Flex Serv Manuf J (2017) 29:286–308
DOI 10.1007/s10696-016-9251-9

Flexibility design in loss and queueing systems: efficiency of *k*-chain configuration

Jingui Xie[1] · **Yiming Fan**[1] · **Mabel C. Chou**[2]

Published online: 7 September 2016

Abstract Process flexibility has altered operations in manufacturing and service companies significantly. For instance, auto-mobile manufacturers use flexible production systems to meet uncertain demands effectively, and workforce flexible systems with cross-training are presently common in service industries. This paper studies *k*-chain configuration in both loss systems and queueing systems. We derive performance measures such as percent of customers loss and average customer waiting time. In the symmetric case, we numerically test the effects of *k* , system size and traffic intensity on flexibility design. The major conclusion is that 2-chain is no longer effective in loss systems although it still performs well in queueing systems.

Keywords Flexibility design · *k*-Chain efficiency · Loss system · Queueing system · Product-form solution

1 Introduction

Flexibility is generally viewed as a firm's ability to match production to uncertain demand. A certain level of flexibility can curb the damage caused by uncertain demand, whereas lacking flexibility can result in significant loss. For example, Chryslers Neon-based PT Cruiser was a very fashionable model in 2000 and 2001. The dedicated plant in Toluca, Mexico, was not able to keep up with its demand, while the plant making the Neon in Belvidere, Illinois, was underutilized but not configured to build the PT Cruiser. The estimated loss was of $240 million in profit

✉ Jingui Xie
xiej@ustc.edu.cn

[1] School of Management, University of Science and Technology of China, Hefei, China

[2] Business School, National University of Singapore, Singapore, Singapore

and another 0.5 points of market share in each of those years (Biller et al. 2006). Later, companies were moving from focused factories to flexible factories (Deng and Shen 2013). The Ford Motor Company, for example, invested $485 million in 2002 in two Canadian engine plants to retool them with a flexible system. The company also developed a plan to convert its engine and transmission plants worldwide into flexible systems. Similar initiatives have also been launched in companies like General Motors and Nissan.

In 1995, Jordan and Graves initiated a stream of research on supply chain flexibility by examining the economic benefits of chaining relative to full flexibility (Jordan and Graves 1995). After that, *k*-chain flexibility has become an important research topic. The efficiency of the long chain, or the sparse process structure in general, has been justified theoretically by Chou et al. (2010) which showed that the performance of 2-chain is close to 96 % that of the full flexibility system. Many studies have also reported that 2-chain is an ideal system. A comprehensive review of recently and significantly related literature is provided in Sect. 2. In the current research, we examine the efficiency of chaining structures under stochastic settings in both loss and queueing systems. The majority of existing studies are based on newsvendor settings (e.g., Chou et al. 2010). In many systems, orders arrive at random and production time is uncertain. However, few studies have been conducted on the chaining configuration in these stochastic systems. In addition, few works analyse queueing systems explicitly due to theoretical intractability. Moreover, the majority of the results from previous studies regarding flexibility system performance is obtained through simulation.

To analyse the effectiveness of *k*-chain, we apply the recent results on skill-based stochastic systems (Adan and Weiss 2012 and Visschers et al. 2012). By redefining the system state space, we are able to design efficient algorithms which can compute the performance of *k*-chain numerically in both loss systems and queueing systems. On the basis of our numerical studies, we observe that 2-chain is no longer effective in loss systems although it still performs well in queueing systems.

The paper is organized as follows. In Sect. 2, we review the most recent and important works on flexibility design. In Sect. 3, we describe our model and provide some basic properties. In Sect. 4 and 5, we design computation algorithms for both loss systems and queueing systems, and summarize our observations from numerical studies. Conclusions are made in Sect. 6.

2 Literature review

In this section, we review the most recent and important works on flexibility design. Jordan and Graves (1995) first examined the economic benefits of chaining relative to full flexibility, thus initiating a stream of research on supply chain flexibility. Graves and Tomlin (2003) then investigated various structures for achieving horizontal flexibility within a single supply chain level. Readers are referred to Chou et al. (2008) for reviews of process flexibility before 2008.

In 2010, significant theoretical progress in exploring efficiency of chaining structures has been made. Chou et al. (2010) showed that a simple chaining structure performs surprisingly well given a variety of realistic demand distributions, even when the system is large. They also identified a class of conditions under which a sparse flexible structure alone is needed so that expected performance is already within the optimality of the full flexibility system given general problems. Chou et al. (2010) examined how range and response dimension interact to affect the performance of the process flexible structure. Deng and Shen (2013) argued that effective flexible process structures are essentially highly connected graphs. They utilized the concept of graph expansion (a measure of graph connectivity) in obtaining various insights into this design problem.

Process flexibility is important in system design. Simchi-Levi and Wei (2012) developed a theory that explains the effectiveness of long chain designs for finite-size systems. Simchi-Levi and Wei (2015) determined that the long chain is superior to a mass of sparse flexibility designs. Simchi-Levi et al. (2013) noted that a 3-chain design is significantly more robust than a 2-chain one and achieves the same robustness as full flexibility does under high uncertainty levels. Furthermore, investment in process flexibility designs alters the optimal inventory placements. Désir et al. (2016) reported that a disconnected network with 2 n edges is optimal under this situation instead of a long chain, even for independent identical demand distributions. Wang and Zhang (2015) assessed the performance of a long chain under different demand distributions from a demand-distribution-free perspective. Iravani et al. (2005) focused on the strategic-level issues of how flexibility can be generated by using multi-purpose resources, such as cross-trained labour, flexible machines and flexible factories. Iravani et al. (2007) emphasised flexible service centres, such as inbound call centres with cross-trained agents, and model these centres as parallel queueing systems with flexible servers. A recent paper (Shi et al. 2015) developed a theory for the design of process flexibility for multi-period product systems and proved that any partial flexibility structure that satisfies Generalized Chaining Condition (GCC) is almost optimal under a class of policies. The authors also proposed that the performance of GCC structures can gain nearly as much as benefit as fully flexible system when the traffic intensity rate is fairly high.

The majority of works focus on symmetric systems. Nonetheless, some studies design process flexibility for unbalanced networks on the basis of chaining structure. Mak and Shen (2009) reported that the heavily advocated chaining heuristic can sometimes perform unsatisfactorily when resources are not perfectly flexible. Deng and Shen (2013) proposed additional flexibility design guidelines for unbalanced networks in which the numbers of plants and products are unequal by refining the well-known Chaining Guidelines. The results of an extensive computational study suggest that the refinements made by the researchers effectively determine flexible configurations with minimal shortfall in unbalanced networks. Besides the works mentioned above, many important studies have also been conducted on flexibility. Readers are referred to Chen et al. (2015), Hopp et al. (2005), Hopp et al. (2009), Iravani et al. (2003), Iravani et al. (2005), Iravani and Teo (2005), Iravani and Krishnamurthy (2007), Kula et al. (2004), Peltokorpi et al. (2015) and Sennott et al. (2006) for other related works.

The study of skill-based systems provides support for our analysis on system performance. Adan et al. (2010) studied an Erlang loss system with multi-type customers and servers and reported that the probabilities of assigning arriving customers to idle servers can be chosen in such a way that the Markov process describing the system is reversible, with a simple product-form stationary distribution. Visschers et al. (2012) examined a system with multi-type jobs and servers in which waiting jobs are served on a first-come-first-served (FCFS) basis, and arriving jobs that encounter several idle servers are assigned to a feasible server at random. These researchers suggested the existence of assignment probabilities under which a system displays product-form stationary distribution and develop explicit expressions for it. Adan and Weiss (2012) detected a simple explicit, steady-state distribution for a loss system with multi-type customers and skill-based servers under assignment to longest idle server (ALIS) policy. These researchers also report that this system is insensitive and that the results hold for general service time distributions as well. Adan and Weiss (2012) also established a product-form solution for two infinite multi-type sequences.

3 Model description

We model the stochastic systems with a set of servers which serve several types of customers. We denote the set of servers by $\mathcal{M}$, and the number of servers by $|\mathcal{M}|$. The system serves several types of customers, and we denote the set of customer types by $\mathcal{C}$. Hence, we have $|\mathcal{C}|$ types of customers. Each server- j can serve a subset of customer types, denoted by $\mathcal{C}(j)$. Equivalently, each type- i customer can be served by a subset of servers $\mathcal{M}(i)$, the union of which is $\mathcal{M}$. This relationship can be specified by a bipartite graph $\mathcal{G} = (\mathcal{C}, \mathcal{M}, E)$, where E denotes the edges of the graph which connects servers to the customer types they can serve. We define $\mathcal{C}(S) = \bigcup_{j \in S} \mathcal{C}(j)$ as the total set of customer types which can be handled by the servers in $S \subset \mathcal{M}$ and $\mathcal{U}(S) = \overline{\mathcal{C}(\overline{S})}$ as the set of customer types which can be served only by the servers in S.

For any two graphs $\mathcal{G} = (\mathcal{C}, \mathcal{M}, E)$ and $\mathcal{G}' = (\mathcal{C}', \mathcal{M}', E')$, if $\mathcal{C} = \mathcal{C}'$, $\mathcal{M} = \mathcal{M}'$ and $E \subset E'$, then $\mathcal{G}$ is regarded as a *spanning subgraph* of $\mathcal{G}'$, as denoted by $\mathcal{G} \subset \mathcal{G}'$. Based on the definition of graphs, we obtain the following lemma. Lemma 1 basically states that if a network exhibits high flexibility, then each server can serve more types of customers and each customer can be assigned to more available servers.

Lemma 1 *If $\mathcal{G} \subset \mathcal{G}'$, then $\mathcal{C}^{\mathcal{G}}(S) \subset \mathcal{C}^{\mathcal{G}'}(S)$ and $\mathcal{U}^{\mathcal{G}}(S) \supset \mathcal{U}^{\mathcal{G}'}(S)$ for any $S \subset \mathcal{M}$.*

Proof $\forall\, i \in \mathcal{C}^{\mathcal{G}}(S), \exists\, j \in S$, such that the server- j can serve type- i customers. That is, the edge $e_{ij} \in E$. Since $\mathcal{G} \subset \mathcal{G}'$, we have $E \subset E'$ according to the definition of spanning subgraph. Thus, $e_{ij} \in E'$. So, $i \in \mathcal{C}^{\mathcal{G}'}(S)$ and hence $\mathcal{C}^{\mathcal{G}}(S) \subset \mathcal{C}^{\mathcal{G}'}(S)$. The proof of the latter part is similar since $\mathcal{U}^{\mathcal{G}}(S) \supset \mathcal{U}^{\mathcal{G}'}(S)$ is equivalent to $\mathcal{C}^{\mathcal{G}}(\overline{S}) \subset \mathcal{C}^{\mathcal{G}'}(\overline{S})$. □

We assume that arrivals are Poisson processes and service times are exponential. Type- i customers arrive according to an independent Poisson process with rate λ_i. Server- j works independently with a rate μ_j. The arrival and service processes are mutually independent of each other. We introduce the notations $\lambda_A = \sum_{i \in A} \lambda_i, \forall A \subset \mathcal{C}$ and $\mu_S = \sum_{j \in S} \mu_j, \forall S \subset \mathcal{M}$. On the basis of Lemma 1, we then derive the following monotonicity properties. Proposition 1 suggests that more servers can take more demands, and less demands can be taken by other servers. Proposition 2 compares two graphs $\mathscr{G}$ and $\mathscr{G}'$ where $\mathscr{G} \subset \mathscr{G}'$. Thus, $\mathscr{G}'$ has more edges (links) than $\mathscr{G}$ which means that the flexibility of $\mathscr{G}'$ is higher. Given a set of servers S , they are capable of serving more customer types in $\mathscr{G}'$. However, the number of customer types which can be served only by the servers in S decrease in $\mathscr{G}'$.

Proposition 1 *Monotonicity property 1: For a graph $\mathscr{G}$ and any $S \subset S' \subset \mathcal{M}$, we have*

$$\lambda_{\mathcal{C}(S)} \leq \lambda_{\mathcal{C}(S')} \tag{1}$$

$$\lambda_{\mathcal{U}(S)} \leq \lambda_{\mathcal{U}(S')}. \tag{2}$$

Proposition 2 *Monotonicity property 2: For any two graphs $\mathscr{G} \subset \mathscr{G}'$ and $S \subset \mathcal{M}$, we have*

$$\lambda^{\mathscr{G}}_{\mathcal{C}(S)} \leq \lambda^{\mathscr{G}'}_{\mathcal{C}(S)} \tag{3}$$

$$\lambda^{\mathscr{G}}_{\mathcal{U}(S)} \geq \lambda^{\mathscr{G}'}_{\mathcal{U}(S)} \tag{4}$$

In our stochastic networks, service discipline is a combination of FCFS and ALIS. Arriving customers which encounter more than one available server are assigned to the server that has been idle for the longest time. However, arriving customers which locate no available servers are either lost or join the queue. We refer to these two systems as *loss systems* and *queueing systems*.

The flexibility design in such stochastic networks involves deciding the flexibility of each server, that is, the subset of customer types which it can serve, $\mathcal{C}(j)$. The flexibility level of server- j is defined as $|\mathcal{C}(j)|$. The system with complete resource pooling displays full flexibility, whereas the system which performs dedicated service has level-1 flexibility. We denote any connected bipartite graph by $\mathscr{G}$, the complete resource pooling graph by $\mathscr{F}$ and the dedicated graph by $\mathscr{D}$.

In this study, we focus on the case in which each server initially serves a dedicated type of customers. Hence, we have $|\mathcal{C}| = |\mathcal{M}|$, let's denote it as K. In particular, we are interested in a bipartite graph structure called *tailored chaining*. Tailored chaining with k-flexibility is called a k-chain, and any server- j can serve k types of customers starting from type- j and followed by type-$(j+1), \ldots, (j+k-1)$ (all numbers larger than K take modular on K) in a k-chain configuration. All servers have the same flexibility as k. Figure 1 shows the bipartite graphs of

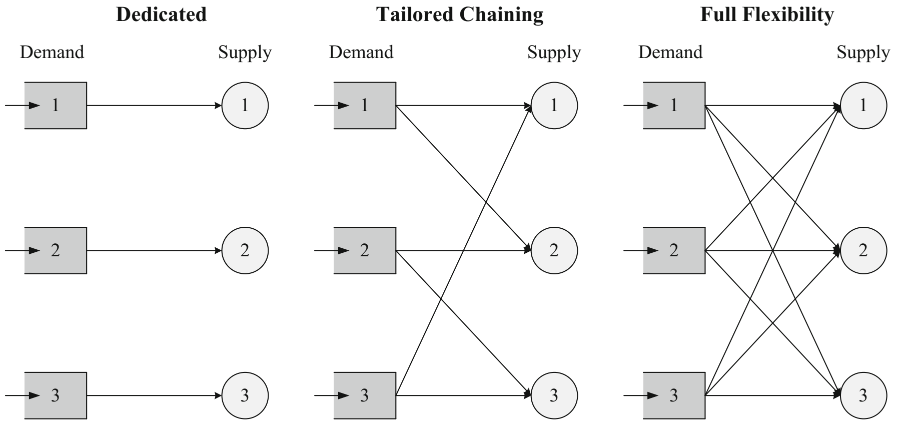

Fig. 1 Structural system flexibility

dedicated, full flexibility and tailored chaining with 2-flexibility when $K = 3$. In the dedicated system, one supplier can supply only one dedicated type of demand. In the tailored chaining (we can also call it 2-chain in this example) setting, each supplier can serve two types of demand. In the full flexibility system, a supplier can serve all types of demand. In the following sections, we study the k-chain efficiency in both loss systems and queueing systems, and compare the performance of a k-chain to the dedicated system and the system with full flexibility.

4 Loss system

In the loss system, arriving customers who do not find available servers leave the system without being served. Customers who find more than one available server are assigned to the server which has been idle for the longest time. The percent of customers who leave without being served, i.e., loss percentage, is the main performance measure for the loss system.

We define the system state at time t as $X(t) = s$, where $s = (j_1, j_2, \ldots, j_k)$ is the list of idle servers at time t , ordered by their order of consecutive idle time. Thus, server j_1 has the longest idle time, and so on. Given this state definition, $X(t)$ is a continuous-time, finite-state Markov chain under the ALIS policy with a stationary distribution (Adan and Weiss 2012):

$$\pi_X(j_1, \ldots, j_k) = \pi_X(\emptyset) \frac{\mu_{j_1}}{\lambda_{\mathcal{C}(j_1)}} \frac{\mu_{j_2}}{\lambda_{\mathcal{C}(\{j_1, j_2\})}} \cdots \frac{\mu_{j_k}}{\lambda_{\mathcal{C}(\{j_1, \ldots, j_k\})}}, \tag{5}$$

which satisfies the partial balance equations

$$\pi_X(j_1, \ldots, j_k) \lambda_{\mathcal{C}(\{j_1, \ldots, j_k\})} = \pi_X(j_1, \ldots, j_{k-1}) \mu_{j_k}, \tag{6}$$

and for all $m \neq j_1, \ldots, j_k$,

$$\begin{aligned}\pi_X(j_1,\ldots,j_k)\mu_m = {} & \pi_X(m,j_1,\ldots,j_k)\lambda_{\mathcal{C}(m)} + \pi_X(j_1,m,\ldots,j_k)\lambda_{\mathcal{C}(m)\setminus\mathcal{C}(j_1)} \\ & + \ldots + \pi_X(j_1,\ldots,j_k,m)\lambda_{\mathcal{C}(m)\setminus\mathcal{C}(\{j_1,\ldots,j_k\})}.\end{aligned} \tag{7}$$

Let θ_i be the fraction of customers of type- i who are lost. According to the property of Poisson arrival see the average (PASTA), a customer of type- i is lost if the set of idle servers does not contain any server from $\mathcal{M}(i)$. Hence,

$$\theta_i = \sum_{\{j_1,\ldots,j_k\}\cap\mathcal{M}(i)=\emptyset} \pi_X(j_1,\ldots,j_k). \tag{8}$$

The total number of states is $K!\sum_{k=0}^{K}\frac{1}{k!}$. A minimum computation time of $O(K!\sum_{k=0}^{K}\frac{1}{k!})$ is required to compute the steady-state distribution and performance measures, which seems to be a huge task.

However, in most cases, the information regarding server idle time is not used in performance analysis, although it is necessary for real-time operations. Therefore, we can redefine the system state at time t as $Y(t) = S$, where $S = \{j_1, j_2, \ldots, j_k\}$ is the set of idle servers at time t , for analysis purposes. We recall that $s = (j_1, j_2, \ldots, j_k)$ represents the list of idle servers at time t , ordered by their order of becoming idle. Thus, s is merely one permutation out of all possible permutations. Let $\mathcal{P}(S)$ be the set of all of the permutations of S , then we have

$$\pi_Y(S) = \pi_Y(\{j_1,j_2,\ldots,j_k\}) = \sum_{(\tilde{j}_1,\ldots,\tilde{j}_k)\in\mathcal{P}(S)} \pi_X(\tilde{j}_1,\ldots,\tilde{j}_k). \tag{9}$$

By applying the partial balance equations (5) and (6), we obtain the recursion of steady-state probabilities

$$\begin{aligned}\pi_Y(S)\lambda_{\mathcal{C}(S)} &= \sum_{(\tilde{j}_1,\ldots,\tilde{j}_k)\in\mathcal{P}(S)} \pi_Y(\emptyset)\frac{\mu_{\tilde{j}_1}}{\lambda_{\mathcal{C}(\tilde{j}_1)}}\frac{\mu_{\tilde{j}_2}}{\lambda_{\mathcal{C}(\{\tilde{j}_1,\tilde{j}_2\})}}\cdots\frac{\mu_{\tilde{j}_k}}{\lambda_{\mathcal{C}(\{\tilde{j}_1,\ldots,\tilde{j}_k\})}} \times \lambda_{\mathcal{C}(\{\tilde{j}_1,\ldots,\tilde{j}_k\})} \\ &= \sum_{(\tilde{j}_1,\ldots,\tilde{j}_k)\in\mathcal{P}(S)} \pi_Y(\emptyset)\frac{\mu_{\tilde{j}_1}}{\lambda_{\mathcal{C}(\tilde{j}_1)}}\frac{\mu_{\tilde{j}_2}}{\lambda_{\mathcal{C}(\{\tilde{j}_1,\tilde{j}_2\})}}\cdots\frac{\mu_{\tilde{j}_{k-1}}}{\lambda_{\mathcal{C}(\{\tilde{j}_1,\ldots,\tilde{j}_{k-1}\})}} \times \mu_{\tilde{j}_k} \\ &= \sum_{j\in S}\pi_Y(S\setminus\{j\})\mu_j.\end{aligned} \tag{10}$$

The loss percentage of type- i customers is given by

$$\theta_i = \sum_{S\cap\mathcal{M}(i)=\emptyset} \pi_Y(S), \tag{11}$$

where

$$\pi_Y(S) = \pi_{\tilde{Y}}(S) = \pi_{\tilde{Y}}(\emptyset)\frac{\mu_{j_1}}{\eta_{j_1}(\{j_1\})}\frac{\mu_{j_2}}{\eta_{j_2}(\{j_1,j_2\})}\cdots\frac{\mu_{j_k}}{\eta_{j_k}(\{j_1,\ldots,j_k\})}, \tag{12}$$

and $\eta_k(S)$ can be recursively calculated by

$$\eta_k(S) = \lambda_{C(S)}\left(1 + \sum_{j\in S\setminus\{k\}} \frac{\eta_j(S\setminus\{k\})}{\eta_k(S\setminus\{j\})}\right)^{-1}, \forall k \in S. \tag{13}$$

Given this state space redefinition, the total number of states is reduced to 2^K. When the recursion equation (10) is used, computation time is reduced to $O(K^2 2^K)$. As a result, the applicability of the results of Adan and Weiss (2012) to our case is enhanced.

In the following paragraphs, we investigate the symmetric case in which $\lambda_i = \lambda$, $\mu_j = \mu$. We define $\rho = \lambda/\mu$. We apply loss rate $\theta(K, k, \rho)$ as a key system performance measure and study the effects of its parameters: system size K , flexibility k (k-chain, $1 \le k \le K$) and traffic intensity ρ. In a dedicated symmetric system ($k = 1$), each server can enter only two possible states: either the server is empty, or it is busy serving a customer. The probability that the server is empty is expressed as $(1 + \rho)^{-1}$, whereas the probability that the server is busy is determined by $\rho(1 + \rho)^{-1}$. In a complete resource pooling system($k = K$), the probability that an arrival customer will be lost is equal to the probability that all servers are busy, known as Erlang loss.

Proposition 3 *The performance of k-chain in loss systems*:

1 *For the dedicated system, the percent of lost is* $\rho(1 + \rho)^{-1}$.

2 *For the complete resource pooling system, the percent of loss is*

$$\frac{(K\rho)^K}{K!}\left(\sum_{k=0}^{K} \frac{(K\rho)^k}{k!}\right)^{-1}. \tag{14}$$

3 *For other k-chain systems, loss percentage can be calculated by* (11).

As a result of computation simplification, we can compute a system with a maximum of 16 servers and 16 types of customers. To compute larger size problems, one needs to further improve the computation algorithm (we leave this issue as a future research topic). For each system with size K , we compute the loss percentage of all possible k-chain configurations, i.e., $k = 1, 2, \ldots, K$. We also study three different scenarios: systems with light load ($\rho = 0.5$), medium load ($\rho = 1$) and heavy load ($\rho = 2$). All numerical results are shown in Tables 1, 2 and 3. We summarize the numerical results in the following observations.

Observation 1 *Loss rate* $\theta(K, k, \rho)$ *is convex and decreases with system size K. In particular, the decrease is insignificant when k is small for all* ρ.

This observation coincides with economies of scale. In any k-chain, loss decreases when system size increases. However, the marginal decrease is reduced, as depicted in Fig. 2.

Table 1 Loss rate of k-chain configurations in Erlang Loss System when $\rho = 1$

K	k															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.5000															
2	0.5000	0.4000														
3	0.5000	0.3913	0.3462													
4	0.5000	0.3902	0.3368	0.3107												
5	0.5000	0.3901	0.3334	0.3021	0.2849											
6	0.5000	0.3901	0.3324	0.2977	0.2772	0.2649										
7	0.5000	0.3901	0.3321	0.2956	0.2725	0.2580	0.2489									
8	0.5000	0.3901	0.3319	0.2947	0.2698	0.2534	0.2427	0.2356								
9	0.5000	0.3901	0.3319	0.2942	0.2683	0.2504	0.2383	0.2300	0.2243							
10	0.5000	0.3901	0.3319	0.2940	0.2674	0.2485	0.2352	0.2259	0.2193	0.2146						
11	0.5000	0.3901	0.3319	0.2939	0.2670	0.2473	0.2330	0.2227	0.2154	0.2100	0.2061					
12	0.5000	0.3901	0.3319	0.2939	0.2667	0.2465	0.2315	0.2204	0.2123	0.2063	0.2019	0.1986				
13	0.5000	0.3901	0.3319	0.2939	0.2665	0.2460	0.2305	0.2188	0.2100	0.2034	0.1985	0.1947	0.1919			
14	0.5000	0.3901	0.3319	0.2938	0.2664	0.2457	0.2298	0.2175	0.2082	0.2011	0.1956	0.1915	0.1883	0.1858		
15	0.5000	0.3901	0.3319	0.2938	0.2664	0.2455	0.2293	0.2166	0.2068	0.1992	0.1933	0.1888	0.1853	0.1825	0.1803	
16	0.5000	0.3901	0.3319	0.2938	0.2663	0.2453	0.2289	0.2160	0.2057	0.1977	0.1915	0.1866	0.1827	0.1797	0.1772	0.1753

Table 2 Loss rate of k-chain configurations in Erlang Loss System when $\rho = 0.5$

K	k															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.3333															
2	0.3333	0.2000														
3	0.3333	0.1875	0.1343													
4	0.3333	0.1857	0.1220	0.0952												
5	0.3333	0.1854	0.1172	0.0849	0.0697											
6	0.3333	0.1854	0.1156	0.0795	0.0614	0.0522										
7	0.3333	0.1854	0.1150	0.0768	0.0563	0.0455	0.0396									
8	0.3333	0.1854	0.1147	0.0755	0.0534	0.0412	0.0344	0.0304								
9	0.3333	0.1854	0.1147	0.0748	0.0516	0.0383	0.0307	0.0263	0.0236							
10	0.3333	0.1854	0.1146	0.0744	0.0505	0.0364	0.0281	0.0232	0.0203	0.0184						
11	0.3333	0.1854	0.1146	0.0742	0.0499	0.0352	0.0263	0.0210	0.0178	0.0158	0.0144					
12	0.3333	0.1854	0.1146	0.0741	0.0495	0.0343	0.0251	0.0194	0.0160	0.0138	0.0123	0.0114				
13	0.3333	0.1854	0.1146	0.0740	0.0492	0.0338	0.0242	0.0183	0.0146	0.0122	0.0107	0.0097	0.0090			
14	0.3333	0.1854	0.1146	0.0740	0.0490	0.0334	0.0235	0.0174	0.0135	0.0111	0.0094	0.0084	0.0076	0.0071		
15	0.3333	0.1854	0.1146	0.0740	0.0489	0.0331	0.0231	0.0167	0.0127	0.0101	0.0085	0.0074	0.0066	0.0061	0.0057	
16	0.3333	0.1854	0.1146	0.0740	0.0488	0.0329	0.0227	0.0162	0.0121	0.0094	0.0077	0.0065	0.0058	0.0052	0.0048	0.0045

Table 3 Loss rate of k-chain configurations in Erlang Loss System when $\rho = 2$

K	k															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.6667															
2	0.6667	0.6154														
3	0.6667	0.6122	0.5902													
4	0.6667	0.6120	0.5872	0.5746												
5	0.6667	0.6120	0.5862	0.5721	0.5640											
6	0.6667	0.6120	0.5860	0.5709	0.5618	0.5561										
7	0.6667	0.6120	0.5860	0.5705	0.5607	0.5543	0.5500									
8	0.6667	0.6120	0.5859	0.5703	0.5601	0.5532	0.5485	0.5452								
9	0.6667	0.6120	0.5859	0.5703	0.5598	0.5526	0.5475	0.5439	0.5413							
10	0.6667	0.6120	0.5859	0.5702	0.5597	0.5522	0.5469	0.5430	0.5401	0.5380						
11	0.6667	0.6120	0.5859	0.5702	0.5596	0.5520	0.5465	0.5424	0.5393	0.5370	0.5352					
12	0.6667	0.6120	0.5859	0.5702	0.5596	0.5519	0.5462	0.5420	0.5387	0.5362	0.5343	0.5328				
13	0.6667	0.6120	0.5859	0.5702	0.5596	0.5519	0.5461	0.5417	0.5383	0.5357	0.5336	0.5320	0.5307			
14	0.6667	0.6120	0.5859	0.5702	0.5596	0.5518	0.5460	0.5415	0.5380	0.5353	0.5331	0.5314	0.5300	0.5289		
15	0.6667	0.6120	0.5859	0.5702	0.5596	0.5518	0.5460	0.5414	0.5378	0.5350	0.5327	0.5309	0.5294	0.5282	0.5272	
16	0.6667	0.6120	0.5859	0.5702	0.5596	0.5518	0.5459	0.5413	0.5377	0.5348	0.5324	0.5305	0.5290	0.5277	0.5267	0.5258

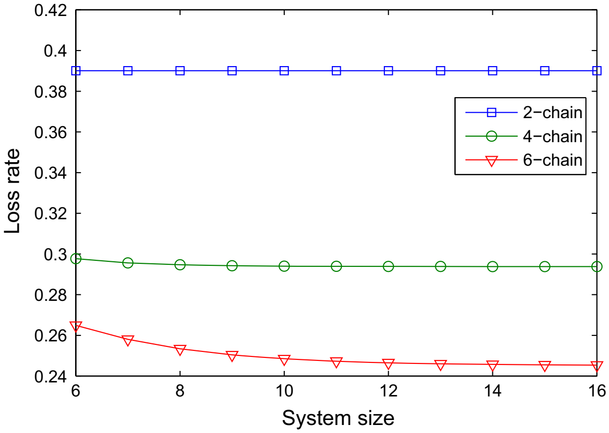

Fig. 2 Performance of k-chain configuration in a loss system with $\rho = 1$

Similar to Simchi-Levi and Wei (2012), the authors observed that the performance of 2-chain improves with K, but the improvement converges to zero exponentially quickly. As shown in Table 1, given a small flexibility system, such as 2-chain, the benefit is insignificant as the system size increased. We can see when $K \geqslant 5$, loss percentage remains the same. The finding above implies that flexibility can be maintained in several separate sub-systems, that is, several short chains, to reduce organizational complexity. For example, the loss rate is approximately 0.3901 for a 2-chain system with size $K = 16$ when $\rho = 1$. If we separate the system into four independent sub-systems with size $K = 4$, then the loss rate for the four systems is 0.3902. The system performance does not change remarkably. That is, the 2-chain is more effective for smaller size systems relative to full flexibility design. Thus, several small closed chains, where each chain connects a substantial number of plants and products, can perform just as well as the long chain.

Observation 2 *Loss rate $\theta(K, k, \rho)$ is convex and decreases with k.*

Generally, increasing the flexibility will reduce the loss. However, in all of the three cases, most of the increase in throughput is achieved by 2-chain since percentage of loss decreases significantly when k changes from 1 to 2. Higher orders of chaining increase throughput only marginally and in progressively diminishing amounts. These results suggest that total flexibility is generally unnecessary if increases in flexibility require massive investments.

Observation 3 *The marginal benefit of increasing the flexibility is insignificant when traffic intensity is high.*

Figure 3 shows that the marginal benefit is much smaller when traffic intensity is high. Take $K = 16$ for example, comparing Table 2 and 3, it is obvious that adding flexibility is more effective in light load systems than systems with heavy load. (When $\rho = 0.5$, loss percentage dropped from 0.3333 to 0.0045, or about 98.64 % as flexibility increases from 1 to 16. However, this rate is only 21.13 % when

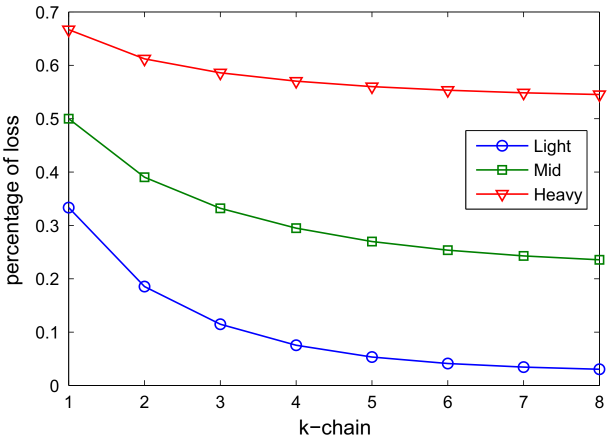

Fig. 3 Performance of *k*-chain configuration in a loss system under different loads

$\rho = 2.$). Thus, there is less incentive to implement flexibility when system is under heavy load.

Observation 4 *2-chain is no longer effective in the loss system.*

Chou et al. (2010) shows that the performance of 2-chain is already close to 96 % of the full flexibility system based on newsvendor settings when $\rho = 1$. Here, we use dedicated and complete resource pooling systems as benchmarks to evaluate *k*-chain efficiency as follow:

$$\psi_{k,K}(\rho) = \frac{L_{1,K}(\rho) - L_{k,K}(\rho)}{L_{1,K}(\rho) - L_{K,K}(\rho)}, \tag{15}$$

where $L_{k,K}(\rho)$ is the customer loss rate in a loss system of size K with *k*-chain configuration and traffic intensity ρ.

As can be seen from Table 4, when $K = 16$, the performance of 2-chain can only reach one third of the full flexibility system. Several explanations have been offered as to why the properties which hold in the newsvendor model cannot be extended to the Erlang loss model: (1) Service times are uncertain which can increase system uncertainty. This situation cannot be handled by the flexibility design alone. (2) Dynamic real-time operations enhance loss. Meanwhile, orders can be held for a period and satisfied at the end of the period in the newsvendor model.

5 Queueing system

The queueing system described in this section is similar to the system presented above, except that customers will join the queue and wait for service if no servers are available upon their arrival. Visschers et al. (2012) analyzed and obtained product-form solutions for a continuous time Markov chain that describes a multi-type customers, multi-type servers queueing system under Assumption 1.

Table 4 Efficiency of k-chain (%) with different sizes in Erlang Loss System when $\rho = 1$

K	k													
	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	70.68													
4	58.00	86.21												
5	51.09	77.45	92.00											
6	46.75	71.29	86.05	94.77										
7	43.77	66.87	81.40	90.60	96.38									
8	41.57	63.58	77.65	87.07	93.27	97.31								
9	39.86	60.97	74.65	84.04	90.53	94.92	97.93							
10	38.51	58.90	72.18	81.50	88.12	92.78	96.04	98.35						
11	37.39	57.20	70.13	79.28	85.98	90.85	94.35	96.84	98.67					
12	36.46	55.77	68.38	77.41	84.11	89.08	92.77	95.45	97.45	98.91				
13	35.67	54.56	66.89	75.79	82.44	87.47	91.27	94.13	96.27	97.86	99.09			
14	34.98	53.50	65.63	74.35	80.94	86.00	89.91	92.87	95.13	96.88	98.19	99.20		
15	34.38	52.58	64.50	73.07	79.61	84.67	88.65	91.71	94.09	95.93	97.34	98.44	99.31	
16	33.85	51.77	63.50	71.97	78.44	83.49	87.47	90.64	93.10	95.01	96.52	97.72	98.64	99.41

Assumption 1 [*Assignment condition in Visschers et al.* (2012)] For $i = 1, \ldots K$, and for every subset $\{M_1, \ldots M_i\} \in \mathcal{M}$, the following holds:

$$\prod_{j=1}^{i} \lambda_{M_j}\left(\{M_1, \ldots, M_{j-1}\}\right) = \prod_{j=1}^{i} \lambda_{\overline{M}_j}\left(\{\overline{M}_1, \ldots, \overline{M}_{j-1}\}\right) \tag{16}$$

for every permutation $\overline{M}_1, \ldots, \overline{M}_i$ of $M_1, \ldots, M_i$, where $\lambda_{M_j}\left(\{M_1, \ldots, M_{j-1}\}\right)$ is the activation rate of machine M_j (See Visschers et al. (2012) for the definition and more details).

Remark 1 Assumption 1 is important for correctness of the analysis, because as Visschers et al. (2012) explicitly constructed examples where no product form solution exists when the random assignment probabilities are not chosen correctly.

Remark 2 As Visschers et al. (2012) conjectured the assignment probability distributions become less relevant when traffic intensity approaches 1. So the assumption on choosing a specific set of random assignment probabilities will not be restrictive and the product form solution will be a good approximation for general assignment probability distributions.

Under Assumption 1, Visschers et al. (2012) obtained the product-form solution:

$$\pi(s) = \alpha_i^{n_i} \ldots \alpha_1^{n_1} \frac{\prod_{j=1}^{i} \lambda_{M_j}\left(\{M_1, \ldots, M_{j-1}\}\right)}{\prod_{j=1}^{i} \mu_{\{M_1, \ldots, M_j\}}} \pi(0), \tag{17}$$

where $s = (n_i, M_i, n_{i-1}, M_{i-1}, \ldots, n_1, M_1)$ and $\alpha_j = \lambda_{\mathcal{U}(\{M_1, \ldots, M_j\})} \mu^{-1}_{\{M_1, \ldots, M_j\}}$.

We define the state as $(M_1, n_1, \ldots, M_i, n_i; M_{i+1}, \ldots, M_K)$, that is, the state in which a system has i busy servers and $K - i$ idle servers with corresponding numbers of customers waiting between the busy servers. $M_1, \ldots, M_K$ is a permutation of $1, \ldots, K$. Servers $M_1, \ldots, M_i$ serve customers with increasing arrival times, and n_j customers wait between servers M_j and M_{j+1}, $1 \leq j \leq i$. It should be noticed that the waiting customers between servers M_j and M_{j+1} can only be handled by the servers $M_1, \ldots, M_j$ and not by any of the servers $M_{j+1}, \ldots, M_i$ or any of the idle servers. This is due to the FCFS processing order (see Fig. 4). Servers $M_{i+1}, \ldots, M_K$ are idle with increasing idle time. Similar to Adan and Weiss (2014), we consider the queueing system under FCFS-ALIS policy. The steady-state probability is determined by [see Adan and Weiss (2014)]

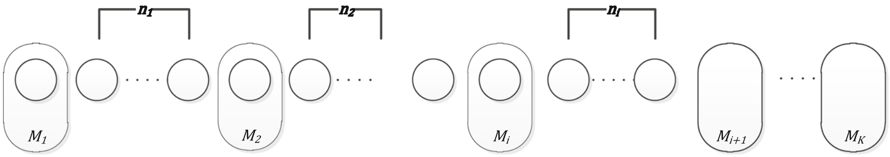

Fig. 4 Graphical illustration of the system state in the queueing system

$$\pi(M_1, n_1, \ldots, M_i, n_i; M_{i+1}, \ldots, M_K) \\ = B\left(\prod_{j=1}^{i} \mu_{\{M_1,\ldots,M_j\}} \prod_{j=i+1}^{K} \lambda_{\mathcal{C}(\{M_j,\ldots,M_K\})}\right)^{-1} \alpha_1^{n_1} \ldots \alpha_i^{n_i}, \tag{18}$$

where B is a normalization constant.

The marginal distribution of attaining server status $\boldsymbol{M}_i = (M_1, \ldots, M_i; M_{i+1}, \ldots, M_K)$ is given by

$$\pi(\boldsymbol{M}_i) = B\left(\prod_{j=1}^{i} \mu_{\{M_1,\ldots,M_j\}} \prod_{j=i+1}^{K} \lambda_{\mathcal{C}(\{M_j,\ldots,M_K\})}\right)^{-1} (1-\alpha_1)^{-1} \ldots (1-\alpha_i)^{-1}. \tag{19}$$

Let $\Phi_{ki} = \{j | 1 \leq j \leq i, k \in \mathcal{U}(\{M_1, \ldots, M_j\})\}$. As aforementioned, n_j is customers waiting between servers M_j and M_{j+1}. Here, we use n_{kj} and N_k to represent the type-k customers among n_j and the total type- k customer in the queue, respectively.

Conditioning on server status $\boldsymbol{M}_i = (M_1, \ldots, M_i; M_{i+1}, \ldots, M_K)$, similar to the discussion in Visschers et al. (2012), n_j is a geometric random variable with parameter α_j,

$$\mathbf{E}[n_j | \boldsymbol{M}_i] = \frac{\alpha_j}{1-\alpha_j}; \tag{20}$$

n_{kj} is a geometric random variable with parameter η_{kj} ,

$$\mathbf{E}[n_{kj} | \boldsymbol{M}_i] = \frac{\eta_{kj}}{1-\eta_{kj}}, \tag{21}$$

where

$$\eta_{kj} = \frac{\lambda_k}{\mu_{\{M_1,\ldots,M_j\}} - \lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})} + \lambda_k};$$

and the expectation of N_k is expressed as

$$\mathbf{E}[N_k | \boldsymbol{M}_i] = \sum_{j \in \Phi_{ki}} \frac{\eta_{kj}}{1-\eta_{kj}}. \tag{22}$$

By law of total probability, we combine Eqs. (19) and (22) and obtain

$$\mathbf{E}[N_k] = B \sum_{\boldsymbol{M}_i} \frac{\sum_{j \in \Phi_{ki}} \frac{\eta_{kj}}{1-\eta_{kj}}}{\prod_{j=1}^{i} (1-\alpha_j) \mu_{\{M_1,\ldots,M_j\}} \prod_{j=i+1}^{K} \lambda_{\mathcal{C}(\{M_j,\ldots,M_K\})}} \tag{23}$$

By applying Little's law, we compute the average waiting time for type- k customers, $\mathbf{E}[N_k] = \lambda \mathbf{E}[W_k]$. Therefore, the average number of customers (waiting time) in the queue is given by

$$\mathbf{E}[N] = \sum_{k=1}^{K} \mathbf{E}[N_k], \quad \mathbf{E}[W] = \mathbf{E}[N]/\lambda_{\mathcal{C}}. \tag{24}$$

Hence, we have

$$\mathbf{E}[N] = B\sum_{k=1}^{K}\sum_{\boldsymbol{M}_i}\frac{\sum_{j\in\Phi_{ki}}\frac{\eta_{kj}}{1-\eta_{kj}}}{\prod_{j=1}^{i}(1-\alpha_j)\mu_{\{M_1,\ldots,M_j\}}\prod_{j=i+1}^{K}\lambda_{\mathcal{C}(\{M_j,\ldots,M_K\})}} \tag{25}$$

$$= B\sum_{\boldsymbol{M}_i}\frac{\sum_{j=1}^{i}\sum_{k\in\mathcal{U}(\{M_1,\ldots,M_j\})}\frac{\eta_{kj}}{1-\eta_{kj}}}{\prod_{j=1}^{i}(1-\alpha_j)\mu_{\{M_1,\ldots,M_j\}}\prod_{j=i+1}^{K}\lambda_{\mathcal{C}(\{M_j,\ldots,M_K\})}}. \tag{26}$$

Proposition 4 *The average waiting time in the queue system is given by*

$$\mathbf{E}[W] = B\lambda_{\mathcal{C}}^{-1}\sum_{\boldsymbol{M}_i}\frac{\sum_{j=1}^{i}\sum_{k\in\mathcal{U}(\{M_1,\ldots,M_j\})}\frac{\eta_{kj}}{1-\eta_{kj}}}{\prod_{j=1}^{i}(1-\alpha_j)\mu_{\{M_1,\ldots,M_j\}}\prod_{j=i+1}^{K}\lambda_{\mathcal{C}(\{M_j,\ldots,M_K\})}}. \tag{27}$$

In particular, if all $\lambda_i = \lambda$ and $\mu_j = \mu$. Let $\rho := \lambda/\mu$. The formula can be simplified as

$$\mathbf{E}[W] = B\lambda_{\mathcal{C}}^{-1}\sum_{\boldsymbol{M}_i}\frac{\sum_{j=1}^{i}\sum_{k\in\mathcal{U}(\{M_1,\ldots,M_j\})}\frac{\rho}{j(1-\alpha_j)}}{\prod_{j=1}^{i}j\mu(1-\alpha_j)\prod_{j=i+1}^{K}\lambda_{\mathcal{C}(\{M_j,\ldots,M_K\})}}, \tag{28}$$

where $\alpha_j = \lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})}/j\mu$.

In the following paragraphs, we consider the symmetric case where all $\lambda_i = \lambda$ and $\mu_j = \mu$. The dedicated system is similar to K parallel and independent M/M/1 queues. The average number of customers in the system is denoted by $K\rho(1-\rho)^{-1}$. For each type of customer, the average response time (the sum of both waiting and service times) is $1/(\mu-\lambda)$, and the average waiting time is $1/(\mu-\lambda) - 1/\mu = \rho/(\mu-\lambda)$.

The full flexibility system is in fact an M/M/K queue. Let $\rho' := \lambda/K\mu$ and p_k denote the stationary probability that k customers are in the system. The system stationary probability is given by

$$p_k = \begin{cases} p_0\dfrac{(K\rho')^k}{k!}, & k \le K \\ p_0\dfrac{K^K\rho'^k}{K!}, & k \ge K \end{cases} \tag{29}$$

where

$$p_0 = \left[\sum_{k=0}^{K-1}\frac{(K\rho')^k}{k!} + \frac{(K\rho')^K}{K!}\frac{1}{1-\rho'}\right]^{-1}. \tag{30}$$

The probability that an arriving customer will be forced to wait in the queue is

$$p_{K_+} = p_0\frac{(K\rho')^K}{K!}\frac{1}{1-\rho'}. \tag{31}$$

The expected number of customers in the system is given by

$$E_K = K\rho' + \frac{\rho'}{1-\rho'} p_{K_+}. \tag{32}$$

Under the queueing system, arriving customers who encounter no available servers will join the queue. The waiting time (or delay) in this queue is the main performance measure. We denote $W_{k,K}(\rho)$ as the customer waiting time in a queueing system of size K with k-chain configuration and traffic intensity ρ. Figure 5 shows the average waiting time given different levels of chaining configuration (or flexibility) and five different traffic intensity levels $\rho = 0.6, 0.7, 0.8, 0.9$ and 0.95. It is intuitive that the heavier traffic load is, the longer waiting time will be. Also, we can obtain that average waiting time drops significantly when flexibility is increased from 1 to 2. Table 5 provides additional details.

Just like loss system, we use the following benchmark to evaluate k-chain efficiency:

$$\psi_{k,K}(\rho) = \frac{\mathbf{E}[W_{1,K}(\rho)] - \mathbf{E}[W_{k,K}(\rho)]}{\mathbf{E}[W_{1,K}(\rho)] - \mathbf{E}[W_{K,K}(\rho)]}. \tag{33}$$

Table 6 shows the efficiency of k-chain with different sizes and traffic loads. For a given K and ρ, $\psi_{k,K}(\rho)$ increases with the flexibility k. As shown in Table 6, efficiency values are getting bigger from the left to right. For a given K and k, $\psi_{k,K}(\rho)$ increases with the traffic intensity ρ. Take $K = 8$ for example, 2-chain cannot achieve >90 % efficiency when $\rho < 0.9$. By contrast, 3-chain attains >93 % efficiency under all investigated traffic loads. A significant increment is observed

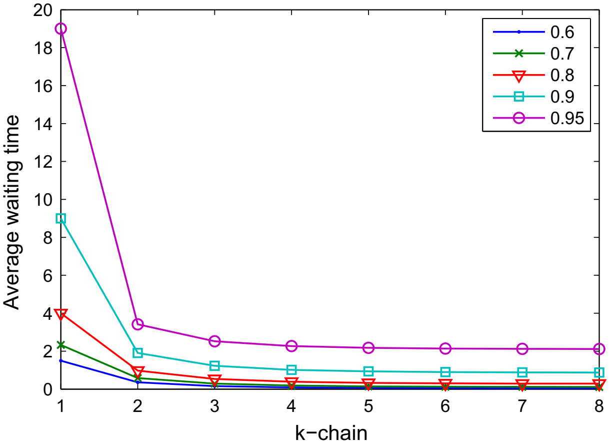

Fig. 5 Performance of k-chain configuration in a queueing system with $K = 8$

Table 5 Average waiting time of *k*-chain with different sizes and traffic loads

k	*K*							
	1	2	3	4	5	6	7	8
$\rho = 0.8$								
1	4							
2	4	1.7778						
3	4	1.2487	1.0787					
4	4	1.0759	0.8006	0.7455				
5	4	1.0116	0.6658	0.5795	0.5541			
6	4	0.9860	0.5964	0.4833	0.4454	0.4315		
7	4	0.9755	0.5587	0.4252	0.3755	0.3556	0.3471	
8	4	0.9711	0.5377	0.3890	0.3291	0.3033	0.2916	0.2860
$\rho = 0.9$								
1	9							
2	9	4.2632						
3	9	2.9267	2.7235					
4	9	2.3907	2.0346	1.9694				
5	9	2.1406	1.6638	1.5555	1.5250			
6	9	2.0140	1.4473	1.2983	1.2506	1.2335		
7	9	1.9468	1.3134	1.1297	1.0645	1.0392	1.0285	
8	9	1.9100	1.2273	1.0148	0.9334	0.8992	0.8841	0.8769
ρ = 0.95								
1	19							
2	19	9.2564						
3	19	6.2681	6.0467					
4	19	4.9330	4.5274	4.4571				
5	19	4.2298	3.6646	3.5442	3.5112			
6	19	3.8238	3.1269	2.9568	2.9039	2.8853		
7	19	3.5756	2.7701	2.5550	2.4812	2.4530	2.4413	
8	19	3.4179	2.5230	2.2681	2.1739	2.1353	2.1184	2.1104
$\rho = 0.99$								
1	99							
2	99	49.2513						
3	99	32.9424	32.7056					
4	99	24.9733	24.5221	24.4476				
5	99	20.3188	19.6662	19.5356	19.5005			
6	99	17.3076	16.4730	16.2840	16.2269	16.2071		
7	99	15.2259	14.2263	13.9819	13.9007	13.8701	13.8576	
8	99	13.7186	12.5685	12.2722	12.1669	12.1245	12.1061	12.0975

from 2-chain to 3-chain or 4-chain; however, the marginal increment is very small when flexibility continues to increase. To facilitate efficient system operation, we suggest employing a 3-chain or 4-chain structure rather than a 2-chain one. Taking

Table 6 Efficiency of k-chain with different sizes and traffic loads(%)

K	k					
	2	3	4	5	6	7
ρ = 0.8						
3	94.18					
4	89.85	98.31				
5	86.72	96.76	99.26			
6	84.46	95.38	98.55	99.61		
7	82.80	94.21	97.86	99.22	99.77	
8	81.55	93.22	97.23	98.84	99.53	99.85
ρ = 0.9						
3	96.76					
4	94.01	99.07				
5	91.76	98.14	99.59			
6	89.95	97.25	99.17	99.78		
7	88.48	96.43	98.73	99.55	99.87	
8	87.28	95.69	98.30	99.30	99.73	99.91
ρ = 0.95						
3	98.29					
4	96.73	99.52				
5	95.36	99.01	99.79			
6	94.18	98.50	99.56	99.88		
7	93.15	98.01	99.31	99.76	99.93	
8	92.26	97.56	99.07	99.62	99.85	99.95
ρ = 0.99						
3	99.64					
4	99.29	99.90				
5	98.97	99.79	99.96			
6	98.67	99.68	99.91	99.98		
7	98.39	99.57	99.85	99.95	99.99	
8	98.13	99.46	99.80	99.92	99.97	99.99

traffic intensity into consideration, 3-chain configuration may be enough when the system is under a heavy traffic load. As for systems with $\rho = 0.8$ or $\rho = 0.9$, 4-chain structure could be better. More details can be seen in Fig. 6. Our observation is similar to Shi et al. (2015), in which the authors proved that in a very different setting, the efficiencies of the chaining structures approach 100 % as traffic intensity approaches 1.

Figure 7 shows the average waiting time under k-chain configuration with an increase in system size K from 2 to 8 and when $\rho = 0.9$. $\mathbf{E}[W_{k,K}(\rho)]$ decreases with K to a certain limit. This result implies that when a large k-chain splits into several independent small chains, system performance does not vary significantly. For example, the average waiting time is approximately 1.91 for a 2-chain system with

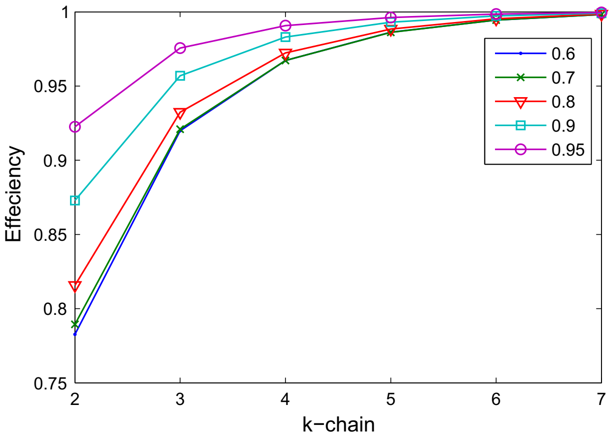

Fig. 6 Efficiency of k-chain configuration in a queueing system with $K = 8$

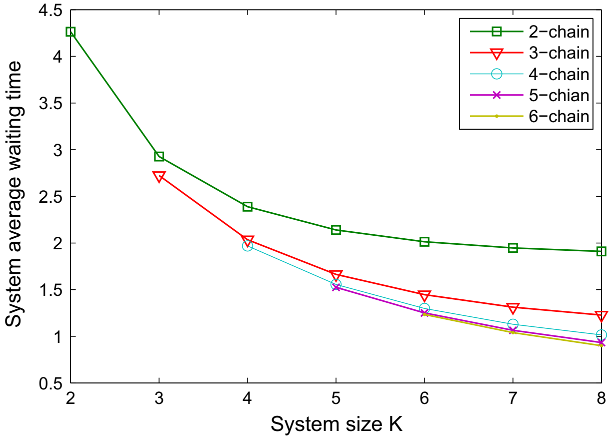

Fig. 7 Average waiting time under k-chain configuration with $\rho = 0.9$

size $K = 8$. If we separate the system into two independent sub-systems with size $K = 4$, then the average waiting time for both systems is 2.39.

6 Conclusion

In this study, we investigate the k-chain efficiency in stochastic networks with uncertainties on both the demand and supply sides. We consider loss systems and queueing systems. The key performance measure for loss systems is the percentage of lost customers. Unlike Chou et al. (2010) which shows that the performance of 2-chain is close to 96 % that of the full flexibility system when $\rho = 1$, we find that 2-chain is no longer effective in the loss system. This situation is even worse when traffic intensity becomes higher. One needs to increase flexibility to achieve better performance. The key performance measure for queueing systems is the average waiting time. In these systems, 2-chain performs rather well when traffic intensity is close to 1. This finding is consistent with the existing literature which shows that 2-chain is an ideal system. However, the efficiency of 2-chain drops when the traffic intensity gets lower. Thus, a higher flexibility level, e.g., 3-chain or 4-chain, is required.

Our work can be extended in a variety of ways. One possible research direction for future works is the discussion on asymmetric systems. In this research, we focus on the symmetric case in which all servers are identical, all customer types are identical, and the number of servers equals to the number of customer types. However, in reality, we are often faced with asymmetric systems where the demands are different for different customer types or the service rates are different for different servers. Another direction may involve investigating structural properties in more general system network design rather than focusing only on the *k*-chain configuration. In addition, one can develop better algorithms to compute the performance of large systems.

Acknowledgments The authors gratefully acknowledge that this work was partially supported by National Nature Science Foundation of China under Grant 71571176, National University of Singapore Academic Research Fund under Grant R314-000-085-112, and Agency for Science Technology and Research under Grant 1122904020, R314-000-091-305.

References

Adan I, Hurkens C, Weiss G (2010) A reversible Erlang loss system with multitype customers and multitype servers. Probab Eng Inf Sci 24(04):535–548

Adan I, Weiss G (2012) A loss system with skill-based servers under assign to longest idle server policy. Probab Eng Inf Sci 26(3):307

Adan I, Weiss G (2012) Exact FCFS matching rates for two infinite multitype sequences. Oper Res 60(2):475–489

Adan I, Weiss G (2014) A skill based parallel service system under FCFS-ALIS steady state, overloads, and abandonments. Stoch Syst 4(1):250–299

Biller S, Muriel A, Zhang Y (2006) Impact of price postponement on capacity and flexibility investment decisions. Prod Oper Manag 15(2):198–214

Chen X, Zhang J, Zhou Y (2015) Optimal sparse designs for process flexibility via probabilistic expanders. Oper Res 63(5):1159–1176

Chou MC, Teo CP, Zheng H (2008) Process flexibility: design, evaluation, and applications. Flex Serv Manuf J 20(1–2):59–94

Chou MC, Chua GA, Teo CP (2010) On range and response: dimensions of process flexibility. Eur J Oper Res 207(2):711–724

Chou MC, Chua GA, Teo CP et al (2010) Design for process flexibility: efficiency of the long chain and sparse structure. Oper Res 58(1):43–58

Deng T, Shen ZJM (2013) Process flexibility design in unbalanced networks. Manuf Serv Oper Manag 15(1):24–32

Désir A, Goyal V, Wei Y et al (2016) Sparse process flexibility designs: is the long chain really optimal? Oper Res 64(2):416–431

Graves SC, Tomlin BT (2003) Process flexibility in supply chains. Manag Sci 49(7):907–919

Harel A (1990) Convexity properties of the Erlang loss formula. Oper Res 38(3):499–505

Hopp WJ, Iravani SMR, Shou B (2005) Serial agile production systems with automation. Oper Res 53(5):852–866

Hopp WJ, Iravani SMR, Shou B et al (2009) Design and control of agile automated CONWIP production lines. Naval Res Logist (NRL) 56(1):42–56

Iravani SMR, Buzacott JA, Posner MJM (2003) Operations and Shipment scheduling of a batch on a felxible machine. Oper Res 51(4):585–601

Iravani SMR, Buzacott JA, Posner MJM (2005) A robust policy for serial agile production systems. Naval Res Logist (NRL) 52(1):58–73

Iravani SM, Van Oyen MP, Sims KT (2005) Structural flexibility: a new perspective on the design of manufacturing and service operations. Manag Sci 51(2):151–166

Iravani SMR, Teo CP (2005) Asymptotically optimal schedules for single-server flow shop problems with setup costs and times. Oper Res Lett 33(4):421–430

Iravani SMR, Krishnamurthy V (2007) Workforce agility in repair and maintenance environments. Manuf Serv Oper Manag 9(2):168–184

Iravani SMR, Kolfal B, Van Oyen MP (2007) Call-center labor cross-training: it's a small world after all. Manag Sci 53(7):1102–1112

Jordan WC, Graves SC (1995) Principles on the benefits of manufacturing process flexibility. Manag Sci 41(4):577–594

Kula U, Duenyas I, Iravani SMR (2004) Estimating job waiting times in production systems with cross-trained setup crews. IIE Trans 36(10):999–1010

Mak HY, Shen ZJM (2009) Stochastic programming approach to process flexibility design. Flex Serv Manuf J 21(3–4):75–91

Peltokorpi J, Tokola H, Niemi E (2015) Worker coordination policies in parallel station systems: performance models for a set of jobs and for continuous arrival of jobs. Int J Prod Res 53(6):1625–1641

Sennott LI, Van Oyen MP, Iravani SMR (2006) Optimal dynamic assignment of a flexible worker on an open production line with specialists. Eur J Oper Res 170(2):541–566

Shi C, Wei Y, Zhong Y (2015) Process flexibility for multi-period production systems. Available at SSRN: http://ssrn.com/abstract=2655790

Simchi-Levi D, Wei Y (2012) Understanding the performance of the long chain and sparse designs in process flexibility. Oper Res 60(5):1125–1141

Simchi-Levi D, Wang H, Wei Y (2013) Increasing supply chain robustness through process flexibility and strategic inventory. Working paper, MIT, Cambridge, MA

Simchi-Levi D, Wei Y (2015) Worst-case analysis of process flexibility designs. Oper Res 63(1):166–185

Visschers J, Adan I, Weiss G (2012) A product form solution to a system with multi-type jobs and multi-type servers. Queueing Syst 70(3):269–298

Wang X, Zhang J (2015) Process flexibility: a distribution-free bound on the performance of k-chain. Oper Res 63(3):555–571

Jingui Xie received the Ph.D. degree in Management Science from Tsinghua University, Beijing, China, in 2010. Currently, he is an Associate Professor at School of Management, University of Science and Technology of China, Hefei, China. He has published several papers on international journals such as IEEE Transactions on Automatic Control, Naval Research Logistics, Queueing Systems, International Journal of Production Research and Operations Research Letters. His specific research interests are stochastic systems, queuing theory, optimization control, data analytics and healthcare management.

Yiming Fan received the B.S degree from Nanjing Forestry University, Nanjing, China, in 2013. Currently, she is a Ph.D. student in Management Science at School of Management, University of Science and Technology of China, Hefei, China. Her research interests include stochastic system and health care management.

Mabel C. Chou received the Ph.D. degree in Industrial Engineering and Management Science from Northwestern University, USA in 2001. Currently, she is an Associate Professor at National University of Singapore. She has published several papers on international journals such as Operations Research, Management Science, European Journal of Operational Research, IEEE Transactions on Industrial Informatics, etc. She has won Department Outstanding Educator Award (NUS) in 2006 and Faculty Outstanding Educator Award (NUS) in 2012. Her specific research interests are Inventory and Warehouse Management, Logistics and Supply Chain Analysis, Operations and Marketing Interface, Operations Research Applications in Health Care, Operations/manufacturing Flexibility Design and Analysis, Optimization with Uncertainty and Production Scheduling.