Structural Flexibility: A New Perspective on the Design of Manufacturing and Service Operations

Seyed M. Iravani[1], Mark P. Van Oyen[2], and Katharine T. Sims[1]

[1] *Department of Industrial Engineering and Management Sciences*
Northwestern University, Evanston, IL 60208

[2] *School of Business Administration*
Loyola University Chicago, Chicago, IL 60611

Abstract

In this paper we present a new perspective on flexibility in manufacturing and service operations by exploring a type of operational flexibility that we term "structural flexibility". We focus on strategic level issues of how flexibility can be created by using multipurpose resources such as cross trained labor, flexible machines, or flexible factories. The proposed structural flexibility method uses the structure of the capability pattern to generate indices that quantify the ability of a system to respond to variability in its environment. Simulations of serial and parallel queueing networks provide evidence that this index is useful in predicting the performance rank of alternative designs for implementing multi-functionality in the face of variability. The proposed methodology supports managerial insight into structural design of manufacturing and service systems at the strategic level.

Keywords: Flexibility; Cross-training; Max Flow Algorithm; Serial and Parallel Production Systems.

1 Introduction

Flexibility is a very general concept that is often viewed as a firm's ability to match production to market demand in the face of uncertainty and variability. The notion of flexibility is also closely linked to the firm's ability to provide niche and customized products to the consumer. Workforce management, supply chain management, and flexible manufacturing are undergoing dramatic development to achieve flexibility using a variety of mechanisms such as cross-trained labor, enhanced use of information systems, improved logistics, small batch sizes, delayed product differentiation, and multi-purpose machines/tools.

A growing literature has focused on developing a deeper understanding of how cross-training and/or flexible equipment can be appropriately used to improve productivity and provide greater performance in servicing a variety of types of demand (see Hopp and Van Oyen 2004b for strategic and tactical frameworks with a literature survey). Behind much of this literature is an intuitive

notion of flexibility. It does not, however, appear to us that there currently exists a mathematical definition of flexibility that is broadly useful. Sethi and Sethi (1990) express the challenge as follows: "The literature makes one thing abundantly clear: flexibility is a complex, multidimensional, and hard-to-capture concept."

Variability in demand (and/or capacity) deteriorates system performance. When shifts in average demand are long-term or permanent shifts, the solution is often to increase source capacities (an expensive option). If the increase in demand is not long-term or not sufficiently large, then increasing capacity may result in underutilization of investment in periods that demand is low. An option receiving growing interest is to enable sources to respond to more than one demand type, so they adapt to changes. Now the question becomes how to add capabilities to sources in order to provide robust performance despite demand shifts and uncertainty (variability). By "capability" we mean the ability of a source to process a demand type. Workers who are cross-trained, machines that are flexible, and factories that can reallocate production across multiple products are all examples of production sources with multiple capabilities. In these examples, classes of tasks, jobs, and products, respectively, are the demand types. Consider Figure 1, which represents three different ways that sources can be given capabilities to serve four demand types. For example, in Figure 1(B), source S_2 has two capabilities which allow it to process both type 1 and type 2 demands. In a call center cross-training application, this would mean that agent S_2 is trained and equipped to handle both type 1 and type 2 calls. Every source has a "capability set" indicating the demand types that it can serve. In this example, the capability set of S_2 is $\{1, 2\}$. The ensemble of capability sets for all the sources creates a graph termed a "structure" (see Figure 1 for three particular structures).

This paper treats a model of an operation with N production sources facing K types of demand. We develop a simple method that computes an index for each alternative structure, which can then be used to predict which structure has better and more robust performance, without needing precise information regarding the patterns of the changes in the system. Our focus is on providing insight into the importance of the system structure as a vehicle toward flexibility.

It is important to emphasize that our method seeks to address applications in which *precise information* regarding the environment (i.e., demand or capacity) is not available, or if it is available, it is *not reliable* for planning purposes since it changes rapidly over time. Examples include call

arrival processes in call centers, which are known to change rapidly throughout the day, and the processing time distributions in a make-to-order mixed-model production system due to shifting in product mix as demand changes. As an example of uncertain changes in capacity, consider a labor-intensive production operation with multiple shifts. From one shift to another, the operation will have different workers with different speeds, as well as fluctuations caused by worker absenteeism. In all the above examples, it is important that the design of the capability structure not be tied to a *specific* demand or capacity pattern, since that would reduce the robustness of the system with respect to changes in its environment. A method such as ours that provides direction in achieving flexibility given some rough idea of the relative demand or capacity levels fits well with strategic level and long-term decisions, where accurate data and forecasts are usually unavailable. It provides managers with a better intuition about the relationship between the selection of a system structure and the resulting flexibility.

Our work addresses key issues such as the following:

1. The authors are aware of the philosophy prevalent at a large U.S. technology company which views employee training as an inherently good thing. The company lacks a corresponding emphasis on strategically selecting the training choices that will yield the most benefit to the company. Is it sufficient for a company to provide good access to training, or is it important that training and cross-training be purposefully and carefully selected so as to create a more flexible system?

2. Can a simple algorithm be developed to assist managers in determining a good strategy for investing in new capabilities in the absence of precise information for system capacity or demand?

3. Are there any canonical capability structures that provide superior flexibility and can serve as a basis for strategy in creating an effective structure of capabilities?

2 System Structure: A Means for Buffering Against Variability

The effects of variability in demand can be mitigated by either increasing capacity or increasing the flexibility of available capacity (e.g., using production sources with multiple capabilities).

To show how production sources with multiple capabilities can be used to create a structure that effectively buffers against variability, consider a system with four such "sources" serving four "demand types." In Figure 1, a link between source S_i and demand type D_j indicates that source S_i is capable of serving demand type D_j (e.g., worker skill, machine capability, plant/product

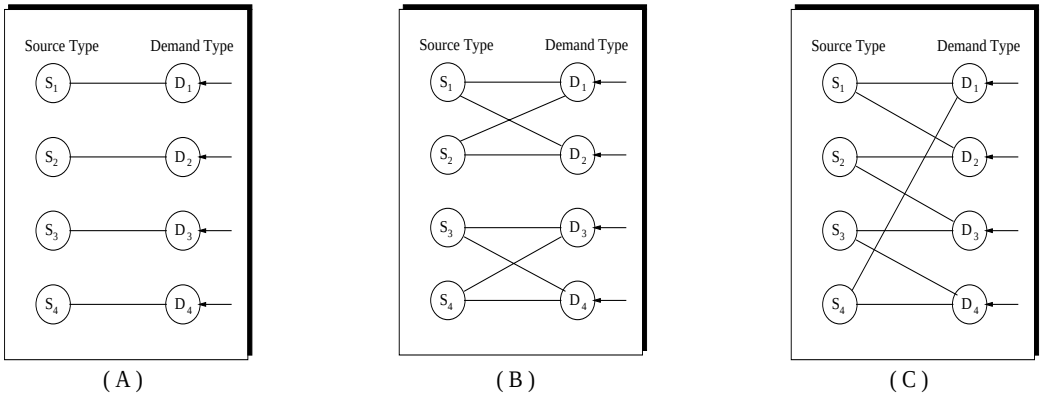

Figure 1: Three different examples of four sources serving four demand types.

allocation, etc.). In Figure 1(A) the capacity of a production source S_i is exclusively devoted to one demand type, while in Figure 1(B) and 1(C), each source can serve two different demand types.

If each source has unit capacity and the demands all require 1 job per unit time, then all three system structures have the same performance in the absence of variability in demand and/or source capacity. Consider now a system with variability in demand, where D_i may equally likely be $1-\delta$, 1, or $1+\delta$, for some $\delta \in [0,1]$. Consider a month in which $D_3 = D_4 = 1$, while demand D_1 is $1+\delta$, and demand D_2 is $1-\delta$. The structure in Figure 1(A) can only respond to the increase in demand D_1 by increasing its capacity at source S_1, (i.e., increasing capacity as the means of buffering against variability). However, the system cannot utilize the δ additional units of unused capacity of source S_2 (since D_2 has fallen by δ). In the structure in Figure 1(B), however, S_2 can use up to δ of its unused capacity for demand type 1 as needed. The structure in Figure 1(C) can also allocate the unused capacity of S_2 to accommodate the increase in D_1 as follows: S_1 can assign its capacity to D_1, while S_2 assigns $1-\delta$ units of its capacity to D_2, and δ units to D_3. S_3 assigns $1-\delta$ units to D_3 and δ units to D_4. S_4 assigns $1-\delta$ units of its capacity to D_4 and δ units to D_1.

Now consider a month, in which $D_2 = D_4 = 1$, while demand D_1 is $1+\delta$, and demand D_3 is $1-\delta$. Under these circumstances, δ units of capacity must be shifted from source S_3 to source S_1 during that month. As it is clear in the figure, only the structure in Figure 1(C) is capable of shifting the unused capacity of source S_3 to S_1. In fact, in $3^4 = 81$ combinations of demand levels with values $1-\delta$, 1, or $1+\delta$ for types 1 through 4, the structure in Figure 1(C) is capable of handling all 50 cases where the total demand from types 1 through 4 is not more than 4, the total available capacity in the system (i.e., $\sum_{i=1}^{4} D_i \leq 4$). Structure (B) can handle 36 cases, and structure (A) can handle only 16 cases. In a stochastic demand environment, structure (C) seems more capable

of dealing with changes in demand and thus is more flexible than the other two structures.

As we described above, in multiple-resource systems, capacity can be used in two different ways to buffer against variability: (i) increasing source capacities, or (ii) directly or indirectly shifting capacity among sources. The ability to shift capacity is the result of capability patterns of sources or what we call "system structure" in this paper. This motivates the fundamental concepts underlying our new "Structural Flexibility."

Definition: *Structural flexibility is a system's ability, provided by its structure of multi-capability sources, to reallocate production to respond to changes in demand (e.g., volume, work content, product mix, etc.) or in source capacity (e.g., absenteeism, breakdowns, rework, etc.).*

Next, we introduce two examples illustrating the concept of structural flexibility. Example 1 deals with the problem of allocating the production capacity of four plants to four different demands (an open parallel network flow). Example 2 looks at workers' training in a serial-flow production line (a closed serial network flow).

2.1 Example 1: Plant Capacity Allocation

Consider a company that has four ($N = 4$) production plants (sources) S_1, S_2, S_3, and S_4 which are used to produce four different products ($K = 4$) with monthly demand type arrival rates D_i, $i = 1, 2, 3, 4$. Interpreting sources as plants and their products as demand types, the capability structures of Figure 1 show three different ways (A, B, and C) of designing the production sources to produce different demand types. According to our capacity shifting argument, we expect that product/plant allocation structure(C) provides more flexibility and performs better than the other two in the face of variability in demand. This has been confirmed by Jordan and Graves (1995), that proposes structure (C) (which they called a "Chain structure") to be a flexible structure for plant capacity in the auto industry.

2.2 Example 2: Cross-training Workers in CONWIP Lines

Figure 2 illustrates a serial production line with four work stations and four workers W_1, W_2, W_3 and W_4 operated under a CONstant Work In Process (CONWIP) release policy (see Hopp and Spearman, 2000). Specifically, CONWIP refers to a job release policy that will release a job to station 1 only upon completion of a job from the line (the total WIP in the system is kept at the

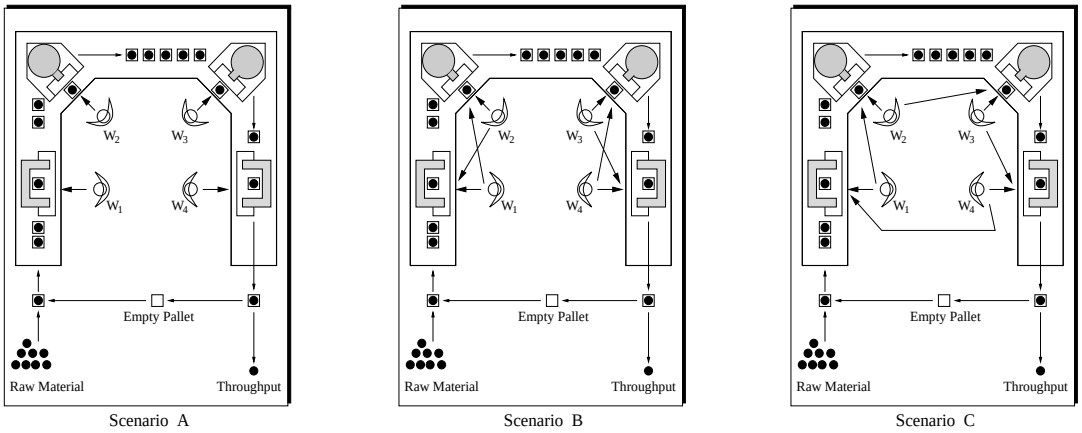

Figure 2: Worker assignment scenarios for CONWIP production line example.

constant CONWIP level).

A similar capacity shifting argument can be used to show that cross-training structure in Figure 2(C) is more flexible than the other two, and therefore it performs better in the face of variability in job processing times. For example, suppose that during a shift job processing times at work stations 2 and 4 are 1 unit, but processing times at work stations 1 and 3 are $1+\delta$ and $1-\delta$, respectively. In that shift, work station 1 becomes the bottleneck, and under the cross-training structures in Figures 2(A) and 2(B) the throughput of the line reduces to $TH = 1/(1+\delta)$ per unit time. However, under the structure in Figure 2(C), the $(\delta \times 100)\%$ idle time (i.e., unused capacity) of worker W_4 can be used (i.e., shifted) to help cover the extra time δ at the bottleneck station. This keeps the throughput of the line steady at $TH = 1$ per unit time in that shift. The system in Figure 2(C) is called a "2-skill chain" by Hopp et al. (2004a, 2004b), and has been shown to be a very flexible and effective cross-training structure for workers in a CONWIP line.

3 Literature Survey

The work on flexibility is very broad, but we cite some of the more germane work here. De Groote (1994) creates a general framework to characterize flexibility and its influences. Sethi and Sethi (1990) provide a survey of notions of flexibility dating back to the 1920's. They define 11 types of flexibility and characterize published notions of flexibility under one of these categories. Our treatment of flexibility can address issues labeled in their taxonomy as "machine flexibility," "product flexibility," "routing flexibility," "volume flexibility," and "market flexibility".

Many studies have been done on the use of cross-trained workers, especially in serial production

lines: Ahn et al. (1999), Andradottir et al. (2001), Bartholdi and Eisenstein et al. (1996), Bartholdi et al. (2001), Berman et al. (1997), Duenyas et al. (1998), Gel et al. (2000, 2001), Iravani et al (1997a, 1997b), and McClain et al. (2000) among many others. For work on parallel systems, Pinker and Shumsky (2000) address cross-training, turnover, and quality in call centers, while Mandelbaum and Reiman (1998) focus on the impact of pooling in general networks. Pooling is referred to as collaboration by Van Oyen et al. (2001), who tackle the optimal control and performance analysis of open and closed production lines.

The notion of a chain was introduced in Jordan and Graves (1995) in the context of process flexibility for a single-stage manufacturing system with random demand and deterministic production. This work was extended to multi-stage manufacturing systems by Graves and Tomlin (2003). Sheikzadeh et al. (1998) used it to analyze equipment flexibility. Hopp et al. (2004a) use queueing models of flexible workers in serial production systems operating under a CONWIP protocol to show that the 2-skill chaining structure possesses strong capacity balancing and variability buffering properties Our methodology gives insight into why chains are so effective.

Gurumurthi and Benjaafar (2001) relate flexibility and throughput under varying parameters, congruent with the observations made in Hopp et al. (2004a). They modeled flexible queueing systems as a connected bipartite undirected graph, a similar representation to the one we develop here (see also Graves and Tomlin (2003) and Aksin and Karaesmen (2002)).

Aksin and Karaesmen (2002) address flexibility in loss systems with parallel flow using a graph-theoretic approach to determine the maximum throughput achievable under a particular network structure. They carefully explore the space of symmetric, connected networks to emphasize the superiority of such structures to alternatives. They show that a throughput bound is increasing and convex in the number of capabilities, and they demonstrate conditions under which it is desirable to balance the number of capabilities of the workers and/or balance the number of workers that are trained for each demand type. Our work differs from theirs in that our work focuses on structure and therefore can be applied to both serial and parallel and both open and closed networks (with an obvious generalization to other network topologies).

4 Structural Flexibility Measures

4.1 The Concept of "Fit": A Requirement for Proper Utilization of Capacity

Most production and service operations are designed with sufficient capacity to meet the average demand for purposes of stability. Consider source capacity vector $\mathcal{C} = (C_1, C_2, \ldots, C_N)$, which represents the (average) capacities of resources 1 to N, and the demand vector $\mathcal{D} = (D_1, D_2, \ldots, D_K)$ which represents the average demand rate for types 1 to K.

Definition: *A structure "fits" an environment with source capacity vector $\mathcal{C}$ and demand vector $\mathcal{D}$ if it is possible to allocate source capacities in such a way that each demand type is satisfied on average.*

Fitness of a structure, also referred to as capacity balancing, has been considered by other studies on parallel queues with resource pooling, mostly in heavy traffic (see Stolyar (2004), Williams (2000), Harrison and Lopez (1999), and references therein). In most of these studies a linear programming model is used to find the resource pooling scenario that ensures that all demands receive the proper capacity. In the case of closed queueing systems such as our CONWIP line example, fitness of a structure is not required to gain stability; rather, to ensure that the system properly utilizes its capacity so that it does not restrict throughput. Note that the condition that a structure must fit its environment is not sufficient for flexibility.

4.2 Structural Flexibility Matrix

We use our example in Section 2 to describe our method. Consider case (C) of Figure 1 where the demand vector is perturbed so that D_3 declines by amount δ in a particular month and D_1 increases by δ. If capacity is tight, then δ units of capacity must be shifted from source 3 to source 1 during that month.

Under this assumption, Figure 3 shows two different paths from node D_3 to node D_1 in the graph of Scenario C. Each path represents a different way of assigning the unused production capacity for product 3 to produce product 1. The first path on the left, $D_3 \to S_2 \to D_2 \to S_1 \to D_1$, corresponds to the following reassignment of production sources: The excess capacity δ available for product 3 releases δ units of capacity of S_2. Therefore, S_2 will be able to produce δ units more of product 2, which in turn releases δ more units of capacity of S_1 that can be used to satisfy the extra

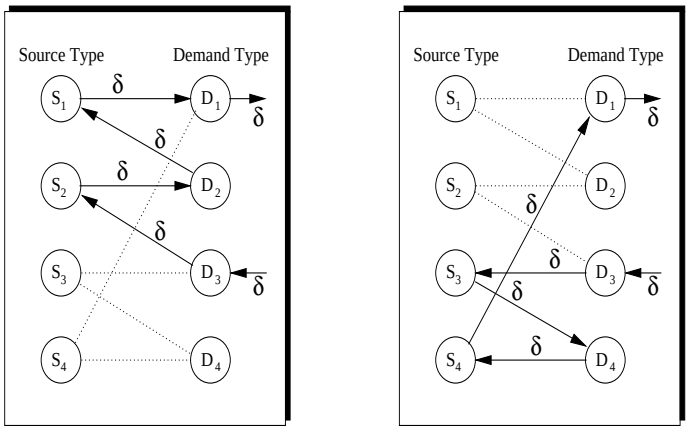

Figure 3: Two paths to shift unused capacity from product 3 to produce product 1 in Scenario 1(C).

demand δ for product 1. The path on the right of Figure 3 is another way of transferring excess capacity for product 3 to respond to the excess demand for product 1: $D_3 \rightarrow S_3 \rightarrow D_4 \rightarrow S_4 \rightarrow D_1$.

The ability of Scenario C to respond to changes using more paths than Scenarios A or B is flexibility inherent in the structure – structural flexibility. Thus, we capture the structural flexibility by counting the total number of nonoverlapping paths a system can use to respond to a particular change in demand. Consistent with the definition of structural flexibility in Section 2, we propose the SF method to quantify the ability to shift internal capacity to respond to shifts in demand using the graph of a structure. The SF method translates a structure into a "Structural Flexibility matrix" (SF matrix), M. Let m_{ij} be the total number of nonoverlapping paths by which the excess capacity for product i can be redirected to produce product j (for the above example $m_{3,1} = 2$). We define matrix M as the matrix with elements m_{ij} for all $i \neq j,\ i, j = 1, 2, \ldots, K$.

Note that the element m_{ij} of the SF matrix can be obtained by solving a max flow problem in which the starting node is demand node D_i and the sink node is demand node D_j. The capacity of each arc is one, and thus m_{ij} is the maximum flow that can be transfered from D_i to D_j. To obtain the SF matrix for a system with K nodes, $K(K-1)/2$ max flow problems must be solved, since the SF matrix is symmetric (see online Appendix II for a summary of the max flow formulation).

Since transferring excess capacity from product i to produce product i does not make clear sense, m_{ii}, the diagonal elements of M are handled differently. A system is more flexible for changes in demand type i if more production sources are capable of supplying demand type i. Therefore, we let m_{ii} be the total number of arcs connected to demand node i. Consequently, the SF matrix for the three scenarios become:

$$M_A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_B = \begin{pmatrix} 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{pmatrix} \quad M_C = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{pmatrix}$$

Our methodology, the structural flexibility (SF) method, is based on M. It seems reasonable to expect that the larger the elements of this matrix, the more flexibility the system has.

4.3 The Structural Flexibility Indices

Our focus is on using the SF matrix to rank the flexibility of connected structures, where the term connected refers to those with at least one path through the structure between any demand node i and j. When the graph of a structure consists of 2 or more connected subgraphs, analysis becomes very complicated and further development is needed in such cases. In order to be able to compare the flexibility of two different structures, we need to further compact the information in our matrix in the form a scalar, which we name "Structural Flexibility index." We develop two candidates: (i) the mean index, and (ii) the eigenvalue index.

4.3.1 Mean Index

It is clear that SF matrices with larger elements (numbers) represent more flexible structures. On the other hand, matrices with larger elements will have a larger mean. We define the *Mean Index* $\mathcal{I}_{me}(M)$ *as the mean of all the elements in* M. Although the mean of the elements of SF matrix is an indicator of how large the elements of the matrix are, it is not sensitive to the location of the larger elements of the matrix. For example, several different matrices can have the same mean. This is one motivator for our next index.

4.3.2 Dominant Eigenvalue

For a system with K demand types, the structural flexibility matrix M has K eigenvalues $\theta_1(M)$, $\theta_2(M), \ldots, \theta_K(M)$ and corresponding eigenvectors $\Theta_1(M), \Theta_2(M), \ldots, \Theta_K(M)$. Since M is real, symmetric, and nonnegative, its eigenvalues are real and nonnegative. We define the *Eigenvalue Index* $\mathcal{I}_{ei}(M)$ *to be the dominant eigenvalue* $\theta^*(M)$ *of* M, *where* $\mathcal{I}_{ei}(M) = \theta^*(M) = \max_i \{\theta_i(M)\}$.

Dominant eigenvalues can also be considered as an indication of the magnitude of elements of a matrix (see on-line Appendix III). It can be shown that, as any element of the SF matrix increases

(i.e., more flexibility), the dominant eigenvalue of the matrix increases. The eigenvalue index is sensitive to the location and variation of the elements of the SF matrix, and different matrices with the same mean index almost always have different dominant eigenvalues.

5 Evaluating the Performance of the SF Indices

In this section, we use a numerical study to evaluate the ability of mean and eigenvalue indices to predict the performance of a structure. We benchmark their performances against two key alternatives, namely the number of arcs index, and the Jordan and Graves (JG) index.

The number of arcs index is based on the simple intuition that each arc adds flexibility. We define the *Number of Arcs Index, $\mathcal{I}_{ar}$, of a structure to be the sum of the number of capabilities of all sources.* Connecting it to the SF matrix M with diagonal elements m_{ii}, we see that the number of arcs is $\mathcal{I}_{ar} = \sum_{i=1}^{K} m_{ii}$; hence, the number of arcs index is actually based only on the diagonal of our SF matrix.

The JG index, $\mathcal{I}_{JG}$, is a detailed heuristic metric developed by Jordan and Graves (1995) for measuring flexibility in parallel systems of flexible factories based on an approximation of the probability of not being able to satisfy demand over a chosen production period (see Jordan and Graves 1995, Page 588). Obtaining this index requires detailed probability distributions of capacity and demand. For deterministic production times and normally distributed demand, Jordan and Graves (1995) provides an expression for computing the JG index. When production times are stochastic, or demand is not normally distributed, this index can only be obtained by complex stochastic modeling or computer simulation (which we used as described in the on-line Appendix IV).

5.1 Structures and Environments

Next, we describe an application of our method, which will also give the rationale for our evaluation approach. Consider the problem of designing a production line and assume that, at the strategic planning level, management intends to use this line for many models of the same products, both now and when future products are launched. The sequence of steps in production is known (there are 10), but precise distributions of work content at each stage cannot be determined. Due to the nature of the products and the processes at each stage, they can only estimate the relative amount

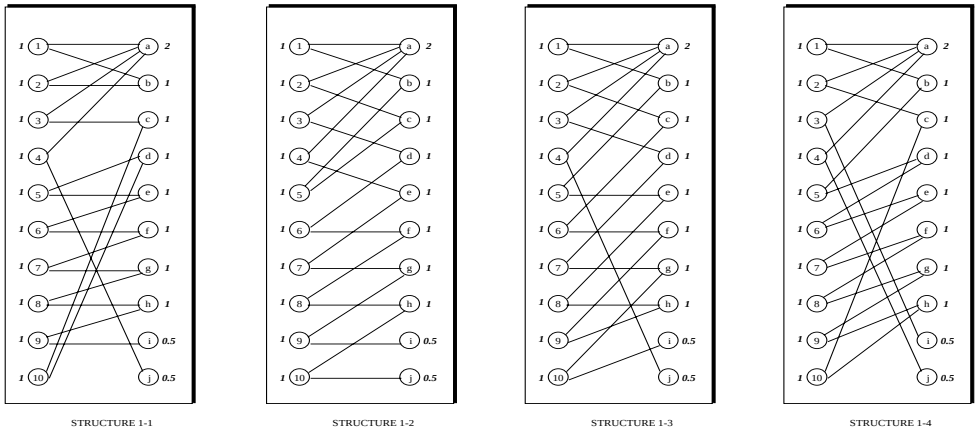

Figure 4: Structures that fit demand vector $\mathcal{D}_1 = (2, 1, 1, 1, 1, 1, 1, 1, 0.5, 0.5)$.

of work (i.e., average processing time for standard workers) as $\mathcal{D}_1 = (2, 1, 1, 1, 1, 1, 1, 1, 0.5, 0.5)$. The focus is on choosing a worker cross-training structure for a 10-worker, 10-station production line under a CONWIP release policy that robustly provides a high throughput under a wide range of WIP levels and workload variability assumptions. The worker cross-training must be chosen independent of the (tactical) decision of setting the WIP level, because it will vary as production schedules change over time. Suppose we desire a cross-training design which for practical reasons (e.g., training costs, learning time, walk times, etc.) uses exactly two skills per worker. Figure 4 shows 4 capability structures that fit the demand environment $\mathcal{D}_1$.

This is an example of a difficult design problem based on limited system data. We will show that our SF method can predict the best capability structure without the need for simulation to perform a rigorous evaluation. We have identified four additional demand vectors to provide a useful test suite within our computational power to benchmark for both closed serial and open parallel systems:

$$\begin{aligned} \mathcal{D}_1 &= (2, 1, 1, 1, 1, 1, 1, 1, 0.5, 0.5) \qquad \mathcal{D}_4 = (0.5, 0.5, 1, 1, 2, 2, 1, 1, 0.5, 0.5) \\ \mathcal{D}_2 &= (1.5, 1.5, 1.5, 0.5, 0.5, 0.5) \qquad \mathcal{D}_5 = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1). \\ \mathcal{D}_3 &= (1.5, 1, 0.5, 0.5, 1, 1.5) \end{aligned}$$

Cases $\mathcal{D}_1$ through $\mathcal{D}_4$ possess variation in demand rate, while $\mathcal{D}_5$ has a uniform demand across types. Figures 4, 5, 6, 7, and 8 show the test suites of structures for our five demand vectors.

For each environment, our test suite was devised to include multiple structures with unit capac-

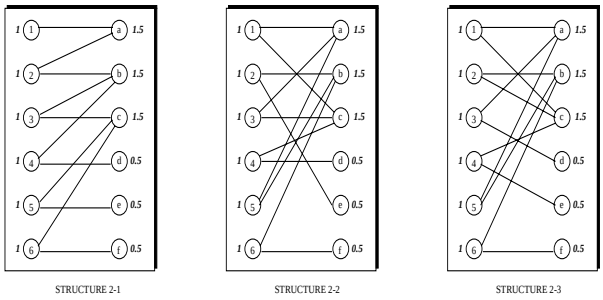

Figure 5: Structures that fit demand vector $\mathcal{D}_2 = (1.5, 1.5, 1.5, 0.5, 0.5, 0.5)$.

ity that fit that environment, and satisfy the condition that a source can allocate its effort equally across its capability set. For example, in Structure 1-1, sources 1, 2, 3, and 4 (each with capacity 1) have two capabilities and in every case one includes the capability to serve demand type 1. Demand type 1, having a magnitude of 2 can be met under our condition that each source must devote 50% of its effort to demand type 1. This approach also ensures that the structure fits the environment. This condition is consistent with the approach taken in Aksin and Karaesmen (2002). Note that while fitness of a structure tends to balance capacity among demands, this condition is made to balance the effort of production sources among their capabilities (arcs). We would like; however, to emphasize that, in half of our test cases (i.e., models with shocks to the arrival rates (parallel cases) or process times (serial cases)), this assumption is violated, since the sources will allocate their effort asymmetrically across their capabilities over time in order to respond to the shocks.

Table 1. Eigenvalue, mean, and number of arcs indices for patterns under study.

Demand Vector	Index	Patterns								
		1	2	3	4	5	6	7	8	9
$\mathcal{D}_1$	Eigenvalue	11.33	11.85	14.73	17.32					
	Mean	1.12	1.16	1.40	1.66					
	# of arcs	20	20	20	20					
$\mathcal{D}_2$	Eigenvalue	6.97	7.65	8.61						
	Mean	1.14	1.22	1.83						
	# of arcs	11	12	12						
$\mathcal{D}_3$	Eigenvalue	7.53	7.61	8.34	8.46	9.58				
	Mean	1.22	1.22	1.33	1.33	1.50				
	# of arcs	12	12	12	12	12				
$\mathcal{D}_4$	Eigenvalue	11.46	12.80	21.04	21.71					
	Mean	1.12	1.22	2.10	2.16					
	# of arcs	20	20	26	28					
$\mathcal{D}_5$	Eigenvalue	16.00	18.25	20.93	23.68	24.00	26.17	32.00	34.12	40.52
	Mean	2.00	2.25	2.53	2.91	3.00	3.25	4.00	4.25	5.06
	# of arcs	16	20	24	26	24	28	32	36	42

The test suite includes "tough cases" in which structures have the same number of total skills and almost the same number of skills per worker. Structures for demand vectors $\mathcal{D}_1$ and $\mathcal{D}_3$ have

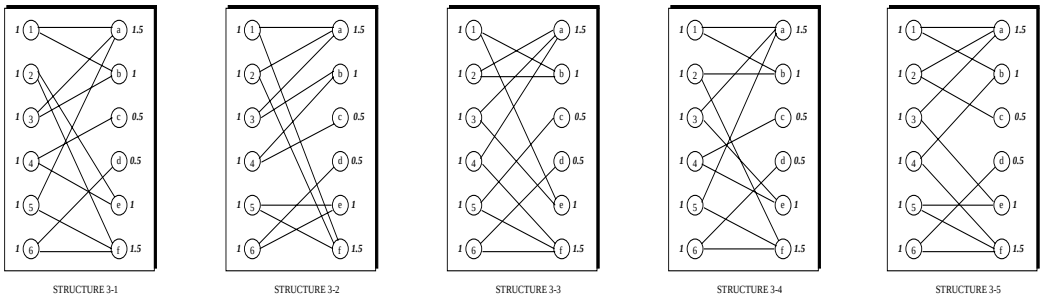

Figure 6: Structures that fit demand vector $\mathcal{D}_3 = (1.5, 1, 0.5, 0.5, 1, 1.5)$.

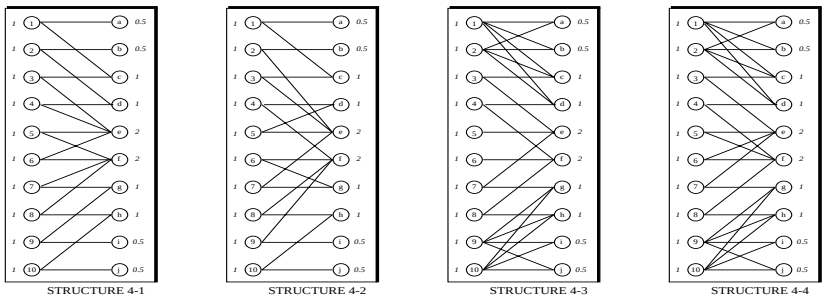

Figure 7: Structures that fit demand vector $\mathcal{D}_4 = (0.5, 0.5, 1, 1, 2, 2, 1, 1, 0.5, 0.5)$

precisely the same total number of capabilities, which requires a powerful index to truly capture the interconnection effects, not merely the number of arcs. These cases are typical of design problems with a fixed "budget" on the number of capabilities. Structure 2-1 for demand vector $\mathcal{D}_2$ has 11 arcs, but 2-2 and 2-3 both have 12. Cases $\mathcal{D}_4$ and $\mathcal{D}_5$ include a greater number of structures, as they allow the number of capabilities to vary widely. The eigenvalue, mean, and number of arcs indices of the structures are given in Table 1. Since the JG index is obtained using the probability distribution of demand and service times, it changes as the CV of the demand interarrival times or service times changes. Table A in on-line Appendix IV presents the JG indices for our test suites. Note that those indices are for the parallel queueing environment, since the JG index does not apply to closed serial environments. Because the 5 demand vectors and their 25 associated structures are often very difficult to distinguish, these examples are sufficient for us to make our point when tested under a variety of operating conditions.

5.2 Environments and Topologies in the Test Suite

Our numerical study examines the performance of our indices in two fundamentally different stochastic environments: (i) the open parallel queueing environment, and (ii) the closed serial queue-

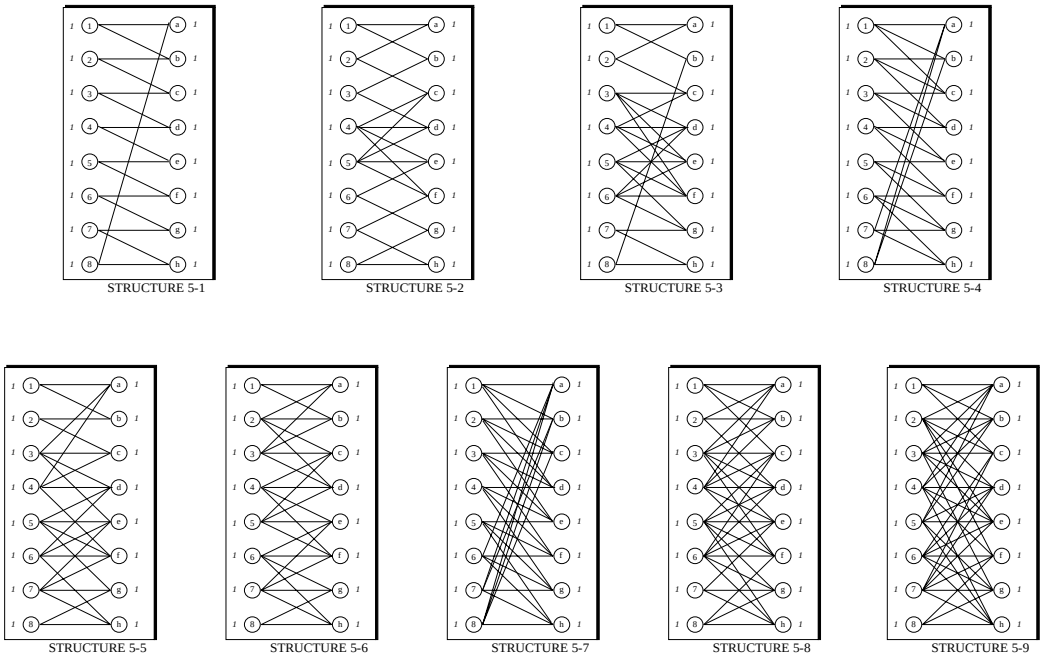

Figure 8: Structures that fit demand vector $\mathcal{D}_5 = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$

ing environment. In all simulations, processing times and interarrival times were generated from a Gamma distribution, which can accommodate any CV.

5.2.1 Open Parallel Queueing Environments

This environment is chosen to represent the many parallel operations which are demand-constrained make-to-order systems. We think of applications such as call centers, paperwork processing in office environments, manufacturing workstations in a job shop, etc. The demand arrival rate vector is $\mathcal{D}_i$, and we used mean process times of 0.9 for all demand types to achieve an average system utilization of 90%, which is in our experience the most practical level in many applications. Moreover, we test shock models that allow utilization to vary in a complex way, thoroughly exercising the transient dynamics without causing instability (see Section 5.2.4). To ensure that our results were not dependent on 90% utilization, we repeated the parallel simulation test suite for 70% utilization (via mean processing times of 0.7). As a widely applicable performance measure, we use average waiting time as our metric. The structure with smaller average demand waiting time will be considered more flexible; however, by Little's Law this is equivalent to minimizing average queue length.

5.2.2 Closed Serial Queueing Environments

The closed serial queueing environment can model a production line that employs a CONWIP release policy to limit the number of jobs (WIP) in the line. Our test examples range from lines with 6 workstations and 6 workers to 10 of each. Each workstation requires a unique skill. Demand vector $\mathcal{D}_i$ denotes the average processing times. Since there is no exogenous demand process, setting the WIP level is analogous to setting the utilization level of the open system. We tested WIP levels corresponding to 2, 6, 10, and 14 jobs per worker. System throughput is the metric of flexibility, and thus systems with higher throughput will be considered more flexible.

5.2.3 Control Policies

In both the closed serial and open parallel environment simulations, we use the "longest queue" policy, which directs an idle source to process the demand type (within its skill set) that has the largest number of waiting jobs. When an arrival finds two or more workers available, the worker that has been idle the longest is selected. We selected the longest queue policy for several reasons. First, the longest queue policy is one of the few policies that can be applied sensibly with any structure and any network flow. Second, Hopp et al. (2004a) have extensively tested a broad suite of policies in CONWIP lines under 2 different cross-training strategies, namely "cherry-picking" and "2-skill chaining." Comparison to both simulations and MDP models revealed the robust effectiveness of the longest queue rule (also called MaxQueue rule). It was found to be, on average, within 3% of optimal in achieving throughput. Third, the longest queue policy is intuitive, easily implemented, and widely used in industry. (See Van Houtum et al. (1997), Van Mieghem (2003), and Stolyar (2004) and references therein for a survey of past work and a recent analysis of the longest queue policy as well as the generalized longest queue policy.)

5.2.4 Uncertainty in the Environments of the Test Suite

Since flexibility is the system's ability to respond to changes in the environment, our simulation study incorporates randomness through the following two mechanisms:

1. Variability: A major source of uncertainty is variability in service times and demand interarrival times. For both of these sources, our simulation study incorporates two different coefficients of variation: $CV = 1$ and $CV = 2$. This creates four different scenarios for the parallel environment in

which the CV for demand interarrival times and service times are all set to be 1 or 2. In the closed (CONWIP) serial environment, it creates 2 scenarios, because there is no exogenous demand.

2. System Shocks: While variability of i.i.d. random variables models short-term fluctuations, the shock models force average demand arrival rates and average job processing times to change over time. The shocks in the CONWIP line represent shifts in service capacity by adjusting mean job processing times, while shocks in the parallel system can be thought of as either demand shifts or, as in the serial case, processing time changes .

We simulate these systems by sequentially providing a shock of type (i, j) for $i = 1$ to N and for $j = 1$ to N. Each shock is preceeded by an equilibrium condition (90% or 70% worker utilization in parallel cases). A shock of type (i, j) with $i \neq j$ boosts type j demand with an absolute rate increase of 0.075 (0.225 in the 70% utilization case) while also dropping type i by 0.075. Shocks of type (i, i) simply boost the demand for type i by an absolute rate of 0.075 (0.225). This represents an average shift in worker utilization of 8.3% (32%). The high-utilization shock periods generate large queues in the open parallel case, so a long shock/equilibrium period of 5,000 jobs was appropriate. On the other hand, the CONWIP system is tightly coupled and queues cannot exceed the WIP level, so the shock and equilibrium periods were ended after 1,000 job completions.

5.3 The Evaluation Process

Our evaluation process is based on pairwise performance rankings of test structures that fit a particular demand vector $\mathcal{D}_i$, $i = 1, 2, 3$. For every pair, we used the four indices to predict the more flexible structure. To base our results only on significant cases, we threw out any comparisons for which the performance outcomes (waiting time in parallel systems and throughput in CONWIP systems) were less than 0.1% different. Clearly, the issue of predicting better performance is almost meaningless once the environments allow two alternatives to achieve performances within 0.1%, especially in light of the fact that these cases only occurred in serial systems in which the realized throughput had almost achieved the capacity asymptote. This threshold also prevents us from including cases in which the confidence intervals on the performance estimates are too close to determine the ranking. If, and only if, a simulation passes the 0.1% criterion and also contradicts the prediction of the index, we count that as a prediction error and calculate the percentage relative

error $\Delta\%$. In the parallel environment the percent error is based on mean waiting time as follows:

$$\Delta = \frac{|Z_{SF} - Z_{Sim}|}{Z_{Sim}}, \tag{1}$$

where Z_{SF} is the simulated performance of the structure chosen by the index, while Z_{Sim} denotes the truly better performance determined by simulation. Since larger throughput is more desirable in a CONWIP line, equation (1) is modified to take the percent error with respect to Z_{SF}.

Our simulation was written in the C++ language. For models without shocks, runs typically ended after 20,000 jobs exited the line, in addition to a warm-up period of 1,500 jobs. Each run was replicated between 25 and 2,000 times, confidence intervals were computed at the 97.5% level, and these allowed us to use the 0.1% significance threshold. With two CV levels for both interarrival and service times and two utilization levels, we have a total of 8 regular (non-shock) scenarios plus the same 8 scenarios under the shock model. For the serial CONWIP environment, we tested four WIP levels at two CV levels — 16 closed serial scenarios when the regular and shock models are combined. Each demand vector (with say n structures) has a number of possible pairwise comparisons ($n(n-1)/2$ comparisons). Since each pairwise structural comparison was tested in these diverse operating environments (i.e., Serial, parallel, regular, shock, different variabilities, different WIP levels, etc.) our experiment included 1585 pairwise comparisons which exceeded our 0.1% difference criterion. Note that in the worst case, the uncounted cases (whose performance differences are insignificant) do not significantly change the maximum error percentages while the average errors would actually improve.

Tables 2 and 3 summarize the results, where the average and maximum percent errors are calculated conditioned upon an error outcome. To calculate the errors in Tables 2 and 3, whenever an index was the same for two structures, we do not count it as a ranked case (the index comparison is indeterminate). Therefore, "% Ranked" in the tables indicates the percentage of valid comparisons an index can distinguish. For this reason, all indices except the eigenvalue and JG indices, which rank all cases, have less than 100% of the comparisons ranked. In this light, the error measures must be interpreted carefully. For example, it is misleading that the number of arcs overall average prediction rate is larger than that of the eigenvalue index, because the number of arcs index fails to distinguish the most difficult cases (i.e., those with equal numbers of capabilities). This is because the average prediction rate of the number of arcs index was computed over 67.13% of the valid

comparisons, while the average prediction rate for the eigenvalue is over all (100%) of them. The number of arcs ignores how those arcs construct a structure. In the next section, we present an example that clearly shows a case where the number of arc index fails even when the two structures differ in the number of arcs.

Table 2. Performance evaluation of the Eigenvalue and the Mean indices.

Demand Vec.	Sim. Environment	Variability Scenario	Num. Comparisons	**Eigenvalue (100% Ranked)**			**Mean Index**			
				% Correct	Error Δ		% Ranked	% Correct	Error Δ	
					Ave.	Max.			Ave.	Max.
$\mathcal{D}_1$	Open Parallel	Regular	48	100.00%	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%
		Shocks	48	100.00%	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%
	Closed Serial	Regular	45	73.33%	1.17%	2.07%	100.00%	73.33%	1.17%	2.07%
		Shocks	38	78.95%	0.75%	1.43%	100.00%	78.95%	0.75%	1.43%
$\mathcal{D}_2$	Open Parallel	Regular	24	100.00%	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%
		Shocks	24	100.00%	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%
	Closed Serial	Regular	20	95.00%	0.57%	0.57%	100.00%	95.00%	0.57%	0.57%
		Shocks	20	95.00%	0.68%	0.68%	100.00%	95.00%	0.68%	0.68%
$\mathcal{D}_3$	Open Parallel	Regular	77	96.10%	0.18%	0.21%	83.12%	100.00%	0.00%	0.00%
		Shocks	78	92.31%	0.25%	0.41%	82.05%	100.00%	0.00%	0.00%
	Closed Serial	Regular	53	79.25%	1.56%	4.18%	77.36%	79.25%	1.56%	4.18%
		Shocks	52	78.85%	1.59%	4.27%	76.92%	78.85%	1.59%	4.27%
$\mathcal{D}_4$	Open Parallel	Regular	48	100.00%	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%
		Shocks	48	100.00%	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%
	Closed Serial	Regular	35	82.86%	1.62%	5.08%	100.00%	82.86%	1.62%	5.08%
		Shocks	35	82.86%	1.63%	5.07%	100.00%	82.86%	1.63%	5.07%
$\mathcal{D}_5$	Open Parallel	Regular	288	89.93%	2.31%	6.32%	100.00%	89.93%	2.31%	6.32%
		Shocks	286	90.21%	2.43%	6.29%	100.00%	90.21%	2.43%	6.29%
	Closed Serial	Regular	159	93.08%	0.71%	1.17%	100.00%	93.08%	0.71%	1.17%
		Shocks	159	93.08%	0.70%	1.13%	100.00%	93.08%	0.70%	1.13%
Total			**1585**	**90.91%**	**1.26%**	**6.32%**	**96.78%**	**91.53%**	**1.26%**	**6.32%**

Although some of the test structures were very close in performance, the performances of other structures in each environment are often significantly different. For example, for demand vector $\mathcal{D}_5$, the difference between the performance of the most and the least flexible structures is up to 111.11%. This number is 12.79%, 17.98%, 11.07%, and 26.22% for demand vectors $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$, respectively. We highlight several observations:

- Overall, Tables 2 and 3 show that the eigenvalue and mean indices resulted in the same average error (given an error was made) of 1.26% and maximum error of 6.32%. They also performed exactly the same for all demand vectors, except $\mathcal{D}_3$, where the mean index could not distinguish patterns 3-1 and 3-2 or 3-3 and 3-4. The eigenvalue index ranks all of the cases and achieves a 90.9% rate of correctly ranking pairs. The mean index predicts correctly in 91.5% of the ranked cases, and 96.8% of the cases are ranked.
- The JG index, which is restricted to parallel systems, ranks all of the pairwise comparisons and performs the same as our eigenvalue and mean indices for Patterns $\mathcal{D}_2$ and $\mathcal{D}_4$. The JG index has the same or worse prediction rate than the eigenvalue index for all patterns except $\mathcal{D}_5$. In addition, it results in worse average and maximum errors compared with our indices. The eigenvalue and JG indices rank 100% of the cases; however, if we compute the prediction rate of the eigenvalue index restricted only to the parallel cases, then we find its prediction rate to be 93.2%, which is close to the 95.5% rate of the JG index.

- The number of arcs index ranked 67.1% of the tests (with a 95% prediction rate among them) with average and maximum errors of 1.56% and 6.32%, respectively. These are good error numbers, but they only apply to 67.1% of the tests. For these test cases, all methods perform about the same.
- The shock models represent highly dynamic environments such as those often seen in practice (e.g., new product launches, economic boom/bust cycles, call center peak/off-peak hours, etc.) The consistent performances of all the tested indices across regular and shock models indicates that the performance rankings are very consistent across a variety of operating conditions. We interpret these results to confirm our belief that a more flexible structure will perform better than a less flexible structure over a range of environments (such as variability levels, demand/load levels, etc.)

Table 3. Performance evaluation of the Number of Arcs and JG indices.

Demand Vec.	Sim. Environment	Variability Scenario	Num. Comparisons	**Number of Arcs Index**				**J G Index (100% Ranked)**		
				% Ranked	% Correct	Error Δ Avg.	Error Δ Max.	% Correct	Error Δ Avg.	Error Δ Max.
$\mathcal{D}_1$	Open Parallel	Regular	48	0.00%	–	–	–	95.83%	3.34%	3.75%
		Shocks	48	0.00%	–	–	–	95.83%	3.43%	3.99%
	Closed Serial	Regular	45	0.00%	–	–	–	N.A.	N.A.	N.A.
		Shocks	38	0.00%	–	–	–	N.A.	N.A.	N.A.
$\mathcal{D}_2$	Open Parallel	Regular	24	66.66%	100.00%	–	–	100.00%	–	–
		Shocks	24	66.66%	100.00%	–	–	100.00%	–	–
	Closed Serial	Regular	20	70.00%	100.00%	–	–	N.A.	N.A.	N.A.
		Shocks	20	70.00%	100.00%	–	–	N.A.	N.A.	N.A.
$\mathcal{D}_3$	Open Parallel	Regular	77	0.00%	–	–	–	87.01%	0.44%	1.13 %
		Shocks	78	0.00%	–	–	–	85.91%	0.45%	0.98%
	Closed Serial	Regular	53	0.00%	–	–	–	N.A.	N.A.	N.A.
		Shocks	52	0.00%	–	–	–	N.A.	N.A.	N.A.
$\mathcal{D}_4$	Open Parallel	Regular	48	83.33%	100.00%	–	–	100.00%	–	–
		Shocks	48	83.33%	100.00%	–	–	100.00%	–	–
	Closed Serial	Regular	35	82.86%	100.00%	–	–	N.A.	N.A.	N.A.
		Shocks	35	82.86%	100.00%	–	–	N.A.	N.A.	N.A.
$\mathcal{D}_5$	Open Parallel	Regular	288	97.22%	92.71%	2.09%	6.32%	96.53%	3.22%	6.98%
		Shocks	286	97.20%	93.01%	2.06%	6.29%	96.85%	3.28%	7.05%
	Closed Serial	Regular	159	96.86%	95.60%	0.63%	1.04%	N.A.	N.A.	N.A.
		Shocks	159	96.86%	95.60%	0.63%	1.04%	N.A.	N.A.	N.A.
Total			**1585**	**67.13%**	**94.98%**	**1.56%**	**6.32%**	**95.46%**	**2.33%**	**7.05%**

The JG index is a little more accurate than the SF indices in this test suite. However, we one should consider that: (*i*) The SF indices ignore the network flow topology and therefore applies to both open parallel and closed serial systems, while the JG index was designed for parallel systems only. (*ii*) In order to obtain JG index, one needs the detailed probability distributions for demand and capacity, while the SF indices do not use any information regarding capacity or demand (provided that structures fit their environment). (*iii*) The SF indices can be obtained using a deterministic maxflow algorithm. If capacity is not deterministic and demand is not normally distributed (as it is in Jordan and Graves [1995]), complex stochastic techniques or computer simulation is needed to calculate the JG index. Our experiments contribute further evidence of the accurate performance of the JG index. For a parallel queueing environments, when detailed information regarding the demand and capacity is available, and if computational effort and complexity is not an issue, the JG index is an efficient metric.

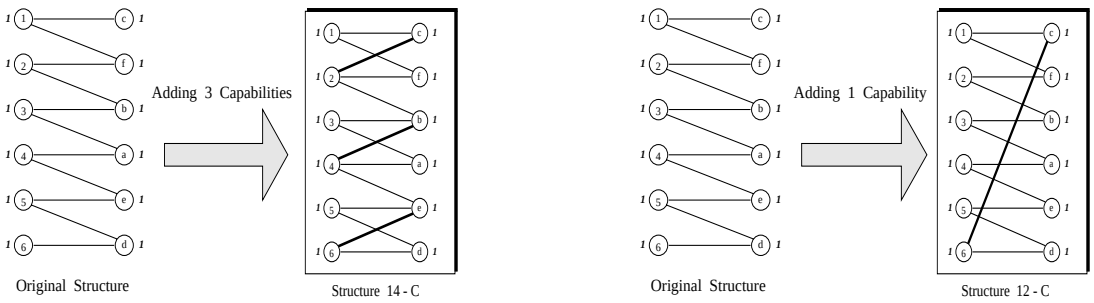

Figure 9: Structures 14-C and 12-C are constructed by adding 3 and 1 capabilities to the original structure.

5.4 The SF-Based Indices and the Number of Arcs Index

The number of arcs index can perform well when the alternatives differ in the number of capabilities. The SF matrix-based indices include not only the number of arcs, but also the interconnection structure. In general, if we compare the SF matrix of one structure with that of another and recognize that the first matrix has larger elements than another, we can immediately conclude that the first matrix has a larger eigenvalue or mean index (a generally known result in Meyer 2000).

Lemma 1 *Given structural flexibility matrices M and M', if $M \leq M'$ ($m_{ij} \leq m'_{ij}$ for all i, j), then $\mathcal{T}_{ei}(M) \leq \mathcal{T}_{ei}(M')$ and $\mathcal{T}_{me}(M) \leq \mathcal{T}_{me}(M')$.*

Intuition suggests that the flexibility of a system should increase as more capabilities are added to the system, which is captured in the following corollary (the proof of this and other technical results can be found in Appendix I). ‘

Corollary 1 *If a capability (arc) is added to a given structure, the structural flexibility indices, $\mathcal{I}_{ei}(M)$, and $\mathcal{I}_{me}(M)$ are non-decreasing.*

Although the SF indices recognize the benefits of having more capabilities, it is also true that a structure with more capabilities can sometimes have a lower flexibility than another one with fewer capabilities. Figure 9 shows two structures with 14 and 12 capabilities which are created by adding 3 and 1 new capabilities, respectively, to an original structure. As can be seen in the figure: (i) structure 14-C has two more capabilities than 12-C, (ii) each source node in 14-C has at least as many capabilities as the corresponding node in 12-C, and (iii) each demand node in 14-C is assigned to at least as many source nodes as the corresponding node in the 12-C.

Structure 14-C dominates structure 12-C with respect to all three of the above features, so it seems intuitive that 14-C would be more flexible. However, both SF-based indices conclude

that structure 12-C is more flexible than 14-C (e.g., eigenvalue indices are $\mathcal{I}_{ei}(M_{12-C}) = 12$ and $\mathcal{I}_{ei}(M_{14-C}) = 8.34$; while $\mathcal{I}_{me}(M_{12-C}) = 2$ and $\mathcal{I}_{me}(M_{14-C}) = 1.39$). To check its validity, we compared their performances in the open parallel and closed serial environments. In both environments 14-C outperformed 12-C in all cases tested (CV's equal to 1 and 2, and CONWIP WIP levels of 12, 36, 60, and 84), which was what both indices correctly predicted. This illustrates the fact that system architecture is important, so it is insufficient to determine flexibility based only upon (*i*) the number of capabilities of each source, (*ii*) the number of sources that cover a demand, or (*iii*) the total number of capabilities in the system.

5.5 D-Skill Chaining Structures

The chain structure in parallel systems (see Figure 1(C) and Structure 12-C of Figure 9 for examples) is analyzed and highlighted in Jordan and Graves (1995), Gurumurthi and Benjaafar (2001), and Sheikhzadeh et al. (1998), while Hopp et al. (2004a) emphasize chains in serial lines (see Figure 2(C)). Complementary to this earlier work, we devote this section to developing a deeper understanding of chain structures (which we call D-skill chaining) and their characteristics. While a formal definition is found in Appendix V, the structure is easily grasped by look at the examples in Figure 8 of chains with D = 2 (Structure 5-1), D = 3 (Structure 5-4), and D = 4 (Structure 5-7). Theorem 1 shows that the D-skill chain generates the maximum eigenvalue and mean indices for systems with a total of DN capabilities. To prove the theorem, we require the following lemma.

Lemma 2 *The structural flexibility matrix of the D-skill chain has value D in every element.*

Theorem 1 *In the class of structures with N demand types, N sources, and DN arcs, the D-skill chain achieves the maximum attainable eigenvalue index of DN, and mean index of D, for $D \geq 2$.*

It is a matter of practical importance to understand what value of D is needed for D-skill chaining to fit a particular demand environment, and we refer the reader to online Appendix V for a mathematical programming definition of the problem and an analytically described search procedure to easily find D.

5.6 Completing The Chain

Hopp et al. (2004a) used Markov decision processes and simulation to demonstrate a surprising performance benefit in N-station serial CONWIP lines when the last capability, number $2N$, com-

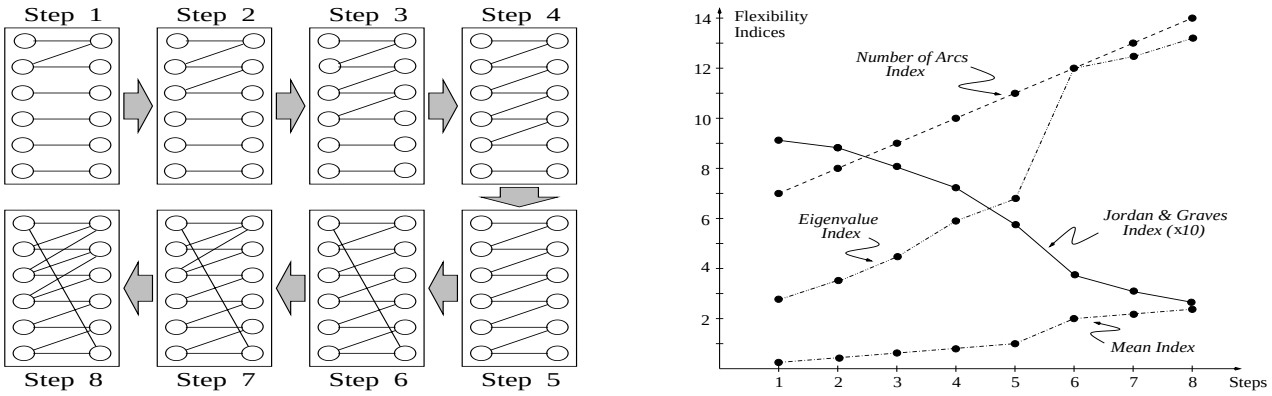

Figure 10: *Left:* The process of skill addition, *Right:* The effect of skill addition on flexibility indices.

pletes a 2-skill chain. We constructed the following experiment in order to: (i) extend their work to parallel systems, and (ii) show that the SF indices can recognize the effect of the chain-completing skill on system's performance (flexibility).

Our experiment starts with a specialist structure with $N = K = 6$. In each step we add one capability toward a chain (see Figure 10-Left). From Hopp et al. (2004a), we learned that the effect of completing the chain is maximized in a balanced line and minimized by the presence of a sharp bottleneck. To create a modest and practical case, we set the workload rate at station one 20% higher than the others, which are all equal. In addition, the most conservative way to add capabilities toward a chain structure to avoid exaggerating the benefit is in the order $(S_2, D_1), (S_3, D_2), \ldots, (S_1, D_N), (S_3, D_1), (S_4, D_2)$, as shown in Figure 10-Left.

Figure 10-Right shows how the four flexibility indices move in the direction of increased flexibility when an additional skill is added. For structures in Figure 10-Left we simulated the open parallel and closed serial environments under a variety of arrival and service process variabilities. As Figure 11 shows, the performance of both parallel and serial environments is very consistent with the behavior of our SF-based indices at every step, but not the number of arcs index. In particular, as the eigenvalue, mean, and JG indices predict, a significant improvement in system performance occurs in step 6 in both parallel (Figure 11-Left) and serial (shown in Figure 11-Right with a CV of 1 for comparison) environments. This is due to the addition of the chain-completing skill, which has been shown to have a significant effect on system performance.

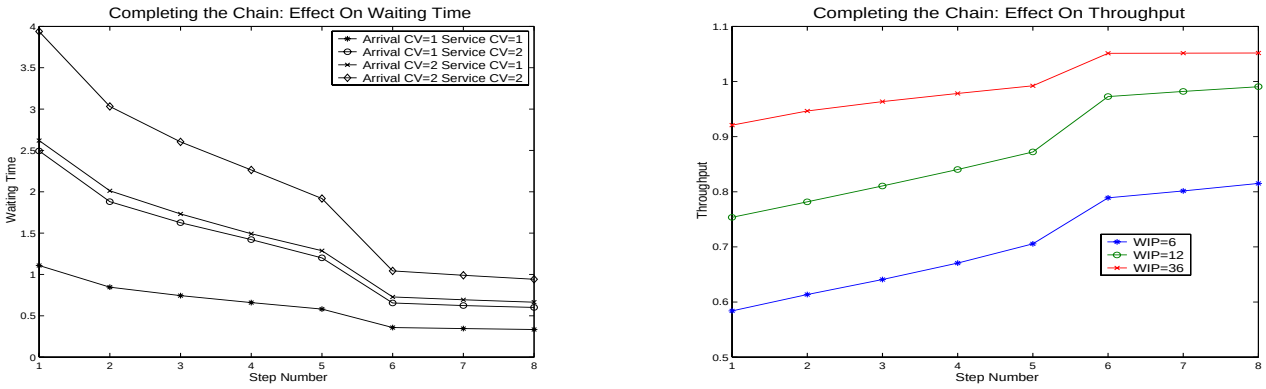

Figure 11: The effect of skill addition on: *Left:* Waiting time reduction in the parallel environment, *Right:* Throughput improvement in the serial environment.

6 Conclusion

In this paper, we have presented the structural flexibility method, a new methodology for assessing flexibility in production and service systems in the face of variability. We have devised the structural flexibility method to use a max flow algorithm to create a *structural flexibility matrix*. We have shown that with almost equal effectiveness the mean and eigenvalue indices, which are extracted from our structural flexibility matrix, capture important information about a structure and how well it can respond to variability.

Our experiments suggest that the structural flexibility indices are a good step towards evaluating the flexibility of alternative structures. The structural flexibility indices are purely deterministic metrics obtained independently of the topology of the system (i.e., open-parallel or closed-serial) and its variability. Nevertheless, our numerical studies suggest that when the first order capacity issues are resolved (i.e., capacity is balanced), the structural flexibility indices are powerful in predicting system performance rankings at different demand arrival or service process variability levels and shock conditions. Moreover, we have identified the striking ability of the structural flexibility indices to rank systems both in parallel as well as serial environments. This confirms our assertion that the contribution of a structure to flexibility depends very little on whether the environment is parallel or serial.

We summarize some of the managerial insights to the questions posed in the introduction.

1. For cross-training applications, we learned that it is not sufficient for a company to provide good access to training. System structure has a major impact on flexibility, and the detailed

structure is important. We have demonstrated that systems with fewer capabilities may outperform systems with more. We have shown how and why the structure of capabilities contributes to the robustness of a system's performance. The structural flexibility approach identifies structures that provide superior performance.

2. It appears very promising that simple algorithms can generate indices such as the eigenvalue or mean index to guide the decision of how to invest in flexibility at the strategic level in the absence of precise information. Intuitive and commonly used metrics such as number of capabilities may not lead to a good decision regarding the system structure. Furthermore, in many cases, restrictions such as a limited cross-training/flexibility budget limits the set of choices to structures that have the same number of capabilities. Under these circumstances, intuitive metrics such as number of arcs cannot distinguish between these choices.

3. Our experience simulating many systems and the analytical result of Theorem 1 indicate that using multi-functionality to connect structures is very beneficial to flexibility and adding a capability increases flexibility. In addition, the SF indices and our simulations confirm that a chain structure is a powerful form of cross-training from the perspective of flexibility, assuming the structure fits the environment. These insights further support and strengthen the work of Jordan and Graves (1995), Gurumurthi and Benjaafar (2001), and Hopp et al. (2004a), who have pointed out the power of the 2-skill chaining structure. Moreover, the SF indices agree with and augment the evidence of the importance of "completing the chain".

This initial success warrants further research to (i) attempt to extract more information from the structural flexibility matrix, and (ii) generate a more robust ranking approach that will compare systems with a richer description. For example, it is important to investigate how to move beyond the structural approach to a method that treats sources with different speeds. It would also be useful to address cases in which only a part of a team of cross-trained workers are trained for a new capability. Capacitated methods can help address the conjecture that an extra skill at a high capacity source is worth much more than at a low-capacity source.

Acknowledgments: The work of all three authors was partially supported by the National Science Foundation under Grant No. DMI-0099821. The authors thank Bora Kolfal and Gary Chao for performing some of the simulations in this paper.

References

Aksin, O.Z., and Karaesmen, F. 2002. Designing flexibility: Characterizing the value of cross-training practices, working paper, INSEAD, Fontainebleau Cedex, France.

Ahn, H-S., Duenyas, I., and Zhang, R. Q. 1999. Optimal scheduling of a two-stage tandem queue with parallel servers, *Advances in Applied Probability* **31**, 1095–1117.

Andradottir, S., Ayhan, H., Down, D.G. 2001. Server assignment policies for maximizing the steady-state throughput of finite queueing systems, *Management Science*, **47:10**, 1421–1439.

Askin, R.G., and Iyer, A. 1993. A comparison of scheduling philosophies for manufacturing cells. *European Journal of Operations Research.* **69** 438–449.

Bartholdi III, J.J., and Eisenstein, D.D. 1996. A production line that balances itself. *Operations Research.* **44**(1) 21–34.

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

Berman, O., Larson, R., and Pinker, E. 1997. Scheduling workforce and workflow in a high volume factory. *Management Science.* **43**(2), 158–172.

de Groote, X. 1994. The flexibility of production processes: A general framework, *Management Science*, **43:2**, 933–945.

Duenyas I., Gupta, D., and Lennon, T.M. 1998. Control of a single server tandem queueing system with setups, *Operations Research*, **46** 218–230.

Gans, N. and van Ryzin, G. 1997. Optimal Control of a Multiclass, Flexible Queueing System, *Operations Research*, **45** 677-93.

Gel, E.G.S., Hopp, W.J., and Van Oyen, M.P. 2002. Factors affecting opportunity of worksharing as a dynamic line balancing mechanism, *IIE Transactions*, **34**, 847–863.

Gel, E.G.S., Hopp, W.J., and Van Oyen, M.P. 2001. Opportunity of hierarchical cross training in serial production, Working paper, Arizona State University, Tempe, AZ.

Graves, S.C. and Tomlin, B.T. 2003. Process flexibility in supply chains. *Management Sciences* **49** 907–919.

Gurumurthi, S. and Benjaafar, S. 2001. Modeling and analysis of flexible queueing systems. Department of Mechanical Engineering, University of Minnesota, Minneapolis.

Harrison, J.M. and Lopez, M.J. 1999 Heavy Traffic Resource Pooling in Parallel-Server Systems, *Queueing Systems* **33** 339–368.

Hopp, W.J., and Spearman, M.L. 2000. *Factory Physics: Foundations of Manufacturing Management.* Second Edition, McGraw-Hill, Burr Ridge, IL.

Hopp, W.J., Tekin, E., and Van Oyen, M.P. 2004a. Benefits of skill-chaining in production lines with cross-trained workers, *Management Science*, **50:1**, 83-98.

Hopp, W.J., and Van Oyen, M.P. 2004b. Agile workforce evaluation: A framework for cross-training and coordination, to appear in *IIE Transactions*, **36**.

Iravani S.M.R., Posner M.J.M., and Buzacott, J.A. 1997a. A Two-Stage Tandem Queue Attended by a Moving Server with Holding and Switching Costs, *Queueing Systems* **26**, 203–228.

Iravani S.M.R., Posner M.J.M., and Buzacott, J.A. 1997b. U-shaped Lines with Switchover Times and Cost, Working paper, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

Jordan, W.J., and Graves, S.C. 1995. Principles on the benefits of manufacturing process flexibility. *Management Science.* **41:4**, 577–594.

Kulkarni, V.G. *Modeling and Analysis of Stochastic Systems,* Chapman & Hall, London, 1995.

Mandelbaum, A., Reiman, M. I. 1998. On pooling in queueing networks, Management Science, 44, 971-981.

McClain, J.O., Schultz, K.L., and Thomas, L.J. 2000. Management of worksharing systems. *Manufacturing & Service Operations Management.* **2:1** 49–67.

Meyer, C.D. *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics. Philadelphia, PA, 2000.

Sethi, A.K. and Sethi, S.P. 1990. Flexibility in Manufacturing: A survey, *The International Journal of Flexible Manufacturing Systems*, **2**, 289–328.

Sheikhzadeh, M., Benjaafar, S., and Gupta, D. 1998. Machine sharing in manufacturing systems: flexibility versus chaining. *International Journal of Flexible Manufacturing Systems*, **10:4**, 351-378.

Pinker, E., and Shumsky, R. 2000. The efficiency-quality tradeoff of crosstrained workers. *Manufacturing and Service Operations Management (M&SOM)*, **2:1**, 32-48.

Stolyar, A. L., 2004 MaxWeight Scheduling in a Generalized Switch: State Space Collapse and Workload Minimization in Heavy Traffic, *Annals of Applied Probability*, **14:1**, 1-53.

Van Houtum, G.J., Adan, I. and Van Der Wal, J. 1997. The symmetric longest queue system, *Communications on Statistics - Stochastic Models*, **13:1**, 105–120.

Van Mieghem, J. A., 2003. Due Date Scheduling: Asymptotic Optimality of Longest Queue and Generalized Largest Delay Rules, *Operations Research*, **51**, 0113–122.

Van Oyen, M. P., Gel, E.G.S., and Hopp, W.J. 2001. Performance opportunity of workforce agility in collaborative and noncollaborative work systems. *IIE Transactions*, **33:9**, 761-777.

Williams, R.J. 2000. On Dynamic Scheduling of a Parallel Server System with Complete Resource Pooling, in *Analysis of Communications Networks: Call Centers, Traffic, and Performance,* D. R. McDonald and S. R. E. Turner, eds., Fields Institute Communications.