Heuristics for Allocation of Reconfigurable Resources in a Serial Line with Reliability Considerations

Cheng-Hung Wu[1] Douglas G. Down[2] Mark E. Lewis[3]

resubmitted January 19, 2007

[1]Professor Cheng-Hung Wu is with the Department of Industrial Engineering and Engineering Management at the National Tsing Hua University, 101 Sec. 2 Kuang-Fu Rd., Hsinchu 30013, Taiwan, wuch@ie.nthu.edu.tw

[2]Professor Down is with the Department of Computing and Software at McMaster University, 1280 Main Street West, Hamilton, ON L8S 4L7, Canada, downd@mcmaster.ca

[3]Professor Lewis is with the School of Operations Research and Information Engineering at Cornell University, 226 Rhodes Hall, Ithaca, New York 14853, mel47@cornell.edu

Abstract

We consider the allocation of reconfigurable resources in a serial line with machine failures. Each station is equipped with non-idling dedicated servers while the whole system is equipped with a finite number of reconfigurable servers that are available to be assigned to any station. We provide conditions for a policy to achieve throughput optimality. We also show in the two-station case that transition monotone optimal policies exist. We discuss heuristics based on the two-server model that reduce average holding costs significantly. These heuristics are compared to several heuristics from the literature via a detailed numerical study.

1 Introduction

Due to rapidly changing market demands and the popularity of customized products, many companies have invested in reconfigurable manufacturing systems (RMSs) [13, 19]. That is, they have developed systems in which some of the capacity can be configured in a relatively short period of time to handle several different job types. To allow for this, the material handling system in the RMS is usually capable of routing work-in-process (WIP). This makes dynamic resource allocation (or dynamic job re-routing) easily implementable without additional costs. On the other hand, while this may be the motivation for the development of RMSs, there is another added benefit; the alleviation of congestion due to machine failures. A typical method of dealing with machine failures is to build enough redundancy into the system so as to have excess capacity at each station. Dynamic allocation of resources allows factory managers to reduce recovery costs by using the reconfigurable resources to "cover" for failed servers in the case of a machine breakdown (cf. [6, 15, 28]).

Unfortunately, due to the "curse of dimensionality" most of the study of optimal allocation policies has been restricted to systems with only a few stations (usually 2) and a few machines. In the particular case where reliability is considered under the average holding cost criterion the system is even less tractable. In order to overcome the dimensionality restriction, we prove that transition monotone policies are average cost optimal in a two-station tandem queueing system with reliability considerations. Based on the intuition obtained from the two station queueing network, heuristic resource allocation policies are developed. This heuristic algorithm significantly reduces calculations and according to our simulation experiments performs very well under both the long run average holding costs and long term average throughput criteria. Along the way we provide conditions for the stability of the network and optimality in terms of average throughput.

By and large the literature on dynamic job re-routing or resource allocation in manufacturing systems focuses on reducing holding costs or increasing throughput without considering machine reliability. One major approach of minimizing holding costs is the clearing system analysis. The goal is, given a fixed number of jobs in the system (a production schedule), how might one empty the system at minimum cost. Under this assumption, Farrar [14] shows the existence of optimal policies that are *transition monotone* (monotone in the number of jobs at the second station) in a two-station tandem queueing system with a flexible server. Ahn et al., [2] proves that allocating both flexible servers to a single station is optimal in a two station queueing system with two identical flexible servers. In [21], Shiefermayr and Weichbold provide a complete solution for the optimal server scheduling policy for a model very similar to that in [2]. In two-station systems with reliability considerations and multiple dedicated and reconfigurable servers, Wu et al., [28] proves the existence of optimal transition monotone policies. With additional assumptions, they also prove that the optimal switching curves that define the optimal transition monotone policy should have slopes of at least -1. When external arrivals are allowed, Hajek [16] shows the existence of transition monotone policies in a queueing network with two flexible servers. Ahn et al. [1] shows that the result in [2] holds in a system with external arrivals under average cost criterion. In [24], Sennott et al. examine a model with several workers dedicated to stations and one "floating" worker and discuss how the floating worker can be used to stabilize the system and to minimize holding costs. They allow for switching times, but do not consider failures. For a more detailed literature review of flexible server allocation problems readers may wish to consult [17].

In addition to minimizing holding costs, several papers have studied the resource allocation problem to maximize throughput. Freiheit and Hu [15] use static allocation policies to examine server and buffer capacity allocation problems and try to improve system throughput. When all servers are reliable, Andradóttir et al., [3, 4] examine tandem queueing systems with finite buffers. Andradóttir et al., [6] studied the dynamic resource allocation problem with both *class* and machine reliability considerations. They show that the maximum throughput is tightly bounded by the solution of a linear program. They also proved that a timed generalized round-robin policy can approximate the maximum capacity. In the current work we study the problem with reliability considerations under the minimum average holding cost criterion. Moreover, we provide conditions that guarantee throughput optimality in the infinite buffer case. It turns out that the heuristic we propose achieves throughput close to maximum and can be slightly modified (as discussed) to achieve this maximum.

The remainder of the paper is organized as follows. A mathematical description of the queueing model and stability conditions are provided in Section 2. In Section 3, we give conditions for policies to guarantee throughput optimality. We use Markov decision processes to prove the optimality of a transition monotone policy in two station queueing networks with reliability considerations in Section 4. Using this result, a heuristic for larger systems (more than 2 stations) is provided in Section 5. Robustness of the heuristic discussed in Section 5 is evaluated by a discrete event simulation in Section 6. The heuristic is also compared to several other policies from the literature. We conclude the paper in Section 7.

2 Preliminaries

We assume that N operations (at N stations) must be performed in a fixed order on every job or customer that enters the system. Job inter-arrival times are assumed to be exponentially distributed with rate λ. The service requirement for each job is exponentially distributed with mean 1 at each station. For the k^{th} station of the system, there are M_k dedicated non-idling servers that can only serve customers at that station. There are M_r additional *generalized* [6] reconfigurable servers that have constant service rates at all stations. These reconfigurable servers can be assigned to any station with zero setup time and costs (see Figure 1). Throughout the rest of the paper, when considering an $N-$station system, label the station that receives arrivals from outside of the system as station 1 and the others sequentially with the last one before exiting the system being labeled station N. We say station n is *upstream (downstream)* from station n' if $n < n'$ $(n > n')$.

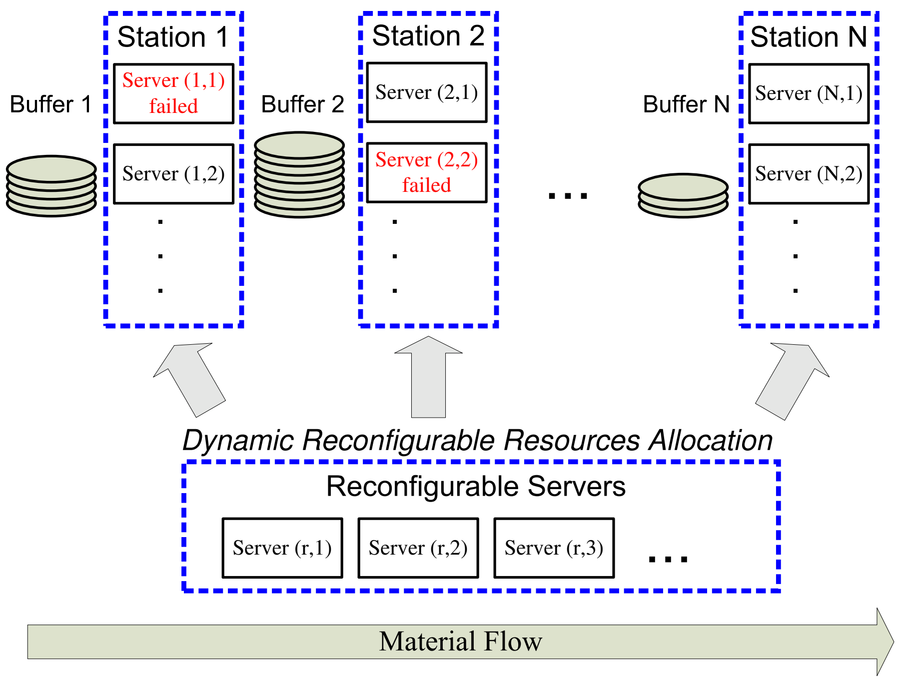


Figure 1: An example of an N-station serial line with reconfigurable servers

Let $(r,\ell), \ell \in \{1,2,...,M_r\}$ be the ℓ^{th} reconfigurable server and (k,ℓ) indicate the ℓ^{th} server at station k, where $k \in \{1,2,...,N\}$ and $\ell \in \{1,2,...,M_k\}$. We assume that the service rate of dedicated server (k,ℓ) is $\mu_{k,\ell}$ and the reconfigurable server (r,ℓ) has service rate $\mu_{r,k,\ell}$ at station k. The failure and repair times are assumed to be exponentially distributed with rates $\alpha_{k,\ell}$ ($\alpha_{r,\ell}$) and $\beta_{k,\ell}$ ($\beta_{r,\ell}$) for dedicated server (k,ℓ) (reconfigurable server (r,ℓ)). When a server failure occurs, the service rate of the corresponding server is reduced to zero. If the number of servers is more than the number of jobs at any station, multiple servers can *collaborate* on a single job; the service rates are additive. This assumption is usual when large components are being produced and more than one machine can be assigned to a single product. Since the only difference between this assumption and the case when servers are not allowed to collaborate is when the system is lightly loaded, we believe that this serves as a reasonable approximation in either case. A brief numerical study to this effect is provided in Section 6.1. The state space is

$$\mathbf{S} = \{q_1, q_2, ..., q_N, m_{1,1}, m_{1,2}, ..., m_{1,M_1}, m_{2,1}, ..., m_{N-1,M_{N-1}}, m_{N,1}, ..., m_{N,M_N}, m_{r,1}, ..., m_{r,M_r}\},$$

where q_k is the queue length at station k, including the one currently in service, $m_{k,\ell} \in \{0 : \text{failed}, 1 : \text{operational}\}$ is the status of server (k,ℓ), and $m_{r,\ell}$ is the status of reconfigurable server ℓ. Let Π be the set of all non-idling, non-anticipating (measurable) policies. Each policy describes where the reconfigurable resources should be allocated for all states, for all time (down servers are not assigned). We next provide conditions under which we can guarantee existence of a policy that achieves stability. This will be useful in both the throughput and average cost analysis to follow. In essence, it is an application of results in [6].

Note that the availability of each server can be modeled as a 2-state (on-off) continuous-time Markov chain. Let $a_{k,\ell}$ be the long-run proportion of time that server (k,ℓ) is operational and $\delta^{\pi}_{r,n,\ell}$ be the proportion of time that reconfigurable server (r,ℓ) is assigned to station n for a policy π. That is

$$a_{k,\ell} = \frac{\alpha_{k,\ell}}{\alpha_{k,\ell} + \beta_{k,\ell}} \qquad \delta^{\pi}_{r,n,\ell} = \limsup_{T\to\infty} \frac{1}{T} \int_0^T 1_{\{\pi \text{ assigns } (r,\ell) \text{ to station } n\}}(t)dt,$$

where $1_A(t)$ is the indicator of event A at time t. Consider the following linear program (LP), which we call $LP(1)$.

$$
\begin{aligned}
&\max \quad \lambda \\
s.t. \quad &\sum_{\ell=1}^{M_k} a_{k,\ell}\mu_{k,\ell} + \sum_{\ell=1}^{M_r} \delta_{r,k,\ell}\mu_{r,k,\ell} \geq \lambda, \quad \text{for all } k \in \{1,2,\dots,N\} \qquad (2.1)\\
&\sum_{k=1}^{N} \delta_{r,k,\ell} \leq a_{r,\ell}, \quad \text{for all } \ell \in \{1,2,...,M_r\} \qquad (2.2)\\
&\delta_{r,k,\ell} \geq 0, \quad \text{for all } k \in \{0,1,...,N\} \text{ and } \ell \in \{1,2,...,M_r\}.
\end{aligned}
$$

The above LP finds the maximal arrival rate, λ^*, such that the average service rate is greater than or equal to λ^* (the first constraint) and guarantees that the total proportion of time each reconfigurable server is allocated does not exceed its availability (the second constraint). Define the following optimality criteria.

Definition 2.1 *Let $D^\pi(t)$ denote the number of departures from the system by time t under policy π. The throughput of policy π is defined by $\lim_{t\to\infty} \frac{D^\pi(t)}{t}$.*

Definition 2.2 *Suppose for each job at station k, there is a linear holding cost accrued at a fixed rate h_k. The expected average cost of policy π is defined by*

$$\lim_{t\to\infty} \frac{\int_0^t \sum_{n=1}^N h_n \, \mathbb{E}\, Q_n^\pi(s) ds}{t},$$

where $Q_n^\pi(s)$ represents the (random) queue length of station n at time s under policy π.

The remainder of the paper is devoted to searching for optimal policies under these two criteria. In the latter case, we pose heuristics that perform very well under the throughput criterion while approximating the average cost case for any finite number of stations.

3 Throughput Optimality of General Scheduling Policies

In this section we show several general results on throughput optimality. In particular, we show that it is not possible to achieve a throughput larger than λ^* under very general conditions. Moreover, if the reconfigurable servers are such that $\mu_{r,k,\ell} = \gamma_{r,\ell}\mu_{r,k}$ for some strictly positive $\gamma_{r,\ell}$ and $\mu_{r,k}$, we show that if a scheduling policy satisfies a condition on the allocation when the queues are "large" it is throughput optimal. This service rate structure includes the important special cases where the reconfigurable servers have service rates that are either independent of the server or independent of the station. Since for most of this section we consider a fixed policy, we suppress the superscript "π".

We first introduce some ideas from [11]. Let $Q_k(t)$ be the queue length at station k at time t. Also, define $T_{k,\ell}(t)$ to be the amount of time that dedicated server ℓ at station k has been busy up until time t and $T_{r,k,\ell}(t)$ to be the amount of time that reconfigurable server ℓ has been busy at station k up until time t. Suppose $\{x_n, n \geq 0\}$ is a sequence of non-negative real numbers such that $\lim_{n\to\infty} x_n = \infty$. If we let a *fluid limit* be defined by

$$\begin{aligned}
\bar{Q}_k(t) &= \lim_{n\to\infty} x_n^{-1} Q_k(x_n t) \\
\bar{T}_{k,\ell}(t) &= \lim_{n\to\infty} x_n^{-1} T_{k,\ell}(x_n t) \\
\bar{T}_{r,k,\ell}(t) &= \lim_{n\to\infty} x_n^{-1} T_{r,k,\ell}(x_n t),
\end{aligned}$$

then as in Section 3.2 of [6], we have that any fluid limit satisfies the following set of conditions, called the *fluid model*:

$$\bar{Q}_1(t) = \bar{Q}_1(0) + \lambda t - \sum_{\ell=1}^{M_1} \mu_{1,\ell}\bar{T}_{1,\ell}(t) - \sum_{\ell=1}^{M_r} \mu_{r,1,\ell}\bar{T}_{r,1,\ell}(t), \tag{3.1}$$

$$\begin{aligned}
\bar{Q}_k(t) = \bar{Q}_k(0) + \sum_{\ell=1}^{M_{k-1}} \mu_{k-1,\ell}\bar{T}_{k-1,\ell}(t) - \sum_{\ell=1}^{M_k} \mu_{k,\ell}\bar{T}_{k,\ell}(t), \\
+ \sum_{\ell=1}^{M_r} \mu_{r,k-1,\ell}\bar{T}_{r,k-1,\ell}(t) - \sum_{\ell=1}^{M_r} \mu_{r,k,\ell}\bar{T}_{r,k,\ell}(t),\ 2 \leq k \leq N,
\end{aligned} \tag{3.2}$$

such that

$$\bar{Q}_k(t) \geq 0,\ 1 \leq k \leq N,$$

$$\bar{T}_{k,\ell}(0) = 0,\ 1 \leq \ell \leq M_k, 1 \leq k \leq N,$$

$$\bar{T}_{r,k,\ell}(0) = 0,\ 1 \leq \ell \leq M_r, 1 \leq k \leq N,$$

$$0 \leq \frac{d}{dt}\bar{T}_{k,\ell}(t) \leq a_{k,\ell}, 1 \leq \ell \leq M_k,\ 1 \leq k \leq N,$$

$$\frac{d}{dt}\bar{T}_{k,\ell}(t) = a_{k,\ell} \text{ whenever } \bar{Q}_k(t) > 0, 1 \leq \ell \leq M_k, 1 \leq k \leq N,$$

$$0 \leq \sum_{k=1}^{N} \frac{d}{dt}\bar{T}_{r,k,\ell}(t) \leq a_{r,\ell},\ 1 \leq \ell \leq M_r,\ \text{and}$$

$$\bar{T}_{k,\ell}(\cdot), \bar{T}_{r,k,\ell_r}(\cdot) \text{ are non-decreasing for } 1 \leq \ell \leq M_k, 1 \leq k \leq N, 1 \leq \ell_r \leq M_r.$$

The derivatives above exist almost everywhere, as $\bar{T}_{k,\ell}(t)$ and $\bar{T}_{r,k,\ell}(t)$ are Lipschitz (note that $|\bar{T}_{k,\ell}(t) - \bar{T}_{k,\ell}(s)| \leq |t - s|$). From this point on, derivatives will be understood to be taken on the condition that they exist. Note that the equations above do not completely specify $\bar{T}_{r,k,\ell}(t)$. Any additional conditions would refine the fluid model. We introduce the following notions of stability for the fluid model (see [11]). The fluid model is said to be *weakly stable* if for each solution to the fluid model with $\bar{Q}_k(0) = 0$, $\bar{Q}_k(t) = 0$ for $t \geq 0$. Conversely, the fluid model is said to be *weakly unstable* if there exists an $s > 0$ such that for *every* solution to the fluid model with $\bar{Q}_k(0) = 0$, $\sum_{k=1}^{N} \bar{Q}_k(s) \neq 0$.

The first result shows that any arrival rate greater than λ^* results in an unstable system, no matter what the scheduling policy.

Proposition 3.1 *If $\lambda > \lambda^*$, with probability one, $\sum_{k=1}^{N} Q_k(t) \to \infty$ as $t \to \infty$. Consequently, the long-run average cost is infinite.*

Proof. By contradiction. Suppose $\lambda > \lambda^*$, and that $\bar{Q}_k(s) = 0$ for all $1 \leq k \leq N$ and all $s > 0$. To prove the result we construct a solution (using λ) to $LP(1)$; this contradicts the definition of λ^*. Since the dedicated servers can only be busy while they are available, set $\bar{T}_{k,\ell}(s) = a_{k,\ell}s$. Suppose we set $\bar{T}_{r,k,\ell}(s) = \delta_{r,k,\ell} s$, (3.1) and (3.2) imply (after dividing through by s),

$$0 = \lambda - \sum_{\ell=1}^{M_1} \mu_{1,\ell} a_{1,\ell} - \sum_{\ell=1}^{M_r} \mu_{r,1,\ell} \delta_{r,1,\ell} \tag{3.3}$$

$$0 = \sum_{\ell=1}^{M_{k-1}} \mu_{k-1,\ell} a_{k-1,\ell} - \sum_{\ell=1}^{M_k} \mu_{k,\ell} a_{k,\ell} + \sum_{\ell=1}^{M_r} \mu_{r,k-1,\ell} \delta_{r,k-1,\ell} - \sum_{\ell=1}^{M_r} \mu_{r,k,\ell} \delta_{r,k,\ell} \tag{3.4}$$

with the constraint that

$$0 \leq \sum_{k=1}^{N} \delta_{r,k,\ell} \leq a_{r,\ell}. \tag{3.5}$$

The fact that (3.3)-(3.5) has a solution for $\lambda > \lambda^*$ contradicts the definition of λ^*, as (3.3)-(3.5) provide a feasible solution for $LP(1)$. Thus, the fluid model is weakly unstable and the result follows from Theorem 2.5.1 of [11]. ■

Now, let all of the reconfigurable servers be such that $\mu_{r,k,\ell} = \gamma_{r,\ell} \mu_{r,k}$ for all $\ell = 1, 2, \ldots, M_r$ and $k = 1, \ldots, N$. Consider the following assumption.

Assumption (LQ). $\sum_{k:\bar{Q}_k(t)>0} \frac{d}{dt} \bar{T}_{r,k,\ell}(t) = a_{r,\ell}$ for $1 \leq \ell \leq M_r$.

Assumption **(LQ)** implies that when some stations have long queues the reconfigurable servers are assigned to serve at one (or some) of the highly loaded stations. It may seem that this assumption is difficult to check, however, we will provide conditions that can be imposed on any policy that ensure that **(LQ)** holds. Before doing so, we show that a policy satisfying **(LQ)** achieves throughput optimality. To do this, we define a policy π to be *rate stable* (cf. [11]) if $\lim_{t\to\infty} D^\pi(t)/t = \lambda$ for $\lambda < \lambda^*$. The notion of rate stability is weaker than stability in the usual sense since no result about the existence of a stationary distribution is obtained. However, since λ^* is the highest throughput rate that can be hoped for, policies satisfying **(LQ)** are called *throughput optimal*.

Proposition 3.2 *If there exists strictly positive $\gamma_{r,\ell}$ and $\mu_{r,k}$ such that $\mu_{r,k,\ell} = \gamma_{r,\ell} \mu_{r,k}$ for all $\ell = 1, 2, \ldots, M_r$ and $k = 1, \ldots, N$, and the (reconfigurable) server scheduling policy π satisfies **(LQ)**, then π is rate stable.*

Proof. If we define $x_{r,k,\ell} = \delta_{r,k,\ell}\gamma_{r,\ell}$, $LP(1)$ may be rewritten as

$$
\begin{aligned}
&\max \quad \lambda \\
&s.t. \quad \sum_{\ell=1}^{M_k} a_{k,\ell}\mu_{k,\ell}/\mu_{r,k} + \sum_{\ell=1}^{M_r} x_{r,k,\ell} \geq \lambda/\mu_{r,k}, \quad \text{for all } k \in \{1,2,\dots,N\} \\
&\qquad \sum_{n=1}^{N} x_{r,n,\ell} \leq a_{r,\ell}/\gamma_{r,\ell}, \quad \text{for all } \ell \in \{1,2,...,M_r\} \\
&\qquad x_{r,k,\ell} \geq 0, \quad \text{for all } k \in \{0,1,...,N\} \text{ and } \ell \in \{1,2,...,M_r\}.
\end{aligned}
$$

Call this $LP(x)$. Now we turn to the fluid model. Differentiating (3.1) and (3.2) and evaluating at time zero yields

$$
\frac{d}{dt}\bar{Q}_1(t)|_{t=0} = \lambda - \sum_{\ell=1}^{M_1} \mu_{1,\ell}\frac{d}{dt}\bar{T}_{1,\ell}(t)|_{t=0} - \sum_{\ell=1}^{M_r} \gamma_{r,\ell}\mu_{r,1}\frac{d}{dt}\bar{T}_{r,1,\ell}(t)|_{t=0} \tag{3.6}
$$

$$
\begin{aligned}
\frac{d}{dt}\bar{Q}_k(t)|_{t=0} = &\sum_{\ell=1}^{M_{k-1}} \mu_{k-1,\ell}\frac{d}{dt}\bar{T}_{k-1,\ell}(t)|_{t=0} - \sum_{\ell=1}^{M_k} \mu_{k,\ell}\frac{d}{dt}\bar{T}_{k,\ell}(t)|_{t=0} \\
&+ \sum_{\ell=1}^{M_r} \gamma_{r,\ell}\mu_{r,k-1}\frac{d}{dt}\bar{T}_{r,k-1,\ell}(t)|_{t=0} - \sum_{\ell=1}^{M_r} \gamma_{r,\ell}\mu_{r,k}\frac{d}{dt}\bar{T}_{r,k,\ell}(t)|_{t=0}.
\end{aligned} \tag{3.7}
$$

Let U be the set of stations such that $\frac{d}{dt}\bar{Q}_k(t)|_{t=0} > 0$ for $k \in U$. We first show by contradiction that U must be empty. Assume U is nonempty. We have by **(LQ)** that $\sum_{k\in U} \frac{d}{dt}\bar{T}_{r,k,\ell}(t)|_{t=0} = a_{r,\ell}$. Also, for each fixed dedicated server ℓ, $\frac{d}{dt}\bar{T}_{k,\ell}(t)|_{t=0} = a_{k,\ell}$ for any $k \in U$. By the definition of U, for all $k \in U$ we have that the instantaneous arrival rate to station k must be greater than the instantaneous service rate (at time 0). Also, λ^* is an upper bound on the instantaneous arrival rate to station k (at time 0). In other words, we have for $k \in U$,

$$
\begin{aligned}
&\sum_{\ell=1}^{M_k} \mu_{k,\ell}\frac{d}{dt}\bar{T}_{k,\ell}(t)|_{t=0} + \sum_{\ell=1}^{M_r} \gamma_{r,\ell}\mu_{r,k}\frac{d}{dt}\bar{T}_{r,k,\ell}(t)|_{t=0} \\
&\qquad = \sum_{\ell=1}^{M_k} \mu_{k,\ell}a_{k,\ell} + \sum_{\ell=1}^{M_r} \gamma_{r,\ell}\mu_{r,k}\frac{d}{dt}\bar{T}_{r,k,\ell}(t)|_{t=0} < \lambda \leq \lambda^*.
\end{aligned} \tag{3.8}
$$

By the definition of λ^*, this combination is not possible as the inequality (3.8) cannot hold for all $k \in U$. Indeed if this were the case, letting $\gamma_{r,\ell}\frac{d}{dt}\bar{T}_{r,k,\ell}(t)|_{t=0}$ play the role of $x_{r,k,\ell}$ in $LP(x)$ would imply that the solution to $LP(x)$ would be less than λ^*. As a result, our assumption on U must be incorrect and U is empty. This means that for $k = 1, \ldots, N$, $\frac{d}{dt}\bar{Q}_k(t)|_{t=0} = 0$.

Since there is nothing special about time 0, we can use the above argument for any time t. It follows that $\bar{Q}_k(t) = 0$ for all t, and in turn the fluid model is weakly stable. The result now follows from Corollary 2.6.4. of [11]. ■

Note that **(LQ)** is not sufficient to guarantee throughput optimality for a more general rate structure, as seen by the following example. There are two stations, two reconfigurable servers, no fixed servers and the reconfigurable servers are not subject to failure. Let $\mu_{r,1,1} = \mu_{r,2,2} = 1$ and $\mu_{r,1,2} = \mu_{r,2,1} = 2$. Suppose that server i gives priority to queue i as much as possible while ensuring **(LQ)** holds. Then if for some $\varepsilon > 0$, the arrival rate is $1 + \varepsilon$, then server i will at some point always remain at queue i (as under the proposed policy, both queues are unstable and as a result **(LQ)** is always satisfied). Thus the maximum throughput in this case can be no more than one. However, it is easy to see by simply assigning server 1 permanently to station 2 and server 2 permanently to station 1, the maximal throughput is two.

It remains to comment on more natural conditions that guarantee **(LQ)**. This will help in constructing heuristics that are near throughput optimal while attempting to minimize holding costs. Consider the following two simple modifications.

Assumption (LQ1). There exists a threshold $L < \infty$ such that if $Q_k(t) \geq L$ for at least one k, then at time t the reconfigurable servers are all serving the station(s) with the longest queue.

Assumption (LQ2). There exists a threshold $L < \infty$ such that if $Q_k(t) \geq L$ for at least one k, then at time t the reconfigurable servers are all serving stations with queue length at least $\gamma Q_{max}(t)$ where $0 < \gamma \leq 1$ and $Q_{max}(t)$ is the largest queue length at time t. (Note that **(LQ1)** is **(LQ2)** with γ = 1.)

Under **(LQ1)** and by the fact that $\bar{Q}_k(t)$ is Lipschitz, if $\bar{Q}_k(t) > 0$ for at least one k, then there exists a subsequence $\{x_n^{-1}Q_k(x_n s), n \geq 1\}$, such that for some $M \geq 0$ and $h > 0$, $Q_k(x_m s) \geq L$ for all $m \geq M$ and $s \in [t, t + h]$. This means that for large m, the longest queue is served on

$[t, t+h]$. Taking limits implies that all of the reconfigurable servers are serving a queue where $\bar{Q}_k(s) > 0$ for $s \in [t, t+h]$. That is, the rate at which the busy time increases is a_r and **(LQ1)** implies **(LQ)**. A similar argument holds for **(LQ2)**.

4 Minimizing Average Holding Costs

In [6], the authors use a fluid model to show that a timed generalized round-robin policy is stable (in the sense of yielding a stationary distribution) when $\lambda < \lambda^*$. Moreover, they showed that any throughput rate less than λ^* can be achieved. In systems with only one server at each station, Theorem 4.1 of [12] shows that all moments of the queue lengths are finite when the fluid model is stable. Since all service times are exponentially distributed, according to [18], their proof carries over to systems with a finite number servers at each station. We conclude that if $\lambda < \lambda^*$,

$$\lim_{t\to\infty} \sum_{n=1}^{N} h_n \, \mathbb{E}_x^{\pi'} \, Q_n(t) < \infty, \tag{4.1}$$

where π' is a timed generalized round-robin policy as described in [6]. That is, there exists a timed generalized round-robin policy with finite average cost. Moreover, applying Theorem 11.1.1 of Puterman [20] yields that there exists a Markovian randomized policy, say $\tilde{\pi}'$, with the same average cost.

Assume now that *uniformization* [25] has been applied so that we consider the discrete time equivalent to the continuous time problem defined. The cost function for each state $x \in \mathbf{S}$ and reconfigurable allocation a (the action) is $c(x, a) := \sum_{n=1}^{N} h_n q_n$. Let $x_n \in \mathbf{S}$ be the state of the system after the n^{th} decision epoch and y_n be the action chosen under policy π. Define the $n-$stage expected total discounted cost for initial state x as follows

$$V_{n,\alpha}(\pi, x) := \mathbb{E}_x^{\pi} \Big[\sum_{k=0}^{n-1} \alpha^k c(x_k, y_k) \Big], \tag{4.2}$$

where $\alpha \in (0, 1]$ is the discount factor and $V_{0,\alpha} = 0$. Let $V_\alpha(\pi, x) := \lim_{n\to\infty} V_{n,\alpha}(\pi, x)$ denote the infinite horizon expected total discounted cost of policy π. The long-run average cost under π is now defined

$$g(\pi, x) = \limsup_{n\to\infty} \frac{V_{n,1}(\pi, x)}{n}. \tag{4.3}$$

A policy π^* is optimal under the respective criterion if $G(\pi^*, x) = \inf_{\pi\in\Pi} G(\pi, x)$ for all $x \in \mathbb{X}$, where $G = V_{n,\alpha},\ V_\alpha,$ or g. It should be clear that any non-idling policy generates a Markov chain such that all states communicate. The following result is now simple but important.

Proposition 4.1 *Suppose $\lambda < \lambda^*$ and consider the discrete-time equivalent to the continuous-time problem originally posed. There exists a Markov, randomized policy, $\tilde{\pi}$ such that $g(\tilde{\pi}) < \infty$.*

Proof. The fact that the Markov chain generated under any non-idling policy is uniformizable is trivial since the transition rate out of each state is bounded. Thus, the average cost under a fixed policy π in the continuous-time problem is the same as that in the discrete-time equivalent modulo a multiplicative constant. Applying (4.1) to Definition 2.2 yields the result. ■

The goal is to find a policy π^* that will minimize the long run average holding cost per unit time over an infinite planning horizon.

4.1 The Two-Station Model

Let $N = 2$ and let i and j represent the number of customers at stations 1 and 2, respectively. From this point, we assume that the service rate for each reconfigurable server is independent of the station, in other words $\mu_{r,k,\ell} \equiv \mu_{r,\ell}$ for all r and ℓ. Denote the server status (0 = failed, 1 = operational) of the ℓ^{th} dedicated server at stations 1 and 2 and the reconfigurable server by ζ_ℓ, η_ℓ and θ_ℓ, respectively. The (now simplified) state space is

$$\mathbb{X} = \{(i, j, s = (\zeta, \eta, \theta))|i, j \in \mathbb{Z}^+, \zeta = (\zeta_1, \zeta_2, ..., \zeta_{M_1}), \eta = (\eta_1, \eta_2, ..., \eta_{M_2}),$$

$$\theta = (\theta_1, \theta_2, ..., \theta_{M_r}), \zeta_\ell \in \{0, 1\}, \eta_\ell \in \{0, 1\}, \theta_\ell \in \{0, 1\}\},$$

Without loss of generality, assume the uniformization rate $\Psi := \lambda + \sum_{k,\ell}\{\mu_{k,\ell} + \alpha_{k,\ell} + \beta_{k,\ell}\} + \sum_\ell\{\mu_{r,\ell} + \alpha_{r,\ell} + \beta_{r,\ell}\} = 1$. Suppose h_1 (h_2) represents the holding cost rate per customer per unit time for station 1 (2). The cost function when in state $x = (i, j, s)$ and choosing the allocation action a is $c(x, a) = ih_1 + jh_2$.

In order to ease notation, we define two operators F and R to encode the failure and repair transitions. Thus, for failure transitions $F_\ell(\zeta) := (\zeta_1, \zeta_2, ..., \zeta_\ell = 0, ..., \zeta_{M_1})$ and repair transitions $R_\ell(\zeta) := (\zeta_1, \zeta_2, ..., \zeta_\ell = 1, ..., \zeta_{M_1})$. Furthermore, let $< x, y >= \sum x_i y_i$ and $\mathbf{1}$ be a vector of all ones. For any real number $a \in \mathbb{R}$, define $a^+ = \max\{0, a\}$; the positive part of a. Then, $min\{a, b\} = a - [a - b]^+$. Define the mapping $Hv(i, j, \zeta, \eta, \theta)$ from $\mathbb{X}$ to $\mathbb{R}$ as follows

$$
\begin{aligned}
Hv(i, j, \zeta, \eta, \theta) = &\sum_{\ell=1}^{M_r} \theta_\ell \mu_{r,\ell} \min\{v(i-1, j+1, \zeta, \eta, \theta), v(i, j-1, \zeta, \eta, \theta)\} \\
&+ u(i, j, \zeta, \eta, \theta) + w(i, j, \zeta, \eta, \theta) \\
=& < \theta, \mu_r > \big(v(i-1, j+1, \zeta, \eta, \theta) - [v(i-1, j+1, \zeta, \eta, \theta) - v(i, j-1, \zeta, \eta, \theta)]^+\big) \\
&+ u(i, j, \zeta, \eta, \theta) + w(i, j, \zeta, \eta, \theta),
\end{aligned} \tag{4.4}
$$

where $v(i, -1, \zeta, \eta, \theta) = v(i, 0, \zeta, \eta, \theta)$ and $v(-1, j, \zeta, \eta, \theta) = v(0, j-1, \zeta, \eta, \theta)$,

$$
\begin{aligned}
u(i, j, \zeta, \eta, \theta) = &\lambda v(i+1, j, \zeta, \eta, \theta) \\
&+ \sum_\ell \mu_{1,\ell} \zeta_\ell v(i-1, j+1, \zeta, \eta, \theta) + \sum_\ell \mu_{2,\ell} \eta_\ell v(i, j-1, \zeta, \eta, \theta) \\
&+ \Big(< \mathbf{1} - \zeta, \mu_1 > + < \mathbf{1} - \eta, \mu_2 > + < \mathbf{1} - \theta, \mu_r > \Big) v(i, j, \zeta, \eta, \theta),
\end{aligned}
$$

and

$$
\begin{aligned}
w(i, j, \zeta, \eta, \theta) = &\sum_\ell \alpha_{1,\ell} v(i, j, F_\ell(\zeta), \eta, \theta) + \sum_\ell \alpha_{2,\ell} v(i, j, \zeta, F_\ell(\eta), \theta) \\
&+ \sum_\ell \alpha_{r,\ell} v(i, j, \zeta, \eta, F_\ell(\theta)) + \sum_\ell \beta_{1,\ell} v(i, j, R_\ell(\zeta), \eta, \theta) \\
&+ \sum_\ell \beta_{2,\ell} v(i, j, \zeta, R_\ell(\eta), \theta) + \sum_\ell \beta_{r,\ell} v(i, j, \zeta, \eta, R_\ell(\theta)).
\end{aligned}
$$

Note that u encodes external arrivals, service completions of dedicated servers and dummy transitions while w encodes the server failures and repair transitions. The following are called the *finite horizon optimality equations* (FHOE), *discounted cost optimality equations* (DCOE) and the *average cost optimality equations* (ACOE), respectively.

$$v_{n,\alpha}(i, j, \zeta, \eta, \theta) = ih_1 + jh_2 + \alpha H v_{n-1,\alpha}(i, j, \zeta, \eta, \theta) \tag{4.5}$$

$$v_\alpha(i, j, \zeta, \eta, \theta) = ih_1 + jh_2 + \alpha H v_\alpha(i, j, \zeta, \eta, \theta) \tag{4.6}$$

$$g + h(i, j, \zeta, \eta, \theta) = ih_1 + jh_2 + Hh(i, j, \zeta, \eta, \theta) \tag{4.7}$$

One might note that the optimality *assume* that all of the reconfigurable servers are to be allocated to one station or the other; never divided between stations. The first result states that in each case the minimum in the optimality equations sufficient when dividing the servers among stations is possible.

Proposition 4.2 *The minimums in* (4.5)*,* (4.6) *and* (4.7) *are sufficient to solve the more general problem where servers can be divided among the two stations.*

Proof. Consider the infinite horizon discounted cost model and allow servers to be divided between stations. The DCOE become

$$v_\alpha(i,j,\zeta,\eta,\theta) = \min_{k_\ell\in\{0,1\},\ell=1,\dots M_r}\{ih_1 + jh_2 + \alpha\Big(\sum_{\ell=1}^{M_r}\theta_\ell\mu_{r,\ell}k_\ell v_\alpha(i-1,j+1,\zeta,\eta,\theta)$$

$$+\sum_{\ell=1}^{M_r}\theta_\ell\mu_{r,\ell}(1-k_\ell)v_\alpha(i,j-1,\zeta,\eta,\theta) + u(i,j,\zeta,\eta,\theta) + w(i,j,\zeta,\eta,\theta)\Big)\},$$

where w and u are as defined above and k_ℓ is the decision variable denoting where the ℓ^{th} server is assigned (to station 1 (1) or 2 (0)). A little algebra yields

$$v_\alpha(i,j,\zeta,\eta,\theta) = \min_{k_\ell\in\{0,1\},\ell=1,\dots M_r}\{\alpha\sum_{\ell=1}^{M_r}\theta_\ell\mu_{r,\ell}k_\ell v_\alpha(i-1,j+1,\zeta,\eta,\theta)$$

$$+\alpha\sum_{\ell=1}^{M_r}\theta_\ell\mu_{r,\ell}(1-k_\ell)v_\alpha(i,j-1,\zeta,\eta,\theta)\}$$

$$+ih_1 + jh_2 + \alpha u(i,j,\zeta,\eta,\theta) + \alpha w(i,j,\zeta,\eta,\theta)$$

$$= \min_{k_\ell\in\{0,1\},\ell=1,\dots M_r}\{\alpha\sum_{\ell=1}^{M_r}\theta_\ell\mu_{r,\ell}k_\ell[v_\alpha(i-1,j+1,\zeta,\eta,\theta) - v_\alpha(i,j-1,\zeta,\eta,\theta)]\}$$

$$+\alpha\sum_{\ell=1}^{M_r}\theta_\ell\mu_{r,\ell}v_\alpha(i,j-1,\zeta,\eta,\theta) + ih_1 + jh_2 + \alpha u(i,j,\zeta,\eta,\theta) + \alpha w(i,j,\zeta,\eta,\theta).$$

Now notice that if $v(i-1,j+1,\zeta,\eta,\theta) - v(i,j-1,\zeta,\eta,\theta) \geq (\leq)\ 0$ indicates whether all of the k_ℓ should be set to 0 or 1 (θ_ℓ and $\mu_{r,\ell}$ are non-negative for all ℓ). In other words, all servers should be assigned to one station or the other as desired. The other cases are similar. ■

It is well-known that a solution to (4.5) exists and satisfies $v_{n,\alpha} = V_{n,\alpha}$. In (4.6), if a solution v_α exists it is such that $v_\alpha = V_\alpha$. In (4.7), it is also well-known (at least when all states communicate) that if a solution, (g, h), to the ACOE exists then $g = g^*(x)$ for all $x \in \mathbb{X}$ and h is called the relative value function (see for example Puterman [20]). We next show the existence of solutions to the DCOE and ACOE under certain assumptions. We also discuss the convergence of the value iterates under both the discounted and average cost criteria. Note that this implies a method for approximating solutions. We then use value iteration to prove the existence of optimal transition monotone policies.

Since the holding costs are non-negative and linear and there are only two actions available for each state, Proposition 4.3 follows directly from Proposition 9.17 of [10].

Proposition 4.3 *For each $\alpha \in (0,1)$, the discounted cost optimality equations (4.6) have a unique solution v_α^* and $v_\alpha^* = \lim_{n\to\infty} v_{n,\alpha}$.*

Proposition 4.4 *Suppose that $\lambda < \lambda^*$ and $x \in \mathbb{X}$, the following hold:*

1. *The optimal average cost may be computed by*

$$g = \lim_{\alpha \uparrow 1} (1-\alpha) V_\alpha(x) \quad \text{or} \quad g = \lim_{m\to\infty} \frac{v_{m,1}(x)}{m}.$$

2. *Let $z \in \mathbb{X}$ be a distinguished state and $V_\alpha(z) < \infty$ for $\alpha \in (0,1)$. There exists a limit function $h(x)$ of $h_{\alpha_n}(x) := V_{\alpha_n}(x) - V_{\alpha_n}(z)$, where $\alpha_n \uparrow 1$ (see Definition 7.2.2 of [23]).*

3. *The average cost optimality equations (4.7) have the solution (g, h) where g and h are defined in parts 1 and 2.*

Proof: Recall from Proposition 4.1 that there exists a Markov randomized policy that has finite average cost, say $J_\pi < \infty$. Since the cost function is linear in the the queue lengths, note that $\{(i,j) \mid \; c((i,j),a) = ih_1 + jh_2 \leq J_\pi\}$ is a finite set. According to Proposition 4.3 of [22] the assumptions of Theorem 7.2.3 (including the postulated finiteness of $V_\alpha(z)$), Proposition 7.2.4 and Theorem 7.4.3 of [23] hold. Applying these theorems directly and noting that the irreducibility of the state space implies that all states are recurrent when $\lambda < \lambda^*$ yields the desired results. ■

4.2 Optimality of Transition Monotone Policies in Two Station Networks

In this section, we extend the results on transition monotone policies of [28] to systems with dedicated servers at station 2 and external arrivals under the discounted and average cost criteria. Throughout the section assume $M_1 = 0$. Note from (4.5) that it is optimal to allocate all available reconfigurable servers to station 1 (2) if $v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta) \leq\ (\geq)\, v_{m,\alpha}(i, j-1, \zeta, \eta, \theta)$.

Definition 4.5 *We say that a policy is transition monotone (cf. [14, 16]) if for any fixed state* $(i, j, \zeta, \eta, \theta)$*,* $v_{m,\alpha}(i, j-1, \zeta, \eta, \theta) \leq v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta)$ *implies* $v_{m,\alpha}(i, j+k-1, \zeta, \eta, \theta) \leq v_{m,\alpha}(i-1, j+k+1, \zeta, \eta, \theta)$ *for all* $k \geq 0$*. In other words, if it is optimal in state* $(i, j, \zeta, \eta, \theta)$ *to allocate the reconfigurable servers to station 2, it is optimal to allocate the reconfigurable servers to station 2 for all states* $(i, j', \zeta, \eta, \theta)$*, where* $j' \geq j$*.*

In the next theorem we show the existence of optimal transition monotone policies for each $m-$stage problem. We then apply Propositions 4.3 and 4.4 to get that the same property holds under the infinite horizon discounted or average cost criteria by taking the appropriate limits.

Theorem 4.6 *Suppose* $\lambda < \lambda^*$ *and* $M_1 = 0$ *(no dedicated server at station 1). For all* $(i, j, \zeta, \eta, \theta) \in \mathbb{X}$*,* $\alpha \in (0, 1]$ *and* $m \in \mathbb{Z}^+$*, the following holds:*

1. $v_{m,\alpha}(i, j, \zeta, \eta, \theta)$ *is non-decreasing in* j*;*

2. $v_{m,\alpha}(i-1, j+2, \zeta, \eta, \theta) - v_{m,\alpha}(i, j, \zeta, \eta, \theta) \geq v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta) - v_{m,\alpha}(i, j-1, \zeta, \eta, \theta)$.

Proof: By induction. The results hold trivially for $m = 0$. Assume that they hold for m and consider $m + 1$. To show that the first assertion holds, let $(i, j, \zeta, \eta, \theta) \in \mathbb{X}$. A little algebra yields

$$
\begin{aligned}
& v_{m+1,\alpha}(i, j+1, \zeta, \eta, \theta) - v_{m+1,\alpha}(i, j, \zeta, \eta, \theta) \\
& \quad = h_2 + \alpha < \theta, \mu_r > [\min\{v_{m,\alpha}(i-1, j+2, \zeta, \eta, \theta), v_{m,\alpha}(i, j, \zeta, \eta, \theta)\} \\
& \qquad - \min\{v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta), v_{m,\alpha}(i, j-1, \zeta, \eta, \theta)\}] \\
& \qquad + \alpha[u_{m,\alpha}(i, j+1, \zeta, \eta, \theta) - u_{m,\alpha}(i, j, \zeta, \eta, \theta)] \\
& \qquad + \alpha[w_{m,\alpha}(i, j+1, \zeta, \eta, \theta) - w_{m,\alpha}(i, j, \zeta, \eta, \theta)]. \qquad (4.8)
\end{aligned}
$$

Since $u_{m,\alpha}$ and $w_{m,\alpha}$ are positive linear combinations of $v_{m,\alpha}$, the induction hypothesis yields that they are non-decreasing in j. Moreover,

$$\min\{v_{m,\alpha}(i-1,j+2,\zeta,\eta,\theta), v_{m,\alpha}(i,j,\zeta,\eta,\theta)\}$$
$$\geq \min\{v_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta), v_{m,\alpha}(i,j-1,\zeta,\eta,\theta)\}$$

since the induction hypothesis yields $v_{m,\alpha}(i-1,j+2,\zeta,\eta,\theta) \geq v_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta)$ and $v_{m,\alpha}(i,j,\zeta,\eta,\theta) \geq v_{m,\alpha}(i,j-1,\zeta,\eta,\theta)$. This completes the proof of the first assertion.

To prove the second assertion, note

$$\begin{aligned}
&v_{m+1,\alpha}(i-1,j+2,\zeta,\eta,\theta) - v_{m,\alpha}(i,j,\zeta,\eta,\theta) \\
&\quad = 2h_2 - h_1 + \alpha < \theta, \mu_r > \big(\min\{v_{m,\alpha}(i-2,j+3,\zeta,\eta,\theta), v_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta)\} \\
&\qquad - \min\{v_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta), v_{m,\alpha}(i,j-1,\zeta,\eta,\theta)\}\big) \\
&\qquad + \alpha[u_{m,\alpha}(i-1,j+2,\zeta,\eta,\theta) - u_{m,\alpha}(i,j,\zeta,\eta,\theta)] \\
&\qquad + \alpha[w_{m,\alpha}(i-1,j+2,\zeta,\eta,\theta) - w_{m,\alpha}(i,j,\zeta,\eta,\theta)],
\end{aligned} \tag{4.9}$$

and

$$\begin{aligned}
&v_{m+1,\alpha}(i-1,j+1,\zeta,\eta,\theta) - v_{m+1,\alpha}(i,j-1,\zeta,\eta,\theta) \\
&\quad = 2h_2 - h_1 + \alpha < \theta, \mu_r > \big(\min\{v_{m,\alpha}(i-2,j+2,\zeta,\eta,\theta), v_{m,\alpha}(i-1,j,\zeta,\eta,\theta)\} \\
&\qquad - \min\{v_{m,\alpha}(i-1,j,\zeta,\eta,\theta), v_{m,\alpha}(i,j-2,\zeta,\eta,\theta)\}\big) \\
&\qquad + \alpha[u_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta) - u_{m,\alpha}(i,j-1,\zeta,\eta,\theta)] \\
&\qquad + \alpha[w_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta) - w_{m,\alpha}(i,j-1,\zeta,\eta,\theta)].
\end{aligned} \tag{4.10}$$

By the induction hypothesis, it is straightforward that $u_{m,\alpha}(i-1,j+2,\zeta,\eta,\theta) - u_{m,\alpha}(i,j,\zeta,\eta,\theta) \geq u_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta) - u_{m,\alpha}(i,j-1,\zeta,\eta,\theta)$ and $w_{m,\alpha}(i-1,j+2,\zeta,\eta,\theta) - w_{m,\alpha}(i,j,\zeta,\eta,\theta) \geq w_{m,\alpha}(i-1,j+1,\zeta,\eta,\theta) - w_{m,\alpha}(i,j-1,\zeta,\eta,\theta)$ for all $j > 1$ because $u_{m,\alpha}$ and $w_{m,\alpha}$ are positive linear combinations of v_α^m.

If $j = 1$ and there is a service completion at station 2, we need to show $v_{m,\alpha}(i-1, 2, \zeta, \eta, \theta) - v_{m,\alpha}(i, 0, \zeta, \eta, \theta) \geq v_{m,\alpha}(i-1, 1, \zeta, \eta, \theta) - v_{m,\alpha}(i, 0, \zeta, \eta, \theta)$ since they are part of the expansion of $u_{m,\alpha}(i-1, 3, \zeta, \eta, \theta) - u_{m,\alpha}(i, 1, \zeta, \eta, \theta)$ and $u_{m,\alpha}(i-1, 2, \zeta, \eta, \theta) - u_{m,\alpha}(i, 0, \zeta, \eta, \theta)$, respectively. However, this follows from the first assertion of the theorem (already proven).

Consider now the terms involving minimums:

$$\begin{aligned}
&\min\{v_{m,\alpha}(i-2, j+3, \zeta, \eta, \theta), v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta)\} \\
&- \min\{v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta), v_{m,\alpha}(i, j-1, \zeta, \eta, \theta)\} \\
&= v_{m,\alpha}(i-2, j+3, \zeta, \eta, \theta) - v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta) \\
&\quad - [v_{m,\alpha}(i-2, j+3, \zeta, \eta, \theta) - v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta)]^+ \\
&\quad + [v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta) - v_{m,\alpha}(i, j-1, \zeta, \eta, \theta)]^+ \qquad (4.11)
\end{aligned}$$

and

$$\begin{aligned}
&\min\{v_{m,\alpha}(i-2, j+2, \zeta, \eta, \theta), v_{m,\alpha}(i-1, j, \zeta, \eta, \theta)\} \\
&- \min\{v_{m,\alpha}(i-1, j, \zeta, \eta, \theta), v_{m,\alpha}(i, j-2, \zeta, \eta, \theta)\} \\
&= v_{m,\alpha}(i-2, j+2, \zeta, \eta, \theta) - v_{m,\alpha}(i-1, j, \zeta, \eta, \theta) \\
&\quad - [v_{m,\alpha}(i-2, j+2, \zeta, \eta, \theta) - v_{m,\alpha}(i-1, j, \zeta, \eta, \theta)]^+ \\
&\quad + [v_{m,\alpha}(i-1, j, \zeta, \eta, \theta) - v_{m,\alpha}(i, j-2, \zeta, \eta, \theta)]^+. \qquad (4.12)
\end{aligned}$$

Define $x^- = min\{0, x\}$ for all $x \in \mathbb{R}$ (note that this is **not** the negative part). Note that for $A, B, C, D \in \mathbb{R}$, $A \geq C$ and $B \geq D$, implies $A - A^+ + B^+ \geq C - C^+ + D^+$. Indeed $A - A^+ + B^+ \geq C - C^+ + B^+ \geq C - C^+ + D^+$.

Let $A = v_{m,\alpha}(i-2, j+3, \zeta, \eta, \theta) - v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta)$, $B = v_{m,\alpha}(i-1, j+1, \zeta, \eta, \theta) - v_{m,\alpha}(i, j-1, \zeta, \eta, \theta)$, $C = v_{m,\alpha}(i-2, j+2, \zeta, \eta, \theta) - v_{m,\alpha}(i-1, j, \zeta, \eta, \theta)$, $D = v_{m,\alpha}(i-1, j, \zeta, \eta, \theta) - v_{m,\alpha}(i, j-2, \zeta, \eta, \theta)$. The fact that $A \geq C$ and $B \geq D$ follows from the induction hypothesis. The previous argument implies $A - A^+ + B^+ \geq C - C^+ + D^+$. Thus the second assertion holds for stage $m + 1$ and the proof of the theorem is complete. ∎

This leads to the following structural result; the main theorem of this section.

Theorem 4.7 *Suppose $\lambda < \lambda^*$ and $M_1 = 0$ (no dedicated server at station 1), the following hold*

1. *There exists a transition monotone policy that is infinite horizon discounted cost optimal.*

2. *There exists a transition monotone policy that is average cost optimal.*

Proof. First note that by taking the limit as $m \to \infty$ in statement 2 of Theorem 4.6 we have

$$v_\alpha(i-1, j+2, \zeta, \eta, \theta) - v_\alpha(i, j, \zeta, \eta, \theta) \geq v_\alpha(i-1, j+1, \zeta, \eta, \theta) - v_\alpha(i, j-1, \zeta, \eta, \theta). \quad (4.13)$$

If we next take the limit along a subsequence $\alpha_n \uparrow 1$ in (4.13), statement 2 of Proposition 4.4 and (4.13) yield

$$h(i-1, j+2, \zeta, \eta, \theta) - h(i, j, \zeta, \eta, \theta) \geq h(i-1, j+1, \zeta, \eta, \theta) - h(i, j-1, \zeta, \eta, \theta), \quad (4.14)$$

where h satisfies the ACOE on recurrent states of the optimal policy.

Recall that it is infinite horizon discounted cost optimal to allocate all available reconfigurable servers to station 1 (2) in state $(i, j, \zeta, \eta, \theta)$ if $v_\alpha(i-1, j+1, \zeta, \eta, \theta) \leq (\geq) v_\alpha(i, j-1, \zeta, \eta, \theta)$. Thus, if it is optimal to allocate the reconfigurable servers to station 2 in state $(i, j, \zeta, \eta, \theta)$, then (4.13) implies it is optimal to allocate all of the resources to station 2 for all $(i, j+k, \zeta, \eta, \theta)$ where $k \geq 0$. Similarly for the average cost case. ∎

Theorem 4.7 implies the existence of a threshold function $\mathbf{L}(i, \zeta, \eta, \theta)$, where for any state $(i, j, \zeta, \eta, \theta) \in \mathbb{X}$, allocating all reconfigurable servers to the second station is optimal if and only if $j \geq \mathbf{L}(i, \zeta, \eta, \theta)$. This simplifies considerably the search for an optimal allocation policy (see Section 8.5.6 of [20]). Moreover, only the curve $\mathbf{L}$ needs to be stored in the lookup table as opposed to actions for every state.

We conclude this section with a remark on the system with dedicated servers at the upstream station. To complete the proof of Theorem 4.6 in systems with dedicated servers at station 1, the additional boundary condition would require $v_\alpha^m(0, j+2, \zeta, \eta, \theta) - v_\alpha^m(0, j+1, \zeta, \eta, \theta) \geq v_\alpha^m(0, j+1, \zeta, \eta, \theta) - v_\alpha^m(0, j, \zeta, \eta, \theta)$ for all j and m. That is to say $v_\alpha^m(0, j, \zeta, \eta, \theta)$ is convex in j. However, sufficient conditions for the convexity of $v_\alpha^m(0, j, \zeta, \eta, \theta)$ in j remain unclear. Consider the following example.

Example 4.8

Suppose there are two stations with two reconfigurable servers and two dedicated servers at each station. As an approximation, assume a fixed buffer capacity of 30 before each station and an external arrival rate $\lambda < 0.5\lambda^*$. Suppose $h_1 = 1$; $h_2 = 3$; $\mu_{1,1} = \mu_{1,2} = 7$; $\mu_{2,1} = \mu_{2,2} = 10$; $\alpha_1 = 0.4$; $\alpha_2 = 0.1$; $\beta_1 = 6$; $\beta_2 = 0.3$; $\mu_r = 5$; $\alpha_r = 0.1$; $\beta_r = 0.3$ and $\lambda = 7$. Let $i = 9$, $j = 10$, $\zeta = \{1, 1\}$, $\eta = \{0, 0\}$, and $\theta = \{0, 0\}$. Then $h(i, j+1, \zeta, \eta, \theta) - 2h(i, j, \zeta, \eta, \theta) + h(i, j-1, \zeta, \eta, \theta) \approx -0.006$. That is to say that h is not convex in j at $(i, j, \zeta, \eta, \theta)$.

We note the non-convexity exhibited in Example 4.8 is not pathological. Other examples in the literature can be found in [16] and [27]. Even within the example there were several other states that exhibited non-convexity. Of course this does not imply that optimal policies are not transition monotone. Quite the contrary, our numerical work suggests that there exists an optimal transition monotone policy when $M_1 > 0$.

5 Heuristic Allocation Policies

We have shown the existence of optimal transition monotone resource allocation policies in a two-station system. This simplifies the search for an optimal policy and the implementation of this policy for small problems. However, finding an average cost optimal policy (or even structural results) for an $N-$station system is still intractable. In this section, we use the two-station solution to develop an easily implementable heuristic. As we will see, it performs well when compared to several heuristics currently in practice.

Instead of solving the resource allocation problem for the whole system, our *Two-pairing* heuristics reduce the computation effort by looking at only two stations at a time. For simplicity, suppose we start with the first two stations and the optimal (two-station) policy is to allocate all of the resources to the first station. The heuristic is to follow this advice, allocate the resources to station 1 and stop. On the other hand, if the optimal policy allocates the resources to the second station, the heuristic considers an optimal policy for two-station model that consists of the second and third station. This process continues until we reach the last station. A formal description of these Two-pairing algorithms follows:

Two-pairing Upstream (TPU) [or *Two-pairing Downstream (TPD)*]: In a system with N stations:

- (1) Initialization.

a. For each positive integer $N > n \geq 1$, consider the two-station sub-system that has only the reconfigurable resources and stations n and $n+1$ of the original system. Assume that the sub-system and the original system have the same service, failure, repair and external arrival rates. Find the optimal resource allocation policy of each sub-system and store the optimal policy in a lookup table π^{n*}.

b. Set $n = 1$ [*for TPD: let* $n = N - 1$].

- (2) Station Evaluation. Suppose the current state $s = (q_1, q_2, ..., q_N, (m_{1,1}, ..., m_{1,M_1}), ..., (m_{N,1}, ..., m_{N,M_N}), (m_{r,1}, ..., m_{r,M_r})) \in \mathbf{S}$.

  a. Let $s' = (q_n, q_{n+1}, (m_{n,1}, ..., m_{n,M_n}), (m_{n+1,1}, ..., m_{n+1,M_{n+1}}), (m_{r,1}, ..., m_{r,M_r}))$. Find the optimal two-station action for state s' in the lookup table $\pi^{n*}(s')$.

  b. If the optimal two-station action in (2)a is to allocate the reconfigurable server at the upstream [*for TPD: downstream*] station, then go to step (3). Otherwise, go to step (2)c.

  c. If $n = N - 1$ [*for TPD: If* $n = 1$], then let $n = N$ [*for TPD: let* $n = 0$] and go to step (3). Otherwise, let $n = n + 1$ [*for TPD: let* $n = n - 1$] and go to step (2)a.

- (3) Allocation. Allocate the reconfigurable server to station n [*for TPD: station* $n + 1$].

One might observe that TPU and TPD choose the allocation of reconfigurable resources without considering non-adjacent stations. We next define a family of Two-pairing heuristics that alleviate this concern.

General Two-pairing heuristics: In a system with N stations:

- (1) Initialization.

  a. For each positive integer $n \in \{1, 2, ..., N\}$ and $n' \in \{n + 1, ..., N\}$ consider a two-station system that has only the reconfigurable resources and stations n and n' of the original system. Assume that the two-station system and the original system have the same service, failure, repair and external arrival rates. Find the optimal resource allocation policies of each two-station system and store the optimal policy in a lookup table $\pi^{n,n'}$.

  b. Let $\mathbf{L}$ be an $N \times N$ matrix and $L_{i,j}$ be the $(i, j)^{th}$ element of $\mathbf{L}$. Initialize $\mathbf{L} = 0$.

c. Let $n = 1$.

- (2) Station Evaluation. Suppose the current state $s = (q_1, q_2, ..., q_N, (m_{1,1}, ..., m_{1,M_1}), ..., (m_{N,1}, ..., m_{N,M_N}), (m_{r,1}, ..., m_{r,M_r})) \in \mathbf{S}$.

  a. For each $n' \in \{n+1, ..., N\}$, let $s' = (q_n, q_{n'}, (m_{n,1}, ..., m_{n,M_n}), (m_{n',1}, ..., m_{n',M_{n'}}), (m_{r,1}, ..., m_{r,M_r}))$. Find the optimal two-station action for state s' in the lookup table $\pi^{n,n'}$.

  b. If the suggested action in (2)a is to allocate the reconfigurable servers at the downstream station, let $L_{n,n'} = 0$ and $L_{n',n} = 1$. Otherwise, $L_{n,n'} = 1$ and $L_{n',n} = 0$.

  c. If $n = N - 1$, continue. Otherwise, $n = n + 1$ and go to step (2)a.

- (3) Station Weights. Step (2) yields pairwise comparisons between any two stations n and n'. Define $f_{n,n'}(h_n, h_{n'}, n'-n)$ to be a set of weight functions calculated from holdings costs h and the distance between n and n'. Let $Y_n = \sum_{n'=1}^{N} L_{n,n'} f_{n,n'}(h_n, h_{n'}, n'-n)$. Y_n is called the weight associated with station n.

- (4) Allocation. Allocate the reconfigurable server to a station with the heaviest weight.

The general two-pairing heuristics deserve more comment. The pairwise comparisons of the second step describe which stations dominate when considered as two-station models. This information is stored in the matrix $\mathbf{L}$. Each station is then weighted based on the holding cost rates and the distance between two stations by the (decision-maker specified) function f. Of course using different weight functions causes the heuristic family to generate different policies. Moreover, it should be clear that TPU and TPD are special cases of general two-pairing heuristics. In Section 6.3 we examine several choices of the function f. In the next section we provide the details of our numerical study comparing each of the heuristics along with several from the literature.

6 Simulation Description and Results

In this section, we analyze the performance of the heuristic policies in a four stage serial line with the buffer capacity before each station fixed at 30. Each station is equipped with two identical dedicated servers. There are two additional reconfigurable servers that can be allocated to any station without setup time or costs. In such systems, 20 different parameters are required to describe the model.

If for each of the twenty parameters required to describe the system we considered two values, a complete design would require $2^{20} > 10^6$ combinations. For each combination, we are required to solve several Markov decision processes with computational complexity of $2|S|^2$ in each iteration ([20], section 4.5, p.93), where $|S|$ denotes the number of different states. In each two station subsystem of this model, $|S|$ equals to $31^2 \times 2^6 = 61,504$. This design is intractable. Thus, in order to efficiently conduct a simulation study, we use randomly generated system parameters. We also restrict the range of each parameter to ensure that reasonable systems are chosen for simulation evaluation. Let $U(0,1)$ be a uniform (0,1) random number. The algorithm that is used to generate system parameters works as follows:

- h_1, h_2, h_3, h_4: randomly choose from {1,2,3,4}

- $\mu_1, \mu_2, \mu_3, \mu_4, \mu_r$: randomly choose from {5,6,7,8,9,10}. This ensures that the system is not extremely asymmetric, which is reasonable for most manufacturing systems. Meanwhile, sufficiently large arrival and service rates also help to ensure that the simulation collects enough data within a fixed time period.

- $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_r$: choose from $0.5 \cdot U(0,1)$. This assumption makes service rates $\gg$ failure rates; a common characteristic in most systems.

- $\beta_1, \beta_2, \beta_3, \beta_4, \beta_r$: randomly choose $\gamma_1, \gamma_2, \gamma_3, \gamma_4$, and γ_r from $\{3, 7, 11, 15, 19\}$, and let $\beta_i = \alpha_i \times \gamma_i$. This forces a server's expected failure time to be equal to $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{12}$, $\frac{1}{16}$, or $\frac{1}{20}$ of the total expected time from repair to repair. In other words, the expected long term average server reliability is in $(0.75, 0.95)$.

- λ: Use the previously generated parameters to obtain the optimal solution λ^* of $LP(1)$. Then, let $\lambda = (0.6 + 0.4 \cdot U(0,1))\lambda^*$. The generated arrival rate allows us to focus on systems with medium to high utilization ; which is realistic in many manufacturing systems.

In each simulation run, the first 500 units of time are deemed the "warm-up" period; no statistics are collected. After the warm-up period, system statistics are collected for 5,000 units of time. According to the simulation run length and the system parameter generation algorithm there are on average between 28,125 and 128,760 external arrivals. (However, in 91% of the generated systems, there were between 50,000 and 100,000 external arrivals.)

6.1 Comparison with Several Heuristics in Practice

Before introducing the simulation results, we first list the policies that we compared to TPU and TPD:

- Upstream First Heuristic (UPF): Allocate the reconfigurable resources to the furthest upstream station which is non-empty. This heuristic is similar in spirit to the "bucket brigade" policies in Bartholdi et. al [8] and [9], except we include the presence of dedicated servers. Bartholdi et. al [8] show that this heuristic has higher throughput than other current industry practices. Of course they do not consider dedicated servers or server reliability.

- Downstream First Heuristic (DTF): Allocate the reconfigurable resources to the furthest downstream station that is non-empty. This heuristic is similar to the expedite heuristic in [26], except for the presence of dedicated servers.

- Trouble Shooting Heuristic (TRS): Allocate the reconfigurable servers to the furthest downstream station when there is no machine failure in the system and that station is non-empty. If there is a machine failure in a station and the station is not empty, use the reconfigurable servers at that station to compensate for the capacity loss. If there are several servers failed in the system, use the reconfigurable servers at the downstream stations when the downstream station with the failed server is not empty [15].

- Timed Generalized Round-Robin Policy 1 (RR1): Reconfigurable servers stay at station n for d_n time units and then move to the next station. When the buffer is empty at station n before the d_n time units are finished, the reconfigurable servers move to the next station to prevent idling. In the timed generalized round-robin policy 1, we assume that $\sum_{n=1}^{N} d_n = \frac{10}{\mu_r}$.[1] As previously noted, in systems with reliability considerations, optimal throughput

[1]In [5], a small value of $\sum_{n=1}^{N} d_n$ is suggested to reduce the average holding costs. However, an extremely small $\sum_{n=1}^{N} d_n$ is not practical in many manufacturing systems. We mention here that $\sum_{n=1}^{N} d_n = \frac{5}{\mu_r}$ does not improve the average holding costs much over $\sum_{n=1}^{N} d_n = \frac{10}{\mu_r}$. On average the difference was less than 1%.

rates can be achieved by timed generalized round-robin policies when buffers have infinite capacity [6].

- Timed Generalized Round-Robin Policy 2 (RR2): This policy is the same as the Timed Generalized Round-Robin Policy 1, except that we assume $\sum_{n=1}^{N} d_n = \frac{20}{\mu_r}$.

- Two-Pairing Downstream (TPD): Since TPD is similar to TPU and can be simulated without additional calculation, we include TPD in our study.

In each simulation, we use common random numbers to reduce the variance of simulation results between policies. For each policy, we use the same random number sequences to generate the next server failure and repair times. We also use common random number sequences to generate the service time of each job. Using common random number sequences should provide better accuracy in comparing different policies.

For each interval of utilizations ρ, $[0.6, 0.7)$, $[0.7, 0.8)$, $[0.8, 0.9)$, and $[0.9, 1.0)$, we simulate 100 different sets of system parameters. For each set of system parameters, we run sufficient replications such that 95% confidence intervals have width less than 5% of the corresponding sample mean. This required 30 replications for 389 systems and 50 replications for the remaining 11. In order to facilitate understanding of the simulation results, we list a complete set of simulation outputs for one example system.

Example 6.1

- Input Parameters:

  – Holding cost rates at each station: $(h_1, h_2, h_3, h_4) = (2, 3, 4, 2)$,

  – Service rates of dedicated servers at each station: $(\mu_1, \mu_2, \mu_3, \mu_4) = (10, 5, 9, 8)$,

  – Failure rates of dedicated servers at each station: $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.450, 0.492, 0.280, 0.023)$,

  – Repair rates of dedicated servers at each station: $(\beta_1, \beta_2, \beta_3, \beta_4) = (4.953, 7.378, 0.839, 0.248)$,

  – Service, failure, and repair rates of the reconfigurable servers: $(\mu_r, \alpha_r, \beta_r) = (9, 0.443, 3.103)$,

  – Arrival rate: λ = 16.18.

- Simulation results of the first replication:

$\lambda^* = 17.764$, $\rho = \lambda/\lambda^* = 0.911$,

Average Holding Costs: TPU = 62.20, TPD = 163.31, UPF = 231.81, DTF = 205.68, TRS =148.14, RR1 = 139.66, RR2 = 145.75,

Average Throughput Rates of each policy: TPU = 16.11, TPD = 14.59, UPF = 14.24, DTF = 14.89, TRS = 10.86, RR1 = 15.71, RR2 = 15.59,

Average Buffer level before each station: see Table 1.

BUFFER LOCATION	TPU	TPD	UPF	DTF	TRS	RR1	RR2
Before Station 1	5.88	16.42	18.39	16.76	26.93	10.46	11.24
Before Station 2	10.36	21.33	22.35	23.07	27.90	16.14	16.95
Before Station 3	2.50	4.62	23.63	13.83	1.92	12.63	13.08
Before Station 4	4.67	23.98	16.76	23.80	1.46	9.91	10.05

Table 1: Average buffer level under different policies

- Average of the 30 replications with 95% confidence intervals:

  Average holding costs of each policy: TPU = 62.07 $\pm$ 0.85 , TPD = 163.08 $\pm$ 1.63, UPF = 230.11 $\pm$ 1.13, DTF = 230.94 $\pm$ 2.07, TRS = 148.11 $\pm$ 0.15, RR1 = 138.12 $\pm$ 2.02, RR2 = 145.60 $\pm$ 1.74,

  Average throughput rates of each policy: TPU = 16.108 $\pm$ 0.019 , TPD = 14.590 $\pm$ 0.057, UPF = 14.244 $\pm$ 0.050, DTF = 14.879 $\pm$ 0.042, TRS = 10.862 $\pm$ 0.015, RR1 = 15.707 $\pm$ 0.038, RR2 = 15.586 $\pm$ 0.037,

  Normalized average performance (Average holding costs and throughput rates are normalized to the results of TPU): Table 2.

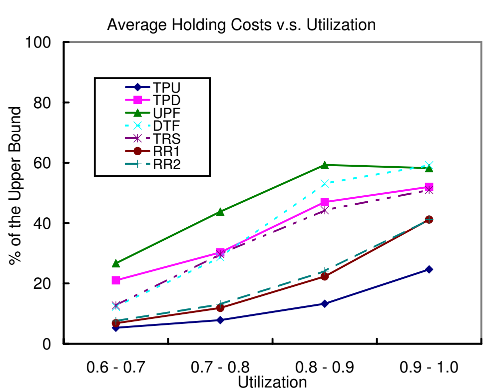
	TPU	TPD	UPF	DTF	TRS	RR1	RR2
Normalized Average Holding Costs	1	2.63	3.71	3.29	2.39	2.23	2.35
Normalized Average Throughput Rates	1	0.91	0.88	0.92	0.67	0.98	0.97

Table 2: Normalized average performance of the example system

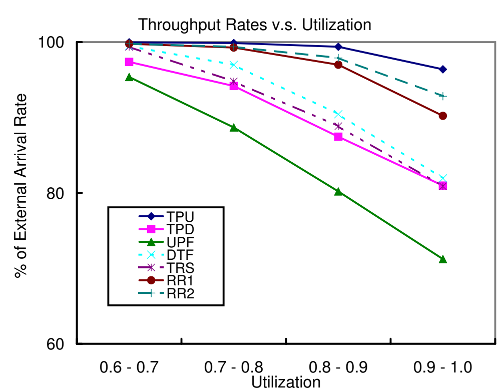
From the previous example, we make several observations. First, the TPU heuristic can detect potential starvation (and blocking) and reallocate WIP to prevent it. WIP reallocation reduces the probability of starving or blocking and improves average system throughput rates. Second, although the RR1 and RR2 heuristics do not perform well under the average holding cost criterion, they can still prevent starving and blocking and perform close to the TPU heuristic in average throughput. Third, intuition does not always work in resource allocation problems. An example is the throughput rate of UPF. Since external arrivals are lost when the first buffer is full, one might think that the UPF heuristic should reduce the number of lost customers and improve the throughput rates. However, this is not the case. We believe this is due to the fact that the UPF heuristic increases the probability of blocking. This also serves to increase the average holding costs.

A summary of the simulation results of the 400 randomly generated systems are presented in several tables/figures. In Figure 2(a), we plot the average holding costs as a percentage of the upper bound $U_c = 30 \cdot \sum_{k=1}^{4} h_k$. When $\rho > 0.8$, the average holding costs of policies TPD, UPF, DTF, and TRS did not grow with ρ. This is intuitive since blocking and starving occurs more frequently at higher utilizations. On the contrary, TPU and the round-robin policies lose fewer arrivals and have higher throughput rates when ρ is increased. The increase in throughput rates causes the increase in holding costs.

The TPU heuristic reduced holding costs by more than 25% in all categories. Since the maximum holding costs are bounded above by U_c, we note that the impact of better resource allocation policies is less significant in normalized costs when $\rho > 0.9$. Compared to DTF, TRS, RR1, and RR2, TPU performs best in normalized holding costs when the system utilization is between 0.7 and 0.9.



(a) Bounded holding costs v. utilization



(b) Throughput rates / arrival rates v. Utilization

Figure 2: Varying Utilization (Exponential Service Times)

We also noticed that RR1 performs better than RR2 in average normalized holding costs. This is similar to the results in Section 5.3 of [5]. This result also suggests that minimizing the time of each cycle in a timed generalized round-robin policy yields lower holding costs.

In Figure 2(a) the improvement of TPU for the average holding cost is not as significant when utilization is greater than 0.9. Figure 2(b) shows that the improvement for the average throughput rate is more significant when the utilization is higher. In some cases (although not on average), the improvement is more than 30%. The increase in throughput rate boosts the utilization of all stations and servers.

In Figure 2(b) the average throughput rate of each policy is plotted as a percentage of the external arrival rate. Categorized by utilization, one should note that TPU effectively prevents loss of external arrivals when utilization is high.

According to [6], timed generalized round-robin policies can achieve optimal throughput rates when the buffers have infinite capacity. Our simulation study confirms similar results in finite buffer systems. The throughput rates of timed generalized round-robin policies are only marginally less than TPU in most cases when the time between two "rounds" is small. One should also note when the utilization is in the region $(.6, .7)$, the difference in the throughput between TPU and the first round-robin policy is negligible. In this region, the finite state approximation closely approximates the infinite state system.

So far, we have studied the average performance of each policy. In addition to the average improvement over the 400 simulated systems, we are also interested in whether TPU is a superior policy in each individual system. We use the statistical methods in Chapter 12 of [7] to study whether TPU is statistically better in each evaluated system. With 95% confidence (p-value $< .05$) the summary of the results is displayed (as percentages) in Table 3.

According to Table 3 TPU does at least as well as any other single policy in more than 97% of the tested systems. When TPU is not superior to other policies, we can usually find strong intuition. For example, when $h_1 = 1, h_2 = 1, h_3 = 1, h_4 = 4$, TPU may allocate the reconfigurable server to the first three stations and ignore the significantly higher holding cost at station 4.

The throughput rates of each evaluated system are also individually analyzed. With 95% confidence (p-value $< .05$), we test whether TPU has higher average throughput rates than the other five policies. The results, categorized by system utilization, are displayed (as percentages) in Table 4.

COMPARE TO THE TWO-PAIRING DOWNSTREAM	UTILIZATION	TPD	UPF	DTF	TRS	RR1	RR2
# of systems where TPU is superior	0.9-1.0	93	99	95	95	93	94
	0.8-0.9	96	100	100	98	88	91
	0.7-0.8	92	100	95	99	92	93
	0.6-0.7	92	100	94	92	85	91
# of systems where TPU is inferior	0.9-1.0	7	1	5	5	6	6
	0.8-0.9	3	0	0	1	10	9
	0.7-0.8	7	0	5	0	7	7
	0.6-0.7	8	0	6	8	14	6
# of systems where the difference are indistinguishable	0.9-1.0	0	0	0	0	1	0
	0.8-0.9	1	0	0	1	2	0
	0.7-0.8	1	0	0	1	1	0
	0.6-0.7	0	0	0	0	1	3

Table 3: Significance of holding cost difference in individual systems

COMPARE TO THE TWO-PAIRING UPSTREAM	UTILIZATION	TPD	UPF	DTF	TRS	RR1	RR2
# of systems where TPU is superior	0.9-1.0	95	100	99	100	90	94
	0.8-0.9	97	100	99	100	92	93
	0.7-0.8	90	100	91	100	87	89
	0.6-0.7	76	97	70	71	59	63
# of systems where TPU is inferior	0.9-1.0	5	0	1	0	9	6
	0.8-0.9	2	0	0	0	7	5
	0.7-0.8	1	0	1	0	2	3
	0.6-0.7	1	0	1	0	0	0
# of systems where the difference are indistinguishable	0.9-1.0	0	0	0	0	1	0
	0.8-0.9	1	0	1	0	1	2
	0.7-0.8	9	0	8	0	11	8
	0.6-0.7	23	3	29	29	41	37

Table 4: Significance of throughput rate difference in individual systems

For any studied range of system utilization the TPU heuristic is at least as good as any other single policy in more than 91% of the tested systems. When system utilization is between 0.6 and 0.7, almost identical throughput rates are observed under the TPU, DTF, TRS, RR1, and RR2 policies (Figure 2(b)). The buffers are rarely full under those policies and only a few arrivals are lost.

We conclude this section with the observation that in spite of the fact that the TPU heuristic is developed under the assumption that servers can collaborate (work together on the same job), even in cases when this assumption does not hold it performs quite well. Suppose now that servers

SERVER COLLABORATION	TPU	TPD	UPF	DTF	TRS	RR1	RR2
Non-collaborative	1	2.206	2.798	2.657	1.978	1.349	1.645
Collaborative	1	3.464	4.301	3.102	3.014	1.530	1.817

Table 5: Average normalized inventory holding costs: with and without server collaboration

SERVER COLLABORATION	TPU	TPD	UPF	DTF	TRS	RR1	RR2
Non-collaborative	1	0.897	0.859	0.878	0.943	0.998	0.988
Collaborative	1	0.923	0.863	0.941	0.921	0.990	0.979

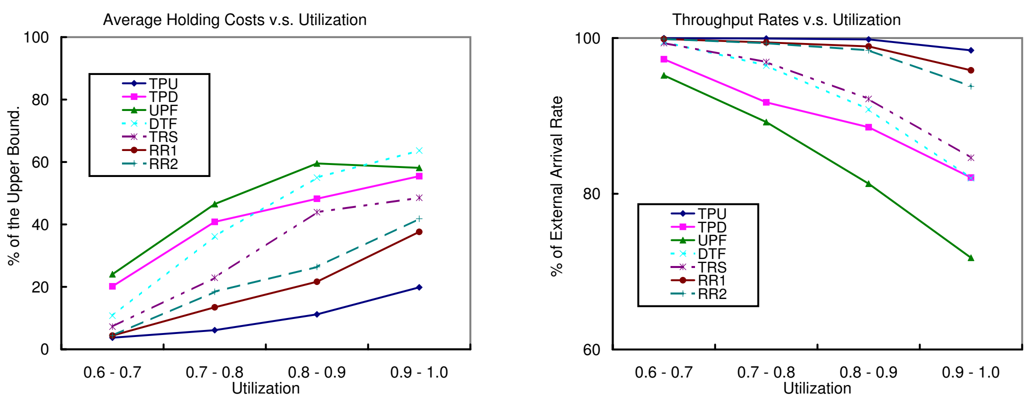
Table 6: Average normalized throughput rates: with and without server collaboration

cannot collaborate. The service rate at each station is the sum of the service rates of (available) servers at that station that currently have work that can be assigned to them (note that dedicated servers are assigned to jobs first). The average normalized holding costs and throughput rates of 100 systems are provided in Table 5 and Table 6. Comparing to systems where servers can collaborate, the performance improvement of TPU is less significant but still outperforms all other heuristics by at least 35% in average holding costs. Among the 100 randomly generated systems, TPU has the lowest average holding cost rates among all heuristics in 78 systems and highest throughput rate in 63 systems. This result confirms the effectiveness of TPU when the collaboration assumption is not valid.

6.2 Generally Distributed Service Times

In this section, we show the effectiveness of the two-pairing heuristic in systems without the memoryless property. Since only the exponential distribution has the memoryless property, finding optimal resource allocation policies using MDPs is not practical in many systems. A typical way to alleviate this problem is to apply results from Markovian systems to non-Markovian systems. For example, in inventory control problems, the (s, S) policy that is derived using MDPs, has been successfully applied to many systems that have non-Poisson or non-discrete customer demand processes. Similarly, we apply our two-pairing heuristics (developed under the exponential service time assumption) in systems with other service time distributions. Our detailed numerical studies show similar performance improvement.

We begin by assuming that the service time distribution is deterministic. In Figure 3(a), the average holding costs in 400 (newly generated) systems for each policy are plotted as a percentage of the upper bound $U_c = 30 \cdot \sum_{k=1}^{4} h_k$. Comparing to Figure 2(a), we see that TPU performs equally well under deterministic service times. In the high utilization region ($\rho > 0.8$), compared to any other policies, TPU reduces the average normalized holding cost by at least 42%. When the utilization is low ($0.6 < \rho < 0.7$), the average normalized holding cost of RR1 is 18.6% more than that of TPU, which is a 10% improvement from that of the exponential case.



(a) Bounded holding costs v. system utilization (b) Throughput rates / arrival rates v. Utilization

Figure 3: Varying Utilization (Deterministic Service Times)

If we compare Figure 2(a) with Figure 3(a), we find that the improvement in holding costs is almost unchanged under deterministic service times. Even when the utilization is high ($\rho > 0.9$), the average buffer levels of TPU are still only 20% of the buffer capacity. Because buffers are relatively empty, TPU should not be sensitive to manufacturing variations such as demand spikes or machine failures. This ability to tolerate variance contributes to the increased average throughput rates plotted as a percentage of the external arrival rate in Figure 3(b).

As stated above, reduced buffer levels increase average throughput rates. When service times are deterministic, the TPU heuristic performs equally well in terms of average throughput rate. The improvement in average normalized throughput ranges from 2.6% to 27.1%. This range of improvement in average throughput rates is almost identical to that of systems with exponentially distributed service times.

To verify the performance improvement of TPU under service time distributions with greater variance, we conducted a smaller simulation study of 100 randomly generated systems with $0.6 \leq \rho \leq 1.0$. For each of the 100 randomly generated systems, we let the service time distribution be first the gamma distribution with $\beta = 3$ and second the truncated normal distribution with $\sigma = 0.5$.

The truncated normal distribution is made symmetric. For example, if $1/\mu$ is the mean service time, a random variable between 0 and $2/\mu$ is accepted by the random-variate generator.

SERVICE TIME DISTRIBUTION	TPU	TPD	UPF	DTF	TRS	RR1	RR2
Gamma	1	3.682	4.885	3.249	2.926	1.496	1.632
Truncated normal	1	4.262	5.11	3.879	3.285	1.547	1.68

Table 7: Average normalized inventory holding costs in the 100 systems

SERVICE TIME DISTRIBUTION	TPU	TPD	UPF	DTF	TRS	RR1	RR2
Gamma	1	0.924	0.846	0.936	0.926	0.991	0.983
Truncated normal	1	0.907	0.832	0.922	0.911	0.993	0.980

Table 8: Average normalized throughput rates in the 100 systems

In Tables 7 and 8, the average normalized holding costs and throughput rates of the 100 systems are provided. Note that TPU is still the best heuristic under both perfomance indices. When the service times follow the gamma distribution, in 87 systems (out of 100), TPU has the lowest average inventory holding costs; in 98 systems, TPU has the highest throughput rates. When the service time distribution is truncated normal, TPU is the top performing heuristic in average holding costs and throughput rates in 83 and 96 systems, respectively. This result confirms that the relative performance of TPU is insensitive to the choice of the service time distribution.

6.3 Comparison Between TPU and Other General Two-pairing Heuristics

In this section, we use simulation to compare TPU, TPD, and several weight functions in the two-pairing heuristic family. As mentioned in Section 5, different weight functions $f_{n,n'}$ for the general two-pairing heuristic generate different policies. We consider 6 weight functions. The heuristic policies generated are compared with TPU and TPD in 400 (newly generated) systems. The weight functions are defined as follows (HF stands for "heuristic function"):

- HF1: $f_{n,n'}(h_n, h'_n, n'-n) = 1$ for all $n, n', h_n, h_{n'}$.
- HF2: $f_{n,n'}(h_n, h'_n, n'-n) = 1_{\{n'-n\leq 1\}} + 0.6 \times 1_{\{n'-n>1\}}$ for all $n, n', h_n, h_{n'}$ (where $1_{\{.\}}$ is the indicator function). HF2 gives higher weight to adjacent stations.
- HF3: $f_{n,n'}(h_n, h'_n, n'-n) = h_n$, for all $n, n', h_n, h_{n'}$.

- HF4: $f_{n,n'}(h_n, h'_n, n'-n) = h_n \times 1_{\{n'-n\leq 1\}} + 0.6h_n \times 1_{\{n'-n>1\}}$ for all $n, n', h_n, h_{n'}$. HF4 gives higher weight to adjacent stations (based on the holding costs).

- HF5: $f_{n,n'}(h_n, h'_n, n'-n) = 10^n$ for all $n, n', h_n, h_{n'}$. HF5 gives higher weight to downstream stations.

- HF6: $f_{n,n'}(h_n, h'_n, n'-n) = 10^{-n}$ for all $n, n', h_n, h_{n'}$. HF6 gives higher weight to upstream stations.

For brevity, the details of each simulated system are not listed. Normalized (to TPU) average holding cost rates and throughput rates (the total number of service completions at the last station divided by the length of the simulation (5,000 after the 500 warm-up period)), are summarized in Table 9. In addition, we give the number of systems that yield holding costs or throughput rates superior to TPU.

	TPU	TPD	HF1	HF2	HF3	HF4	HF5	HF6
Average Holding Costs (Normalized to TPU)	1	3.16455	3.38441	3.290828	2.763799	2.760721	2.844213	4.990953
Average Throughput Rate (Normalized to TPU)	1	0.923139	0.859389	0.879412	0.887469	0.909762	0.857001	0.860244
Average Holding Costs < TPU	-	1	1	4	2	5	1	0
Average Holding Costs < TPD	29	-	9	8	12	12	17	2
Throughput Rate > TPU	-	3	3	2	2	4	0	2
Throughput Rate > TPD	27	-	7	8	7	8	7	4

Table 9: Comparison Between Heuristics

Note that TPU is better than each of the tested heuristics in terms of average throughput rates. Compared to any of the other policies, the average throughput rate of TPU is improved between 1% to 20%. However, compared to TPU, HF1 and HF2 reduce average holding cost by 7% and 13%, respectively. The general two-pairing idea was not fully explored, we chose what we believe to be reasonable weight functions. For sets of system parameters where TPU or TPD do not perform well, we believe that identifying appropriate two-pairing heuristics is a promising direction for future research.

7 Conclusions

In this paper we have studied the optimal allocation of reconfigurable resources in a serial line with reliability considerations. When the number of stations is greater than 2, computing an optimal policy is computationally intractable. This is due to the well-known "curse of dimensionality" that limits the applicability of Markov decision processes. To alleviate this problem, we introduced a class of heuristics based on the two-station model that performs well under the average holding cost criterion. Somewhat surprisingly, the same class of policies performs well under the average throughput criterion despite the fact that the policies are developed for average holding costs.

In the special case when there are no dedicated servers at the first station, we show the existence of optimal transition monotone policies in the two-station model. Since several MDPs must be solved to implement our heuristics, this stands to reduce the computational requirements by allowing the decision-maker to restrict attention to a smaller set of policies.

We note that the numerical study was performed on a finite state system. However, the conditions that guarantee throughput optimality in Section 3 are for the infinite buffer system. The heuristics discussed can be implemented in an infinite buffer system. If the decision-maker would like to guarantee throughput optimality, the heuristic can be implemented when the buffer levels are not "too high" and could serve at the longest queue (recall **(LQ1)**) when the queue lengths grow. In the systems studied in the numerical study, setting $L > 30$ would suffice.

Acknowledgment

This research was supported in part by the NSF Engineering Research Center for Reconfigurable Manufacturing Systems part of the Engineering Research Centers Program of the National Science Foundation under NSF Award Number EEC 95-29125. The work of the 2nd author is supported by the Natural Sciences and Engineering Research Council of Canada. The research of the 3rd author is supported in part by the NSF grant DMI-0132811.

References

[1] H. Ahn, I. Duenyas, and M. E. Lewis. The optimal control of a two-stage tandem queueing system with flexible servers. *Probability in the Engineering and Informational Sciences*, 16(4):453–469, 2002. 2

[2] H. Ahn, I. Duenyas, and R. Zhang. Optimal stochastic scheduling of a two-stage tandem queue with parallel servers. *Advances in Applied Probability*, 31:1095–1117, 1999. 1, 2

[3] S. Andradóttir and H. Ayhan. Throughput maximization for tandem lines with two stations and flexible servers. *Operations Research*, 53(3):516–531, 2005. 2

[4] S. Andradóttir, H. Ayhan, and D. G. Down. Server assignment policies for maximizing the steady-state throghput of finite queueing sytems. *Management Science*, 47:1421–1439, 2001. 2

[5] S. Andradóttir, H. Ayhan, and D. G. Down. Dynamic server allocation for queueing networks with flexible servers. *Operations Research*, 51(6):952–968, 2003. 24, 28

[6] S. Andradóttir, H. Ayhan, and D. G. Down. Compensating for failures with flexible servers. Submitted, 2005. 1, 2, 3, 4, 6, 11, 25, 28

[7] J. Banks, J. Carson, B. Nelson, and D. Nicol. *Discrete-Event System Simulation, Third Edition*. Prentice Hall, 2000. 28

[8] J. Bartholdi and D. Eisenstein. A production line that balances itself. *Operations Research*, 44(2):21–34, 1996. 24

[9] J. Bartholdi, D. Eisenstein, and R. Foley. Performance of bucket brigades when work is stochastic. *Operations Research*, 49(5):710–719, 2001. 24

[10] D. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete Time Case*. Optimization and Neural Computation Series. Athena Scientific, 1996. 15

[11] J. Dai. *Stability of Fluid and Stochastic Processing Networks*. Centre for Mathematical Physics and Stochastics, publication no. 9 edition, 1999. 6, 7, 8, 10

[12] J. Dai and S. Meyn. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transactions on Automatic Control*, 40:1889–1904, 1995. 11

[13] J. DeGaspari. All in the family: Flexible machining systems give manufacturers a hedge on their bets. *Mechanical Engineering*, 124(2):56–58, February 2002. 1

[14] T. Farrar. Optimal use of an extra server in a two station tandem queueing network. *IEEE Transactions on Automatic Control*, 38:1296–1299, August 1993. 1, 16

[15] T. Freiheit, M. Shpitalni, and J. Hu. Productivity of paced parallel-serial manufacturing lines with and without crossover. *Journal of Manufacturing Science and Engineering*, 126(2):361–367, May 2004. 1, 2, 24

[16] B. Hajek. Optimal control of two interacting service stations. *IEEE Transactions on Automatic Control*, AC-29(6):491–499, 1984. 2, 16, 20

[17] W. Hopp and M. Van Oyen. Agile workforce evaluation: A framework for cross-training and coordination. *IIE Transactions*, 36(10):919–940, 2004. 2

[18] D. Kaufman, H. Ahn, and M. E. Lewis. On the introduction of an agile, temporary workforce into a tandem queueing system. *Queueing Systems*, 51(1-2):135–171, 2005. 11

[19] W. Narongwanich, I. Duenyas, and J. Birge. Optimal portfolio of reconfigurable and dedicated capacity under uncertainty. Preprint, 2003. 1

[20] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons Inc., 1994. 11, 15, 19, 23

[21] K. Schiefermayr and J. Weichbold. A complete solution for the optimal stochastic scheduling of a two-stage tandem queue with two flexible servers,. *Journal of Applied Probability*, 42(3):778–796, 2005. 2

[22] L. Sennott. The convergence of value iteration in average cost Markov decision chains. *Operations Research Letters*, 19:11–16, 1996. 15

[23] L. Sennott. *Stochastic Dynamic Programming and the Control of Queueing Systems*. Wiley Interscience. John Wiley and Sons, 1999. 15

[24] L. Sennott, M. van Oyen, and S. Iravani. Optimal dynamic assignment of a flexible worker on an open production line with specialists. *European Journal of Operational Research*, 170:541–566, 2006. 2

[25] R. Serfozo. An equivalence between continuous and discrete time Markov decision processes. *Operations Research*, 27(3):616–620, 1978. 11

[26] M. Van Oyen, E. Gel, and W. Hopp. Performance opportunity for workforce agility in collaborative and non-collaborative work systems. *IIE Transactions*, 33(9):761–777, 2001. 24

[27] R. Weber and S. Stidham. Optimal controls of service rates in networks of queues. *Advances in Applied Probability*, 19:202–218, 1987. 20

[28] C. Wu, M. E. Lewis, and M. Veatch. Dynamic allocation of reconfigurable resources in a two-stage tandem queueing system with reliability considerations. *IEEE Transactions on Automatic Control*, 51(2):309–314, 2006. 1, 2, 16