta}(i+1,j)-\tilde{v}_{n,\delta}(i,j)]-\mu_2[\tilde{v}_{n,\delta}(i+1,j)-\tilde{v}_{n,\delta}(i+1,j-1)]\le 0$, where the inequality follows by applying the inductive hypothesis at state $(i+1,j)$. The case where the decision-maker routes to station 2 in all 3 states is similar by using the inductive hypothesis at $(i,j+1)$.

If the decision-maker routes station 2 in state $(i,j-1)$ and to station 1 in states (i,j) and $(i-1,j)$ we have $(\lambda_1+\lambda_2)(\mu_c-\mu_1-\mu_2)[\tilde{v}_{n,\delta}(i+1,j)-\tilde{v}_{n,\delta}(i,j)]\le 0$ since $\mu_1+\mu_2\ge\mu_c$. Similarly, if the decision-maker routes to station 2 in state $(i,j-1)$ and (i,j) and to station 1 in state $(i-1,j)$

the arrival terms are $(\lambda_1+\lambda_2)(\mu_c-\mu_1-\mu_2)[\tilde{v}_{n+1,\delta}(i,j+1)-\tilde{v}_{n+1,\delta}(i,j)]\leq 0$. Thus, the result is proven since all other cases are precluded by the restriction to only monotone routing policies. For example, routing to station 2 in $(i,j-1)$, to station 1 in (i,j) and station 2 in $(i-1,j)$ is precluded since routing to station 1 in (i,j) implies that we route to station 1 in $(i-1,j)$. ∎

Proof of Theorem 4.5. Note that the finite horizon discounted expected cost function in the balanced case can be written (for a particular policy π)

$$v^{\pi}_{\alpha,t}(i,j)=h\int_0^t e^{-\alpha s}\,\mathbb{E}^{\pi}_{(i,j)}(Q_1(s)+Q_2(s))ds+\text{routing costs}$$

Without loss of generality assume $h=1$. For $k=1,2$ let $A_k(t)$ be a Poisson process of rate λ_k. Similarly, define $B^{\mu}_k(x)$ as independent Poisson processes with rate μ and $C^{\mu_c}_k(x)$ as (again) independent Poisson processes with rate μ_c representing potential services when the servers are split or collaborating at each respective station. For a policy π let $d_{s_k}(x-)=1$ or 2 denote the place where arrivals to station k (for $k=1,2$) are routed at time x. Suppose $d_a(x-)=1,2$, or 3 describes the action of assigning both servers to station 1, 2, or split, respectively. The queue length processes under policy π can be written

$$\begin{aligned}Q_1(s)=Q_1(0)&+\int_0^s 1_{\{d_{s_1}(x-)=1\}}dA_1(x)+\int_0^s 1_{\{d_{s_2}(x-)=1\}}dA_2(x)\\&-\int_0^s 1_{\{Q_1(x-)>0\}}1_{\{d_a(x-)=1\}}dC^{\mu_c}_1(x)-\int_0^s 1_{\{Q_1(x-)>0\}}1_{\{d_a(x-)=3\}}dB^{\mu}_1(x),\end{aligned}$$

and

$$\begin{aligned}Q_2(s)=Q_2(0)&+\int_0^s 1_{\{d_{s_2}(x-)=2\}}dA_2(x)+\int_0^s 1_{\{d_{s_1}(x-)=2\}}dA_1(x)\\&-\int_0^s 1_{\{Q_2(x-)>0\}}1_{\{d_a(x-)=2\}}dC^{\mu_c}_1(x)-\int_0^s 1_{\{Q_2(x-)>0\}}1_{\{d_a(x-)=3\}}dB^{\mu}_2(x).\end{aligned}$$

Taking expectations and summing yields

$$\begin{aligned}\mathbb{E}^{\pi}_{(ij)}(Q_1(s)+Q_2(s))=i+j+\lambda_1 s+\lambda_2 s&-\mu_c\int_0^s \mathbb{E}^{\pi}_{(i,j)}\,1_{\{Q_1(x-)>0\}}1_{\{d_a(x-)=1\}}dx\\&-\mu_c\int_0^s \mathbb{E}^{\pi}_{(i,j)}\,1_{\{Q_2(x-)>0\}}1_{\{d_a(x-)=2\}}dx\\&-\mu\int_0^s \mathbb{E}^{\pi}_{(i,j)}(1_{\{Q_1(x-)>0\}}+1_{\{Q_2(x-)>0\}})1_{\{d_a(x-)=3\}}dx.\end{aligned}\tag{A.3}$$

Define now two processes on the same probability space so that they see the same arrivals and workload requirements. Assume each process starts in state (i,j) such that $i,j\geq 1$. Process 1

uses a policy that splits the servers (sets $d_a(x-) = 3$) whenever possible and assigns both servers to the station with customers there when one station is empty. Suppose Process 2 uses a policy that assigns both servers to station 1 initially. Process 1 also uses whatever routing policy Process 2 uses. That is, when a customer arrives, Process 1 assigns the customers to the same station that Process 2 does (regardless of the current queue lengths). Note that since both processes see exactly the same (total) amount of work in the time $[0, s)$ the cost is decreased most during times when both stations have work available, by assigning a server to each station (at rate 2μ as opposed to μ_c). That is $\mathbb{E}_{(ij)}(Q_1(s) + Q_2(s))$ is minimized at every time point. The result follows for any fixed and finite t. Taking limits as t approaches ∞ yields the result for the infinite horizon case. Similarly for the average cost case (when the limit exists). ∎

Proof of Proposition 4.7. We present the proof of Statement 1; Statement 2 follows by symmetry. The case where $i = j$ follows immediately from symmetry: If $i = j$, $\Delta v_{n+1,\delta}(i, i) = v_{n+1,\delta}(i + 1, i) - v_{n+1,\delta}(i, i + 1) = 0$. Thus, we focus on the case that $j > i \geq 0$. From (4.2) and Theorem 4.5, we have (after a little algebra)

$$\begin{aligned}
\Delta v_{n+1,\delta}(i,j) &= H^m v_{n,\delta}(i+1,j) - H^m v_{n,\delta}(i,j+1) \\
&= \lambda\big[\min\{\Delta v_{n,\delta}(i+1,j), r\} - \min\{\Delta v_{n,\delta}(i,j+1), r\} + \Delta v_{n,\delta}(i,j+1)\big] \\
&\quad + \lambda\big[\min\{-\Delta v_{n,\delta}(i+1,j), r\} - \min\{-\Delta v_{n,\delta}(i,j+1), r\} + \Delta v_{n,\delta}(i+1,j)\big] \\
&\quad + \begin{cases} \mu[\Delta v_{n,\delta}(i-1,j) + \Delta v_{n,\delta}(i,j-1)] & \text{for } j \geq 1, \\ (2\mu - \mu_c)[v_{n,\delta}(i+1,j) - v_{n,\delta}(i,j)] + \mu \Delta v_{n,\delta}(i-1,j) & \text{for } j = 0. \end{cases}
\end{aligned} \tag{A.4}$$

Suppose first that $j \geq 2$. Using the inductive hypothesis yields and the fact that $j > i$ we have

$$\begin{aligned}
\Delta v_{n+1,\delta}(i,j) &= H^m v_{n,\delta}(i+1,j) - H^m v_{n,\delta}(i,j+1) \\
&= \lambda \Delta v_{n,\delta}(i+1,j) + \lambda\big[\min\{-\Delta v_{n,\delta}(i+1,j), r\} \\
&\quad - \min\{-\Delta v_{n,\delta}(i,j+1), r\} + \Delta v_{n,\delta}(i+1,j)\big] \\
&\quad + \mu[\Delta v_{n,\delta}(i-1,j) + \Delta v_{n,\delta}(i,j-1)].
\end{aligned} \tag{A.5}$$

There are now 4 cases to consider. If $-\Delta v_{n,\delta}(i, j + 1) \geq r$ and $-\Delta v_{n,\delta}(i + 1, j) \geq r$ then (A.5) becomes $2\lambda \Delta v_{n,\delta}(i + 1, j) + \mu[\Delta v_{n,\delta}(i - 1, j) + \Delta v_{n,\delta}(i, j - 1)] \leq 2(\lambda + \mu)r = r$, where the inequality is due to the inductive hypothesis and the equality follows from the uniformization constant being equal to 1. Suppose $-\Delta v_{n,\delta}(i, j + 1) \leq r$ and $-\Delta v_{n,\delta}(i + 1, j) \leq r$ then (A.5) becomes $\lambda[\Delta v_{n,\delta}(i + 1, j) + \Delta v_{n,\delta}(i, j + 1)] + \mu[\Delta v_{n,\delta}(i - 1, j) + \Delta v_{n,\delta}(i, j - 1)] \leq 2(\lambda + \mu)r = r$,

where again the inequality follows from the inductive hypothesis. If $-\Delta v_{n,\delta}(i,j+1) \leq r$ and $-\Delta v_{n,\delta}(i+1,j) \geq r$ then (A.5) is

$$\lambda[2\Delta v_{n,\delta}(i+1,j) + r + \Delta v_{n,\delta}(i,j+1)] + \mu[\Delta v_{n,\delta}(i-1,j) + \Delta v_{n,\delta}(i,j-1)]$$

$$\leq \lambda[-2r + r + \Delta v_{n,\delta}(i,j+1)] + \mu[\Delta v_{n,\delta}(i-1,j) + \Delta v_{n,\delta}(i,j-1)] \leq 2(\lambda+\mu)r = r,$$

where the first inequality follows from the assumption that $-\Delta v_{n,\delta}(i+1,j) \geq r$. Finally, suppose $-\Delta v_{n,\delta}(i,j+1) \geq r$ and $-\Delta v_{n,\delta}(i+1,j) \leq r$ then (A.5) becomes $\lambda[\Delta v_{n,\delta}(i+1,j) - r] + \mu[\Delta v_{n,\delta}(i-1,j) + \Delta v_{n,\delta}(i,j-1)] \leq 2(\lambda+\mu)r = r$, where the inequality holds using the inductive hypothesis.

Suppose now $j = 1$ (so that $j > i$ implies $i = 0$). The previous discussion yields a bound of $2\lambda r$ for the terms with coefficient λ. The terms related to services in $\Delta v_{n+1,\delta}(0,1)$ are

$$\mu[v_{n,\delta}(0,1) + v_{n,\delta}(1,0)] - \mu_c v_{n,\delta}(0,1) - (2\mu - \mu_c)v_{n,\delta}(0,2)$$

$$= (2\mu - \mu_c)[v_{n,\delta}(0,1) - v_{n,\delta}(0,2)] \leq 0,$$

where the equality holds since $v_{n,\delta}(0,1) = v_{n,\delta}(1,0)$ by symmetry. This clearly leads to an upper bound of r. The result follows. ∎