Open Queueing Networks: Optimization and Performance Evaluation Models for Discrete Manufacturing Systems

by
Gabriel R. Bitran
Reinaldo Morabito

WP #3743-94 November 1994

Open Queueing Networks: Optimization and Performance Evaluation Models for Discrete 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

November 23, 1994

Abstract

In this survey we review methods to analyze open queueing network models for discrete manufacturing systems. We focus on design and planning models for job-shops. The survey is divided in two parts: in the first we review exact and approximate decomposition methods for performance evaluation models for single and multiple product class networks. The second part reviews optimization models of three categories of problems: the first minimizes capital investment subject to attaining a performance measure (WIP or leadtime), the second seeks to optimize the performance measure subject to resource constraints, and the third explores recent research developments in complexity reduction through shop redesign and products partitioning.

Keywords: *open queueing networks, optimization and performance evaluation, decomposition methods, discrete manufacturing systems, job-shop design*

1 Introduction

A large fraction of products is manufactured in *discrete* systems, where items are processed either individually or in lots. Therefore, an important strategic problem is the *design* and *planning* of discrete manufacturing systems. Examples of decisions involved are: selection of products and technology, choice of equipment and capacity, and allocation of products to plants. For the purpose of this paper, we group the design problems in three classes proposed in Bitran and Dasu (1992): (i) *targeted system performance* (SP1), (ii) *optimal system performance* (SP2), and (iii) *partitioning of the facility* (SP3). We present problems of classes SP1, SP2 and SP3 formulated as optimization problems.

In class SP1 the objective is to minimize the investment in the manufacturing system subject to the constraints of the desired system performance. Typical performance measures are *work-in-process* WIP (in-process inventory), job leadtime (cycle time), throughput (production rate) and equipment utilization (traffic intensity). In the sequel we choose WIP as a performance measure. Consider the following example of class SP1:

(SP1.1) Targeted WIP level:
Objective: minimize cost of equipment acquisition
Decision variables: capacity of each workstation, technology
Constraints: upper bound on WIP level.

In class SP2 we want to optimize system performance subject to a limited budget for investment in the system. An example of class SP2 is given below:

(SP2.1) Optimal WIP level:
Objective: minimize WIP level
Decision variables: capacity of each workstation, technology
Constraints: upper bound on cost of equipment acquisition.

Note that SP1.1 and SP2.1 involve a trade-off between investment capital and working capital. Finally, in class SP3 we seek to subdivide the manufacturing system into smaller units (which can be thought of as plants within the plant) to improve the overall performance. However, partitioning may require duplication of equipment and resources. Consider the following example of class SP3:

(SP3.1) Targeted number of products and WIP level in each plant:
Objective: minimize cost of equipment acquisition
Decision variables: number of plants, product mix in each plant,
capacity of each workstation
Constraints: upper bound on number of products in each plant and WIP level.

Note that SP3.1 also involves a trade-off between cost of adding capacity and reduction of managerial complexity in the system. It may be seen as a special case of class SP1. The decisions involved are: number of plants into which we subdivide the original system, allocation of products to the plants, and choice of capacity in each plant. In fact, problems of class SP3 are special cases of both classes SP1 and SP2. We consider partitioning problems separately in order to emphasize their importance in the design of production systems (see section 4).

This paper reviews the developments of *optimization models* for classes SP1, SP2 and SP3, combining techniques of mathematical programming and the theory of *open queueing networks* (OQNs). We focus on design and planning models for *job-shops.* For completeness, the paper is divided in two parts. In the first part we review the so called *performance evaluation models* to compute performance measures for OQNs, such as WIP and leadtimes, and in the second part we review the optimization models. Roughly speaking, the difference between a performance evaluation and an optimization model for OQNs is that the first one describes performance measures under certain condition while the second prescribes decisions.

In a previous survey, Bitran and Dasu (1992) analyzed several optimization and performance evaluation models for job shops. In this present review that survey is updated and extended with a more quantitative flavor. We present multiple product class OQN models in more detail, emphasizing the importance of interference among classes and light-traffic approximations. Solution algorithms are presented based on marginal analysis and greedy heuristics. We include also recent developments such as products partitioning, and suggest perspectives for future research.

1.1 Network of Queues Representation of Discrete Manufacturing Systems

Job-shops are complex discrete manufacturing systems that process a wide variety of products or *jobs* in low volumes (Chase and Aquilano, 1992). In general, job-shops involve complex job flows through the workstations (or simply *stations*) and waiting queues in front of the machines. We can represent a job-shop as a network of queues, where nodes correspond to the stations and arcs correspond to job flows between the stations.

The study of queueing networks began basically with the work of Erlang (1917) in telephony. Since then, various examples appeared in different areas; for example, communication, computation, transportation, production, maintenance, biology (neural networks), health (behavior models), chemistry and materials (polymerization), among others; see Disney and Konig (1985). Hsu et al (1993) and Suri et al (1993) provided a broad description of the use of queueing networks to represent manufacturing systems. Each node contains the following elements: (i) arrival process, (ii) service process, and (iii) waiting queue. Figure 1 illustrates this representation.

The *arrival process* at a station is described by job interarrival times, which can be deterministic (D) or probabilistic. If the arrival process is probabilistic, it may either depend on other interarrival times and the service process, or consist of *independent and identically distributed* (*iid*) interarrival times. The former case is called a G-arrival process and the latter case, a GI-arrival process or *renewal process.* An example of a G-arrival process dependent on the service process occurs if an arriving job balks when the waiting queue is too long, or if the job is removed from the queue after waiting for a long time. An instance of a particular GI-arrival process is when the interarrival times are exponential (memoryless or Markovian process M). We can have all jobs belonging to a single class or product family, or different jobs belonging to multiple classes (sometimes one job class may be artificially used to model interruptions such as machine breakdowns). All jobs of the same class are assumed statistically identical. We may also have either individual job arrivals or bulk-arrivals at the station.

The *service process* at a station is described by job service times, classified as deterministic (D) or probabilistic. If the service process is probabilistic, it may either depend on other service times and the arrival process, or consist of *iid* service times. Some authors have called the former

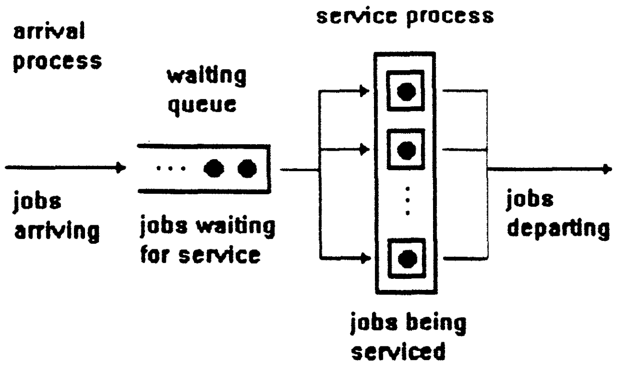

Figure 1: A station with identical machines and a single queue

case as a G-service process and the latter case as a GI-service process (Disney and Konig, 1985). Here we will refer to both as a G-service process since it is more usual in the literature. An example of the G-service process dependent on the arrival process occurs when the service time varies according to the number of jobs in the queue. Similarly to interarrival times, service times may be independent and identically exponential random variables (M-service process). Stations may have only one machine (single server), or many machines (multiple servers). The machine may execute an operation for each job individually or in batch (bulk-service), and eventually breaks down. Each machine may represent a set of resources like different machines, operators, tools, etc. We use, for example, the notation $GI/G/m$ to denote a single queueing system where the arrival process is a renewal process (the first GI), the service times are *iid* random variables (the second G), and the number of servers in the system is m.

Finally, the *waiting queue* (or buffer queue) of a station may have either limited or unlimited capacity for the number of jobs in the queue, which is generally determined by the available physical space. If it is full, new job arrivals at the queue are blocked. The queue has a discipline or rule to sequence jobs waiting for service. Examples of queueing disciplines are: *first-come first-served* (FCFS), job priority, shortest-processing-time-first, and largest-processing-time-first. When the rule is based on job priority, preemption may or may not be allowed. In the preemptive case, the highest priority job in the queue starts service as soon as it arrives, even if a job with lower priority is already in service. In the non-preemptive case, a job already in service can not be interrupted until it is completed.

A set of nodes, arcs and jobs constitute a *network of queues* with the following characteristics: (i) number of stations (nodes), (ii) sequence of operations (routing), and (iii) type of queueing network: open, closed and mixed. The number of nodes in the network (greater or equal to 1) corresponds to the number of stations offering different operations. The sequence of operations or routing through the stations may be sequential, sequential with feedback, assembly, arborescent, acyclic, and cyclic. Feedback arcs can be used to represent, for example, rework in manufacturing, and probabilistic routings can model, for example, machine failures.

A queueing network can be classified as either open, closed, or mixed. In an *open* queueing network (OQN), jobs enter the network, receive service at one or more nodes, and eventually leave the network. The number of jobs flowing among the nodes is a random variable. In a *closed* queueing network (CQN), there are no external job arrivals or departures. The departure rate from any designated node is a random variable but the number of jobs flowing among the nodes is fixed. However, we can artificially represent external job arrivals and departures in a station of a CQN by defining this station as an instantaneous load-unload station. The load-unload station switches at the same moment an internal job arrival with an external job arrival without varying the total number of jobs in the CQN. Note that, in this way, we can represent a *flexible manufacturing system* (FMS) as a CQN. If the network has multiple classes, we can redefine open sub-networks for some classes and closed sub-networks for other classes. In this case, the resulting network of queues is called *mixed.*

The queueing network models analyzed in this paper assume that the system attains equilibrium or *steady-state.* The arrival processes are probabilistic with *iid* interarrival times at stations. Jobs may belong to a single class or to multiple classes, and arrive individually. There is no limit on the number of jobs in each class, but jobs can not change from one class to another. The service processes are also probabilistic with *iid* service times at stations. Each station may have one or more identical machines and each machine serves only one job at a time. Jobs can not be combined or created in the network, and the waiting queues have unlimited capacity with discipline FCFS. All models discussed in this paper correspond to OQNs with acyclical or cyclical structures and deterministic or probabilistic routings (note that OQN models are analytically more tractable than CQN models, and may approximate CQN models; see e.g. Whitt (1984) and Calabrese (1992)). Other models for the various cases not considered in this review can be found in the literature discussed below and in the references cited there.

1.2 Related Literature Reviews

Disney and Konig (1985) presented an extensive survey of queueing network theory, covering the seminal works of Jackson and the extensions of Kelly, including a bibliography of more than 300 references. Other surveys are Lemoine (1977) and Koenigsberg (1982). Buzacott and Yao (1986) discussed the developments of CQN models before 1986 (oriented to FMS applications) and classified the approaches based on the different research groups. Suri et al (1993) examined performance evaluation models for different manufacturing systems such as single stage systems (single queues), production lines (tandem queues), assembly lines (arborescent queues), job-shops (OQN), and FMS (CQN). Suri et al commented on the use of queueing theory in topics like MRP II, JIT, Kanban, and suggested alternative approaches such as sensitivity analysis in simulation, models based on Petri net, and hierarchical queueing networks.

Buzacott and Shanthikumar (1992, 1993), Hsu et al (1993) and Bitran and Dasu (1992) analyzed both performance evaluation models and optimization models for queueing networks. Buzacott and Shanthikumar presented an extensive analysis oriented to the design of different manufacturing systems such as flow lines, automated transfer lines, job shops, FMS and multicellular systems. They analyzed optimal design problems and, in particular, considered some optimization models in job shops that will not be covered here, such as optimal allocation of workers to stations, optimal number of operators in the system, optimal allocation of jobs to stations, and analysis of routing and time diversity effects in job processing. Hsu et al examined optimization models for FMS based in CQNs; they also suggested the use of alternative techniques like algebra max-plus, fuzzy sets and expert systems. Bitran and Dasu discussed strategic, tactical and operational problems of manufacturing systems based on the OQN methodology, with a special attention to design and planning models for job-shops. As we mentioned earlier, this focus is extended in this paper.

Most approaches to optimization models are based on decomposition methods (see below) to evaluate performance measures for an OQN. More recently, alternative approaches have been explored (*Brownian models*) based on heavy-traffic limit theorems to evaluate performance measures. In section 3.2.1 we describe an example of this approach (Wein, 1990) without exploring further developments on this topic since Harrison and Nguyen (1993) recently reviewed the state of the art of Brownian models for multiple-class OQNs.

1.3 Structure and Notation

In section 2 we examine decompositon methods for performance evaluation models for OQNs. In section 2.1 we briefly review exact decomposition methods for *Jackson networks* ($M/M/m$ queueing networks), and in section 2.2 we review approximate decomposition methods for *generalized Jackson networks* ($GI/G/m$ queueing networks). In section 3 we deal with problems SP1.1 and SP2.1, and review solution methods based on some convexity results and performance evaluation models of section 2. In section 3.1 we present algorithms to solve SP1.1 and SP2.1 for Jackson networks, and in section 3.2 we present algorithms to solve SP1.1 and SP2.1 for generalized Jackson networks. Finally, in section 4 we emphasize the importance of problem SP3.1 in the manufacturing environment and suggest some perspectives for future research.

In the following sections, we generally use the indices i and j to indicate station, the index k to indicate product class, and the index l to indicate class operation at stations. The notations $E(x)$, $V(x)$ and cx denote respectively the expected value, the variance and the *square coefficient of variation* (scv) of a random variable x. The scv is defined as $cx = \frac{V(x)}{E(x)^2}$.

2 Performance Evaluation Models for OQNs

Performance evaluation models have been addressed using : (i) exact methods, (ii) approximate methods, and (iii) simulation and related techniques. *Exact methods* exist for Jackson networks (section 2.1). The main result is that if the external interarrival and service times are exponentially distributed, we can define the equilibrium distribution (if it exists) of the number of jobs in the network as a product form, and decompose the network in a set of stocastically independent stations. Thus each station is individually analyzed as an independent $M/M/m$ system.

In most manufacturing systems the arrival and service processes are generally less variable than the Poisson process, and the assumption above does not apply. In the absence of exact methods for general OQNs, we can use *simulation and related techniques* (Law and Haider, 1989, Law and McComas, 1989). These approaches allow the use of more elaborate assumptions that are close to reality. The main drawback is the computational requirements that limit the number of alternatives to be considered. Techniques like perturbation analysis suggest possible ways to reduce the computational cost. These techniques are beyond the scope of this work and are described in Ho and Cao (1983), Ho (1987) and Suri (1989).

The limitations imposed by exact methods and simulation led authors to develop *approximate methods.* These are classified in five categories: (i) diffusion approximations, (ii) mean value analysis, ((iii) operational analysis, (iv) exponentialization approximations, and (v) decomposition methods.

Diffusion approximations are motivated by heavy-traffic limit theorems and have generated new solution methods for OQNs (e.g., Reiman, 1990, Harrison and Nguyen, 1990). They have been applied to scheduling and operational control problems. *Mean value analysis* (Seidmann et al, 1987, Suri et al, 1993), *operational analysis* (Denning and Buzen, 1978, Dallery and David, 1986) and *exponentialization approximations* (Yao and Buzacott, 1986, Hsu et al, 1993) have been basically used to analyze CQNs. The most frequently used approximate methods to analyze OQN models for job-shops have been *decomposition methods.* In this paper we only review decomposition methods (section 2.2).

2.1 Jackson Networks (Exact Decomposition Methods)

Consider a network of queues composed of n stations, each one with one or more identical machines and infinite waiting capacity. Stations $1, 2, \ldots, n$ are internal stations and station 0 is the external station of the system. For each internal station j, jobs arrive from station 0 with *iid* interarrival times a_{0j}, wait in queue for an available machine, and are processed with *iid* service times s_j. After being processed, jobs leave station j with interdeparture times d_j and go to station i, $i = 0, \ldots, n$, with transition probability defined by a Markov chain. We assume that any sequence of external interarrival times, service times and routing decisions are mutually independent, and that jobs are serviced at each station according to a FCFS discipline.

We refer to the network above as a *Jackson open queueing network* when we have exponential distributed external interarrival and service times (Poisson processes). Otherwise we have a *generalized Jackson open queueing network* (or simply, a general OQN). Jackson networks have elegant exact solutions in a product form which were shown by Jackson (1957, 1963), as we will see below.

2.1.1 Single Class $M/M/m$ OQNs

Assume that all jobs belong to the same class. Consider the following notation for the input data:

n number of internal stations in the network.

For each station j, $j = 1, \ldots, n$:

m_j number of identical machines at station j, $m_j \geq 1$

λ_{0j} expected external arrival rate at station j ($\lambda_{0j} \doteq \frac{1}{E(a_{0j})}$)

μ_j expected service rate of each machine at station j ($\mu_j = \frac{1}{E(s_j)}$).

For each pair $(i,j), i = 0, \ldots, n, j = 0, \ldots, n$:

r_{ij} probability of a job going to station j after completing service at station i.

Thus our input data has $(n+1)^2 + 3n$ numbers and each station j is described by 3 parameters: $\{m_j, \lambda_{0j}, \mu_j\}$. Successive stations are visited according to an absorbing Markov chain with transition matrix $R = \{r_{ij}, 0 \leq r_{ij} \leq 1, i = 0, \ldots, n, j = 0, \ldots, n\}$, where $\sum_{j=0}^{n} r_{ij} = 1, i = 0, \ldots, n$, and

$r_{00} = 1$ by definition. Note that $r_{00} = 1$ eliminates any chance of a job returning to the system and so, it reduces the input data to $n^2 + 3n$ numbers.

Let $Q = \{q_{ij} \in R, i = 1, \ldots, n, j = 1, \ldots, n\}$ and $q_{i0} = 1 - \sum_{j=1}^{n} q_{ij}$. Q is the matrix R without line 0 (probability of a job entering the system at station j) and without column 0 (probability of a job leaving the system by station i). Similarly, $q_{i0}, i = 1, \ldots, n$, is the column 0 of matrix R without the element of line 0. A deterministic job routing may be also described by Q and q_{i0} for all i, since it is a particular case of a probabilistic routing. If $q_{jj} > 0$, we say that station j has an *immediate feedback* arc. To illustrate the transition matrix, consider a symmetric job shop (Shanthikumar and Buzacott, 1981) for which $Q = \{q_{ij} = \frac{1}{n}; q_{ii} = 0, i \neq j, i = 1, \ldots, n, j = 1, \ldots, n\}$ and $q_{i0} = \frac{1}{n}, i = 1, \ldots, n$. Now consider a deterministic flow-shop with all stations in series in the sequence $1, 2, \ldots, n$, for which $Q = \{q_{i,i+1} = 1, i = 1, \ldots, n-1; q_{ij} = 0 \; otherwise\}$, $q_{i0} = 0, i = 1, \ldots, n-1$ and $q_{n0} = 1$. Note that these two examples do not contain immediate feedback arcs.

Traffic Rate Equations

The traffic rate equations provide the expected arrival rate at each station. Let λ_j be the expected arrival rate at station j, defined as $\lambda_j = \frac{1}{E(a_j)}$, where a_j is the interarrival time at station j. Under the assumption of steady-state, λ_j is obtained from the following system of linear equations:

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{n} q_{ij}\lambda_i \text{ for } j = 1, \ldots, n \tag{1}$$

Given that $q_{i0} \geq 0$ for $i = 1, \ldots, n$, it can be shown that (1) has a unique solution satisfying $\lambda_j \geq 0$ for all j. Using this solution we can calculate the *expected utilization* ρ_j (or traffic intensity) of station j, defined as:

$$\rho_j = \frac{\lambda_j}{\mu_j m_j} \tag{2}$$

where $0 \leq \rho_j < 1$. The ratio $\frac{\lambda_j}{\mu_j}$ in (2) is called *offered load* (or workload), and corresponds to the expected number of busy machines at station j, also denoted by $\alpha_j = \rho_j m_j$. The expected arrival rate at station j from station i is given by:

$$\lambda_{ij} = \lambda_i q_{ij} \tag{3}$$

and the expected (external) departure rate to station 0 from station j is given by:

$$\lambda_{j0} = \lambda_j(1 - \sum_{i=1}^{n} q_{ji}) \tag{4}$$

Adding λ_{j0} (or λ_{0j}) for all j, we obtain the *throughput* λ_0 (or production rate) of the network. The *expected number of visits* $E(V_j)$ of an arbitrary job to station j is then evaluated by:

$$E(V_j) = \frac{\lambda_j}{\lambda_0} \tag{5}$$

Let L be a state of the system defined as a vector $L = (L_1, L_2, \ldots, L_n)$, where L_j corresponds to the number of jobs in queue and in service at station j. Assuming that the system reaches steady-state, let $\pi(L)$ be the probability of the system being in equilibrium at state L. Jackson showed that $\pi(L)$ is given by the following product form:

$$\pi(L) = \prod_{j=1}^{n} \pi_j(L_j) \tag{6}$$

with

$$\pi_j(L_j) = \begin{cases} \frac{\pi_j(0)\lambda_j^{L_j}}{\mu_j^{L_j} L_j!} & \text{if } L_j \leq m_j \\ \frac{\pi_j(0)\lambda_j^{m_j} \rho_j^{L_j - m_j}}{\mu_j^{m_j} m_j!} & \text{if } L_j > m_j \end{cases}$$

where $\pi_j(L_j)$ is the probability of station j having L_j jobs ($L_j = 0, 1, \ldots$) and $\pi_j(0)$ is a normalizing constant. These results imply that in order to compute the equilibrium probability for a given state L, we may consider each station independently (note that (6) is the product of the probabilities of each $M/M/m_j$ queue in the network). Thus, after applying the linear system (1) to determine each λ_j we may decompose the network into n individual $M/M/m_j$ stations, each one described by $\{m_j, \lambda_j, \mu_j\}$. To evaluate performance measures we just consider each station individually and independently of the others. For example, the *expected waiting time* $E(Wq_j)$ (or mean delay) of a job in the $M/M/m_j$ queue of station j can be derived from (6), given by (Tijms, 1986, p.333):

$$E(Wq_j) = \frac{(m_j\rho_j)^{m_j}\pi(0)}{\mu_j m_j (1-\rho_j)^2 m_j!} \tag{7}$$

where

$$\pi(0) = \{ \sum_{t=0}^{m_j - 1} \frac{(m_j\rho_j)^t}{t!} + \frac{(m_j\rho_j)^{m_j}}{(1-\rho_j)m_j!} \}^{-1}$$

Note in (7) that if $m_j = 1$, then $E(Wq_j) = \frac{\rho_j}{\mu_j(1-\rho_j)}$. The *expected number of jobs* $E(Lq_j)$ in queue of station j can be easily obtained applying Little's law: $E(Lq_j) = \lambda_j E(Wq_j)$.

The *expected leadtime* $E(T)$ (or cycle time) for an arbitrary job, including waiting times and service times spent in the network from the first arrival until the final departure, is given by:

$$E(T) = \sum_{j=1}^{n} E(V_j)(E(Wq_j) + E(s_j)) \tag{8}$$

where $E(V_j)$ is the expected number of visits at station j defined by (5), $E(Wq_j)$ is defined by (7), and $E(s_j)$ is the expected service time of a job at station j. The expected number of jobs in the network can be defined in a similar way. An interesting observation is that, even though the number of jobs at the stations are statistically independent at a given instant of time, the waiting times at different stations are in general not independent random variables. The variance of the leadtime in the network is usually approximated by ignoring the correlation in the waiting times (Shanthikumar and Buzacott, 1984). Buzacott and Shantikhumar (1993) provided an analysis of Jackson networks considering the correlation above.

2.1.2 Multiple Class $M/M/m$ OQNs

Kelly (1975, 1979) extended Jackson's product form solutions to multiple class queueing networks for the case in which service times are independent of job class (see also Baskett et al, 1975). The model allows the definition of a deterministic routing for each class. Even if we impose other service disciplines (e.g., priority queues), we obtain a solution in product form. Consider the state $S = (S_1, S_2, \ldots, S_n)$, where S_j denotes the state of station j. Each S_j corresponds to a row vector $(s_{j1}, s_{j2}, \ldots, s_{j,Lq_j})$, where Lq_j is the queue length of station j and $s_{jl}, l = 1, \ldots, Lq_j$, specifies the class of the job at the l-th position in the queue. The equilibrium probability $\pi(S)$ has the following product form:

$$\pi(S) = K\, f(S)\, g_1(S_1)\, g_2(S_2) \ldots g_n(S_n) \tag{9}$$

where K is a normalizing constant, $f(.)$ is a function of state S, and $g_j(.)$ is a function of station j. Although these results are interesting, practical implementations are difficult due to the size of the state space in (9). Furthermore, the assumptions underlying Jackson networks are very restrictive for general job shops and other manufacturing systems. For instance, Bitran and Tirupati (1988) suggested that exponential distributions overstate the variability in the service times found in many manufacturing operations. For further details regarding Jackson networks the readers are referred to the surveys of Disney and Konig (1985), Walrand (1990), Suri et al (1993) and Buzacott and Shanthikumar (1993), and the references cited there.

2.2 Generalized Jackson Networks (Approximate Decomposition Methods)

Decomposition methods may be seen as efforts to extend Jackson's product form solution and the "independence" between stations to general OQNs (generalized Jackson networks). The arrival and departure processes are approximated by renewal processes, and each station is analyzed as a stocastically independent $GI/G/m$ queue. The complete decomposition procedure is essentially described in three steps:

Step 1: Analysis of interaction between stations of the OQN,
Step 2: Decomposition of the OQN into systems of individual and independent stations,
Step 3: Recomposition of the decomposition results to analyze the general performance of the OQN.

In step 1 we determine the internal arrival flows for each station. In step 2 we compute the performance measures for each station separately. In step 3 we compute the performance measures for the whole network. Step 1 is fundamental in this procedure and involves three basic processes, namely: (i) merging or superposition of arrivals, (ii) departures, and (iii) decomposition or splitting of departures.

Figure 2 illustrates each one of these processes. The *superposition process* merges the individual arrival flows from other stations (including the external station), producing a merged arrival flow to the station. The *departure process* analyzes the effects of the merged arrival flow in the queue of a station, producing a merged departure flow from this station. Finally, the *splitting process* decomposes the merged departure flow from a station into individual departure flows to other stations (including the external station).

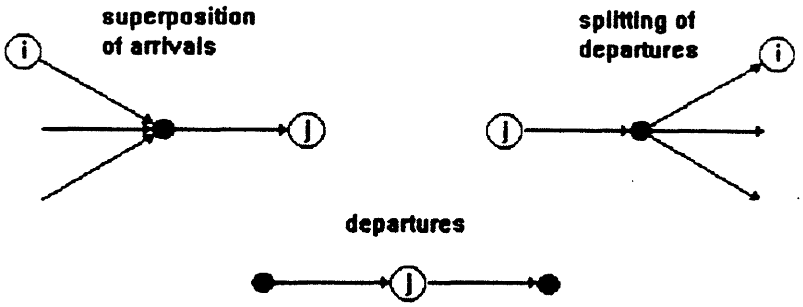

Figure 2: Superposition of arrivals, departures, and splitting of departures

In general, just the first two moments of the distributions (typically the mean and the scv) are sufficient to provide a good approximation, and they have been often utilized to describe the flows above. This approach was initially proposed by Reiser and Kobayashi (1974) and was improved by Sevcik et al. (1977), Kuehn (1979), Shanthikumar and Buzacott (1981), Albin (1982), Whitt (1983a), Bitran and Tirupati (1988), Segal and Whitt (1989), Whitt (1994), among others. Shanthikumar and Buzacott were the first to apply this method to manufacturing systems. In section 2.2.1 we present steps 1, 2 and 3 for a single class $GI/G/1$ queueing networks with probabilistic routing. In section 2.2.2 we extend these steps to $GI/G/m$ queueing networks, and in section 2.2.3 we deal with multiple class $GI/G/m$ queueing networks with deterministic routing for each class. This last case is widely considered in practice to model job-shop systems.

2.2.1 Single Class $GI/G/1$ OQNs with Probabilistic Routing

In this section we assume that all jobs belong to the same class and move through stations according to a probabilistic routing. Both external interarrival times and service times are *iid* but now we consider general distributions. Initially we assume single server stations. Consider the following notation for the input data:

n number of internal stations in the network.

For each station $j, j = 1, \ldots, n$:

λ_{0j} expected external arrival rate to station j ($\lambda_{0j} = \frac{1}{E(a_{0j})}$)

ca_{0j} scv or variability of external interarrival time at station j ($ca_{0j} = \frac{V(a_{0j})}{E(a_{0j})^2}$)

μ_j expected service rate at station j ($\mu_j = \frac{1}{E(s_j)}$) .

cs_j scv or variability of service time at station j ($cs_j = \frac{V(s_j)}{E(s_j)^2}$).

For each pair $(i, j), i = 1, \ldots, n, j = 1, \ldots, n$:

q_{ij} probability of a job going to station j after completing service at station i.

Thus our input data has $n^2 + 4n$ numbers and each station j is described by 4 parameters: $\{\lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$. Similarly to section 2.1, let the transition matrix $Q = \{q_{ij}, i = 1, \ldots, n, j = 1, \ldots, n\}$ and $q_{i0} = 1 - \sum_{j=1}^{n} q_{ij}$. In what follows we consider only OQNs with no immediate feedback arcs (i.e., $q_{ii} = 0, i = 1, \ldots, n$). If an OQN originally contains immediate feedback arcs, then we can easily remove them from the OQN adjusting the initial parameters (see Whitt, 1983a). This procedure improves the quality of the approximations. All assumptions for the Jackson networks are also assumed here except, of course, exponential distributions for external arrival and service processes (which result in $ca_{0j} = 1$ and $cs_j = 1$ for each station j).

Step 1

In step 1 we want to determine two parameters for each station j: (i) the expected arrival rate λ_j and (ii) the scv or variability of interarrival time ca_j (note that for Jackson networks we obtain $V(a_j) = E(a_j)^2$ and, therefore, $ca_j = 1$). In other words, starting with the initial parameters $\{\lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ and the matrix Q, we want to describe each station j by the parameters $\{\lambda_j, ca_j, \mu_j, cs_j\}$. The two parameters λ_j and ca_j are determined solving two linear systems. Firstly, we obtain *exact* expected arrival rates from the traffic rate equations (1), similarly to the Jackson networks. These are used to obtain *approximate* variability parameters from the traffic variability equations defined below. These systems can be shown to have a unique non-negative solution.

Traffic Variability Equations

The traffic variability equations involve the three processes discussed earlier: superposition of arrivals, departures, and splitting of departures. They provide approximations for the interarrival time variability ca_j at each station j. These approximations combine two basic methods: the *asymptotic method* (Sevcik et al, 1977) and the *stationary-interval method* (Kuehn, 1979).

Superposition of Arrivals

In the superposition process (figure 2), the expected arrival rates and interarrival time variability parameters at station j are combined, producing the merged expected arrival rate λ_j (from (1)) and the merged interarrival time variability ca_j (remember that $\lambda_j = \frac{1}{E(a_j)}$ and $ca_j = \frac{V(a_j)}{E(a_j)^2}$). The asymptotic method and the stationary-interval method may be used to determine ca_j (or $V(a_j)$). These methods are also called *macro* and *micro*, respectively, because of the macroscopic and the microscopic view of the arrival process (Whitt, 1982).

Assume that the arrivals are occurring at station j since $t = -\infty$, and a new arrival occurs at $t = 0$. Let S_p be the elapsed time until the p-th arrival after $t = 0$. Both methods yield the same expected time interval $E(a_j)$, but may yield very different variances $V(a_j)$. The asymptotic method takes a macroscopic view and try to match process behavior over a relatively long time interval, yielding $V(a_j) = \lim_{p\to\infty} \frac{V(S_p)}{p}$. The stationary-interval method takes a microscopic view and try to match process behavior during a relatively short time interval. It yields $V(a_j) = V(S_1)$, where S_1 is referred as the stationary interval. Moreover, the asymptotic method is asymptotically correct as $\rho_j \to 1$ (heavy-traffic intensity), and the interval-stationary method is asymptotically correct as $p \to \infty$, when the arrival process tends to a Poisson process.

Let ca_{ij} be the interarrival time variability at station j from station i. Based on the asymptotic method, the superposition ca_j is a convex combination of ca_{ij} given by (Sevcik et al, 1977):

$$ca_j = \frac{\lambda_{0j}}{\lambda_j} ca_{0j} + \sum_{i=1}^{n} \frac{\lambda_{ij}}{\lambda_j} ca_{ij} = \sum_{i=0}^{n} \frac{\lambda_{ij}}{\lambda_j} ca_{ij} \tag{10}$$

where λ_{ij} and λ_j are obtained from (3) and (1), respectively. Based on the interval-stationary method, the superposition ca_j results in a non-linear function (Kuehn, 1979). Note that if the arrival process is Poisson (i.e., $ca_{ij} = 1, i = 0, \ldots, n$), then (10) is exact and returns $ca_j = 1$.

The asymptotic approximation (10) does not reflect the convergence to the Poisson process as $p \to \infty$; on the other hand, the stationary-interval approximation deteriorates as $\rho_j \to 1$. Albin (1982, 1984) suggested a more refined approximation to ca_j with a relative error around 3% in comparison to simulation. This approximation is based on the convex combination between the value obtained by (10) and the value obtained by the stationary-interval method. Whitt (1983b) simplified Albin's refinement substituting the stationary-interval method by a Poisson process obtaining:

$$ca_j = w_j \sum_{i=0}^{n} \frac{\lambda_{ij}}{\lambda_j} ca_{ij} + 1 - w_j \tag{11}$$

where

$$w_j = \frac{1}{1 + 4(1 - \rho_j)^2 (v_j - 1)}$$

$$v_j = \frac{1}{\sum_{i=0}^{n} (\frac{\lambda_{ij}}{\lambda_j})^2}$$

The approximation (11) yields results very close to Albin's hybrid approximation.

Departures

In the departure process (figure 2), the merged expected arrival rate and the merged interarrival time variability at station j, together with the service time variability cs_j, are used to determine the merged expected departure rate and the merged interdeparture time variability from station j. If station j is not saturated (i.e., $\rho_j < 1$) and is in steady-state, then we have the expected departure rate equal to the expected arrival rate. However, the evaluation of the interdeparture time variability is not so easy.

Let cd_j be the scv or variability of interdeparture time at station j. Based on the stationary-interval method and using Marshall's formulae for a $GI/G/1$ system we obtain (Kuehn, 1979):

$$cd_j = ca_j + 2\rho_j^2 cs_j - \frac{2\rho_j(1 - \rho_j)E(Wq_j)}{\mu_j} \tag{12}$$

where $E(Wq_j)$ is the expected waiting time in queue at station j. Substituting in (12) the Kraemer&Lagenback-Belz approximation for $E(Wq_j)$ (see equation (16) below), we obtain the interdeparture time variability as a convex combination of ca_j and cs_j given by:

$$cd_j = \rho_j^2 cs_j + (1 - \rho_j^2) ca_j \tag{13}$$

where ρ_j is known from (2). Note that if the arrival process and the service process are Poisson (i.e., $ca_j = cs_j = 1$), then (13) is exact and produces $cd_j = 1$. Note also that if $\rho_j \to 1$, then we obtain $cd_j \to cs_j$, suggesting that the interdeparture time variability tends to the service time variability as the expected utilization of station j becomes very high (i.e., long queues at station j tend to diminish the effect of the interarrival time variability). On the other hand, if $\rho_j \to 0$, then we obtain $cd_j \to ca_j$, suggesting that the interdeparture time variability tends to the interarrival time variability as the expected utilization of station j is very low and no waiting queues are expected.

Based on the asymptotic method we obtain an elementary approximation to cd_j, given by (Whitt, 1983a):

$$cd_j = ca_j \tag{14}$$

The approximation (14) is also exact if the arrival and service processes are Poisson. Furthermore, it becomes more accurate at station j as the expected utilization increases in the subsequent stations to station j. For example, consider an OQN composed of two stations in series, say station 1 and 2 with parameters $\{\lambda_{01}, ca_{01}, \mu_1, cs_1\}$ and $\{0, 0, \mu_2, cs_2\}$, respectively, and $q_{12} = 1$ and $q_{11} = q_{22} = q_{21} = 0$. Using (1) we obtain $\lambda_1 = \lambda_2 = \lambda_{01}$. In addition, if we have $\mu_2 \to \lambda_2$ and μ_1 constant, we obtain $\rho_2 \to 1$. Based on heavy-traffic limit theorems, Whitt (1983a) observed that the performance measures of station 2 are asymptotically the same as if station 1 is removed (i.e., $\frac{1}{\mu_1} \equiv 0$). In other words, the arrival process at station 2 is the same arrival process at station 1. Under these conditions, (14) is asymptotically correct for station 1 resulting $ca_1 = cd_1 = ca_2$, while a *heavy-traffic bottleneck* phenomenon occurs at station 2.

A possible refinement is to combine the approximations from the two methods above, similarly to the superposition process. However, Whitt observed that this refinement is not as critical as for the superposition case, and suggested the use of (13). More recently, Suresh and Whitt (1990) observed that the heavy-traffic bottleneck can occur in practice at reasonable expected utilization levels. Experiments with various stations in series and different parameters revealed limitations in the use of the approximations (13) and (14) separately. Suresh and Whitt suggested that it should be appropriate to consider hybrid approximations to the departure process, combining the stationary-interval method and the asymptotic method. They observed that the expected waiting time $E(Wq_j)$ at station j does not reflect the heavy-traffic phenomenon because ca_j is assumed totally independent of ρ_j (see (16) below). Then, they suggested that ca_j should be a function of $ca_1, cs_1, cs_2, \ldots, cs_{j-1}$ and $\rho_1, \rho_2, \ldots, \rho_j$. For instance, ca_j could be a convex combination of $ca_1, cs_1, cs_2, \ldots, cs_{j-1}$ with weights that are continuous functions of $\rho_1, \rho_2, \ldots, \rho_j$.

Splitting of Departures

In the splitting process (figure 2), the merged expected departure rate and merged interdeparture time variability from station j are decomposed, producing the expected rates λ_{ji} according to (3). The interdeparture time variability cd_{ji} between stations j and i is defined below as a function of cd_j (Sevcik et al, 1977, or Kuehn, 1979):

$$cd_{ji} = q_{ji} cd_j + 1 - q_{ji} \tag{15}$$

If the departure process is Poisson (i.e., $cd_j = 1$), then (15) is exact and gives $cd_{ji} = 1$. Note that if $q_{ji} \to 1$, then (15) results in $cd_{ji} \to cd_j$. That is, as the expected departure rate from station j to station i tends to the merged expected departure rate from station j, the interdeparture time variability from station j to station i also tends to the merged interdeparture time variability from station j. Furthermore, if $q_{ji} \to 0$, then (15) results in $cd_{ji} \to 1$, indicating that as the proportion of flow between stations j and i tends to zero, the departure process between these two stations tends to a Poisson process. Note also that cd_{ji} in (15) is equal to ca_{ji} in (11), that is, the interdeparture time variability from station j to station i is exactly the same as the interarrival time variability at station i from station j. Assuming that the departure process is renewal and $q_{ji}, i = 1, \ldots, n$, represents independent events (Markovian routing), then (15) is exact and the stationary-interval approximation and the asymptotic approximation coincide.

Combining equations (11), (13) and (15), we obtain the second linear system as a function of ca_j, cd_j, and ca_{ij} (or cd_{ij}) to approximate the scv ca_j for each station $j, j = 1, \ldots, n$. The solution of the two linear systems discussed so far (traffic rate and traffic variability equations) allows the description of each station j by the desired parameters $\{\lambda_j, ca_j, \mu_j, cs_j\}$. We can then proceed to steps 2 and 3. Note that if the network is acyclic (i.e., the job routing do not form cycles), then the stations $1, 2, \ldots, n$ can be relabeled as $j_1, j_2, \ldots, j_n$, such that jobs visit station j_i after station j_k for $j_i > j_k$. Since there are no cycles, the parameters λ_j and ca_j can be easily computed for each station j following the increasing order of the station labels.

Steps 2 and 3

In step 1 we decomposed the OQN into a collection of independent stations, each one described by $\{\lambda_j, ca_j, \mu_j, cs_j\}$. In step 2 we want to evaluate performance measures for each station, such as expected waiting time in queue, expected length of queue, and so on. These measures may be approximated by formulas from queueing theory (e.g., Kleinrock, 1975, Tijms, 1986). Whitt (1983a) observed that since the arrival process is usually not a renewal process and only two parameters (mean and scv) are known for each distribution, then there is little to be gained from more elaborate procedures.

For illustration, the expected waiting time $E(Wq_j)$ in the $GI/G/1$ queue of station j may be estimated by the Kraemer&Lagenbach-Belz formulae (modified by Whitt), defined as:

$$E(Wq_j) = \frac{\rho_j(ca_j + cs_j)g(\rho_j, ca_j, cs_j)}{2\mu_j(1-\rho_j)} \qquad (16)$$

where

$$g(\rho_j, ca_j, cs_j) = \begin{cases} exp\{\frac{-2(1-\rho_j)(1-ca_j)^2}{3\rho_j(ca_j+cs_j)}\} & \text{if } ca_j < 1 \\ 1 & \text{if } ca_j \geq 1 \end{cases}$$

Note that for a $M/M/1$ queue, (16) and (7) produce the same result. For a study and comparison of other approximations for $E(Wq_j)$, see for example Shanthikumar and Buzacott (1981) and Buzacott and Shanthikumar (1993).

Finally, in step 3 we want to evaluate performance measures for the whole network, for example, the expected job leadtime, expected number of jobs, and production rate. For instance, let's consider the expected leadtime for an arbitrary job, including waiting times and service times spent in the network. Similarly to (8), we obtain:

$$E(T) = \sum_{j=1}^{n} E(V_j)(E(Wq_j) + E(s_j)) \qquad (17)$$

where $E(V_j)$ is the expected number of visits at station j defined by (5), $E(Wq_j)$ is defined by (16), and $E(s_j)$ is the expected service time at station j. As in the Jackson networks, the variance of the leadtime is usually approximated by ignoring the correlation in the waiting times at different stations. Further details of steps 2 and 3 may be found in Whitt (1983a, 1983b) and Suri et al (1993).

2.2.2 Single Class $GI/G/m$ OQNs with Probabilistic Routing

The model above considers single machine stations. The more general case with one or more identical machines in each station can be derived from the previous one. Let $m_j, m_j \geq 1$, be the number of machines in station j, now defined by 5 parameters: $\{m_j, \lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$. In step 1 equation (13) is replaced by:

$$cd_j = 1 + (1-\rho_j^2)(ca_j - 1) + \frac{\rho_j^2(cs_j - 1)}{\sqrt{m_j}} \tag{18}$$

Note that if $m_j = 1$, then (18) reduces to (13), and that for $M/M/m_j$ ($ca_j = 1$, $cs_j = 1$) and $M/G/\infty$ ($ca_j = 1$, $m_j \to \infty$) systems, (18) correctly results in a Poisson process (i.e., $cd_j = 1$). However for a $M/D/1$ ($ca_j = 1$, $cs_j = 0$) system, (18) or (13) incorrectly produce an interdeparture time variability less than the interarrival time variability (i.e., $cd_j = 1 - \rho_j < 1$). In fact, Shanthikumar and Buzacott (1981) did not find good results (relative to simulation) upon applying (13) to $M/D/1$ and $GI/D/1$ queueing networks. In order to reduce this distortion, Whitt (1983a) suggested to modify (18) to:

$$cd_j = 1 + (1-\rho_j^2)(ca_j - 1) + \frac{\rho_j^2(max\{cs_j, 0.2\} - 1)}{\sqrt{m_j}} \tag{19}$$

where ρ_j is known from (2). Finally, combining equations (11), (19) and (15) we obtain the following linear system as a function of ca_j, cd_j and cd_{ji} (or ca_{ji}) to determine the scv ca_j for each station j:

$$ca_j = \alpha_j + \sum_{i=1}^{n} ca_i \beta_{ij} \text{ for } j = 1, \ldots, n \tag{20}$$

where

$$\alpha_j = 1 + w_j\{(p_{0j}ca_{0j} - 1) + \sum_{i=1}^{n} p_{ij}[(1 - q_{ij}) + q_{ij}\rho_i^2 x_i]\}$$

$$\beta_{ij} = w_j p_{ij} q_{ij}(1 - \rho_i^2)$$

with w_j defined according to (11) and

$$p_{ij} = \frac{\lambda_i}{\lambda_j} q_{ij} \text{ (where } p_{ij} = \tfrac{\lambda_{ij}}{\lambda_j} \text{ and } q_{ij} = \tfrac{\lambda_{ij}}{\lambda_i})$$

$$x_i = 1 + \frac{(max\{cs_i, 0.2\} - 1)}{\sqrt{m_i}}$$

Steps 2 and 3 are similar to the previous section, using performance measure formulas derived from $GI/G/m$ queueing theory. For example, the expected waiting time $E(Wq_j)$ at station j can be approximated by:

$$E(Wq_j) = \frac{(ca_j + cs_j)}{2} E(Wq_j(M/M/m_j)) \tag{21}$$

where $E(Wq_j(M/M/m_j))$ is the expected waiting time for a $M/M/m_j$ queue defined in (7). Note that if the arrival and service processes are Poisson, then (21) reduces to (7). Moreover, if $m_j = 1$ and $ca \geq 1$, then (21) reduces to (16) (for improved approximations of $E(Wq_j)$, see e.g. Whitt (1993) and Buzacott and Shanthikumar (1993)). The expected job leadtime $E(T)$ in the network can be defined similarly to (17).

2.2.3 Multiple Class $GI/G/m$ OQNs with Deterministic Routings

In this section we modify the prior model (sections 2.2.1 and 2.2.2) to deal with multiple job class OQNs. Each job class has its own routing which defines the sequence of stations to be visited. For a class routing, each visit to a station corresponds to a different operation, and we may have various visits to the same station. For example, the sequence (2, 3, 1, 3, 4) defines a class routing whose jobs visit four different stations for five operations (the two operations produced at station 3 may be different). Contrary to the previous sections, now routing is deterministic. Consider the following notation for the input data:

n number of internal stations in the network
r number of classes in the network.
For each station $j, j = 1, \ldots, n$:
m_j number of machines at station j.
For each class $k, k = 1, \ldots, r$:
n_k number of operations in the routing of class k
λ'_k expected external arrival rate of class k
ca'_k scv or variability of external interarrival time of class k.
For each class $k, k = 1, \ldots, r$, and for each operation $l, l = 1, \ldots, n_k$ in the routing of class k:
n_{kl} station visited for operation l in the routing of class k
$E(s_{kl})$ expected service time for operation l in the routing of class k
or μ_{kl} expected service rate for operation l in the routing of class k (i.e., $\mu_{kl} = \frac{1}{E(s_{kl})}$)
cs_{kl} scv or variability of service time for operation l in the routing of class k.

The routing of class k is now described by n_k and n_{kl} (instead of matrix Q), and may have a different service time distribution for each operation. Whitt (1983a) presented a procedure to aggregate all classes in a single one and utilize the single class model discussed before (sections 2.2.1 and 2.2.2). Note that in this way the original multiple class OQN is reduced to a single *aggregate class* OQN. After this aggregate class OQN has been analyzed, we return to the original network and estimate the performance measures for each class individually. This procedure is described below.

Firstly we obtain the initial parameters $\{m_j, \lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ of the aggregate class for each station $j, j = 1, \ldots, n$ and then, we utilize step 1 from previous sections to obtain the final parameters $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ of the aggregate class. Let $1H(x) = 1$ if $x \in H$ and $1H(x) = 0$ otherwise.

We calculate λ_{0j} simply by adding the expected external arrival rates of all classes whose first operation occurs at station j, that is,

$$\lambda_{0j} = \sum_{k=1}^{r} \lambda'_k 1\{k : n_{k1} = j\} \tag{22}$$

where $H = \{k : n_{k1} = j\}$. Each element of the aggregate transition matrix Q is obtained by:

$$q_{ij} = \frac{\sum_{k=1}^{r} \sum_{l=1}^{n_k - 1} \lambda'_k \, 1\{(k,l) : n_{kl} = i, n_{k,l+1} = j\}}{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k \, 1\{(k,l) : n_{kl} = i\}} \tag{23}$$

where the numerator of (23) corresponds to λ_{ij} and the denominator corresponds to λ_i (compare to (3)). The aggregate expected service time at station j is estimated by:

$$E(s_j) = \frac{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k E(s_{kl}) \, 1\{(k,l) : n_{kl} = j\}}{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k \, 1\{(k,l) : n_{kl} = j\}} \tag{24}$$

and the aggregate expected service rate μ_j at station j is simply $\frac{1}{E(s_j)}$. The aggregate service time variability at station j is estimated using (24) and the property that the second moment of a mixture of independent distributions is the mixture of the second moments:

$$cs_j = \frac{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k E(s_{kl})^2 (cs_{kl} + 1) \, 1\{(k,l) : n_{kl} = j\}}{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k \, 1\{(k,l) : n_{kl} = j\} E(s_j)^2} - 1 \tag{25}$$

Finally, we may utilize the hybrid method (see (11)) to obtain the aggregate external interarrival time variability to station j. Using (2) and merging all external interarrival time variability parameters at station j, we obtain:

$$ca_{0j} = w_j \sum_{k=1}^{r} \frac{\lambda'_k \, 1\{k : n_{k1} = j\} \, ca'_k}{\sum_{l=1}^{r} \lambda'_k \, 1\{l : n_{l1} = j\}} + 1 - w_j \tag{26}$$

where

$$w_j = \frac{1}{1 + 4(1 - \rho_j)^2 (v_j - 1)}$$

$$v_j = \frac{1}{\sum_{k=1}^{r} \left(\frac{\lambda'_k \, 1\{k : n_{k1} = j\}}{\sum_{l=1}^{r} \lambda'_k \, 1\{l : n_{l1} = j\}} \right)^2}$$

Expressions (22)-(26) produce the initial parameters $\{m_j, \lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ for each station j and matrix Q so that we have all input data for the single class OQN of section 2.2.2. In step 1 we describe each station j by $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ after solving (1) (or (29) defined below) and (20). In steps 2 and 3 we obtain the performance measures for the aggregate class in the same way as before. Then, we return to the original network and estimate the performance measures for each class individually. For example, the expected leadtime $E(T_k)$ for a job of class k is given by:

$$E(T_k) = \sum_{l=1}^{n_k} E(Wq_{n_{kl}}) + \sum_{l=1}^{n_k} E(s_{kl}) \tag{27}$$

where $E(Wq_{n_{kl}})$ is the expected waiting time at station n_{kl} (i.e., the station relative to the l-th operation in the routing of class k), and can be evaluated by (21). Note that the first term in (27) corresponds to the total expected waiting time for a job of class k, and the second term corresponds to the total expected service time for a job of class k. Similarly, the variance of leadtime $V(T_k)$ for a job of class k is given by:

$$V(T_k) = \sum_{l=1}^{n_k} V(Wq_{n_{kl}}) + \sum_{l=1}^{n_k} E(s_{kl})^2 cs_{kl} \tag{28}$$

where the first and second terms correspond to the variances of waiting times and service times, respectively.

Interference Among Classes

Bitran and Tirupati (1988) showed that expression (15) may be less effective for the splitting process when we have multiple class OQNs with deterministic routings. Note in (15) that if $q_{ji} \to 0$, then $cd_{ji} \to 1$. Bitran and Tirupati extended (15) to represent the *interference among classes.* For each class at a station, the analysis is reduced to two classes: (i) the class of interest itself and (ii) the aggregation of all other classes arriving between two successive arrivals of the class of interest. We call this second class the *aggregate class* (do not confuse with the aggregate class of the previous section).

Let class k be the class of interest at a certain station j in a multiple class OQN with deterministic routings. For convenience, assume that class k has one and only one operation at station j, say operation l (the approximations below are also valid for the case when class k has more than one operation at station j). Hence, $n_{kl} = j$. Assume also that the interarrival and interdeparture times of all classes at station j are *iid.* Since we have only deterministic routings in the network, we can easily obtain λ_j by adding the expected arrival rates from all classes (including class k) which operations occur at station j, that is,

$$\lambda_j = \sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k 1\{(k,l) : n_{kl} = j\} \tag{29}$$

It is also easy to obtain the proportion of the class of interest k at station j, $q_{kl} = \frac{\lambda'_k}{\lambda_j}$ (recall that $j = n_{kl}$). Let d_{kl} be the interdeparture time of class k from station j, d_j be the interdeparture time of all classes from station j, and z_{kl} be the number of jobs of the aggregate class that arrive at station j during one interarrival time of class k. Note that d_{kl} is the sum of $z_{kl} + 1$ *iid* random variables. Define $z'_{kl} = z_{kl} + 1$. Since the expectation and variance of the sum of z'_{kl} *iid* random variables are $E(d_{kl}) = E(z'_{kl})E(d_j)$ and $V(d_{kl}) = E(z'_{kl})V(d_j) + V(z'_{kl})E(d_j)^2$, where $E(d_{kl}) = \frac{1}{\lambda'_k}$ and $E(d_j) = \frac{1}{\lambda_j}$, it follows that $E(z'_{kl}) = \frac{\lambda_j}{\lambda'_k} = \frac{1}{q_{kl}}$ and cd_{kl} is given by:

$$cd_{kl} = q_{kl} cd_j + cz'_{kl} \tag{30}$$

Bitran and Tirupati observed that the first term on the right hand side of (30) reflects the effect of the queue process at station j, while the second term does not depend on the service process. It captures the effect of the aggregate class arrival process between two successive arrivals of class k.

Bitran and Tirupati proposed two approximations to cz'_{kl} based on the assumption that z_{kl} has a Poisson and Erlang distributions, respectively. Assuming that z_{kl} has a Poisson distribution with rate $\lambda_j(1-q_{kl})$, it follows that $cz'_{kl} = (1-q_{kl})[q_{kl} + (1-q_{kl})ca_{kl}]$, and expression (30) (splitting process) can be written as:

$$cd_{kl} = q_{kl}cd_j + (1-q_{kl})q_{kl} + (1-q_{kl})^2 ca_{kl} \tag{31}$$

where $j = n_{kl}$, $q_{kl} = \frac{\lambda'_k}{\lambda_j}$, and ca_{kl} is the interarrival time variability of the class of interest k at station j. Note that $ca_{kl} = cd_{k,l-1}$. We may also rewrite expression (11) (superposition process) as a function of ca_{kl}:

$$ca_j = w_j \sum_{k=1}^{r} \sum_{l=1}^{n_k} \frac{\lambda'_k}{\lambda_j} ca_{kl}\, 1\{(k,l), n_{kl} = j\} + 1 - w_j \tag{32}$$

where

$$w_j = \frac{1}{1+4(1-\rho_j)^2(v_j-1)}$$
$$v_j = \frac{1}{\sum_{k=1}^{r}\sum_{l=1}^{n_k}(\frac{\lambda'_k}{\lambda_j})^2\, 1\{(k,l), n_{kl} = j\}}$$

and λ_j is obtained from (29). Combining (32), (19) and (31), we obtain an alternative linear system as a function of ca_j, cd_j, and cd_{kl} (or $ca_{k,l+1}$) to determine ca_j. After obtaining the parameters $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ from step 1, we proceed to steps 2 and 3 as before. This approach based on (31) produces much better estimates for ca_j than (20), which is based on (15) (see the computational results in Bitran and Tirupati, 1988, 1989b).

In fact, expression (31) can be seen as a generalization of (15). To see that, consider a particular situation where jobs of the class of interest k enter the network at station j, wait in line together with jobs of other classes and, after finishing service, only the jobs of class k proceed to a certain station i. Thus, the expected departure rate of arc (j,i) is $\lambda_{ji} = \lambda'_k$, and the rate proportion (or probability) of jobs going from station j to station i is $q_{ji} = \frac{\lambda'_k}{\lambda_j}$. Following the same steps as above, we can define d_{ji}, z_{ji}, z'_{ji} and so on, and rewrite (30) as: $cd_{ji} = q_{ji}cd_j + cz'_{ji}$. Assuming that z_{ji} has a Poisson distribution with rate $\lambda_j(1-q_{ji})$, it follows that $cz'_{ji} = (1-q_{ji})[q_{ji} + (1-q_{ji})ca'_k]$, and (31) can be written as:

$$cd_{ji} = q_{ji}cd_j + (1-q_{ji})q_{ji} + (1-q_{ji})^2 ca'_k \tag{33}$$

Note that if the arrival process of class k is Poisson (i.e., $ca'_k = 1$), then (33) reduces to (15). In fact, it can be shown that (15) is the special case of (33) when z'_{ji} is geometrically distributed with parameter q_{ji}, yielding $cz'_{ji} = 1 - q_{ji}$. Note also that if $q_{ji} \to 1$, then (15) and (33) lead to $cd_{ji} \to cd_j$ but, if $q_{ji} \to 0$, then only (33) leads to $cd_{ji} \to ca'_k$. This last result is asymptotically exact (Bitran and Tirupati, 1988, remark 1), and permitted two important approximations to multiple class OQNs with deterministic routings, presented by Whitt (1988). Initially, consider a certain station in the network (Whitt, 1988, p.1335):

> *"If the arrival rate of one class upon one visit to some queue is a small proportion of the total arrival rate there, then the departure process for that class from that visit to that queue tends to be nearly the same as the arrival process for that class for that visit to that queue."*

Whitt observed that this principle may be seen as a *light-traffic approximation*, where only the class of interest must have low utilization (i.e., the overall utilization of the station need not be low). Consider now a certain class with deterministic routing in the network (Whitt, 1988, p.1335):

> *"If the contribution to the arrival rate by this class at each visit to each queue is a small proportion of the total arrival rate at that queue, then the departure process of that class from each visit to each queue, and thus from the entire network, is nearly the same as the external arrival process of this class to the network."*

Based on (33) and the approximations above, Segal and Whitt (1989) proposed an alternative expression for the splitting process of multiple class OQNs with deterministic routings. Let ce_j be the average of the external interarrival time variability parameters of the classes at station j, weighted by the expected number of visits of each class at station j (see expression (5)). We define:

$$ce_j = \frac{\sum_{k=1}^{r}\sum_{l=1}^{n_k}\lambda'_k 1\{(k,l) : n_{kl} = j\} ca'_k}{\sum_{k=1}^{r}\sum_{l=1}^{n_k}\lambda'_k 1\{(k,l) : n_{kl} = j\}} \tag{34}$$

Using (34), the variability parameter cd_{ji} between stations j and i is redefined by (compare to (15)):

$$cd_{ji} = q_{ji}cd_j + (1 - q_{ji})q_{ji}ca_j + (1 - q_{ji})^2 ce_j \tag{35}$$

Segal and Whitt suggested to replace (15) by (35) if the classes follow purely deterministic routings (they also suggested the use of a convex combination of (15) and (35) to capture the effect of probabilistic routings). Note that as we substitute (15) by (35), we must modify the linear system (20) with (11), (19) and (35). Steps 2 and 3 are as before. However, we are not aware of any computational experience comparing the performance of this approximation based on (35) and the previous one based on (31).

Recently, Whitt (1994) proposed an extension of (31) defined as:

$$cd_{kl} = q_{kl}cd_j + (1 - q_{kl})q_{kl}\bar{ca}_{kl} + (1 - q_{kl})^2 ca_{kl} \tag{36}$$

where $\bar{ca}_{kl}$ is the interarrival time variability of the aggregation of all classes arriving between two successive arrivals of class k for operation l at station j, $j = n_{kl}$. Whitt presented computational results suggesting that (36) is more effective for the splitting process than (31). Note that (31) can be seen as a special case of (36) when we assume that the arrival process of the aggregate class is Poisson (i.e., $\bar{ca}_{kl} = 1$). Whitt proposed also other approximations for the splitting process under the assumption that the server is continuously busy, which will not be discussed here.

The approximate decomposition methods can be used to evaluate the performance measures of OQNs modeling real manufacturing networks. In addition to the instances discussed in this section, more complex situations including batch service and overtime (Bitran and Tirupati, 1989c, 1991), and machine breakdowns, changes in lot sizes, product testing and repairing (Segal and

Whitt, 1989), may be also incorporated to these methods with little modification. The effect of material handling in manufacturing networks is discussed in Buzacott and Shanthikumar (1993). The potential of practical applications motivated the development of various software packages based on these methods, such as the *QNA - Queueing Network Analyzer* (Whitt, 1983a, 1983b; Segal and Whitt, 1989), *ManuPlan* (Suri et al., 1986, Brown, 1988), *MPX* (Suri and De Treville, 1991), *QNAP - Queueing Network Analysis Package* (Pujolle and Ai, 1986), *Operations Planner* (Jackman and Johnson, 1993) and *X-FLO* (Karmarkar, 1993). For references of major corporation applications and case studies, see for example Suri et al (1993).

3 Optimization Models for OQNs

In section 2 we reviewed models to evaluate the performance of a given OQN representing a job-shop system. In this section we analyze models to either design an OQN or redesign an existing OQN representing a job-shop. Clearly, if the design is one of selecting from a small number of alternatives, then we may utilize the models from section 2 to choose the alternative with the best performance, otherwise we need models based on optimization techniques. Bitran and Dasu (1992) classified optimization models for OQNs in: (i) optimal design and (ii) optimal control.

Optimal design models assume a simple operational rule to optimize the system design, for example, the FCFS discipline. *Optimal control* models determine the optimal operational rule for the system. This paper reviews only optimal design models. For a recent discussion of optimal control models based on Brownian motion, see Harrison and Nguyen (1993).

Problems SP1.1, SP2.1 and SP3.1 presented in section 1 are examples of optimal design problems. As discussed in section 1, various performance measures may be utilized such as WIP, leadtime or throughput. In what follows we formulate problems SP1.1 and SP2.1 choosing WIP as a performance measure. Since WIP and leadtime are linearly related through Little's law, the algorithms presented below also apply to leadtime. The readers are referred to Bitran and Sarkar (1994a) for a similar study utilizing the throughput as a performance measure. For simplicity, we adopt the notation $L_j(.)$ and $W_j(.)$, instead of $E(L_j(.))$ and $E(W_j(.))$, to denote the expected number of jobs and waiting time in queue and service at station j. Let:

μ_j	expected service rate of each machine at station j
m_j	number of identical machines at station j
$F_j(\mu_j, m_j)$	cost of allocating capacity (μ_j, m_j) at station j
F_T	available budget for the network capacity
$L_j(\mu_1, m_1; \ldots$ $\ldots; \mu_n, m_n)$	expected number of jobs at station j as a function of the network capacity
v_j	mean monetary value of a job at station j (independent of the job class)
L_T	upper bound on the network WIP.

Recall that the WIP is a mean monetary value of the expected number of jobs in the network defined as $\sum_{j=1}^{n} v_j L_j(\mu_1, m_1; \mu_2, m_2; \ldots; \mu_n, m_n)$. Each monetary value v_j associated to a job at station j can be 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, 1986). Obviously, if $v_j = 1$ for all j, then the WIP corresponds to the expected number of jobs in the network.

The targeted WIP performance problem SP1.1 is the problem of determining capacity (μ_j, m_j) for each station j in such a way to minimize total cost and satisfy a WIP target constraint for the network. SP1.1 is formulated as:

$$\text{(SP1.1)} \quad \min \sum_{j=1}^{n} F_j(\mu_j, m_j)$$

$$\text{subject to:} \quad \sum_{j=1}^{n} v_j L_j(\mu_1, m_1; \mu_2, m_2; \ldots; \mu_n, m_n) \leq L_T$$

$$\text{with:} \quad (\mu_j, m_j) \in P_j, j = 1, \ldots, n$$

where P_j is a given domain of the variables. Similarly, the optimal WIP performance problem SP2.1 is the problem of determining capacity (μ_j, m_j) for each station j in such a way to minimize total WIP and satisfy a budget constraint for the network. SP2.1 is formulated as:

$$\text{(SP2.1)} \quad \min \sum_{j=1}^{n} v_j L_j(\mu_1, m_1; \mu_2, m_2; \ldots; \mu_n, m_n)$$

$$\text{subject to:} \quad \sum_{j=1}^{n} F_j(\mu_j, m_j) = F_T$$

$$\text{with:} \quad (\mu_j, m_j) \in P_j, j = 1, \ldots, n$$

Different authors have presented solution methods for the two problems above. In the following sections, we review some of these approaches. In order to present them in a more structured way, we adopt the notation suggested by Bitran and Dasu (1992) denoting each instance by $\alpha/\beta/\chi/\delta$, where $\alpha \in$\{SP1.1, SP2.1, SP3.1 \}, $\beta \in$\{ J, G \}, $\chi \in$\{ S, M \} and $\delta \in$\{ R, N \}. The symbol α indicates problem type, β indicates if the problem is applied to a Jackson OQN (J) or to a general OQN (G), χ indicates if the stations have a simple machine (S) or multiple machines (M), and δ indicates the decision variable: expected service rate (R) or number of machines (N) in each station.

Problems SP1.1 and SP2.1 are considered in both sections 3.1 and 3.2. In section 3.1 we review models and solution methods to Jackson OQNs and in section 3.2, to general OQNs. For related approaches to CQNs, the readers are referred to Shanthikumar and Yao (1987, 1988), Dallery and Stecke (1990), and Calabrese (1992).

3.1 Models ./J/./. (Jackson Networks)

As we saw in section 2.1, we can analyze exactly each station j of a Jackson network as a stocastically independent system. Thus, L_j in SP1.1 and SP2.1 becomes a function of μ_j and m_j only, instead of a function of $\mu_1, m_1; \mu_2, m_2; \ldots; \mu_n, m_n$.

3.1.1 Models ./J/./R

Kleinrock (1964, 1976) initially studied the problem of minimizing the expected number of jobs in a single-class single-server Jackson queueing network. Consider again the input data from section 2.1.1 with $m_j = 1$ for all j. Kleinrock chose the service rates $\mu_j, j = 1, \ldots, n$, as decision variables, and assumed that the cost F_j is proportional to μ_j for each station j: $F_j(\mu_i) = f_j\mu_j$, where f_j is the unit cost of capacity at station j. Applying Little's law ($Lq_j = \lambda_j Wq_j$) in (7) and adding the offered load ($\alpha_j = \frac{\lambda_j}{\mu_j}$), we obtain the expected number of jobs at a $M/M/1$ system defined as $L_j(\mu_i) = \frac{\lambda_j}{\mu_j - \lambda_j}$, where λ_j is computed from (1). The model SP2.1/J/S/R is then formulated as:

$$\text{(SP2.1/J/S/R)} \quad \min \sum_{j=1}^{n} L_j(\mu_i)$$

$$\text{subject to:} \quad \sum_{j=1}^{n} f_j\mu_j = F_T$$

$$\text{with:} \quad \mu_j \geq 0, j = 1, \ldots, n$$

Kleinrock showed that SP2.1/J/S/R has an optimal solution $\mu_j^*, j = 1, \ldots, n$, defined in a closed form by:

$$\mu_j^* = \lambda_j + \frac{\sqrt{f_j\lambda_j}}{\sum_{i=1}^{n}\sqrt{f_i\lambda_i}} \frac{(F_T - \sum_{i=1}^{n} f_i\lambda_i)}{f_j} \tag{37}$$

Note that if the unit cost of capacity is the same for all stations, (37) first allocates enough capacity to station j to satisfy the expected arrival rate, and then allocates capacity to station j in proportion to the square root of its expected arrival rate. As Bitran and Dasu (1992) observed, five conditions are satisfied in the model above:
(i) $L_j(\mu_j)$ is a convex function of μ_j (the expected number of jobs at station j is a convex function of the capacity at station j),
(ii) $L_j(\mu_j)$ is not dependent of $\mu_i, i \neq j, i = 1, \ldots, n$ (capacity additions at other stations have no effect on the expected number of jobs at station j),
(iii) μ_j is continuous (the decision variables are continuous variables),
(iv) $F_j(\mu_j)$ is a convex function of μ_j (the cost of capacity.at station j is a convex function of the capacity at station j),
(v) $L_j(\mu_j)$ (and $W_j(\mu_j)$) can be expressed in closed form.

Conditions (i)-(iv) reduce SP2.1/J/S/R to a convex program that can be optimally solved via local-optimal methods (Bazaraa et al, 1993). Condition (v) enables a closed form solution to the problem. These conditions will be extensively used in the remaining of section 3.

We may formulate SP2.1/J/M/R exactly as SP2.1/J/S/R, where $L_j(\mu_j)$ is redefined for a $M/M/m_j$ queue (compare to (7)). Harel and Zipkin (1987) showed that the expected waiting time $W_j(\mu_j)$ (and the expected number of jobs $L_j(\mu_j)$) in a $M/M/m_j$ queue is also a convex function of μ_j. Thus, conditions (i)-(iv) are again satisfied and SP2.1/J/M/R may be also reduced to a convex program.

Models SP1.1/J/S/R and SP1.1/J/M/R can be defined and analyzed in a similar way.

3.1.2 Models ./J/M/N

In sequel, we discuss models SP1.1/J/M/N and SP2.1/J/M/N. Now we have *integer* decision variables corresponding to the number of machines in each station. Boxma et al (1990) presented a heuristic and an exact algorithm to solve both problems. The manufacturing network is represented by a multiple-class multiple-server Jackson OQN with a different deterministic routing for each class (see sections 2.1.2 and 2.2.3). Consider again the input data described in section 2.2.3 with $ca_k = 1$ and $cs_k = 1$ for each class k, and $m_j \geq 1$ for each station j.

Aggregating all classes into a unique class, we obtain each station j described by 3 parameters $\{m_j, \lambda_j, \mu_j\}$ (see section 2.1.2). Kelly (1979) showed that the equilibrium distribution of the number of jobs in the network can be expressed as a product form, and that each station j in steady-state behaves as a $M/M/m_j$ system. Applying Little's law in (7) and adding the offered load, we obtain the expected number of jobs L_j as a function of m_j, λ_j and μ_j, given by:

$$L_j(m_j, \lambda_j, \mu_j) = \frac{\frac{\lambda_j}{\mu_j m_j}(\frac{\lambda_j}{\mu_j})^{m_j}\pi(0)}{(1 - \frac{\lambda_j}{\mu_j m_j})^2 m_j!} + \frac{\lambda_j}{\mu_j} \tag{38}$$

where

$$\pi(0) = \{\sum_{t=0}^{m_j-1} \frac{(\frac{\lambda_j}{\mu_j})^t}{t!} + \frac{(\frac{\lambda_j}{\mu_j})^{m_j}}{(1 - \frac{\lambda_j}{\mu_j m_j})m_j!}\}^{-1}$$

Boxma et al considered $m_j, j = 1, \ldots, n$, as decision variables in models SP1.1/J/M/N and SP2.1/J/M/N, and observed that $L_j(m_j, \lambda_j, \mu_j)$ in (38), namely $L_j(m_j)$, is a convex decreasing function of m_j (conditions (i) and (ii) are satisfied). They chose WIP as a performance measure for the network. Note that this analysis is easily extended to leadtime since WIP and leadtime are linearly related through Little's law. Let m be the vector of decision variables $(m_1, m_2, \ldots, m_n)$. The network WIP, $L(m)$, is given by:

$$L(m) = \sum_{j=1}^{n} v_j L_j(m_j) \tag{39}$$

The choice of m_j in each station must satisfy the condition $\rho_j < 1$, in order to prevent system instability. Let $|z|$ denote the largest integer number less than z. Using this condition and (2), it follows that m_j must be an integer number greater than or equal to the lower bound m_j^0, defined as:

$$m_j^0 = |\frac{\lambda_j}{\mu_j}| + 1 \tag{40}$$

Model SP1.1/J/M/N

In model SP1.1/J/M/N we want to find a minimal cost solution satisfying a WIP level less than or equal to the specified limit L_T, where $L_T < L(m^0)$. Let $F_j(m_j)$ be the cost of allocating m_j machines at station j, defined as a convex non-decreasing function of m_j (condition (iv) is satisfied). Using (39) and (40) we obtain the targeted WIP level problem (also called *server* *allocation problem*):

$$(\text{SP1.1/J/M/N}) \;\; \min \;\; F(m) = \sum_{j=1}^{n} F_j(m_j)$$

$$\text{subject to: } \; L(m) \leq L_T$$

$$\text{with: } \; m_j \geq m_j^0, m_j integer, j = 1, \ldots, n$$

Note that since SP1.1/J/M/N is a convex program with integer variables, the use of *marginal analysis* schemes do not lead necessarily to optimality (condition (iii) is not satisfied). Let $PI_j(m_j)$ be a *priority index* defined as the quotient of the increase of cost and the decrease of WIP at station j, given by:

$$PI_j(m_j) = \frac{\Delta F_j(m_j + 1)}{-v_j \Delta L_j(m_j + 1)} \tag{41}$$

where

$$\Delta F_j(m_j + 1) = F_j(m_j + 1) - F_j(m_j) \geq 0$$

$$\Delta L_j(m_j + 1) = L_j(m_j + 1) - L_j(m_j) < 0$$

PI_j is a result from the marginal analysis of F_j and L_j. Boxma et al (1990) presented a simple heuristic algorithm (algorithm 1) based on the *greedy method* to solve problem SP1.1/J/M/N (see also Sundarraj et al (1994). for a related approach to a similar problem). The algorithm starts with the smallest possible machine allocation (40) for each station. At every iteration it then adds one machine at the station where the priority index (41) is the smallest. The algorithm terminates as soon as adding a machine makes the allocation feasible.

Algorithm 1

1. Start with the allocation $m_j = m_j^0, j = 1, \ldots, n$. This solution is infeasible ($L(m^0) > L_T$) and its cost $F(m^0)$ is less than the minimum cost of SP1.1/J/M/N.

2. At each iteration, update the cost $F(m)$, WIP $L(m)$ (using (38) and (39)), and $PI_j(m_j)$ (using (41)). Add one machine at the station j^* which results in the smallest quotient PI_{j^*} (greedy strategy), given by:

$$PI_{j^*} = min\{PI_j(m_j), j = 1, \ldots, n\} \tag{42}$$

3. Stop as soon as $L(m)$ reaches the target L_T (feasible solution).

Note that station j^* chosen in (42) produces the smallest increase of $F(m)$ per unit of decrease of $L(m)$, indicated by PI_{j^*}. From the convexity of F_j and L_j, we obtain:

$$\frac{\Delta F_j(m_j + 1)}{-v_j \Delta L_j(m_j + 1)} \geq \frac{\Delta F_j(m_j)}{-v_j \Delta L_j(m_j)} \tag{43}$$

An interesting result from (43) is that we can verify the quality of the heuristic solution generated by algorithm 1, just comparing the solutions generated in the last two iterations. Let p be the last iteration, and $m^1, m^2, \ldots, m^{p-1}, m^p$ be the solution generated in each iteration. Obviously,

m^{p-1} is infeasible and m^p is feasible. Let's denote the optimal solution of SP1.1/J/M/N by m^*. Boxma et al (theorems 1 and 2) showed that:

$$F(m^{p-1}) < F(m^*) \leq F(m^p)$$

and so, $F(m^{p-1})$ and $F(m^p)$ are bounds for the optimal solution value. Computational experiments using two real-life manufacturing networks resulted in a relative error of 5% between $F(m^p)$ and $F(m^{p-1})$. These experiences suggest that algorithm 1 generates an allocation sufficiently close to the optimal allocation of SP1.1/J/M/N.

Model SP2.1/M/N

In model SP2.1/M/N we want to allocate (or reallocate) machines in order to optimize a performance measure, for instance, the WIP of the network. We assume a total of M homogeneous machines to be allocated to the stations, where $M > \sum_{j=1}^{n} m_j^0$. This situation occurs for example in the FMS design, where we may have identical machines performing different operations as we install different tools. Using (39) and (40) we obtain the optimal WIP level problem (also called *server reallocation problem*):

$$\text{(SP2.1/J/M/N)} \quad \min \quad L(m)$$

$$\text{subject to:} \quad \sum_{j=1}^{n} m_j = M$$

$$\text{with:} \quad m_j \geq m_j^0, m_j integer, j = 1, \ldots, n$$

Again, we have a convex program with integer variables (conditions (i), (ii) and (iv) are satisfied but condition (iii) is violated), and the use of marginal analysis schemes may not produce the optimal solution of SP2.1/J/M/N. Let $PI_j(m_j)$ be a priority index defined now as the decrease of WIP per machine at station j, given by:

$$PI_j(m_j) = -v_j \Delta L_j(m_j + 1) \tag{44}$$

where $\Delta L_j(m_j + 1) = L_j(m_j + 1) - L_j(m_j) < 0$, in accordance with the previous section.

Boxma et al (1990) presented a simple algorithm (algorithm 2), similar to algorithm 1, also based on the greedy method to solve SP2.1/J/M/N. The algorithm starts with the smallest possible machine allocation (40) for each station. At every iteration it then adds one machine to the station where the priority index (44) is the largest. The algorithm terminates when all the M machines have been allocated.

Algorithm 2

1. Start with the allocation $m_j = m_j^0, j = 1, \ldots, n$. This is an infeasible solution ($\sum_{j=1}^{n} m_j^0 < M$) and its WIP $L(m^0)$ is greater than the minimum WIP of SP2.1/J/M/N.

2. At each iteration, update the WIP $L(m)$ (using (38) and (39)) and $PI_j(m_j)$ (using (44)). Add one machine at the station j^* which results in the largest PI_{j^*} (greedy strategy), given by:

$$PI_{j^*} = max\{PI_j(m_j), j = 1, \ldots, n\} \tag{45}$$

3. Stop as soon as the total number of allocated machines reaches the limit M (feasible solution).

Note that station j^* chosen in (45) produces the largest decrease of $L(m)$ per machine, indicated by PI_{j^*}. From the convexity of L_j, we obtain:

$$v_j \Delta L_j(m_j + 1) \geq v_j \Delta L_j(m_j) \tag{46}$$

Using (46), Boxma et al (theorem 3) proved that algorithm 2 is exact and terminates with the optimal solution of SP2.1/J/M/N (in spite of the fact that condition (iii) is not satisfied). Furthermore, this solution is found within a time span bounded by a polynomial function in the number of stations of the network, that is, within $O(Mn)$ steps.

3.2 Models ./G/./. (Generalized Jackson Networks)

In this section we study models SP1.1/G/S/R and SP2.1/G/S/R (Bitran and Tirupati, 1989a, Bitran and Sarkar, 1994a, and Wein, 1990), SP1.1/G/M/R with discrete variables (Bitran and Tirupati, 1989b), and SP1.1/G/M/N and SP2.1/G/M/N (Van Vliet and Rinnooy Kan, 1991).

3.2.1 Models ./G/./R

Initially we present two algorithms introduced by Bitran and Tirupati (1989a) to solve models SP1.1/G/S/R and SP2.1/G/S/R. These algorithms can be easily extended to deal with models SP1.1/G/M/R and SP2.1/G/M/R. The manufacturing network is represented by a multiple class $GI/G/1$ OQN with a deterministic routing for each class (see section 2.2.3). In the previous section performance measures such as WIP in (39), were easily evaluated because of the exact results for Jackson networks. In the absence of exact methods for generalized Jackson networks, approximate decomposition methods are then utilized to estimate the variability parameters at each station. Let's consider again the input data of section 2.2.3 with $m_j = 1$ for all j. Step 1 of the decomposition method results in the system of equations (13) plus (29), (31) and (32), represented below simply by:

$$\Phi(\lambda, ca, \mu, cs) = 0 \tag{47}$$

where vectors λ, ca, μ and cs denote $\{\lambda_j, ca_j, \mu_j, cs_j\}$ for all j. Applying Little's law in (16) and adding the offered load, we obtain the expected number of jobs L_j as a function of λ_j, ca_j, μ_j and cs_j, given by:

$$L_j(\lambda_j, ca_j, \mu_j, cs_j) = \frac{(\frac{\lambda_j}{\mu_j})^2 (ca_j + cs_j) g(\lambda_j, ca_j, \mu_j, cs_j)}{2(1 - \frac{\lambda_j}{\mu_j})} + \frac{\lambda_j}{\mu_j} \tag{48}$$

where $g(\lambda_j, ca_j, \mu_j, cs_j)$ is defined according to (16). Since L_j is a function of λ_j, ca_j, μ_j and cs_j in (48) and λ, ca, μ and cs are related in (47), we obtain L_j as a function of λ, ca, μ and cs. Bitran and Tirupati considered each capacity $\mu_j, j = 1, \dots, n$, as a continuous decision variable (condition (iii) is satisfied), assuming that additional capacity may be added to a station by small increments when compared to the total capacity (remember that we are assuming only one machine at each station). For a given λ, (47) and (48) suggest that changes in the capacity μ result in changes in ca and cs. Therefore, L_j is a function of $\mu_1, \mu_2, \dots, \mu_n$. However, this functional relationship is complex since the system of equations in (47) is non-linear and not easy to analyze.

Bitran and Tirupati assumed that ca and cs are independent of changes in capacity μ. In this way, L_j is not dependent on $\mu_i, i \neq j$ (condition (ii) is satisfied). They assumed that as we modify μ, the mean and variance of the service time vary in the same proportion and hence, cs remains nearly constant. Furthermore, the sensitivity of ca to changes in μ seems to be small, as we increase the number of classes, and the proportion of load due to each class decreases (see the numerical results in Bitran and Tirupati (1988) and the discussion in Whitt (1988)). The consequence of these assumptions is that we can first solve system (47) for a given initial capacity, and then treat the resulting ca as known parameters in (48). Under these assumptions, Bitran and Tirupati also showed that $L_j(\lambda_j, ca_j, \mu_j, cs_j)$ in (48) is a convex function of μ_j, now denoted simply by $L_j(\mu_j)$ (condition (i) is satisfied). The WIP of the network may be expressed as (compare to (39)):

$$L(\mu) = \sum_{j=1}^{n} v_j L_j(\mu_j) \tag{49}$$

Finally, we denote by μ_j^0 a lower bound on the capacity at station j. Note that this bound must satisfy the condition $\rho_j < 1$ to avoid system instability:

$$\mu_j^0 > \lambda_j \tag{50}$$

Model SP1.1/G/S/R

Similarly to section 3.2.1, let L_T be a target WIP level of the network, such that $L_T < L(\mu^0)$. Let also $F_j(\mu_j)$ be the cost of allocating capacity μ_j to station j, defined as a convex non-decreasing differentiable function of μ_j (condition (iv) is satisfied). Using (49) and (50), we obtain the following convex programming problem:

$$\text{(SP1.1/G/S/R)} \quad \min \quad F(\mu) = \sum_{j=1}^{n} F_j(\mu_j)$$

$$\text{subject to:} \quad L(\mu) \leq L_T$$

$$\text{with:} \quad \mu_j \geq \mu_j^0, j = 1, \ldots, n$$

Bitran and Tirupati (1989a) presented a heuristic algorithm (algorithm 3) to solve SP1.1/G/S/R and to generate trade-off curves between $F(\mu)$ and $L(\mu)$. Let $PI_j(\mu_j)$ be a priority index, now defined as the quotient of the increase of marginal cost and the decrease of marginal WIP at station j, given by:

$$PI_j(\mu_j) = \frac{\partial F_j(\mu_j)/\partial \mu_j}{-v_j \partial L_j(\mu_j)/\partial \mu_j} \tag{51}$$

Algorithm 3 is similar to algorithm 1 from section 3.2.1. Let Δ be a previously specified increment of capacity at each iteration. We start with an initial capacity satisfying (50) for all stations. At each iteration, we increase by Δ the capacity of the station with the minimum priority index (51). The procedure is repeated until the target L_T is achieved.

Algorithm 3

1. Start with the allocation $\mu_j = \mu_j^0$ (sufficiently small), and compute ca_j and cs_j (using (47)) for each station j, j=1,...,n. This solution is infeasible ($L(\mu^0) > L_T$) and its cost $F(\mu^0)$ is less than the minimum cost of SP1.1/G/S/R.

2. At each iteration, update the cost $F(\mu)$, WIP $L(\mu)$ (using (48) and (49)) and $PI_j(\mu_j)$ (using (51)). Add capacity Δ to the station j^* which results in the smallest PI_{j^*} (greedy strategy), given by:
$$PI_{j^*} = min\{PI_j(\mu_j), j = 1, \dots, n\} \tag{52}$$

3. Stop as soon as $L(\mu)$ reaches the target L_T (feasible solution).

As we choose smaller values for Δ, algorithm 3 generates more precise trade-off curves. Bitran and Tirupati (proposition 2) showed that in the limit $\Delta \to 0$, algorithm 3 optimally solves SP1.1/G/S/R (remember that we have assumed that all conditions (i)-(iv) are satisfied), and $PI_j(\mu_j)$ obtained in the last iteration corresponds to the dual multiplier associated with the WIP constraint of station j.

Bitran and Tirupati (proposition 3) also presented an error bound for the approximate solution value produced by algorithm 3. Let's suppose that algorithm 3 finds a feasible solution after p iterations, and let's denote this solution by μ^p and the optimal solution of SP1.1/G/S/R by μ^*. Then:
$$0 \leq F(\mu^p) - F(\mu^*) \leq \frac{(L_T - L(\mu^p))}{PI_{j^*}^p} + \delta \tag{53}$$
where $\delta = \Delta \sum_{i=1}^{p}(1 - \frac{PI_{j^*}^i}{PI_{j^*}^p})$ and $PI_{j^*}^i$ is the quotient obtained by (52) at iteration $i, i = 1, \dots, p$. Computational experiences with $\Delta = 0.1$ applied to a real-life example with 13 stations and 10 product classes resulted in a relative error of 0.6% between $F(\mu^p)$ and $F(\mu^*)$. This is acceptable in many practical situations. These experiences also indicated that the previous assumption of considering *ca* and *cs* independent of changes in μ is reasonable (observe in algorithm 3 that *ca* and *cs* are maintained constant). As an illustration, as the WIP was reduced from an initial value of 70000 to a final value of 30000 in the example above, the maximum change found in *ca* was 3%. This change was obtained by updating *ca* according to (47) at the final network configuration.

A refinement of algorithm 3 is to update *ca* in (47). In fact, Bitran and Sarkar (1994b) have explored this alternative. When the *ca* are not considered independent of capacity, there is no guarantee that SP1.1/G/S/R is a convex program. We do not know if L in (49) stays convex in μ because *ca* now changes as μ changes, according to (47). Thus, this alternative procedure may not converge to an optimal solution or even to a feasible solution. Nevertheless, Bitran and Sarkar (op.cit.) showed that the procedure converges under certain conditions for the initial data.

Model SP2.1/G/S/R

In this section, we analyze the problem of redistributing the existing capacity in the stations to minimize WIP. This redistribution is meaningful in networks with homogeneous capacity, that is, resources that can be shared by different stations (e.g., labor). Let μ_j^1 be the initial existing capacity at station j, such that $\mu_j^1 \geq \mu_j^0$. Using (49) and (50), we obtain:

$$(\text{SP2.1/G/S/R}) \quad \min \quad L(\mu)$$

$$\text{subject to:} \quad \sum_{j=1}^{n} \mu_j = \sum_{j=1}^{n} \mu_j^1$$

$$\text{with:} \quad \mu_j \geq \mu_j^0, j = 1, \ldots, n$$

Bitran and Tirupati (1989a) presented a heuristic algorithm (algorithm 4) also based on the greedy method to solve SP2.1/G/S/R (again, conditions (i)-(iv) are satisfied). Let Δ be defined as before (see algorithm 3), and let $PI_j(\mu_j)$ be defined now as the decrease of marginal WIP at station j, given by:

$$PI_j(\mu_j) = -v_j \frac{\partial L_j(\mu_j)}{\partial \mu_j} \tag{54}$$

Algorithm 4

1. Start with the feasible allocation $\mu_j = \mu_j^1$, and compute ca_j and cs_j (using (47)) for each station j, j=1,...,n. Define J_0 as the set of available stations, J_1 as the set of stations to which capacity is increased, and J_2 as the set of stations to which capacity is reduced. Initially, $J_0 = \{1, 2, \ldots, n\}$, and J_1 and J_2 are empty. Compute ε_j such that:

$$PI_j(\mu_j)(\lambda_j + \varepsilon_j) = max\{PI_j(\mu_j), j \in J_0\} \tag{55}$$

2. At each iteration, update the WIP $L(\mu)$ (using (48) and (49)), and $PI_j(\mu_j)$ (using (54)). Find the station j_1 which results in the smallest PI_{j_1} given by:

$$PI_{j_1} = min\{PI_j(\mu_j), j \in J_0\} \tag{56}$$

and the station j_2 which results in the largest PI_{j_2} given by:

$$PI_{j_2} = max\{PI_j(\mu_j), j \in J_0\} \tag{57}$$

2a. If $j_1 \in J_1$, then make $J_0 \leftarrow J_0 - \{j_1\}$.

2b. If $j_2 \in J_2$, then make $J_0 \leftarrow J_0 - \{j_2\}$.

2c. If $j_1 \notin J_1$ and $j_2 \notin J_2$, then define $\Delta_1 = min\{\Delta, \mu_{j_1} - \lambda_{j_1} - \varepsilon_{j_1}\}$ and make $\mu_{j_1} \leftarrow \mu_{j_1} - \Delta_1$, $\mu_{j_2} \leftarrow \mu_{j_2} + \Delta_1$, $J_1 \leftarrow J_1 \cup \{j_2\}$ and $J_2 \leftarrow J_2 \cup \{j_1\}$.

3. Stop if J_0 is empty or unitary, or $PI_{j_1} = PI_{j_2}$.

Note that, at each iteration, (56) and (57) correspond to the stations that produce the largest and smallest marginal reduction in $L(\mu)$, respectively. Expression (55) together with Δ_1 guarantees that the solution generated by algorithm 4 satisfies $\mu_j \geq \lambda_j + \varepsilon_j, j = 1, \ldots, n$. Hence, it satisfies (50) and is feasible. Bitran and Tirupati (1989a, remark) showed that in the limit $\Delta \to 0$, this solution is optimal to SP2.1/G/S/R (remember that we have assumed all conditions (i)-(iv) satisfied), and all $PI_j(\mu_j)$ in the last iteration have the same value. As before, each $PI_j(\mu_j)$ can be interpreted as the dual multiplier associated to the capacity constraint at station j. It represents the rate of decrease in WIP due to marginal additions in the capacity of this station.

Bitran and Tirupati (proposition 4) also presented an error bound for the approximate solution value produced by algorithm 4. Let μ^p denote the heuristic solution found at the last iteration p, and μ^* denote the optimal solution of SP2.1/G/S/R. Then:

$$0 \leq F(\mu^p) - F(\mu^*) \leq n\Delta PI^p_{j_2} \tag{58}$$

where $PI^p_{j_2}$ is the priority index obtained from (57) at the last iteration p. Note in (58) that the solution μ^p is optimal in the limit $\Delta \to 0$. Computational experience with $\Delta = 0.02$ applied to the same practical example of the previous section resulted in a relative error smaller than 2% between $F(\mu^p)$ and $F(\mu^*)$, indicating that algorithm 4 is a good approximation to SP2.1/G/S/R. Bitran and Tirupati reported an interesting result from this example: the WIP is reduced from a initial value of 70000 to a final value around 40000 just by redistributing the initial capacity of the network (note however that they have assumed that the capacity of one station is completely transferable to other station). In order to test the hypothesis of independence of *ca* to capacity changes, they recomputed *ca* according to (47) at the final network configuration (remember that algorithm 4 maintains *ca* and *cs* fixed during the iterations). The largest variation found in *ca* was around 3%.

Note that algorithms 2 and 4 help balancing a manufacturing system, whereas algorithms 1 and 3 efficiently add resources to that system. One may generate trade-off curves between working capital (WIP) and investment capital by, firstly, applying algorithm 4 to the original system configuration and then, utilizing the solution obtained, μ^p, as a starting capacity for algorithm 3 (i.e., $\mu^0 \leftarrow \mu^p$, where μ^0 is the initial capacity at step 1 of algorithm 3). For a computational experience and analysis of trade-off curves, see e.g. Bitran and Tirupati (1989a) and Bitran and Morabito (1994).

Wein (1990) analyzed the model SP2.1/G/S/R for a single-class $GI/G/1$ OQN with all jobs following a probabilistic routing. Starting from the *Brownian model* proposed by Harrison and Williams (1987), which is based on heavy-traffic approximations (Reiman, 1984), Wein obtained the expected number of jobs at station j (in equilibrium) given by:

$$L_j(\mu_j) = \frac{\sigma_j}{2(\mu_j - \lambda_{0j})} \tag{59}$$

where

$$\sigma_j = \lambda_{0j}ca_j + \lambda_j cs_j + \sum_{i=1}^{n} \lambda_i q_{ij}(cs_i q_{ij} + 1 - q_{ij})$$

Note that expression (59) is not derived from the approximate decomposition methods discussed in section 2.2, such as expression (48). Furthermore, (59) is valid only if a certain condition, called *skew-symmetry*, is satisfied (see expression (4) in Wein, 1990). Let's consider again the budget constraint used by Kleinrock (1964) and discussed in the section 3.1.1. Assuming that the skew-symmetry condition is satisfied and using (59), the model SP2.1/G/S/R may be formulated as below:

$$\text{(SP2.1/G/S/R')} \quad \min \sum_{j=1}^{n} L_j(\mu_i)$$

$$\text{subject to: } \sum_{j=1}^{n} f_j \mu_j = F_T$$

$$\text{with: } \mu_j \geq 0, j = 1, \ldots, n$$

After deriving the Lagrangean function of this problem, Wein obtained a closed form solution $\mu_j^*, j = 1, \ldots, n$, given by:

$$\mu_j^* = \lambda_j + \frac{\sqrt{f_j \sigma_j}}{\sum_{i=1}^{n} \sqrt{f_i \sigma_i}} \frac{(F_T - \sum_{i=1}^{n} f_i \lambda_i)}{f_j} \tag{60}$$

Wein observed that the skew-symmetry condition is satisfied for Jackson networks (i.e., $M/M/1$ systems with $ca_j = 1$ and $cs_j = 1, j = 1, \ldots, n$), and (60) reduces to (37) which is the optimal solution to SP2.1/J/S/R'. Note that if f_j is equal for all stations, then (60) first allocates enough capacity to station j just to compensate λ_j, and then allocates capacity to station j in proportion to the square root of the parameter σ_j.

Wein presented computational experiences from a simple network example satisfying the skew-symmetry condition. These results showed that (60) produces a solution very close to the optimal solution found by simulation. Although (60) is derived under heavy-traffic conditions ($\rho_j \geq 0.9$), it may also produce good approximations for low traffic intensities. An important question is to investigate the quality of the solution generated by (60) in situations where the skew-symmetry condition is not satisfied.

3.2.2 Model SP1.1/G/M/R with Discrete Variables

Bitran and Tirupati (1989b) presented a heuristic algorithm to solve the model SP1.1/G/M/R with *discrete alternatives* for capacity changes at each station. Jobs belong to multiple classes and each class follows a different deterministic route, according to section 2.2.3. Similarly as we have done in section 3.2.1, the system of equations (19) plus (29), (31) and (32) from step 1 of the decomposition method is described below simply as:

$$\Phi(m, \lambda, ca, \mu, cs) = 0 \tag{61}$$

where vectors m, λ, ca, μ and cs denote respectively the parameters $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ for all stations $j, j = 1, \ldots, n$. Applying Little's law in (21) and adding the offered load, we obtain the expected number of jobs at station j, given by:

$$L_j(m_j, \lambda_j, ca_j, \mu_j, cs_j) = \frac{\lambda_j(ca_j + cs_j)}{2} Lq_j(M/M/m_j) + \frac{\lambda_j}{\mu_j} \tag{62}$$

where $Lq_j(M/M/m_j)$ denotes the expected number of jobs in queue at a $M/M/m_j$ system. Note that $Lq_j(M/M/m_j)$ corresponds to the first term on the right hand side of (38). Similarly to (48), $L_j(m_j, \lambda_j, ca_j, \mu_j, cs_j)$ in (62) is a function of m_j, λ_j, ca_j, μ_j and cs_j satisfying (61) (for other approximations of L_j, see e.g. Whitt (1993)). Instead of choosing m or μ as the decision variables of the model, Bitran and Tirupati considered a finite number of alternatives for capacity change at each station. Let's define n_j as the total number of alternatives at station j. For each alternative $k, k = 1, \ldots, n_j$, the following input data is given:

m_{jk} number of identical machines at station j in alternative k
μ_{jk} expected service rate of each machine at station j in alternative k
f_{jk} cost of station j in alternative k.

Define u_{jk} as a 0-1 decision variable (condition (iii) is not satisfied), such that:

$$u_{jk} = \begin{cases} 1 & \text{if alternative } k \text{ is chosen for station } j \\ 0 & \text{otherwise} \end{cases}$$

where $\sum_{k=1}^{n_j} u_{jk} = 1$. For each station j, the capacity choice is represented by the vector $(u_{j1}, u_{j2}, \ldots, u_{j,n_k})$ where all elements are null except one. In this way, we have $m_j = \sum_{k=1}^{n_j} m_{jk}u_{jk}$ and $\mu_j = \sum_{k=1}^{n_j} \mu_{jk}u_{jk}$ and hence, (61) and (62) depend on u_{jk}. Let $u = \{u_{jk}, j = 1, \ldots, n; k = 1, \ldots, n_j\}$. Similarly to section 3.2.1, Bitran and Tirupati assumed that ca and cs are independent of capacity changes in the network (see the discussion in the section 3.2.1). As a consequence, we may first solve system (61) for a given u (i.e., a given capacity m and μ), and then treat the resulted values of ca and cs as fixed parameters in (62). Furthermore, as we choose an alternative k at station j (i.e., $u_{jk} = 1$ and $u_{jl} = 0, l \neq k$), we may refer to $L_j(m_j, \lambda_j, ca_j, \mu_j, cs_j)$ in (62) simply as L_{jk}, where $L_{jk} = L_j(m_{jk}, \lambda_j, ca_j, \mu_{jk}, cs_j)$. Note that, in this way, we can compute L_{jk} for every alternative k and every station j using (62). Without loss of generality, we assume that if $L_{jk} > L_{jl}$, then $f_{jk} < f_{jl}, k \neq l, k, l = 1, \ldots, n_j$. Similarly to (49), the network WIP can be written as:

$$L(u) = \sum_{j=1}^{n} \sum_{k=1}^{n_j} v_j L_{jk} u_{jk} \tag{63}$$

where, as before, v_j is the mean value of a job at station j. Using (63), we obtain the following problem:

$$\text{(SP1.1/G/M/R)} \ \min \ F(u) = \sum_{j=1}^{n} \sum_{k=1}^{n_j} f_{jk} u_{jk}$$

$$\text{subject to: } L(u) \leq L_T$$

$$\sum_{k=1}^{n_j} u_{jk} = 1, \ \text{for } j = 1, \ldots, n$$

$$\text{with: } u_{jk} \in \{0,1\}, j = 1, \ldots, n, k = 1, \ldots, n_j$$

where L_T is a given target for the network WIP. Note that $L(u)$ and $F(u)$ are assumed to be linear functions of u. SP1.1/G/M/R models situations in which the target L_T is achieved by increasing capacity either by means of additional machines, workers, or by increasing the availability with overtime, or additional shift operations. Bitran and Tirupati proposed a heuristic algorithm (algorithm 5) to solve the integer linear program SP1.1/G/M/R above. They showed that: (i) the optimal solution of the linear program relaxation of SP1.1/G/M/R has either *zero* or *two* different variables u_{jk} with fractional values (proposition 3.1), and (ii) if this optimal solution has *two* variables with fractional values, then they correspond to the same station (corollary). The next algorithm described below produces the approximate solution u^1:

Algorithm 5

1. Let u^0 be the optimal solution to the LP relaxation of SP1.1/G/M/R. If u^0 is a feasible solution to SP1.1/G/M/R, then $u^1 = u^0$ is an optimal solution to SP1.1/G/M/R, otherwise go to step 2.

2. Let i be the station which variables are fractional values for some k_1 and k_2 ($0 < u^0_{ik_1} < 1$, $0 < u^0_{ik_2} < 1$). A feasible solution to SP1.1/G/M/R is given by:

$$u^1_{jk} = u^0_{jk}, j \neq i, j = 1, \ldots, n, k = 1, \ldots, n_j$$

$$u^1_{ik} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise} \end{cases}$$

where l is such that:

$$L_{il} = max\{L_{ik} | L_{ik} \leq L_{ik_1} u^0_{ik_1} + L_{ik_2} u^0_{ik_2}, k = 1, \ldots, n_i\}$$

The authors also presented an error bound on the value of approximate solution u^1 generated by algorithm 5. Without loss of generality assume that $L_{ik_1} > L_{ik_2}$, and denote by u^* the optimal solution of SP1.1/G/M/R. Then:

$$0 \leq F(u^1) - F(u^*) \leq f_{ik_2} - f_{ik_1} \leq max\{f_{jk}, j = 1, \ldots, n, k = 1, \ldots, n_j\}$$

Computational experiences from a real-life network example of 13 stations and 10 product classes indicate that algorithm 5 is a good approximation to SP1.1/G/M/R when the number of classes is relatively large. In this example, as the network WIP was reduced from an initial value of 80000 to a final value below 30000, the relative error between $F(u^1)$ and $F(u^*)$ was less than 0.08%. The largest change in *ca* was equal to 4.6%, corresponding to a change of 0.5% in the WIP (remember that the values of *ca* and *cs* are also kept constant in algorithm 5). Bitran and Tirupati left the development of approaches to situations involving a small number of classes and mixtures of deterministic and probabilistic routings for future investigation.

3.2.3 Models ./G/M/N

Van Vliet and Rinnooy Kan (1991) presented two algorithms to solve models SP1.1/G/M/N and SP2.1/G/M/N, based on marginal analysis and greedy methods. These algorithms are closely related to the two algorithms presented by Boxma et al (1990) to solve models SP1.1/J/M/N and SP2.1/J/M/N (described in the section 3.1.2). Again, jobs belong to multiple classes and each class follows a different deterministic route. In contrast with section 3.2.2, the decision variables are the number of machines at the stations.

Let's consider again the system of equations (61), and expression (62) for the expected number of jobs in a $GI/G/m_j$ queue at station j. Van Vliet and Rinnooy Kan considered each capacity $m_j, j = 1, \ldots, n$, as an *integer* decision variable. Given λ and μ, (61) and (62) suggest that changes in capacity m result in changes in *ca* and *cs* (L_j is a function of $m_1, m_2, \ldots, m_n$). Note, however, that this functional relationship is not easy to analyze.

Based on the results from Bitran and Tirupati (1989a) (see section 3.2.1), Van Vliet and Rinnooy Kan assumed that ca and cs are independent of changes in capacity m. Therefore, L_j is not dependent on $m_i, i \neq j$ (condition (ii) is satisfied). They argued that as we modify m, the mean and variance of the service time vary in the same proportion and hence, cs remains nearly constant. Furthermore, the sensitivity of ca to changes in m seems to be small as the number of classes increases and the proportion of load due to each class decreases. Hence, once the set of equations (61) is calculated, we can regard ca and cs as parameters for (62). This means that $L_j(m_j, \lambda_j, ca_j, \mu_j, cs_j)$ in (62) can be seen as a function of m_j only, now denoted by $L_j(m_j)$. Since $Lq_j(M/M/m_j)$ is a convex function of m_j, and we are assuming ca and cs as parameters, then $L_j(m_j)$ becomes a convex function of m_j (condition (i) is satisfied). Following the same steps of section 3.1.2, the network WIP $L(m)$ is defined according to (39) (where $L_j(m_j)$ is given by (62), instead of (38)), and the initial number of machines m^0 must satisfy (40).

Model SP1.1/G/M/N

The model SP1.1/G/M/N is formulated exactly as SP1.1/J/M/N described in section 3.1.2, where $F(m)$ is a convex non-decreasing function of m (condition (iv) is satisfied), and $L(m)$ is assumed to be convex in m, in accordance with the discussion above. SP1.1/G/M/N, also called the server allocation problem, is a convex program with integer variables and the use of marginal analysis do not lead necessarily to optimality (condition (iii) is not satisfied). The problem can be regarded as the minimum-cost allocation of machines so that the WIP is less than a given target WIP.

Van Vliet and Rinnooy Kan utilized algorithm 1 (section 3.1.2) to solve SP1.1/G/M/N. The algorithm starts with the smallest possible allocation of machines m^0 for all stations (infeasible allocation). At every iteration, it adds one machine at the station with the smallest priority index (i.e., the quotient of the increase of the objective function and the decrease of the network WIP). Note that this priority index is a result of marginal analysis. It is obtained by substituting (62) in (41). The algorithm terminates as soon as adding a machine makes the allocation feasible.

The error bound provided by Boxma et al (1990) (discussed in section 3.1.2) can also be applied here. For instance, if p is the last iteration of algorithm 1 and m^* is the optimal solution of SP1.1/G/M/N, we have: $F(m^{p-1}) < F(m^*) \leq F(m^p)$. Van Vliet and Rinnooy Kan generated trade-off curves between cost and WIP similar to those discussed by Bitran and Tirupati (1989a). Using different computational results from two network examples, they found a relative error of 7% in the solution of the first example and 5% in the solution of the second example. This relative error decreases as the target WIP imposed to the problem decreases. Van Vliet and Rinnooy Kan also recalculated ca for the final configuration of the network to verify the sensitivity of ca to changes of m. They obtained an error in ca below 6%, suggesting that this approach is a good approximation to problem SP1.1/G/M/N.

Model SP2.1/G/M/N

Similarly, the model SP2.1/G/M/N is formulated exactly as the model SP2.1/J/M/N described in section 3.1.2, where M is the number of available machines such that $M > \sum_{j=1}^{n} m_j^0$. Again, we have the objective function and the constraint of the problem as convex functions of m. SP2.1/G/M/N, also called the server reallocation problem, is a convex program with integer variables and it can be regarded as the minimum-WIP redistribution of the M machines over the network.

Van Vliet and Rinnooy Kan utilized algorithm 2 (section 3.1.2) to solve SP2.1/G/M/N. The algorithm starts with the smallest possible allocation of machines m^0. At every iteration, it adds one machine to the station with the largest priority index (i.e., the greatest decrease in the network WIP per machine). Note that this priority index is obtained from marginal analysis by substituting (62) in (44). The algorithm terminates when all the M machines have been allocated.

Since we are assuming $L(m)$ as a convex function of m, algorithm 2 terminates after $O(Mn)$ steps with an optimal reallocation of machines (see section 3.1.2 for details). Computational experiments with the two examples above indicated that the sensitivity of ca to changes in m is small. Therefore, the optimal solution produced by algorithm 2 to the assumed convex problem may be utilized as a good approximation to the original problem.

4 Perspectives for Further Research

In this section, we suggest topics for future research. The first one is concerned with the relevance of class SP3 (see section 1) to the focused factory design. Then, we discuss the importance of light-traffic approximations to manufacturing networks with multiple classes and deterministic routings (according to section 2.2.3). Finally, we return to the critical question of the existence of steady-state for an OQN representing a discrete manufacturing system.

4.1 Relevance of Class SP3

Different authors have pointed out that modern manufacturing systems are becoming more complex due to: (i) the large number of product classes competing for common resources, (ii) the uncertainty of product demands, and (iii) the reduction of life cycles. In addition to developing efficient methods to analyze more complex systems, we may also try to reduce the complexity of the manufacturing environment. Much of the success of JIT and other related methods comes from simplification. Examples of alternatives to reduce complexity include partitioning existing production lines, duplication of resources, and redesigning products and manufacturing processes. Note that problems of class SP3 (e.g., problem SP3.1 in section 1) can be regarded in this context.

Recent attempts based on OQN models have been made to analyze the trade-off between the partitioning of product lines and duplication of machines (Bitran and Sarkar, 1994c; see also Tang and Yoo (1991) for a related study of customer partition and server allocation applied to a single node service system). The idea is to relate *complexity* and *predictability* of a system, suggesting that more complex systems tend to have poorer predictability. Therefore, we can use system predictability of job completion as a measure of complexity. For instance, as we increase the number of product classes manufactured in a system, complexity tends to increase and predictability tends to decrease due to product interference at stations. Of course we may reduce complexity by *adding capacity* to stations. We may also reduce complexity by simply *partitioning the shop.* In what follows, we suggest possible measures of complexity to capture this notion. We consider: (i) measures of complexity from a product management point of view, (ii) measures of complexity from a station management point of view.

4.1.1 Measures of Complexity from a Product Management Point of View

Managers should be able tc predict product leadtimes as accurately as possible. In other words, it is desirable to have small leadtime variances. We may reduce variances by adding additional machines to the stations. Let T_k be the leadtime of a product from class k, w_k be a weight associated with a product from class k, and T be an upper bound on the weighted leadtime of all classes in the network. Then, we can formulate the following complexity constraint:

$$\sum_{k=1}^{r} w_k V(T_k) \leq T \tag{64}$$

Note that the smaller the bound T, the higher is the predictability of the system. Each variance $V(T_k)$ in (64) is defined as the sum of the variances of waiting times and service times of all stations in class k routing. For general OQNs, $V(T_k)$ can be estimated by using the approximate decomposition methods discussed in section 2.2. For simplicity, let's assume that service times are deterministic at all stations and hence, their variances are null. Then, we obtain (Bitran and Sarkar, 1994c):

$$V(T_k) = \sum_{l=1}^{n_k} \sum_{j=1}^{n} V(Wq_j)\, 1\{j : j = n_{kl}\} \tag{65}$$

where $V(Wq_j)$ is the waiting time variance at station j, given by:

$$V(Wq_j) = [Wq_j(M/M/m_j)]^2 \frac{(ca_j + cs_j)}{4} \tag{66}$$

where $Wq_j(M/M/m_j)$ is the expected waiting time for a $M/M/m_j$ system. Since $Wq_j(M/M/m_j)$ is a convex decreasing function of m_j and assuming ca_j and cs_j independent of capacity changes in the network (see section 3.2.1 for a detailed discussion), it follows from (66) that $V(Wq_j)$ decreases as we increase m_j. Therefore, as we add machines to the stations in the routing of class k, we reduce the variance $V(T_k)$ in (65) and hence, the system complexity on the left side of (64).

Expressions (64)-(66) also suggest that for the same overall shop capacity, we might reduce system complexity by appropriately partitioning the shop into sub-shops or production lines with more homogeneous product mix. In this way, we might obtain smaller variability parameters at the stations of each sub-shop, such that the overall variance of each class in (65) would be reduced and hencė, the system complexity in (64).

4.1.2 Measures of Complexity from a Station Management Point of View

Bitran and Sarkar (1994c) have pointed out that as the number of machines increases at one station, we expect to obtain more *flexibility* (in terms of scheduling and maintenance) to operate that station. In order to determine the number of machines for each station we should consider the uncertainties of interarrival and service times of products visiting the station.

Whitt (1992) discussed some heuristics that may be useful to describe complexity constraints with respect to the *grade of service* of each station. A grade of service is a measure that, when fixed, maintains a certain *congestion measure* nearly constant in the station (later we point out the relationship between the grade of service and the flexibility of the station). For example, let γ_j be a grade of service for station j, given by:

$$\gamma_j = (1-\rho_j)\sqrt{m_j} \tag{67}$$

Expression (67) suggests an *economy of scale*, that is, for the same grade of service, the expected utilization level increases as we increase the number of machines and the expected arrival rate λ_j at the station (recall that $\rho_j = \frac{\lambda_j}{m_j \mu_j}$). Note, however, that the rate of increase of the number of machines is smaller than the rate of increase of the expected arrival rate. Whitt showed that if we maintain γ_j constant in (67), then we also maintain the congestion measure $P(W_j > 0)$ nearly constant (i.e., the probability of a positive waiting time). This result is supported by heavy-traffic limit theorems, and was observed in computational experiments. In particular, Whitt showed that, for a $GI/G/m_j$ queue, we have the following approximation:

$$\gamma_j \approx \frac{(ca_j + cs_j)}{2\sqrt{m_j}E(W_j|W_j > 0)} \tag{68}$$

where $E(W_j|W_j > 0)$ is also a congestion measure. It corresponds to the expected waiting time in the queue of station j, given that the waiting time is greater than zero. Combining (67) and (68), we obtain another example of grade of service for station j, η_j, defined as:

$$\eta_j = \frac{(1-\rho_j)m_j}{(ca_j + cs_j)} \approx \frac{1}{2E(W_j|W_j > 0)} \tag{69}$$

Equation (69) implies that given a grade of service η_j, we maintain the congestion measure $E(W_j|W_j > 0)$ nearly constant. Assuming that ca_j and cs_j are independent of capacity changes in the network (see section 3.2.1 for a detailed discussion), it follows from (69) that as we add machines and increase the expected arrival rate at station j, the expected utilization increases for the same grade of service. Let's define the following constraint for each station j of the network:

$$\frac{(1-\rho_j)m_j}{(ca_j + cs_j)} \geq G_j \tag{70}$$

where G_j is a lower bound on the grade of service of station j (note that G_j is also an upper bound on the congestion measure $E(W_j|W_j > 0)$). The parameter G_j can be also viewed as the minimum desired flexibility for station j. For larger values of ca_j and cs_j, we should increase the number of machines at station j in order to satisfy the desired flexibility.

Expression (70) suggests that in some cases we can satisfy the minimum desired flexibility at stations without changing the overall number of machines in the network. Partitioning appropriately the shop into sub-shops with more homogeneous product mix, we may obtain smaller variability parameters, say ca_j^i and cs_j^i for each station j of each sub-shop i, such that the left-hand side in (70) would be increased for all j and i.

4.1.3 Focused Factory Design

The *focused factory design* involves product allocation to production lines, and capacity allocation to the stations of each line. This problem may be regarded as an instance of class SP3 (see problem

SP3.1 in section 1), and it is different from problems discussed in section 3, where only capacity allocation was involved.

An interesting research issue is developing optimization models to analyze the trade-off between the partitioning of product lines and the duplication of machines in the focused factory design. We might incorporate to these models the complexity constraints from the product and station management points of views, such as (64) and (70) discussed above. These constraints can help us represent desired product leadtimes and desired station flexibility in the system.

A recent research based on these ideas (Bitran and Sarkar, 1994c) reveals an unexpected result:

> *Contrary to the commonly view, the number of machines required can be less when production lines are partitioned (compared to when they are put in one shop only).*

This result suggests that we may reduce the complexity of the network (left side of (64)), or increase the flexibility of the stations (left side of (70)), by only partitioning the facility into product lines. Further research might investigate the stability of optimal partitions. For instance, the solution sensitivity to changes on the expected arrival rate of products, or to changes on the desired grade of service of stations. We might also consider particular situations where privileged product classes must have lower leadtimes, or follow routings through high service grade stations.

4.2 Light-Traffic Approximations

In this section, we emphasize the importance of *light-traffic approximations* to analyze OQNs with multiple classes and deterministic routings. Upon certain conditions, these approximations may effectively simplify performance evaluation of large and complex manufacturing networks such as job-shops.

As discussed in section 2.2.3, Bitran and Tirupati (1988, remark 2) suggested that if the number of product classes processed at each station is large enough, we may ignore the interaction between stations and analyze each station individually. The mean and variance of each product class are preserved throughout the network, and we may assume that at every station, the mean and variance are the same as those of the external arrival process of this class. In other words, as the number of classes increases, we expect that $q_{kl} \rightarrow 0$ and $cd_{kl} \rightarrow ca_{kl}$ in (31) for all k and l, and hence, $cd_{kl} \rightarrow ca'_k$. The network may be decomposed in a set of independent stations, each one analyzed as a single queue system (similarly to the Jackson networks discussed in section 2.1). Note that, in this way, we have virtually no limits on the size of the OQN that could be analyzed. However, practical manufacturing networks usually have one or more stations working under heavy-traffic intensity condition.

Based on the argument above, Whitt (1988) observed that if the arrival rate of one class upon one visit at a station is a small proportion of the total arrival rate there (i.e., q_{kl} is small for class k during visit l to a certain station), then the departure process for that class from that station should be nearly the same as the arrival process for that class for that visit (i.e., $cd_{kl} \rightarrow ca_{kl}$). Note that this can be regarded as a light-traffic approximation, where the class of interest has light-traffic at the station; however, the overall traffic intensity of the station does not need to be low. This observation can be extended to the network. If the light-traffic condition occurs at every station in the routing of the class, then the arrival and departure processes for this class at the stations in its routing should be nearly the same as its external arrival process (i.e., $ca_{kl} \rightarrow ca'_k$ and so,

$cd_{kl} \rightarrow ca'_k$). Again, note that only the class of interest must be in light-traffic at the stations. This principle becomes more meaningful as the size and complexity of the network increases.

Whitt also observed that there is another condition to be satisfied for the light-traffic approximation. The offered load for the class of interest must also be small, that is, its service time can not grow indefinitely at the same time. In other words, the light-traffic condition assumes that the time scale for arrivals and departures from the class of interest is much longer than the time scale for the aggregate of all other classes (aggregate class) at the station. For example, the interarrival times for jobs of the designated class are days or months (and the service times are hours or minutes), whereas interarrival times for jobs of the aggregate class are hours or minutes. Thus, the waiting times and service times for the class of interest are negligible if compared to their interarrival times at the station.

The use of light-traffic approximations, therefore, might permit enormous simplifications for the analysis of classes in light-traffic in OQNs. However, an important pragmatical question remains: *When could these approximations be actually applied to manufacturing networks?* For instance, which practical values of traffic intensity and proportion of a class at a station would satisfy the light-traffic condition for that class at that station? How to combine light-traffic approximations with busy stations?

4.3 Steady-State Assumption

In all OQN models discussed in this study, we have assumed that the system attains equilibrium or *steady-state.* The system is supposed to travel through different *transient states* until it achieves steady-state. However, the steady-state may not exist, or if it exists it may not be attainable by the system during its life time. In particular, many authors have criticized the steady-state assumption for discrete manufacturing systems. Changes often happen in this environment due to new products, premature obsolescence of current products, capacity changes in the workstations, process technology updates, and so on. Examples of critical questions are: Do manufacturing systems maintain their characteristics during a time long enough to attain steady-state? How is the sensitivity of the steady-state to changes in the system configuration? Further research exploring these themes would be helpful to characterize instances where the steady-state hypothesis can be assumed.

Acknowledgments

The authors would like to thank Dr. Deb Sarkar of the AT&T Bell Laboratories and Luis C. Pedrosa of the Sloan School of Management for their helpful comments and suggestions. This research was partially supported by grant #93/0891-7 from Fundação de Amparo a Pesquisa do Estado de São Paulo, Brazil.

References

Albin, S. L. (1982). Poisson approximations for superposition arrival processes in queues. *Mgmt. Sci.* 28(2), 126-137.

Albin, S. L. (1984). Approximating a point process by a renewal process, II: Superposition arrival processes of queues. *Oper. Res.* 32, 1133-1162.

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

Baskett, F., K. M. Chandy, R. R. Muntz and F. G. Palacios (1975). Open, closed, and mixed networks of queues with different classes of customers. *J. ACM* 22(2), 248-260.

Bazaraa, M. S., H. D. Sherali and C. M. Shetty (1993). *Nonlinear programming: Theory and algorithms*, 2nd.ed., Wiley, NY.

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

Bitran, G. R. and R. Morabito (1994). Manufacturing systems design: Trade-off curve analysis. *Working Paper*, Sloan School of Management, MIT.

Bitran, G. R. and D. Sarkar (1994a). Throughput analysis in manufacturing networks. *EJOR* 74, 448-465.

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

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

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

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

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

Bitran, G. R. and D. Tirupati (1989c). Approximations for product departures from a single server station with batch processing in multi-product queues. *Mgmt. Sci.* 35, 851-878.

Bitran, G. R. and D. Tirupati (1991). Approximations for network of queues with overtime. *Mgmt. Sci.* 37(3), 282-300.

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

Brown, E. (1988). IBM combines rapid modeling technique and simulation to design PCB factory-of-the-future. *Ind. Eng.*, June 88, 23-90.

Buzacott, J. A. and J. G. Shanthikumar (1992). Design of manufacturing systems using queueing models. *Queueing Syst.* 12, 135-214.

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

Buzacott, J. A. and D. D. Yao (1986). Flexible manufacturing systems: A review of analytical models. *Mgmt. Sci.* 32(7), 890-905.

Calabrese, J. M. (1992). Optimal Workload Allocation in Open Networks of Multiserver Queues. *Mgmt. Sci.* 38(12), 1792-1802.

Chase, R. B. and N. J. Aquilano (1992). *Production and operations management - A life cycle approach*, Irwin, Homewood, MA.

Dallery, Y. and R. David (1986). Operational analysis of multiclass queueing networks. *Proc. 25th IEEE Conf. Decision and Control*, 1728-1732.

Dallery, Y. and K. E. Stecke (1990). On the optimal allocation of servers and workloads in closed queueing networks. *Oper. Res.* 38(4), 694-703.

Denning, P. J. and J. P. Buzen (1978). The operational analysis of queueing networks. *ACM Computing Surveys* 10(3), 225-261.

Disney, R. L. and D. Konig (1985). Queueing networks: A survey of their random processes. *SIAM Rev.* 27(3), 335-403.

Erlang, A. K. (1917). Solution of some problems in the theory of probabilities of some significance in automatic telephone exchanges. *Post Office Electrical Engineer's Journal* 10, 189-197.

Gershwin, S. B. (1994). *Manufacturing systems engineering*, PrenticeHall, Englewood-Cliffs, NJ.

Harel, A. and P. Zipkin (1987). Strong convexity results for queueing systems. *Oper. Res.* 35, 405-418.

Harrison, J. and V. Nguyen (1990). The QNET method for two-moment analysis of open queueing networks. *Queueing Syst.* 6, 1-32.

Harrison, J. and V. Nguyen (1993). Brownian models of multiclass queueing networks: Current status and open problems. *Queueing Syst.* 13, 5-40.

Harrison, J. and R. Williams (1987). Brownian models of open queueing networks with homogeneous customer populations. *Stochastic* 22, 77-115.

Ho, Y. C. (1987). Performance evaluation and perturbation analysis of discrete event dynamic systems. *IEEE Trans. Automat. Control* 32, 563-572.

Ho, Y. C. and X. Cao (1983). Perturbation analysis and optimization of queueing networks. *J. Optimization Theory and Applications* 40(4), 559-582.

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

Jackman, J. and E. Johnson (1993). The role of queueing network models in performance evaluation of manufacturing systems. *J. Opl. Res. Soc.* 44(8), 797-807.

Jackson, J. R. (1957). Networks of waiting lines. *Oper. Res.* 5, 518-521.

Jackson, J. R. (1963). Job shop-like queueing systems. *Mgmt. Sci.* 10, 131-142.

Karmarkar, U. S. (1993). Manufacturing lead times, order release and capacity loading. *Handbooks in OR/MS*, S. C. Graves (ed.), vol 4, Elsevier, North-Holland, Amsterdam.

Kelly, F. P. (1975). Networks of queues with customers of different types. *J. Appl. Prob.* 12, 542-554.

Kelly, F. P. (1979). *Reversibility and Stochastic Processes*, Wiley, NY.

Kleinrock, L. (1964). *Communication nets: stochastic message flow and delay*, Dover Publ., NY.

Kleinrock, L. (1975). *Queueing systems*, vol 1: Computer applications, Wiley, NY.

Kleinrock, L. (1976). *Queueing systems*, vol 2: Computer applications, Wiley, NY.

Kobayashi, H. (1974). Application of the diffusion approximation to queueing networks I: Equilibrium queue distributions. *J. ACM* 21(2), 316-328.

Koenigsberg, E. (1982). Twenty five years of cyclic queues and closed queue networks: A review. *J. Opl. Res. Soc.* 33, 605-619.

Kuehn, P. J. (1979). Approximate analysis of general networks by decomposition. *IEEE Trans. Commun.* 27(1), 113-126.

Law, A. M. and S. W. Haider (1989). Selecting simulation software for manufacturing applications: practical guidelines & software survey. *Ind. Eng.*, May 89, 33-46.

Law, A. M. and M. G. McComas (1989). Pitfalls to avoid in the simulation of manufacturing systems. *Ind. Eng.*, May 89, 28-69.

Lemoine, A. J. (1977). Networks of queues - a survey of equilibrium analysis. *Mgmt. Sci.* 24, 464-481.

Pujolle, G. and W. Ai (1986). A solution for multiserver and multiclass open queueing networks. *INFOR* 24(3), 221-230.

Reiman, M. I. (1984). Open queueing networks in heavy-traffic. *Math. of Oper. Res.* 9, 441-458.

Reiman, M. I. (1990). Asymptotically exact decomposition approximations for open queueing networks. *Oper. Res. Lett.* 9, 363-370.

Reiser, M. and H. Kobayashi (1974). Accuracy of diffusion approximations for some queueing systems. *IBM J. Res. Dev.* 18, 110-124.

Ross, S. M. (1985). *Introduction to Probability Models*, Academic Press, Inc., Orlando, FL.

Sevcik, K. C., A. I. Levy, S. K. Tripathi and J. L. Zahorjan (1977). Improving approximations of aggregated queueing network systems. *Computer Performance* (eds. K. Chandy and M. Reiser), North-Holland, 1-22.

Segal, M. and W. Whitt (1989). A queueing network analyzer for manufacturing. *Teletraffic Science for New Cost-Effective Systems, Networks and Services*, ITC-12, M. Bonatti (ed.), Elsevier, North-Holland, Amsterdam, 1146-1152.

Seidmann, A., P. J. Schweitzer and S. Shalev-Oren (1987). Computerized closed queueing network models of flexible manufacturing systems: A comparative evaluation. *Large Scale Systems* 12, 91-107.

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

Shanthikumar, J. G. and J. A. Buzacott (1984). The time spent in a dynamic job shop. *EJOR* 17, 215-226.

Shanthikumar, J. G. and D. D. Yao (1987). Optimal server allocation in a system of multi-server stations. *Mgmt. Sci.* 33(9), 1173-1180.

Shanthikumar, J. G. and D. D. Yao (1988). On server allocation in multiple center manufacturing systems. *Oper. Res.* 36(2), 333-342.

Sundarraj, R. P., P. S. Sundararaghavan, D. R. Fox (1994). Optimal server acquisition in open queueing networks. *J. Oper. Res. Soc.* 45(5), 549-558.

Suresh, S. and W. Whitt (1990). The heavy-traffic bottleneck phenomenon in open queueing networks. *Oper. Res. Lett.* 9, 355-362.

Suri, R. (1989). Perturbation analysis: The state of the art and research issues explained via the G/G/1 queue. *Proc. IEEE* 77, 114-137.

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

Suri, R., G. W. Diehl and R. Dean (1986). Quick and easy manufacturing systems analysis using ManuPlan. *Proc. Spring IIE Conf.*, Dallas, TX, 195-205.

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

Tang, C. S. and S. Yoo (1991). System planning and configuration problems for optimal system design. *EJOR* 54, 163-175.

Tijms, H. C. (1986). *Stochastic modeling and analysis: A computational approach*, Wiley, NY.

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

Walrand, J. (1990). Queueing networks. *Handbooks in OR/MS*, D. P. Heyman and M. J. Sobel (ed.), vol 2, Elsevier, North-Holland, Amsterdam.

Wein, L. M. (1990). Capacity allocation in generalized Jackson networks. *Oper. Res. Lett.* 15, 215-242.

Wein, L. M. (1990). Scheduling networks of queues: Heavy traffic analysis of a two-station network with controllable inputs. *Oper. Res.* 38(6), 1065-1078.

Whitt, W. (1982). Approximating a point process by a renewal process, I: Two basic methods. *Oper. Res.* 30, 125-147.

Whitt, W. (1983a). The queueing network analyzer. *Bell Syst. Tech. J.* 62, 2779-2815.

Whitt, W. (1983b). Performance of the queueing network analyzer. *Bell Syst. Tech. J.* 62, 2817-2843.

Whitt, W. (1984). Open and closed models for networks of queues. *AT&T Bell Lab. Tech. J.* 63(9), 1911-1979.

Whitt, W. (1988). A light-traffic approximation for single-class departures from multi-class queues. *Mgmt. Sci.* 34(11), 1333-1346.

Whitt, W. (1992). Understating the efficiency of multi-server service systems. *Mgmt. Sci.* 38(5), 708-723.

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

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

Yao, D. D. and J. A. Buzacott (1986). The exponentialization approach to flexible manufacturing system models with general processing times. *EJOR* 24, 410-416.