An Overview of Tradeoff Curve Analysis in the Design of Manufacturing Systems

by
Gabriel R. Bitran
Reinaldo Morabito

WP #3806-95 January 1995

An Overview of tradeoff curve analysis in the design of manufacturing systems

Gabriel R. Bitran
Massachusetts Institute of Technology
Sloan School of Management

Reinaldo Morabito
Universidade Federal de São Carlos, Brazil
Dept. Engenharia de Produção

January 1995

Abstract

Uncertainty in manufacturing systems has long been the source of managerial complexity. In this paper we discuss the impact of different sources of uncertainty and present a methodology to assess their impact on system behavior. We introduce the concept of tradeoff curves as a characteristic of a manufacturing system and illustrate their use to make decisions concerning the amount and type of capacity necessary to manage the system efficiently, to assess the impact of products arrival and processing uncertainties, as well as the consequences of changes in throughput and product mix. The methodology is illustrated with an example derived from an actual application in the semiconductor industry.

Keywords: *tradeoff curve analysis, manufacturing system design, open queueing networks, optimization and performance evaluation*

1 Introduction

Authors have pointed out that modern manufacturing systems are becoming very complex to manage due to the wide variety of products competing for common resources, uncertainty in demand, and reduction of product leadtimes. One of the current paradigm in the manufacturing environment is to *reduce complexity*, that is, to "simplify life". Several alternatives have been proposed to simplify a manufacturing system and hence, reduce its complexity. They include: (i) *product classification*, (ii) *uncertainty reduction*, and (iii) *knowledge of system relationships*. The first defines more homogeneous product classes in order to treat them in a more specific way. The second tries to reduce uncertainty and thus, improve the capacity of predicting system behavior (more predictable systems tend to be less complex). Finally, the third emphasizes the importance of deeply understanding relationships in the system in order to simplify it.

In the next section we analyze these three alternatives with special focus on the last two ones (sections 2.1 and 2.2, respectively). We emphasize the connections among complexity, predictability and uncertainty of a system, and discuss endogenous and exogenous factors that contribute to uncertainty. We are particularly concerned on the effects of uncertainty reduction on the performance of the system. In general manufacturing systems can be seen as dynamic systems. As we reduce uncertainty, we also reduce the system variability and hence, we expect to obtain better performance measures (e.g., shorter product leadtimes, lower work-in-process). The effect of uncertainty reduction can then be evaluated considering the variation of those performance measures. In this context we can also visualize the *just-in-time* (JIT) strategy as the limit state of a dynamic system from which all uncertainty has been eliminated.

The manufacturing system is modeled as an *open queueing network* where the nodes correspond to the stations and the arcs connecting nodes correspond to the product flows between stations. Queueing network models have been applied to the design of manufacturing systems by several authors in this last decade; see e.g. the references cited in the recent surveys of Suri et al [18], Buzacott and Shanthikumar [11], Hsu et al [12], and Bitran and Dasu [2]; see also the discussion in Suri and De Treville [17]. Design decisions should consider the tradeoff among performance measures for different system configurations. One way to describe this tradeoff is with the so called *tradeoff curves*; we say that these curves are the *signature* of the system.

In this paper we emphasize the importance of tradeoff curves to the analysis of manufacturing systems such as *job shops*. In section 3 we discuss how to generate these curves: In section 3.1 we review performance measure evaluation, in sections 3.2 and 3.3 we present, respectively, the problem of minimizing WIP without adding resources to the system and the problem of minimizing resources without increasing system WIP, and in section 3.4 we discuss how to utilize the solutions of these problems to generate tradeoff curves. In section 4 we use tradeoff curves to analyze the effects of reducing network uncertainty (section 4.1), changing the throughput (section 4.2), and the product mix (section 4.3) of the network. In

sections 5 and 6 we extend this analysis to the cases where we have a finite set of discrete alternatives for capacity change and where we can not approximate each station as a single machine. Finally, in section 7 we present the conclusions of this paper.

In order to illustrate the presentation of this topic, we chose an example derived from a real situation of a job-shop system with 10 product classes and 13 stations. We generated different tradeoff curves to analyze this network as presented in sections 3, 4 and 5.

2 Complexity reduction

One of the procedures that contributes toward reducing the complexity of manufacturing systems is the *classification of products.* As we define more homogeneous classes, we may treat them in a more specific way. In marketing this procedure is known as market segmentation. Another well-known example is the classification of animals in zoology. Without classifying them into species we may say that they are living creatures, which is not very helpful for operational purposes. However, as we classify them into species we may be more specific about each one.

The concept of product classification plays an important role in the *group technology* approach in manufacturing; see e.g. Krajewski and Ritzman [13] and Kusiak [14]. Parts and products with similar characteristics are grouped into *families* (or *product classes*) and processed by dedicated groups of machines. These similarities may be in size, shape, raw materials, operations, sequence of operations, and other characteristics. The goal is to define product classes with similar processing requirements to minimize machine changeover and setup time.

Besides product classification, other alternatives to simplifying a manufacturing system include: (i) *uncertainty reduction* and (ii) *knowledge of system relationships.* In what follows we discuss in more depth these two alternatives.

2.1 Uncertainty reduction

The complexity of a manufacturing system can be measured in different ways. In a job-shop system, for example, we may measure complexity as the diversity of product routings in the network, or the diversity of processing times at the stations. Bitran and Sarkar [5] proposed the *predictability* of some characteristic of the system as a more powerful measure of complexity: *Less complex systems tend to be more predictable.* The authors observed that: (i) predictability reflects the impact of specific features of the system (e.g., the more similar the processing times, the more predictable is the system), (ii) predictability is a useful measure for managers who need to predict, for example, when a product will be completed at the shop, and (iii) predictability helps organizational learning. As we reduce uncertainty, we expect our ability of predicting the behavior of the system to increase.

There are many factors that contribute to uncertainty. One possible classification is: (i) *endogenous factors* and (ii) *exogenous factors.* Examples of endogenous factors are poorly

trained operators, machine breakdowns, maintenance failures, shortages, etc. These sources of uncertainty may be controlled, for example, by investing in labor training and process improvement. In general we have more control over endogenous factors than exogenous factors. But it is also often possible to manage the uncertainty of exogenous factors, as illustrated in the following example: Consider a product with total cycle time of 6 months (including design and production). The planning horizon for its demand forecast must be larger than its cycle time, let us say 12 months. Hence, its forecast uncertainty depends on a 12 month period. If we reduce the product cycle time to 3 months, we may also reduce the planning horizon, say to 6 months. Note that now the forecast uncertainty should be smaller, since it depends on a shorter period.

There are several creative ways to reduce uncertainty and therefore, simplify the manufacturing environment. An alternative explored in Bitran and Sarkar [5] is *products partitioning*, which relates to the concept of *focused factory* (Skinner [15]). Consider a manufacturing system with a large number of product classes. If we allocate all classes to a single production line, we may reduce system predictability due to the effects of class interference at stations. Bitran and Sarkar showed that, for a given system capacity, we may obtain better performance measures by appropriately allocating classes to different production lines (or focused factories) instead of allocating all classes to a single line.

The alternative of partitioning products to reduce uncertainty can also be applied to situations where the system has a purely deterministic behavior. Consider a manufacturing system with a large number of product classes arriving at a particular station, and assume that their interarrival times are deterministic. The repetition cycle of class arrivals at that station can be fully determined since the arrival process is deterministic; however, if the cycle length is large, an observer at the station may have the illusion of a random arrival process.

For illustration, consider a small example with only three product classes, named 1, 2 and 3, with interarrival times of 2, 3 and 5 time units at a certain station. The repetition cycle of class arrivals has a length of 30 time units (i.e., the minimum-common-multiple between 2, 3 and 5), presented in table 1. The first row indicates the time unit $t, t = 1, \ldots, 30$, and the second row lists the classes arriving at t. Observe that, despite of the simplicity of this example, it is not a trivial task to identify and memorize the repetition cycle of table 1. This phenomenon generates a perception of uncertainty at the station even if the sequence of class arrivals is perfectly predictable. A similar phenomenon occurs with *random generators* of computers, which generate *deterministic* sequences of numbers with long repetition cycles giving us the illusion of a pure random generation.

The manufacturing system as a queueing system

So far we have discussed the concepts of complexity, predictability and uncertainty of a manufacturing system, and remarked that uncertainty can be reduced in different ways. In order to assess the impact of reducing uncertainty we consider manufacturing systems that can be modeled as a queueing system. Consider the simple example of one product class and

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Class		1	2	1	3	1, 2		1	2	1, 3		1, 2		1	2, 3
t	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Class	1		1, 2		1, 3	2	1		1, 2	3	1	2	1		1, 2, 3

Table 1: Repetition cycle of class arrivals

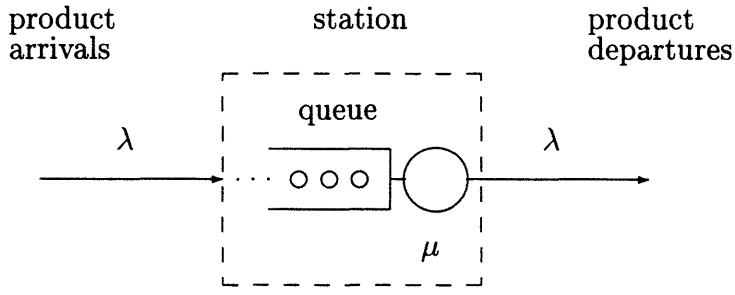

Figure 1: Single-stage queueing system

one station depicted in figure 1 as a single-stage queueing system (the queue corresponds to the waiting line of products at the station). We want to analyze the effects of reducing uncertainty to the mean product leadtime (queue time plus processing time).

Let A and S be two random variables representing, respectively, the product interarrival times and the service (or processing) times at the station. Denote by $E(A)$ and $E(S)$, and $V(A)$ and $V(S)$, their expected values and variances, respectively, and let $\lambda = \frac{1}{E(A)}$ be the mean product arrival rate and $\mu = \frac{1}{E(S)}$ be the mean service rate at the station. We assume that the system is *stable* (i.e., $\lambda < \mu$) and is in *steady-state*.

Consider initially that $\lambda = 0.5$ and $\mu = 1.0$ products per hour; therefore the mean utilization at the station, defined as $\rho = \frac{\lambda}{\mu}$, is equal to 0.5. If product interarrival times and processing times were deterministic and uniform (e.g., products arrive every 2 hours and are processed in 1 hour), we would never have waiting lines at the station. But as the variances $V(A)$ and $V(S)$ increase, we expect longer and more frequent queues. Furthermore, as the mean product arrival rate increases, the mean utilization also increases and the queues become even longer and more frequent. For instance, if instead of $\lambda = 0.5$ we have $\lambda = 0.95$ products per hour, the mean utilization jumps to 0.95. Figure 2 illustrates the *tradeoff curve* between the mean product leadtime and the mean utilization at the station. For simplicity, this curve was generated assuming that both the arrival and service processes are Poisson; therefore, the leadtime is defined as $\frac{1}{\mu(1-\rho)}$ and is asymptotic in the limit as ρ tends to 1

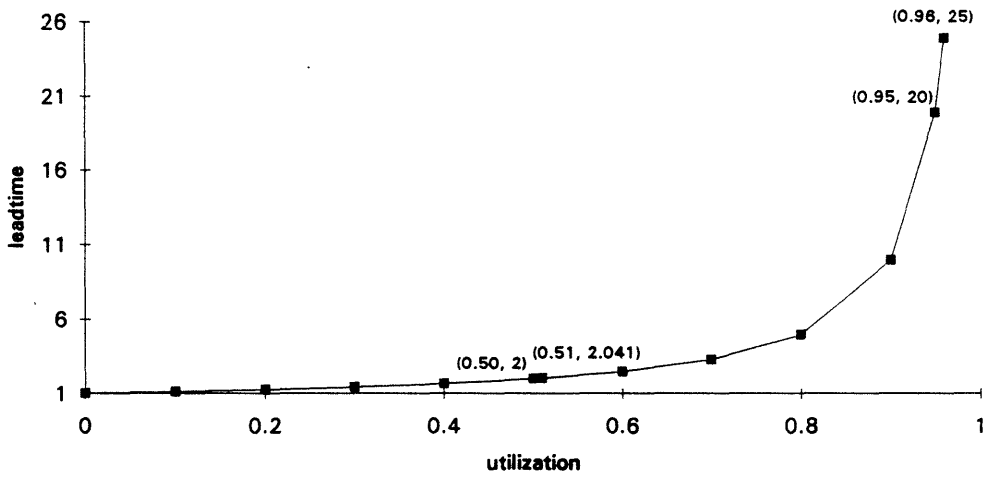

Figure 2: Tradeoff curve between mean product leadtime and mean utilization, and impact of small perturbations

(notice in the figure the leadtime jump from 2 to 20 hours as we increase the mean utilization from 0.5 to 0.95).

Impact of small perturbations

What is the sensitivity of the system to small perturbations? Perturbations may arise from unexpected tool failures, energy shortages, supply delays, etc., contributing to a small reduction in the mean processing rate (capacity) of the system. Note that as the capacity is reduced, the mean utilization increases for the same arrival rate λ.

Consider the two mean utilization levels $\rho = 0.50$ (with leadtime equal to 2 hours) and $\rho = 0.95$ (with leadtime equal to 20 hours) depicted in figure 2. Let Δ be a small increment of utilization due to unexpected loss of capacity. What are the new leadtimes if these two utilization levels are incremented by, let's say, $\Delta = 0.01$? The leadtime variation in the first case increases just 0.041 hours, but in the second case it increases 5 hours! Therefore, at high utilization, a small perturbation caused by unexpected events can trigger a great crisis in the system. The physics of manufacturing systems is not different from the physics of a dynamic system: A broken car at 3:00 am and a broken car at 5:00 pm may cause much different traffic jams in the same tunnel.

Impact of uncertainty reduction

Several manufacturing systems operate at high utilization levels due to high capacity acquisition costs. To illustrate, the *break-even point* of some semi-conductor factories corresponds to a utilization higher than 0.7. Under the same utilization, one may reduce product leadtime without adding capacity to the system by reducing the *variability* of the system.

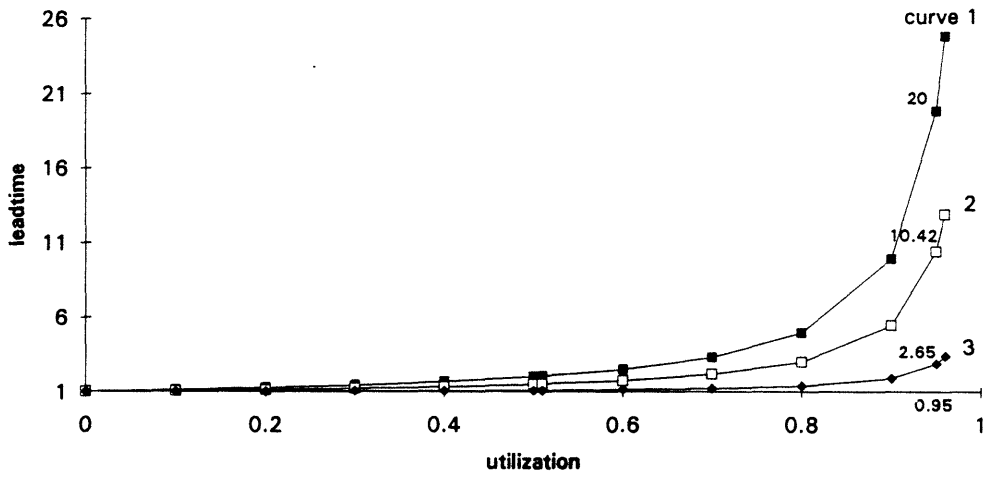

Figure 3: Impact of uncertainty reduction: Curve 1 ($ca = 1, cs = 1$), curve 2 ($ca = 0.5, cs = 0.5$) and curve 3 ($ca = 0.1, cs = 0.1$)

Let's consider again the example of figures 1 and 2. In order to manipulate a *dimensionless* measure of variability, let us denote by $ca = \frac{V(A)}{E(A)^2}$ and $cs = \frac{V(S)}{E(S)^2}$ the *squared coefficient of variation*, or simply the *variability parameters*, of A and S. Observe that ca and cs correspond respectively to the external (interarrival time) and internal (service time) variability at the station. The system in figure 1 can be approximately described with the 4 parameters: $\{\lambda, ca, \mu, cs\}$, which are then used as input data to generate the curve of figure 2. Authors have shown that performance measure estimates based on these 4 parameters lead to good approximations; see e.g. Whitt [21] and Buzacott and Shanthikumar [11].

As mentioned earlier, we may reduce the variances $V(A)$ and $V(S)$ by controlling the endogenous and exogenous factors. As we reduce $V(A)$ and $V(S)$ and hence the variability parameters ca and cs, we obtain "flatter" tradeoff curves than the one in figure 2. Figure 3 depicts three curves for different values of ca and cs and $\mu = 1$; these curves were generated based on Kraemer & Lagenbach-Belz's formulae [20]. Curve 1 is the same curve of figure 2 with $ca = 1$ and $cs = 1$ (Poisson process), and curves 2 and 3 were generated with $ca = 0.5$ and $cs = 0.5$, and $ca = 0.1$ and $cs = 0.1$, respectively. Note that, for the same mean utilization $\rho = 0.95$, we obtain very different leadtimes as a function of system variability (leadtimes of 20, 10.42 and 2.65 hours for curves 1, 2 and 3, respectively). In the limit as ca and cs tend to 0, the curve coincides with the horizontal axis, corresponding to a purely deterministic system with leadtime equal to the mean service rate $E(S) = 1$ hour for all $\rho, 0 \leq \rho < 1$. In this case we say that all system uncertainty was eliminated; note that this can be seen as a "perfect" JIT (*just-in-time*): the limit state of a dynamic system when all variability is removed.

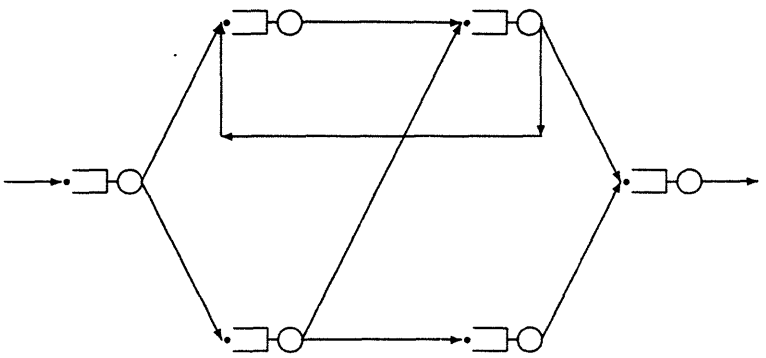

Figure 4: Open queueing network system

2.2 Knowledge of system relationships

It is much easier to complicate a system than to simplify it. To simplify a system requires a profound understanding of its structure and relationships among its components. The more we simplify a system the easier it is to understand it, hence creating a cycle that stimulates learning.

Queueing network systems

A manufacturing system is called *discrete* if products are processed individually or in batches. A large portion of the United States manufactured products are processed in discrete systems. These systems can often be modeled as *open queueing networks*, where nodes represent the stations and arcs represent the product flows between stations (figure 4). If the product arrival process and/or the service process at the stations are stochastic, we may have waiting lines of products in front of the stations. In general queueing network systems are complex (i.e., poorly predictable) and the ideal is to simplify them, or if possible avoid them.

In many cases, however, queueing network systems can not be avoided, and several performance measures should be evaluated in order to analyze them. Examples of these measures are *work-in-process* (WIP), product leadtimes, number of finished products on time, capacity utilization, throughput, costs and capital investments. System designers and managers should establish parameters for these performance measures as a function of how to compete in the market, that is, as a function of the corporate strategies. In order to describe the relationships between these parameters, we can utilize *tradeoff curves* similar to those of figure 2 and 3. These curves (to be discussed in the following sections) can be used to analyze strategic objectives as a function of the investment required.

3 Tradeoff curve generation

Every manufacturing system has a *signature* describing the most effective way of allocating or removing its resources. As we mentioned earlier, we say that *tradeoff curves are the signature of the system.* They depict the relationship among performance measures at each station and for the whole network. For example, the curves presented in figures 2 and 3 relate to only one station and describe the tradeoff between the mean product leadtime and the mean capacity utilization at the station.

In this section we discuss how to generate tradeoff curves among performance measures, in particular, between the resources and the WIP of the network (this methodology can be easily extended to other performance measures, such as product leadtimes). For convenience, the resources are measured by a cost function of capacity defined below. In section 3.1 we discuss how to evaluate network performance measures; our focus is on WIP. In sections 3.2 and 3.3 we discuss respectively how to minimize network WIP without adding resources to the system and how to minimize the resources without changing network WIP. As we will see, the solutions to these problems correspond to points on the tradeoff curve. Finally in section 3.4 we discuss how to use these results to generate the remaining points on the curve.

3.1 Performance measures evaluation

Performance measures for open queueing networks can be evaluated by applying the so called *decomposition method* described below. This method computes performance measures assuming that the network is in *equilibrium* or *steady-state.* Consider that the following input data are given:

- λ'_k mean external arrival rate of product class k
- ca'_k squared coefficient of variation of external interarrival times of product class k
- n_{kl} station that produces the l-th operation in the routing of class k
- r number of product classes
- n_k number of operations in the routing of class k
- μ_j mean processing rate (or capacity) at station j
- cs_j squared coefficient of variation of the processing times at station j
- n number of stations in the network.

Tables 2 and 3 present these parameters for a manufacturing network example with 10 product classes and 13 stations. This is derived from a real example of a semi-conductor factory and was analyzed in Bitran and Tirupati [9]. For simplicity, we consider each station j as a single machine with mean processing rate μ_j. In section 6 we make some comments of how to extend this discussion and consider each station as a set of machines. In addition to the parameters above, table 3 also presents the parameters v_j, a_j and b_j to be defined below. Note that the network *throughput* (or mean production rate), defined as the sum of all class arrival rates, is equal to 10 products per time unit (see table 2) (recall that all measures are evaluated assuming that the system is in steady-state).

Class k	λ'_k	ca'_k	n_{kl}	n_k
1	1.0	0.500	1, 2, 4, 2, 9, 10, 11	7
2	1.0	0.500	1, 2, 5, 2, 8, 9, 10, 11	8
3	1.0	0.333	1, 2, 6, 4, 2, 9, 12, 11	8
4	1.0	0.333	1, 2, 7, 4, 2, 9, 10, 11	8
5	1.0	0.333	1, 2, 4, 12, 2, 9, 2, 13	8
6	1.0	0.333	1, 2, 5, 12, 2, 9, 7, 13	8
7	1.0	0.250	1, 2, 6, 12, 2, 8, 2, 13	8
8	1.0	1.000	1, 2, 3, 7, 4, 12, 2, 8, 6, 9, 2, 13	12
9	1.0	1.000	1, 2, 3, 5, 4, 6, 12, 2, 8, 2, 10, 6, 13	13
10	1.0	0.333	1, 2, 3, 6, 2, 4, 12, 7, 2, 9, 11, 5, 13	13
total	10.0			93

Table 2: Input data for the product classes of the network example

Station j	μ_j	cs_j	v_j	a_j	b_j
1	13.004	0.500	100	5.68	-51.69
2	27.778	0.250	1612	2.59	-50.40
3	3.160	0.333	733	74.77	-165.40
4	10.000	0.500	1052	6.93	-48.53
5	5.631	0.333	912	12.62	-49.73
6	9.225	0.250	1683	7.51	-48.54
7	5.999	1.000	1662	11.11	-46.67
8	4.500	0.333	1812	27.66	-87.11
9	10.000	0.333	1730	7.47	-52.27
10	5.711	0.333	1600	15.34	-61.30
11	5.441	0.333	1882	27.03	-102.94
12	7.440	0.500	1486	13.01	-67.74
13	7.502	0.500	3250	14.22	-74.67
total	115.391				

Table 3: Input data for the stations of the network example

Station j	λ_j	ca_j	ρ_j	L_j	W_j	F_j
1	10.0	0.492	0.769	1.974	197.377	288.325
2	25.0	0.601	0.900	4.298	6929.058	598.457
3	3.0	0.760	0.949	10.694	7838.522	223.823
4	7.0	0.608	0.700	1.569	1650.825	207.700
5	4.0	0.613	0.710	1.500	1367.930	120.093
6	6.0	0.583	0.650	1.118	1881.392	191.332
7	4.0	0.619	0.667	1.715	2850.717	119.836
8	4.0	0.665	0.889	4.403	7979.090	168.193
9	8.0	0.642	0.800	2.327	4025.622	224.300
10	4.0	0.662	0.700	1.489	2382.308	150.240
11	5.0	0.684	0.919	6.194	11656.346	240.055
12	7.0	0.614	0.941	9.226	13709.098	216.225
13	6.0	0.677	0.800	2.653	8620.970	240.110
total				49.160	71089.253	2988.689

Table 4: Parameters and performance measures of the network example

The decomposition method involves basically three steps. In the first step all products are aggregated into a single class, called the *aggregate class*, and its flow is analyzed through the network. The parameters $\{\lambda_j, ca_j, \mu_j, cs_j\}$ are determined for each station j from the initial parameters, where λ_j and ca_j denote respectively the mean product arrival rate and the squared coefficient of variation of product interarrival times at station j. These parameters are estimated considering the interference among product classes at the stations. In the second step performance measures of the aggregate class are evaluated for each station, analyzed as a single queueing system. Finally in the third step performance measures are evaluated for the network and for each product class by decomposing the aggregate class. For more details of the decomposition method, the readers are referred to Shanthikumar and Buzacott [16], Whitt [20, 22], Bitran and Tirupati [7], and Bitran and Morabito [3]. Table 4 presents the parameters λ_j and ca_j computed for the network example of tables 2 and 3 (the remaining columns of table 4 are defined below).

We estimate performance measures for each station j by substituting the 4 parameters $\{\lambda_j, ca_j, \mu_j, cs_j\}$ into the formulas of queueing theory. For example, the mean utilization at station j, defined as $\rho_j = \frac{\lambda_j}{\mu_j}$, can be easily calculated substituting λ_j and μ_j. We can also calculate the WIP (in monetary value) at station j, defined as:

$$W_j = v_j L_j(\lambda_j, ca_j, \mu_j, cs_j) \tag{1}$$

where v_j is a unit monetary value of an arbitrary product at station j, and $L_j(\lambda_j, ca_j, \mu_j, cs_j)$ is the mean number of products (in queue and in processing) at station j. Each value v_j is

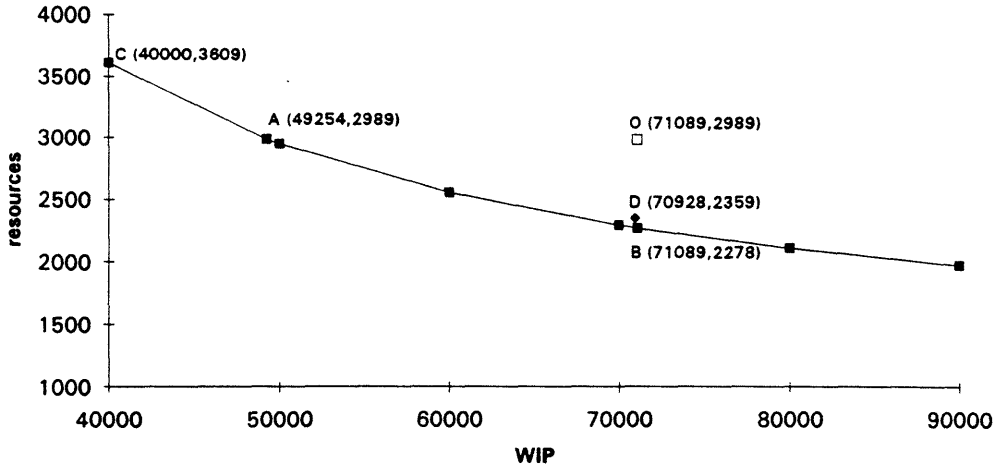

Figure 5: Points O, A, B, C, D and efficient frontier

estimated using practical experience, or as a weighted average proportional to the expected arrival rate and expected waiting time of each class (the expected waiting time may be computed approximately by a procedure given in Albin [1]). Obviously, if $v_j = 1$ then W_j corresponds to the mean number of products at station j. Each value $L_j(\lambda_j, ca_j, \mu_j, cs_j)$ can be estimated by the Kraemer & Lagenbach-Belz's formulae [20].

Table 3 presents the values of v_j and table 4, the computed values of the mean number of products, the WIP, and the mean utilization at each station j of the network example. Adding the WIP of all stations, we obtain the network WIP, equal to 71089 (see table 4).

Another useful performance measure is the value of the resources utilized in the system, which can be measured by the cost of capacity acquisition (or capacity investment). This cost is a function of the capacity at each station j, μ_j. An example of such a cost function is:

$$F_j(\mu_j) = a_j\mu_j^2 + b_j\mu_j + c_j \tag{2}$$

where a_j, b_j and c_j are known coefficients. We are assuming that it is possible to add capacity to station j by amounts small enough to consider μ_j as a continuous variable. In section 5 we analyze the more general case where capacity changes are limited to a finite set of discrete alternatives.

Table 3 presents the values of a_j and b_j for each station of the network example (for simplicity, we assume that $c_j = 0$). Adding the capacity cost of all stations, we obtain the network resources. Table 4 presents these values utilizing the data of table 3. Note that the WIP value 71089 and the resource value 2989 define the point O depicted in figure 5.

For convenience, we assume that the network capacity is *homogeneous* and *interchangeable* among stations. An example is a trained labor-force that can be transferred from one station to other. The algorithms discussed below can also be applied when the capacity of

Station j	ca_j	μ_j	ρ_j	L_j	W_j	F_j
1	0.492	10.604	0.943	8.609	860.861	90.541
2	0.602	28.041	0.892	3.966	6392.554	623.270
3	0.761	3.421	0.877	4.279	3136.301	309.222
4	0.610	8.712	0.804	2.587	2721.365	103.173
5	0.621	5.081	0.787	2.141	1952.440	73.096
6	0.589	7.818	0.767	1.787	3007.797	79.567
7	0.624	5.828	0.686	1.874	3115.223	105.356
8	0.665	4.999	0.800	2.369	4292.979	255.741
9	0.643	9.918	0.807	2.415	4178.749	216.376
10	0.666	5.296	0.755	1.891	3026.341	105.638
11	0.682	6.050	0.826	2.795	5260.484	366.620
12	0.611	8.403	0.833	3.101	4607.640	349.405
13	0.678	7.987	0.751	2.062	6701.743	310.684
total		112.158		39.876	49254.477	2988.689

Table 5: Parameters and performance measures relative to point A

a station is not transferable to all other stations. At the end of this section we discuss this more general case.

3.2 Efficient redistribution of resources

Let us assume that the system is at point O (71089, 2989). Considering the data of tables 2 and 3, we can formulate the following question: Is it possible to reduce the network WIP to under 71089 without adding resources to the network? In other words, is it possible to redistribute the resources of 2989 (by interchanging capacity among stations) such that the network WIP is reduced? And if this reduction is possible, what is the redistribution that leads to the minimum network WIP?

The *optimal resource redistribution problem* can be solved by an exact iterative algorithm proposed in Bitran and Sarkar [6]. This algorithm, here called *algorithm 1*, starts from the input data of tables 2 and 3 and solves a convex program at each iteration. In spite of utilizing this algorithm in this present paper, we will not describe it in more details; its complete description can be found in [6] and Bitran and Morabito [4].

Applying algorithm 1 to the network example, we obtain point A (49254, 2989) depicted in figure 5. Table 5 presents the final values of ca_j, μ_j, ρ_j, L_j, W_j and F_j for each station. Note that the values of ca_j in tables 4 (point O) and 5 (point A) are almost the same in spite of the capacity changes at stations.

Point A indicates that we can substantially improve performance (i.e., reduce the network

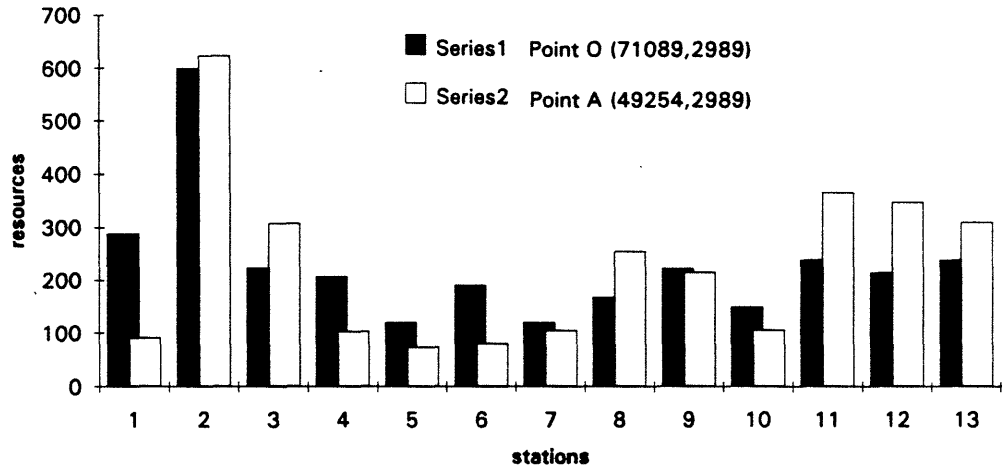

Figure 6: Resources for each station at points O and A

WIP from 71089 to 49254) without changing the resources (2989). This reduction is obtained by appropriately redistributing the resources among stations (compare tables 4 and 5); it does not imply in a process or technology modification. The throughput is also maintained, equal to 10 units of product per time unit (see table 2). Figure 6 compares, for each station, the resources before (point O) and after (point A) the redistribution.

As we move the system from point O to point A, we must "sell" capacity of stations 1, 4, 5, 6, 7, 9 and 10 in order to "buy" capacity for stations 2, 3, 8, 11, 12 and 13. Although the network capacity at point O, 115.4, is different of the network capacity at point A, 112.2 (compare tables 3 and 5), their cost are exactly the same: 2989. Figure 7 shows the impact of capacity change to the WIP of each station. Note that the WIP increases a little at stations 1, 4, 5, 6, 7, 9 and 10, but decreases substantially at stations 3, 8, 11 and 12.

Figure 8 compares the mean utilization at the stations of points O and A. Stations 3, 8, 11 and 12, with high utilization at point O, had their utilization reduced at point A. On the other hand, station 1 had its utilization substantially increased at point A (from 0.769 to 0.943); however, the effect on network WIP is not so large (from 197 to 861) since v_1 is small compared to other stations (see table 3 and figure 7).

3.3 Efficient redistribution of WIP

Let's assume that the system is again at point O (71089, 2989). Considering the data of tables 2 and 3, we can formulate the following (second) question: Is it possible to reduce the network resources to under 2989 without changing the network WIP of 71089? In other words, is it possible to redistribute the WIP of 71089 (by interchanging capacity among stations) such that the necessary network resources are reduced? And if this reduction is

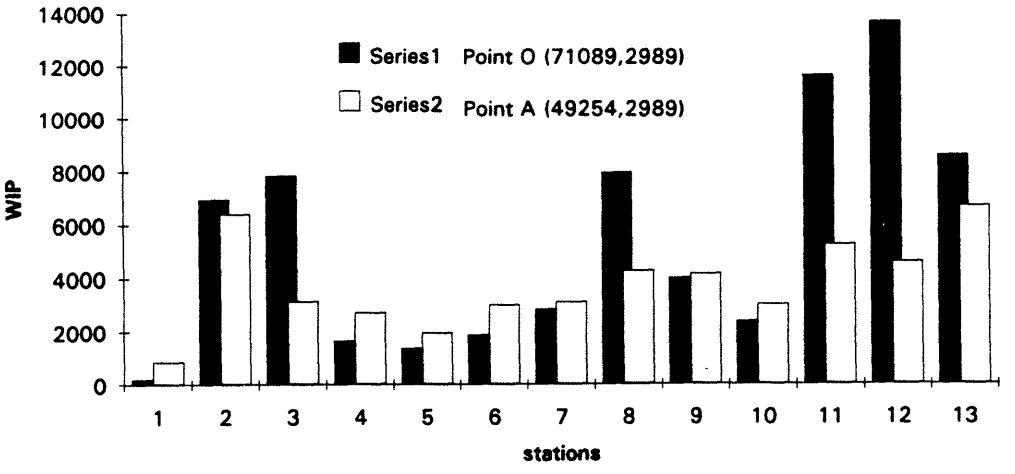

Figure 7: WIP for each station at points O and A

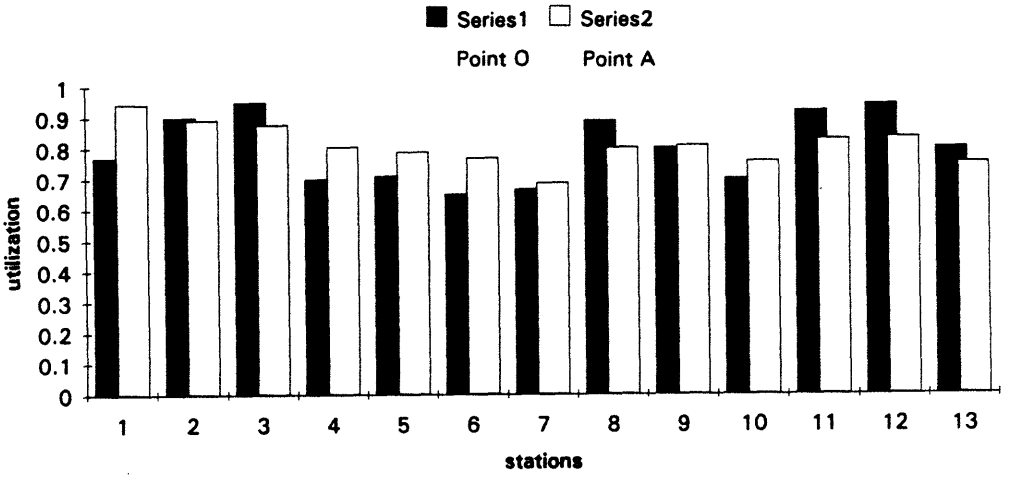

Figure 8: Utilization for each station at points O and A

Station j	ca_j	μ_j	ρ_j	L_j	W_j	F_j
1	0.492	10.390	0.962	13.114	1311.434	76.112
2	0.598	26.978	0.927	5.841	9415.943	525.354
3	0.760	3.275	0.916	6.355	4658.410	260.353
4	0.607	8.143	0.860	3.730	3923.519	64.327
5	0.616	4.720	0.847	3.039	2772.018	46.459
6	0.581	7.215	0.832	2.489	4189.145	40.719
7	0.617	5.255	0.761	2.686	4463.872	61.536
8	0.657	4.660	0.858	3.402	6163.572	194.739
9	0.638	9.270	0.863	3.465	5994.499	157.403
10	0.657	4.868	0.822	2.664	4262.773	65.111
11	0.672	5.690	0.879	4.047	7616.165	289.389
12	0.604	7.923	0.884	4.537	6741.802	279.960
13	0.668	7.330	0.819	2.946	9576.101	216.651
total		105.717		58.315	71089.253	2278.113

Table 6: Parameters and performance measures relative to point B

possible, what is the redistribution that leads to the minimum value of network resources?

The *optimal WIP redistribution problem* can be solved by an exact iterative algorithm, similar to algorithm 1. This algorithm, here called *algorithm 2*, solves a convex program at each iteration and is described in details in Bitran and Morabito [4].

Applying algorithm 2 to the network example, we obtain point B (71089, 2278) depicted in figure 5, which parameters and performance measures appear in table 6. Note that, similarly to point A, the values of ca_j at point B (tables 5) are very close to the values at point O (table 3) in spite of the capacity changes at the stations.

Point B indicates that we can reduce the network resources from 2989 to 2278 without changing the network WIP of 71089 (compare tables 4 and 6). Similarly to the efficient redistribution of resources, the efficient redistribution of WIP does not imply a process, technology, or throughput change. Figure 9 presents for each station the WIP obtained before (point O) and after (point B) the redistribution.

As we move the system state from point O to point B, we "transfer" WIP from stations 3, 8, 11 and 12 to other stations (compare points O and B in figure 9). This transference is obtained by appropriately interchanging capacity between stations such that the network WIP of 71089 is maintained. Figure 10 illustrates the resources at each station after the WIP transfer. Note that the resources increase a little at stations 3, 8, 11 and 12, but decrease by more than half of their initial values at stations 1, 4, 6 and 10. The mean utilization obtained before and after the WIP redistribution are depicted in figure 11.

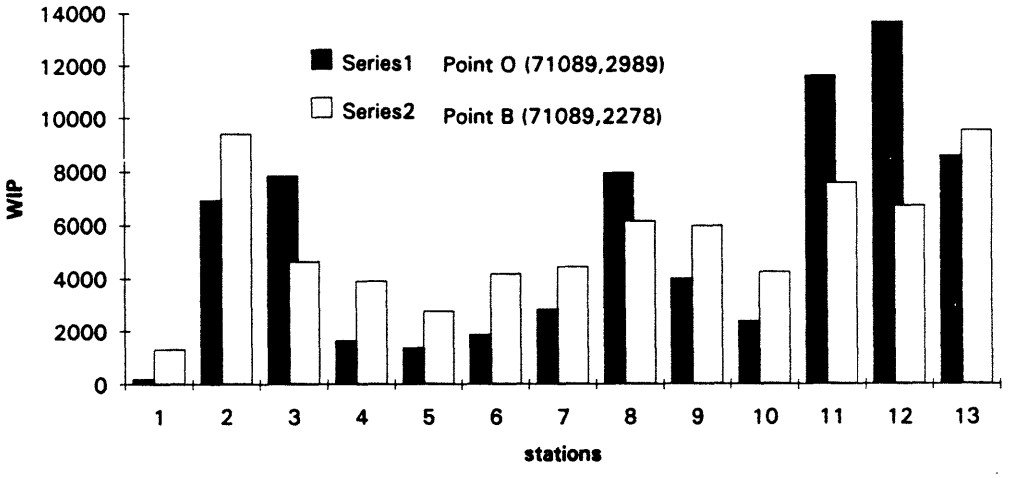

Figure 9: WIP for each station at points O and B

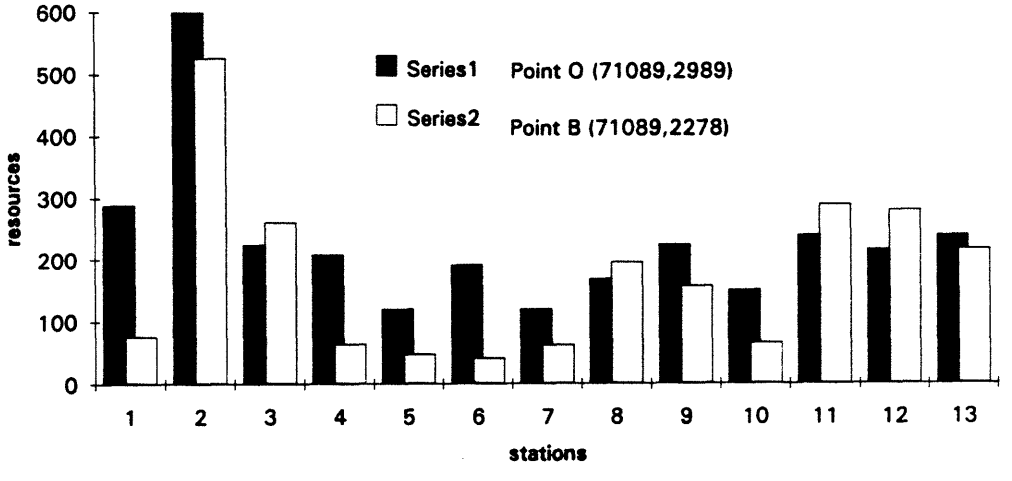

Figure 10: Resources for each station at points O and B

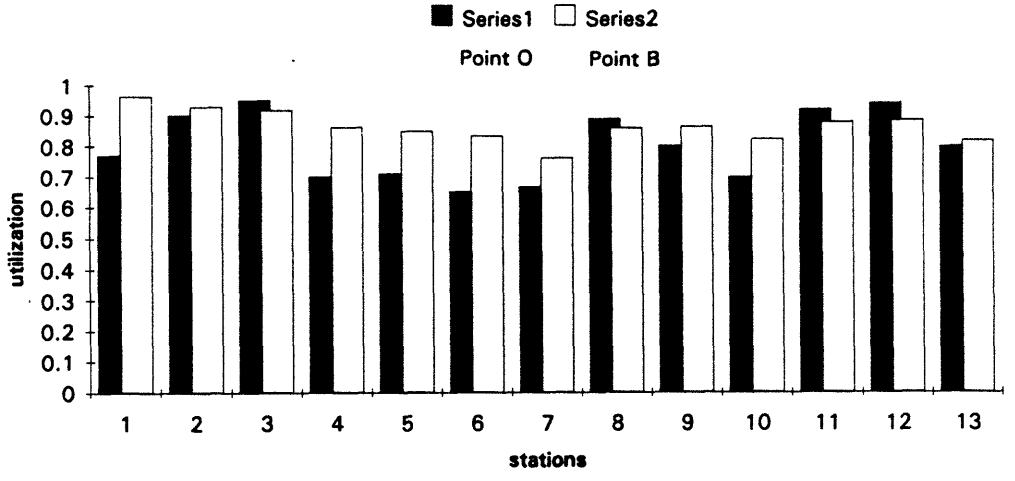

Figure 11: Utilization for each station at points O and B

3.4 Efficient frontier

The algorithms 1 and 2 move the system to a point on the tradeoff curve depicted in figure 5 (recall that points A and B belong to the curve). Algorithm 1 moves the system from point O to point A by efficiently redistributing the resources of point O, while algorithm 2 moves the system from point O to point B by efficiently redistributing the WIP of point O. The remaining points on the curve can also be obtained applying algorithms 1 and 2 for arbitrary values of network resources and WIP. In particular, the curve of figure 5 was traced applying algorithm 2 to the network WIP values 40000, 50000, ..., 90000 indicated in the figure (see the corresponding dots in the figure that originate the curve).

Alternatively, this curve can also be generated with less computational effort using a heuristic algorithm proposed in Bitran and Tirupati [8]. The algorithm assumes that the system is at a point on the curve, and employs a simple and intuitive *greedy heuristic* to find the remaining points. This procedure is illustrated in the following example: Consider that we want to add 100 labor hours of capacity to the stations of a network. For simplicity, assume that the cost of adding 1 hour to any station is constant, let's say \$1 (and hence, allocating 100 hours is equivalent to allocating \$100 to the network). The question is: How should we distribute this extra capacity to the stations such that the network WIP is minimized?

Given that we can add capacity to the stations in small quantities, let's partition these 100 hours in sufficiently small increments and add them, one after another, according to the following *greedy rule*: The next increment is added to the station that results in the largest reduction of the network WIP, and so on, until all increments have been added to the network. The smaller the increments, the more accurate is the solution generated by this procedure. The complete description of the heuristic algorithm can be found in Bitran

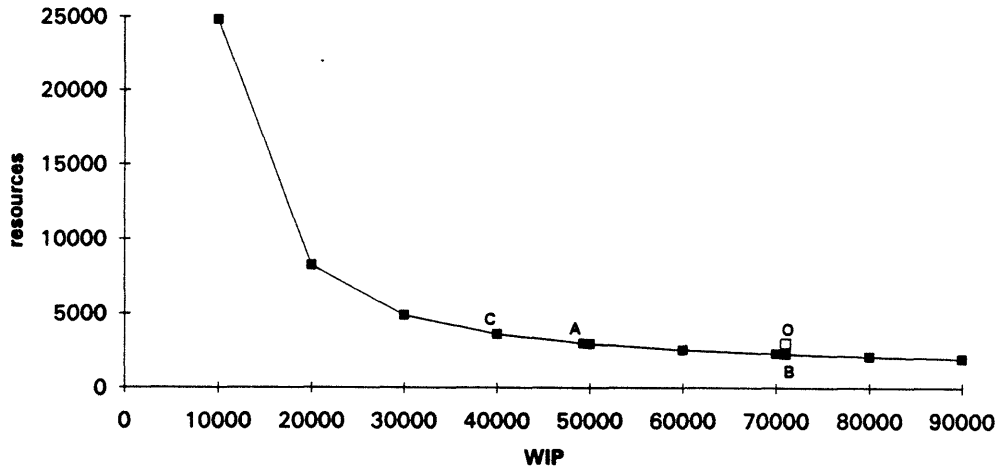

Figure 12: Points O, A, B, C, D and efficient frontier in a smaller scale

and Tirupati [8], and Bitran and Morabito [3].

Figure 12 presents the tradeoff curve of figure 5 in a smaller scale, showing its behavior as either resource or WIP value are reduced. This curve defines an *efficient frontier*, that is, the minimum resource value necessary to produce each WIP or, equivalently, the minimum WIP produced by each value of resources. Let's take for example point A (49254, 2989) and consider that, according to the competitive strategy, the system should operate with a WIP less than or equal to 40000. What is the minimum resource requirement to reduce the WIP from 49254 to 40000? As we travel through the points on the curve to the left of point A, we find point C (40000, 3609). Hence, the system needs an additional capacity investment of 620 (3609-2989).

Heterogeneous and non-interchangeable capacity

So far we have assumed that the network capacity is homogeneous and totally interchangeable between stations. Algorithms 1 and 2 can also be applied when part of the capacity μ_j at station j is not interchangeable; in this case it is enough to impose a lower bound on the variable μ_j. The more general case where capacity need not be homogeneous nor interchangeable involves additional considerations. If the unit cost of "selling" capacity (i.e., removing capacity) is equal to the unit cost of "buying" capacity (i.e., adding capacity) at each station, then algorithms 1 and 2 can be applied without any modification. Otherwise, they can be still applied but with some changes (see the discussion in Bitran and Morabito [4] for the particular case where the unit disposal cost of a station is a fraction of its unit purchase cost). In all cases the concepts of points A, B and the efficient frontier remain valid.

In the next section we assume that the efficient redistribution of resources (or WIP) has

been done and the system state is at a certain point on the curve of figure 5, let us say at point A (49254, 2989). In the preceding discussion we have shown that, starting from a point on the curve, we can reduce WIP by adding capacity to the network. In the sequel we discuss other alternatives to reduce WIP, such as uncertainty reduction. The next tradeoff curves to be presented were also generated with algorithms 1 and 2.

4 Changing the variability parameters, throughput and product mix

The tradeoff curve of figures 5 and 12 was generated with the data of tables 2 and 3, where the variability parameters (i.e., the ca'_k for all product classes and the cs_j for all stations), the throughput and the product mix remained fixed. We varied the capacity μ_j at each station and consequently, we varied the resources, the WIP and the mean utilization at each station. In this section we analyze what happens to that tradeoff curve as we change the variability parameters, the throughput and the product mix of the network.

4.1 Changing the variability parameters

In figure 3 we show that as we reduce the variability parameters ca and cs of a single-stage queueing system, we obtain "flatter" tradeoff curves between the mean utilization and the product leadtime at the station. In the limit as ca and cs tend to 0, every variability is eliminated, the curve tends to the horizontal axis, and the product leadtime tends to the expected value of the processing time at the station, $E(S)$. We can also extend this observation to queueing network systems. As we reduce the variability parameters $ca'_k, k = 1, \ldots, r$, and $cs_j, j = 1, \ldots, n$, we expect the same flattening effect of the tradeoff curves between network resources and WIP.

We may reduce the variability parameters, for example, by working closer with suppliers, investing in labor training, and process improvement. In this way we expect to obtain lower WIP levels without additional capacity investments. An immediate question is: Under what conditions uncertainty reduction produces better performance (e.g. lower WIP levels) than simply investing in capacity expansion?

Figure 13 presents the tradeoff curve of figure 5 (curve 1) next to three other curves generated with smaller values of ca'_k and cs_j. In the first (curve 2) we reduce by half all ca'_k values of table 2, in the second (curve 3) we reduce by half all cs_j values of table 3, and in the third (curve 4) we reduce by half all ca'_k and cs_j values.

Starting from point A (49254, 2989), the points $B1$ (45584, 2989), $B2$ (40645, 2989) and $B3$ (36948, 2989) can be obtained with algorithm 1, and the points $C1$ (49254, 2803), $C2$ (49254, 2537) and $C3$ (49254, 2351) with algorithm 2. Consider that the system is originally at point A and let's take, for example, the curve 4. Define V as the required investment to reduce by half all variability parameters. As we invest V, we move the system state

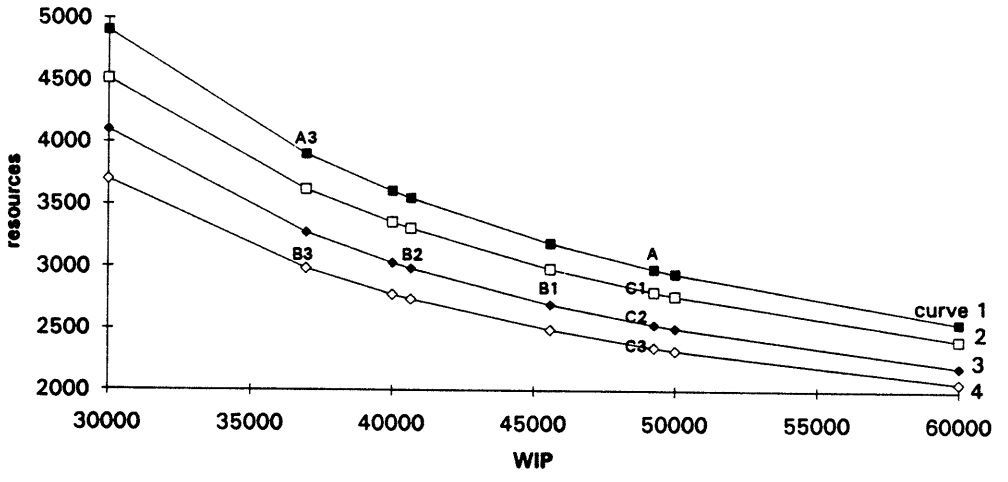

Figure 13: Changing the variability parameters: Curve 1 (ca'_k, cs_j), curve 2 $(\frac{ca'_k}{2}, cs_j)$, curve 3 $(ca'_k, \frac{cs_j}{2})$ and curve 4 $(\frac{ca'_k}{2}, \frac{cs_j}{2})$

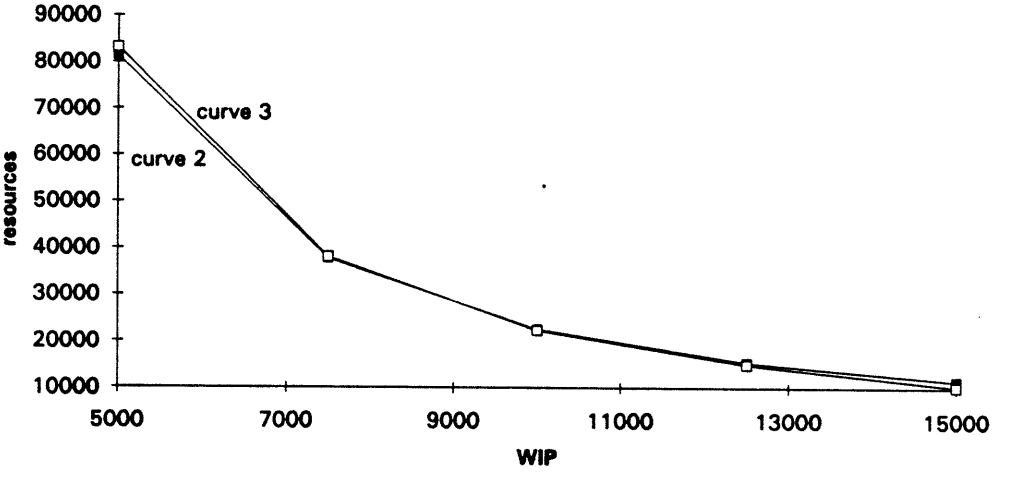

Figure 14: Variability parameter changes for small WIP values: Curve 2 $(\frac{ca'_k}{2}, cs_j)$ and curve 3 $(ca'_k, \frac{cs_j}{2})$

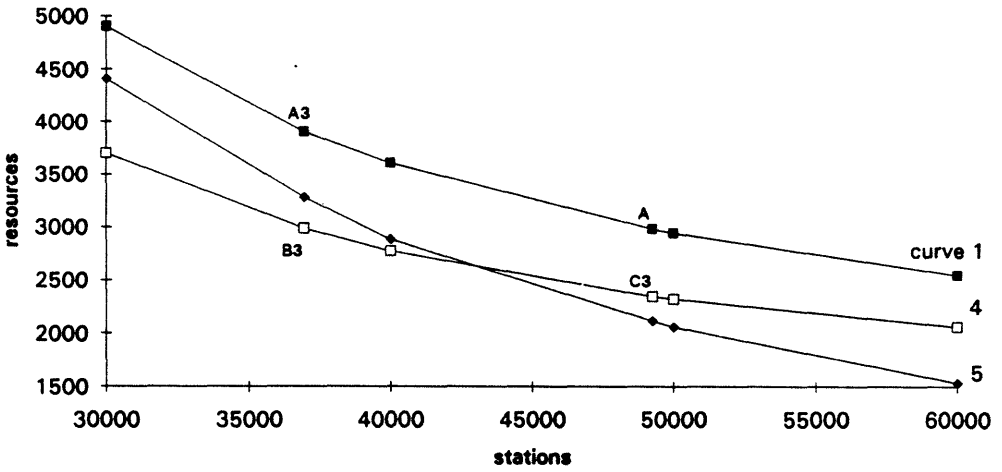

Figure 15: Tradeoff between uncertainty reduction and technology changing: Curve 1 (ca'_k, cs_j), curve 4 $(\frac{ca'_k}{2}, \frac{cs_j}{2})$ and curve 5 (new technology)

from point A to point $B3$ and hence, we reduce the WIP to 36948. This WIP level could also be attained by investing 918 (i.e., 3907-2989) in additional network capacity, instead of variability reduction, to reach point $A3$ (36948, 3907). The value 918 becomes an upper bound on the investment V. Note that with the curves of figure 13 at hand, we can now measure the tradeoff between investing in capacity *versus* investing in variability reduction.

Curve 4 is flatter than curve 3, which is flatter than curve 2, which is flatter than curve 1. For high utilization levels, the effect of reducing cs_j is more sensible than that of reducing ca'_k (compare curves 2 and 3). Nevertheless, we expect the inverse for low utilization. In order to illustrate this effect, figure 14 presents curves 2 and 3 for small WIP values (much less than 30000) and hence, low utilization at stations. Note that for WIP values less than the crossing point between curves 2 and 3, the effect of reducing ca'_k becomes more sensible than that of reducing cs_j.

Technology substitution

As we have seen, utilizing the tradeoff curves of figure 13 we can assess the tradeoff between adding capacity and investing in uncertainty reduction, without changing neither the technology, the throughput, nor the product mix of the network. Now let's assume that we have an alternative technology that allows us to produce the same mix of products at the same throughput rate. Figure 15 depicts its hypotethical curve (curve 5) together with the curves of the current technology (curve 1) and the current technology with uncertainty reduction (curve 4). These two last ones correspond respectively to the curves 1 and 4 of figure 13. Note that now we have a new tradeoff analysis: the tradeoff between buying this substitute technology *versus* investing in uncertainty reduction in the current system.

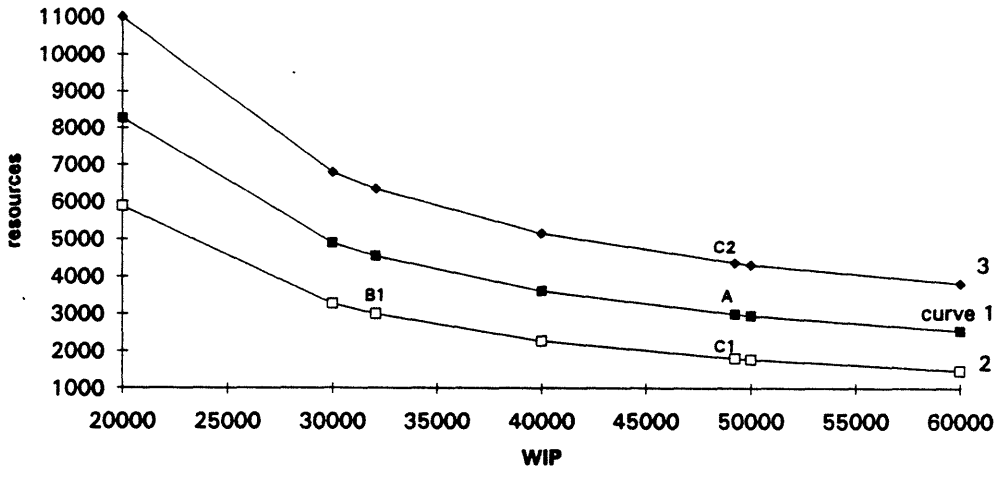

Figure 16: Changing the throughput: Curve 1 (10 products/hour), curve 2 (9 products/hour) and curve 3 (11 products/hour)

4.2 Changing the throughput

The tradeoff curves also helps the analysis of throughput changing in the network. Figure 16 presents curve 1 of figure 5, together with two other curves generated by varying the original network throughput, equal to 10 products per time unit (table 2). In the first (curve 2), we reduce by 10% the mean external arrival rates of all product classes in the network (so that the network throughput becomes 9 products per time unit), and in the second (curve 3), we increase them by 10% (11 products per time unit). Note in the figure that curve 2 is flatter than curve 1, while curve 1 is flatter than curve 3. The throughput variation apparently *translates* the curve and the smaller the throughput, the flatter is the curve.

Starting at point A (49254, 2989), we obtain points $B1$ (32079, 2989) and $B2$ (98107, 2989) (the latter does not appear in figure 16) with algorithm 1, and points $C1$ (49254, 1798) and $C2$ (49254, 4378) with algorithm 2. Consider again that the system is at point A and take, for example, curve 3. Note that it is unlikely that the system will survive the 10% growth of the throughput without additional resources (point $B2$). However, even a 50% increase of the current resources is not sufficient to maintain the same WIP level of point A (point $C2$).

4.3 Changing the product mix

The effects of changes in the product mix, such as removing old products, modifying the proportion among products, including new products, can also be analyzed with the tradeoff curves. Figure 17 presents curve 1 of figure 5, together with three other curves generated by modifying the product mix. In the first (curve 2) we eliminate product class 1 (i.e., $\lambda_1 = 0$),

Class k	λ'_k	ca'_k	n_{kl}	n_k
11	1.0	0.500	13, 1, 11, 3, 9, 5, 7	7

Table 7: Input data of product class 11

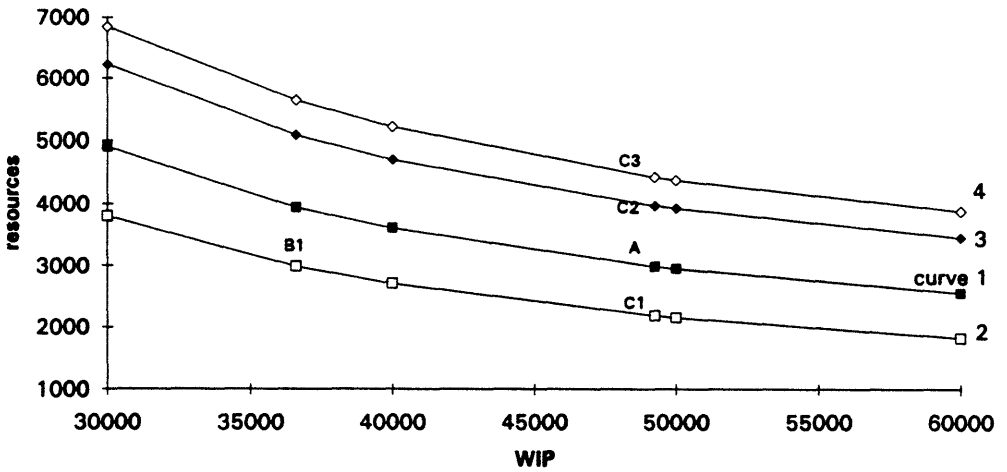

Figure 17: Changing the product mix: Curve 1 (original curve), curve 2 (class 1 deleted), curve 3 (class 1 duplicated) and curve 4 (class 11 added)

in the second (curve 3) we duplicate the mean arrival rate of product class 1 (i.e., $\lambda_1 = 2$), and in the third (curve 4) we introduce a new product class (class 11) with the same mean arrival rate of class 1 (i.e., $\lambda_{11} = 1$) but with a very different routing. Table 7 presents the input data for product class 11.

Note that curve 2 corresponds to a throughput of 9 products per time unit, while curves 3 and 4 to a throughput of 11 products per time unit. Curve 2 is flatter than curve 1, whereas curve 1 is flatter than curves 3 and 4. This result is consistent with our discussion of throughput changing in section 4.2. However, curve 3 is flatter than curve 4 in spite of having the same throughput (recall that in curve 3 we duplicate the mean arrival rate of class 1 and in curve 4 we introduce class 11 with the same mean arrival rate, variability parameter and number of operations of class 1). This shows that, in this example, the routing of class 11 produces higher WIP than the routing of class 1.

Starting from point A (49254, 2989), we obtain points $B1$ (36607, 2989), $B2$ (77768, 2989) and $B3$ (103211, 2989) (points $B2$ and $B3$ do not appear in figure 17) with algorithm 1, and points $C1$ (49254, 2184), $C2$ (49254, 3973) and $C3$ (49254, 4435) with algorithm 2. Note that if we eliminate product class 1, we reduce the mean utilization of the original network, and the effect on network WIP corresponds to the horizontal distance between

points A and $B1$. On the other hand, if we duplicate product class 1 or introduce product class 11, we raise the mean capacity utilization, yielding to a substantial increase in the network WIP, as shown by points $B2$ and $B3$. Similar results were found in section 4.2 as we increased by 10% the throughput of the network (compare figures 16 and 17).

5 Discrete alternatives for capacity changes

In sections 3 and 4 we have assumed that capacity can be added or removed from each station by amounts small enough to consider the total capacity at the station, μ_j, as a continuous variable. This is not always valid. In this section we briefly analyze the more general case where capacity changes at each station is limited to a finite set of discrete alternatives. In a previous paper, Bitran and Tirupati [9] presented a heuristic algorithm that considers capacity at each station as a discrete variable. The algorithm utilized in this section, here named *algorithm 3*, is an extension of that algorithm. Basically it is an iterative algorithm that, at each iteration, solves an integer linear program and updates the variability parameters; for a complete description of algorithm 3 the readers are referred to Bitran and Morabito [4].

Consider that, instead of choosing any value for the capacity μ_j, we are limited to a finite set of n_j discrete alternatives at each station j. This set is described by the vector $\{\mu_{j1}, \mu_{j2}, \ldots, \mu_{jn_j}\}$, where μ_{ji} denotes the capacity at station j under alternative i and satisfies $\mu_{ji} > \lambda_j$ for all i. Table 8 presents a set with 5 possible capacity alternatives for each station of the network example. Note that the first alternative corresponds to point O of figure 5 (compare to table 3).

Each alternative $i, i = 1, \ldots, n_j$, requires the resource level $F_{ji}(\mu_{ji})$ at station j, defined similarly to expression (2) as:

$$F_{ji}(\mu_{ji}) = a_j \mu_{ji}^2 + b_j \mu_{ji} + c_j \tag{3}$$

Note that for each alternative we can calculate the resource requirements at the station. After choosing a capacity alternative for each station j, we may apply the decomposition method (section 3.1) to obtain the 4 parameters $\{\lambda_j, ca_j, \mu_j, cs_j\}$. Let us assume that we chose alternative i at station j (i.e., $\mu_j = \mu_{ji}$) and so, we obtain $\{\lambda_j, ca_j, \mu_{ji}, cs_j\}$. Using these parameters we can also calculate the WIP at station j under alternative i, W_{ji}, defined similarly to expression (1) as:

$$W_{ji} = v_j L_{ji}(\lambda_j, ca_j, \mu_{ji}, cs_j) \tag{4}$$

where, as before, v_j is a given monetary value of an arbitrary product at station j, and $L_{ji}(\lambda_j, ca_j, \mu_{ji}, cs_j)$ is the mean number of products (in queue and in processing) at station j under alternative i. L_{ji} can be calculated in the same way as L_j in expression (1). Note that, once we have chosen an alternative at each station, we may obtain the WIP for each

	Alternative i				
Station j	1	2	3	4	5
1	13.004	10.5	11.0	14.0	15.0
2	27.778	26.0	27.0	28.0	30.0
3	3.160	3.5	3.5	4.0	4.5
4	10.000	7.5	8.0	9.0	11.0
5	5.631	4.5	4.7	5.0	6.0
6	9.225	6.5	7.0	9.0	12.0
7	5.999	5.0	5.5	6.0	6.5
8	4.500	4.5	5.0	5.5	6.0
9	10.000	8.5	9.0	10.0	11.0
10	5.711	4.5	4.7	5.0	6.0
11	5.441	5.3	5.6	5.9	6.0
12	7.440	7.5	8.0	8.5	9.0
13	7.502	6.5	7.0	7.5	8.0
total	115.391				

Table 8: Five discrete alternatives for capacity changes at each station

station and for the whole network. We can use algorithm 3 mentioned above to search for the best capacity choice at the stations.

Applying algorithm 3 to the network example of the previous sections (with the 5 available alternatives of table 8), we obtain point D (70927, 2359) indicated in figure 5 and presented in table 9. Note that the algorithm selected different alternatives for the stations (see the second column of the table). The columns ca_{ji}, μ_{ji}, ρ_{ji}, and so on, indicate the parameters and performance measures at station j under alternative i relative to point D.

Note that the ca_j values at point D, as well as at point B, are very close to the ca_j values at point O (compare tables 4, 6 and 9). The value of the required resources at point B, 2278, becomes a lower bound on the value of the required resources at point D (equal to 2359).

Similarly to algorithm 2, we can also apply algorithm 3 to trace the efficient frontier of the problem with discrete alternatives for capacity changes. Naturally this efficient frontier now is not defined as a continuous curve anymore, but as a set of discrete points.

6 Multiple machines

In sections 3, 4 and 5 we defined the capacity at each station as the mean processing rate μ_j. We considered each station j as a "single machine", or a set of machines, operators, tools, etc., that can be approximated by a single machine with mean processing rate μ_j. This approximation is not always reasonable. We may have situations where we must describe

Station j	Alt.i	ca_{ji}	μ_{ji}	ρ_{ji}	L_{ji}	W_{ji}	F_{ji}
1	2	0.492	10.500	0.952	10.315	1031.498	83.475
2	3	0.598	27.000	0.926	5.782	9321.135	527.310
3	2	0.760	3.500	0.857	3.652	2676.847	337.032
4	3	0.610	8.000	0.875	4.229	4448.631	55.280
5	4	0.619	5.000	0.800	2.285	2084.087	66.850
6	3	0.584	7.000	0.857	2.954	4971.318	28.210
7	3	0.622	5.500	0.727	2.266	3765.365	79.392
8	1	0.660	4.500	0.889	4.386	7947.184	168.120
9	3	0.637	9.000	0.889	4.300	7438.939	134.640
10	4	0.654	5.000	0.800	2.348	3756.943	77.000
11	3	0.671	5.600	0.893	4.597	8651.750	271.197
12	3	0.606	8.000	0.875	4.216	6265.355	290.720
13	4	0.666	7.500	0.800	2.637	8568.902	239.850
total			106.100		53.967	70927.955	2359.077

Table 9: Parameters and performance measures of point D

the capacity at each station as a set of machines, each one with a given mean processing rate.

In the case where we have identical machines at each station (i.e., with the same mean processing rate), algorithms very similar to algorithms 1, 2 and 3 can be applied. Such algorithms consider the decision variable at each station as the number of machines, instead of the mean processing rate. Furthermore, performance measures such as the WIP defined in expression (1) must be redefined according to the multi-machine formulas of queueing theory. For more details of these algorithms, see e.g. Boxma et al [10], Van Vliet and Rinnooy Kan [19], Bitran and Tirupati [9], and Bitran and Morabito [3].

The more general case when we may have distinct machines at the same station involves additional difficulties, and is a topic for future research.

7 Conclusions

In this paper we emphasized the application of *tradeoff curves* to the analysis of discrete manufacturing systems. Initially we discussed the importance of reducing uncertainty and understanding system relationships. We presented the manufacturing environment as a dynamic system, and modeled it as an open queueing network.

In order to generate tradeoff curves between network WIP and resources, we solved optimization problems of these measures applying algorithms known in the literature. To illustrate we presented several tradeoff curves for a manufacturing network example of a semi-

conductor factory, and used them to analyze the effects of uncertainty reduction, throughput variation and product mix changes.

Tradeoff curves describe the relationship between performance measures, and can be effectively used to analyze strategic objectives as a function of the resource requirements to meet them. Therefore, we can design a new manufacturing system or redesign an existing one in such a way to reflect our decision of how to compete in the market.

Acknowledgments

The authors would like to thank Luis A. C. Pedrosa of the Sloan School of Management for his helpful comments. This research was partially supported by a post-doctoral fellowship from Fundação de Amparo a Pesquisa do Estado de São Paulo, Brazil.

References

[1] Albin, S. L. Delays for customers from different arrival streams to a queue. *Mgmt. Sci.* 32, 1986, 329-340.

[2] Bitran, G. R. and S. Dasu. A review of open queueing network models of manufacturing systems. *Queueing Syst.* 12, 1992, 95-134.

[3] Bitran, G. R. and R. Morabito. Open queueing networks: Optimization and performance evaluation models for discrete manufacturing systems. *Working paper*, Sloan School of Management, MIT, 1994, 45p.

[4] Bitran, G. R. and R. Morabito. Manufacturing systems design: Tradeoff curve analysis. *Working paper*, Sloan School of Management, MIT, 1994, 33p.

[5] Bitran, G. R. and D. Sarkar. Targeting problems in manufacturing queueing networks - An iterative scheme and convergence. *EJOR* 76, 1994, 501-510.

[6] Bitran, G. R. and D. Sarkar. Focused factory design: Complexity, capacity and inventory tradeoffs. *Technical Memorandum*, AT&T Bell Lab., 1994, 36p.

[7] Bitran, G. R. and D. Tirupati. Multiproduct queueing networks with deterministic routing: Decomposition approach and the notion of interference. *Mgmt. Sci.* 34(1), 1988, 75-100.

[8] Bitran, G. R. and D. Tirupati. Tradeoff curves, targeting and balancing in manufacturing queueing networks. *Oper. Res.* 37, 1989, 547-564.

[9] Bitran, G. R. and D. Tirupati. Capacity planning in manufacturing networks with discrete options. *Annals of Oper. Res.* 17, 1989, 119-136.

[10] Boxma, O. J., A. Rinnooy Kan and M. Van Vliet. Machine allocation problems in manufacturing networks. *EJOR* 45, 1990, 47-54.

[11] Buzacott, J. A. and J. G. Shanthikumar. *Stochastic models of manufacturing systems*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[12] Hsu, L. F., C. S. Tapiero and C. Lin. Network of queues modeling in flexible manufacturing systems: A survey. *RAIRO* 27(2), 1993, 201-248.

[13] Krajewski, L. J. and P. L. Ritzman. *Operations management: Strategy and analysis*, 2nd.ed., Addison-Wesley, Reading, MA, 1990.

[14] Kusiak, A. *Intelligent manufacturing systems*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[15] Skinner, W. The focused factory. *Harvard Bus. Rev.*, May-June, 1974, 113-121.

[16] Shanthikumar, J. G. and J. A. Buzacott. Open queueing network models of dynamic job shops. *Int. J. Prod. Res.* 19, 1981, 255-266.

[17] Suri, R. and S. De Treville. Full speed ahead: A look at rapid modeling technology in operations management. *OR/MS Today* 18, 1991, 34-42.

[18] Suri, R., J. L. Sanders, M. Kamath. Performance evaluation of production networks. *Handbooks in OR/MS*, S. C. Graves (ed.), vol 4, Elsevier, North-Holland, Amsterdam, 1993.

[19] Van Vliet, M. and A. Rinnooy Kan. Machine allocation algorithms for job shop manufacturing. *Journal of Intelligent Manufacturing* 2, 1991, 83-94.

[20] Whitt, W. The queueing network analyzer. *Bell Syst. Tech. J.* 62, 1983, 2779-2815.

[21] Whitt, W. Approximations for the GI/G/m queue. *Production and Oper.Mgmt.* 2(2), 1993, 114-161.

[22] Whitt, W. Towards better multi-class parametric decomposition approximations for open queueing networks. *Annals of Oper.Res.* 48, 1994, 221-248.