Computerized closed queueing network models of flexible manufacturing systems: A comparative evaluation

Abraham SEIDMANN and Paul J. SCHWEITZER
W.E. Simon Graduate School of Business Administration, The University of Rochester, Rochester, NY 14627, U.S.A.

Sarit SHALEV-OREN
Department of Industrial Engineering, Tel Aviv University, Tel Aviv 69978, Israel

Received February 1986
Revised February 1987
Communicated by R. Suri

Advanced closed queueing network models are available for stochastic performance evaluation of Flexible Manufacturing Systems (FMS). These models are particularly useful during the planning and design phases of FMS, since they can be applied to study gross tradeoffs between principal design parameters. This paper reviews and analyzes several computerized models for evaluating complex FMS facilities with respect to the desired allocation of the following key resources: manufacturing centers, transporters, pallets and tools. It focuses on such issues as the structure of the mathematical system models, the variety of performance measures (model outputs), model inputs, accuracy of results and computational effort. In addition, original alternatives to overcome part of the common limitations of the models are developed and tested empirically.

1. Introduction

Flexible Manufacturing Systems (FMS) are designed to produce mid-sized batches of several different part types with the efficiency of automated mass production and the flexibility of job-shops. The number of new FMS facilities is expected to grow rapidly; in fact, Barash [1] estimates that about 5000 such systems will be in existence by the year 2000.

There are many complex design and planning problems associated with constructing and managing an FMS. Careful planning is required because the versatility of the machines generates many more options to consider than in conventional production systems [2]. Since the number of variables and possibilities are overwhelming, mathematical planning and management tools are required [11, 26].

The use of very detailed models may be unneccesary in the early design stages of various manufacturing systems and for strategic management decisions, e.g. part-mix changes, system modifications/expansions and performance monitoring. In recent years, several mathematical tools have been developed for the analysis of product flow in FMS. Most of these tools use either analytic closed queueing network (CQN) models or simulation [4, 12, 27]. With CQN models, one assumes that a fixed number of items (or pallets) circulate throughout the facility in accordance with prescribed routing requirements. Finished parts are immediately replaced at the load/unload stations by raw parts. The computed steady-state throughputs are determined by the complex relationships between the stochastic processing times at the machines and the random arrivals of parts to them; the expected manufacturing lead times are closely correlated with the expected sojourn times at the machines and at the robotic transporters [25]. Most of the CQN models are computationally very efficient: that is, they require relatively little input data, and consume little computer time. Thus, they are applied interactively to quickly obtain gross tradeoffs between principle design parameters and performance measures. One can, for instance, assess the impact of process selection, the number of machines, transporters and pallets on the expected machine utilization, throughputs and leadtimes. Rathmill, Green-

North-Holland
Large Scale Systems 12 (1987) 91–107

0167-420X/87/$3.50

wood and Housmand [17], Solberg [24] and Suri [28] all applied industrial operational data and detailed simulations to discuss the merits of using this approach for modelling FMS. Their conclusion was that closed queueing network models provided accurate predictions even when more detailed models would seem to be required. Some of these models were programmed and distributed as design tools to several industries, universities and research institutes [12]. For example, Rathmill et al. [17] used both mathematical model (CAN-Q) and simulation to support the design effort of the 'SCAMP' FMS in the United Kingdom; also, the MVAQ model developed recently by Suri and Hildebrant is used at the optronic FMS of Hughes Aircraft Company in El Segundo, California [31].

This paper presents a comparative evaluation of several closed queueing network models, as these have been the most widely used models for practical designs. Some of them were developed especially for FMS, and others are generic models for large complex queueing networks. The emphasis, as reflected in the title, will be on the specific issues relevant to FMS applications. Special attention is given, therefore, to the characterization of the transporters, computations of throughputs, the determination of the production mix and to the routing of parts. Some of the models discussed here suffer from limitations due to the inherent structures of their underlying mathematical models. A few alternatives to overcome part of these limitations will also be discussed.

In Section 2 we begin with a brief presentation of the various models and design tools. Then in Section 3 we describe the numerical experimentation procedure and present their results. Next, in Section 4 we focus on the interpretation of results regarding predictive accuracy and computational effort. Several concluding remarks on the relative capabilities of each approach are provided in Section 5.

2. The design and evaluation models

All the models discussed here assume that the manufacturing system consists of M stations (manufacturing centers) labeled $m = 1, 2, \ldots, M$ and a common storage area for in-process inventory. This area is assumed to be large enough to prevent blocking effects. The M centers are viewed as the service stations of a queueing network with R classes of part types (customers) and $K(r) \geq 1$ parts of class $r, 1 \leq r \leq R$. Each station m operates $J(m) \geq 1$ parallel identical servers, all fed by a common queue.

The service (manufacturing)time of a type r part at station m is taken as exponentially distributed with mean service time $S(r, m), 1 \leq r \leq R$, $1 \leq m \leq M$. The various processing sequences are accounted for through the routing probabilities between stations. The routing of part type r from station i to station j is represented by a Markovian routing matrix $P_r(i, j)$. From this matrix the steady-state probabilities $\pi(r, m)$ are calculated. The $\pi(r, m)$ values are non-negative and may be scaled arbitrarily [14].

The common input parameters are:

(1) the number of stations in the system (M),

(2) the number of part types produced (R),

(3) the number of type $r, 1 \leq r \leq R$, parts in the system ($K(r)$), *or* the total number of parts in the system K,

(4) the average processing time at each station for each part type ($S(r, m)$),

(5) the number of parallel identical servers at each station ($J(m)$),

(6) the routing probabilities or the mean number of visits $\pi(r, m)$ of part type r to station m,

(7) the number of robotic transporters $J(TRAN)$,

(8) the average time it takes the transporter to move a part type r from one station to the next $S(r, TRANS)$, and

(9) the service discipline $SD(m)$ at station $m, 1 \leqq m \leqq M$. This may be First Come First Served (FCFS), Head of Line (HOL) or Ample Server (AS) [14].

Ample server is a *fiction*, since it corresponds to a number of parallel servers at least equal to the number of pallets which can visit this station. It is useful for design evaluation purposes, since it describes a best case with no queueing delays. It is also useful for analyzing conveyor apparatuses.

Given these parameters, the performance measures that the system designer requires are (typically):

$G(r, m)$ = *throughput* of class r parts at station m,

and

$W(r, m)$ = mean time spent by a class r part *on queue* at station m.

From these basic variables, one can derive various other measures. For example:

$Q(r, m) = G(r, m)W(r, m)$

= mean number of class r parts on queue at station m,

$$Q(m) = \sum_{r=1}^{R} Q(r, m)$$

= mean number of parts on queue at sta tion m,

$$RO(m) = \sum_{r=1}^{R} G(r, m)S(r, m)/J(m)$$

= utilization of a typical server at station m,

$GS(r) = G(r, L/UL)$

= class r FMS throughput,

and

$TFT(r) = K(r)/GS(r)$

= mean manufacturing lead time for class r parts.

Due to considerations of space, this paper investigates only five models representing a broader set of models which could not be analyzed here. Fig. 1 delineates a brief taxonomy of these models. All these models were programmed in FORTRAN and tested on a CDC CYBER 170/855 Computer.

Next, we introduce the reader to the CAN-Q, MVAN, MVHEUR, PMVA and PSIM software models. Each of these software models is associated with some computational algorithm. For simplicity of exposition, most of the mathematical details regarding these algorithms are omitted from our presentation. Instead, the emphasis is placed on the relative modeling capabilities of these five models and on their implications for FMS studies.

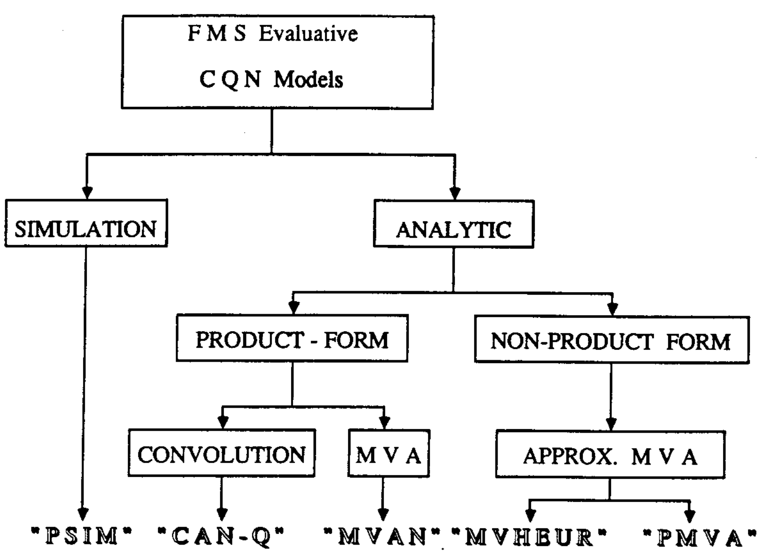

Fig. 1. Taxonomy of the CQN Models discussed.

2.1. The CAN-Q model

This model was developed by Solberg [24] who applied it successfully to capacity planning for industrial FMS. An extension by Stecke [26] enables CAN-Q to handle multiple part type systems. It is based on an exact model of a central server CQN satisfying the conditions for a product form solution [14]. (In product form networks, the steady state distribution of the jobs in the network can be written as a product of factors for each station, all divided by a normalizing constant.) The transport system is central in the sense that every part must pass through it before (and after) every operation. A fixed number of parts circulate throughout the system in accordance with prescribed routing probabilities. In CAN-Q, as well as in all other subsequent models, these probabilities are taken to be independent of previous operating decisions.

If the FMS produces multiple part types, then the model is given the output mix, i.e. the desired *fraction* of the total output of each type. Then, whenever a part leaves the L/UL station it is immediately replaced by a new part type. Each new part type is selected probabilistically according to the prescribed fraction for each part type. This implies that the *total* number of parts in the system is fixed but the instantaneous fraction of each part type is not constant, i.e. a *single* aggregate part type replaces a multiplicity of part types.

All stations can have multiple servers and employ FCFS service discipline. The principal performance measures computed are mean queue lengths and mean delays per station, utilizations of stations and transporters, and system throughputs. Sensitivity analysis depicts the impact of changing the total number of parts in the system (K) on its overall performance by presenting tables of the average flow times and of the production rates for a range of values of K from $K-5$ to $K+5$. Recall that the product form solution employed in CAN-Q holds only when the processing times at each FCFS station are *independent* of the part type. In this case, CAN-Q gives exact results. In all other cases it provides only approximate solutions as follows: CAN-Q handles several product types with *distinct* processing times at a FCFS station by replacing the exact processing time of each class by a weighted average processing time at this station (weighted according to the $\pi(r, m)$'s). This weighted average is taken over all visits to the station, whether they be the same part making repeated visits, or by different part types. CAN-Q does not compute the performance measures *per product type* at the stations (i.e. $Q(r, m)$ or $W(r, m)$).

The solution procedure is based on an extension of Buzen's [5] recursive algorithm. Buzen's algorithm can be applied also to a closed queueing network with topologies different from the central transport system, provided they have a product form solution. (Only the CAN-Q package is restricted to central server models, A simple way to overcome this difficulty is later presented in this paper.) The computational effort for single server stations is proportional to the product of the *number of stations* by the *number of parts* in the system.

2.2. The MVAN model

This model is based on the *exact* Mean Value Analysis (MVA) technique introduced by Reiser and Lavenberg [19]. It is used for analyzing generic product form CQN and it was not developed, originally, for FMS modeling. The users of this MVA model have to explicitly define the transporters and the load/unload (L/UL) stations. Modeling the transporters is not a trivial task since they cannot be represented as separate stations appearing in every other place on the part's routes. The appendix suggests a way to model the transporters with such generic CQN models. The numbers of parts per product type ($K(r)$) are given by the modeler and the MVA algorithm computes the resulting throughputs and the output mix.

The model is based on a set of 3MR equations – handling only single server FCFS, or AS stations. The model yields exact solutions for product-form queueing networks where the service demands at each FCFS station are *independent* of the customer's class (type) membership. In this paper we also use the *heuristic* extension to this model by Hildebrant [13] which can admit class-dependent service time requirements.

Using this extension, the mean service times ($S(r, m)$) are distinguished in the recursive model.

The model computes *detailed* performance measures per station such as the throughput and the mean waiting times *per product type* and the utilization of a typical server due to each part type. Thus, system performance is predicted on a part-by-part basis. Other measures are aggregated station, part type and system statistics.

The solution procedure involved is a recursive algorithm which starts from the trivial solution of the network with zero parts and gradually incrementing the population size, culminating in performance measures to construct the solutions for the population of interest. The computational effort is proportional to the *number of stations*, and to the product $K(1) \cdot K(2) \ldots K(R)$. Because this recursive solution technique requires excessive time and space for large networks, an approximate MVA algorithm is often suggested instead for practical use. Such a model is discussed next.

2.3. *The MVHEUR model*

This is an approximate MVA algorithm which provides fast and accurate solutions of very large networks. The original algorithm presented by Schweitzer [20] can handle single server FCFS or AS networks where the various part types may require distinct routing and distinct service demands at each station.

The variety of performance measures computed by the model is identical to that of the MVAN model. The solution algorithm is based on solving a set of 3MR equations by successive substitutions. The algorithm needs an initial guess for a starting point and it iterates until the convergence criteria are met. Proofs of convergence or uniqueness of solution are not available but these hold empirically. When properly implemented the computational effort is proportional to the sum $M + R$ of the *number of stations* and the *number of part types*, and is *independent* of the number of parts in the network. Several studies indicate that since this approximation technique is reasonably accurate, it may become useful as a general technique, even for networks that could be solved exactly [16]. More accurate extensions, at the expense of increased computational effort, are also available [7].

2.4. *The PMVA model*

PMVA, a model recently developed by Shalev-Oren, Seidmann and Schweitzer [23] extends the general principles of MVHEUR to model FMS with *parallel stations* having several identical machines; service discipline at each station can be either AS, FCFS or HOL priority scheduling scheme [10]. At HOL stations the same priority level may be assigned to several product types, and the relative priority level assigned to a given product type may change from station to station. The load/unload stations and the transportation system are explicitly modeled. Optionally, PMVA can handle an FMS with several transportation mechanisms operating with partitioned service responsibilities (i.e. each mechanism serves a distinct zone in the facility). The transportation times may be distinguished according to the actual distance among machine groups and the part type attributes (i.e. pallets' weight).

The variety of performance measures computed by PMVA is similar to that of the MVAN and the MVHEUR models. The solution algorithm used in PMVA solves by successive substitutions a set of 2MR simultaneous non-linear equations. An initial guess is provided by the model. The MVHEUR equations constitute a special case of the PMVA model for the single server FCFS or AS stations.

2.5. *The PSIM model*

This is a generic GPSS simulation model of FMS developed by Shalev-Oren, Seidmann and Schweitzer to support their research efforts in the field of FMS modeling [23]. The input file used is *identical* to that of PMVA. It was designed to allow easy experimentation with new layout configurations and to study the interplay of transportation times, part mix and other production control alternatives. It can model FMS similar to those handled by PMVA with a central transporter. The output production mix of the FMS is not predetermined and is computed by PSIM as a function of the number of parts per type used in the system ($K(r)$). PSIM may, alter-

Table 1
Comparative summary of attributes

	CAN-Q	MVAN	MVHEUR	PMVA	PSIM
Input data					
Part routing	Routing proportion: $\pi(r, m)$	Routing proportion: $\pi(r, m)$	Routing proportion: $\pi(r, m)$	Routing proportion: $\pi(r, m)$	Routing matrix: $Pr(i, j)$
Process times	Exponential[a]	Exponential[a]	Exponential[b]	Exponential[b]	Arbitrary[b]
Output mix	Pre-determined	Computed	Computed	Computed	Computed or predetermined
Number of part types:	One[c]	Many	Many	Many	Many
System					
Structure	Central server	Generic network	Generic network	FMS[d] network	Central server
Scheduling	FCFS[e]	FCFS, AS	FCFS, AS	FCFS, AS HOL	FCFS, HOL[e]
Parallel server stations	Yes	No	No	Yes	Yes
Transporter	Central	Not explicit	Not explicit	Central & partitioned	Central
Pallets	Identical type	Multiple types	Multiple types	Multiple types	Multiple types
Solution algorithm					
Approach	Finite recursion on population	Finite recursion on population	Iteration	Iteration	Digital simulation
Accuracy	Exact[f]	Exact[f]	Heuristic	Heuristic	Simulated
Sensitivity analysis	Yes	No	No	No	No
Computational effort depends on	M, R, K	$M, R, K(1) \cdot K(2) \ldots K(R)$	$M + R$	$M + R$	–
Principal performance measures					
$G(r, m)$, $W(r, m)$	No[g]	Yes	Yes	Yes	Yes
$Q(m)$, $RO(m)$	Yes	Yes	Yes	Yes	Yes
$GS(r)$	Yes	Yes[h]	Yes[h]	Yes[h]	Yes[h]
Probability distribution of $Q(m)$	Yes	No	No	No	Yes

[a] Part type – independent process times at each FCFS station are required to guarantee exact results, otherwise results are approximate.
[b] The mean processing times at the stations can be part type-dependent.
[c] Single aggregate part type replaces multiplicity of actual part types.
[d] Both generic CQN and FMS with transporters can be modeled.
[e] As can be modeled by assigning K parallel servers to a station.
[f] Type-independent process times at each FCFS station are required to guarantee exact results.
[g] In principle, it is possible to compute these measures but the results are only approximate since inaccurate process time averages are used.
[h] A special Load/Unload station, defined as the last station visited on a part route, is used to measure the throughputs.

natively, model a system similar to CAN-Q where the output mix is predetermined.

In PSIM the user can define the desired distribution of manufacturing and processing times. It computes the same performance measures as PMVA and, in addition, it provides the transient response and the steady-state distributions of selected output variables. Changes in the physical layout such as addition/deletion of stations and products are facilitated by changes in the numerical input fields of PSIM. No rewriting of the GPSS statements is required since it is intended for non programmer users.

Table 1 presents a comparative summary of attributes for these programs.

3. Numerical experimentation and predictive capabilities

The relative predictive capabilities of the different models is based upon examining their results for the same design problem. There are, as discussed earlier, several inherent differences in the expressivity and applicability of the various models. The common denominator for all models is an FMS with *single* server FCFS or AS stations and with *identical* processing times for all product types at each station. Unfortunately, this is a trivial case, not appropriate for real industrial applications. This paper analyzes, therefore, a more involved system and investigates few alternatives for modeling it despite existing limitations in some of the models.

The system examined here produces three product types ($R = 3$) labeled: $r = 1$, 2 and 3. Assume that there are three machining centers ($M = 3$): *three* machines of type A in the first center; *two* machines of type B in the second; and *one* type C machine in the third. *One* transporter (TRANS) is used for material handling and there is *one* load/unload (L/UL) station (Fig. 2). Ten percent of the produced items are routed randomly to the inspection station (INS), before reaching the L/UL station. The mean transport time is assumed to be 2 time units. The system operates with 6, 6 and 9 pallets for product types 1, 2 and 3, respectively. Table 2 presents the process requirements (i.e. the routes and service times) for the three product types. Denote the FMS configuration detailed above as *Configuration I.*

Using Configuration I, the CAN-Q model does not lead to exact results. Thus, define *Configuration Ia* which is identical to I except that in Ia the processing times are *independent* of the part types. Table 3 presents the processing requirements for Ia.

3.1. Modeling parallel server stations

The MVAN and MVHEUR cannot analyze either I or Ia because some of the stations have several servers. (An extension of MVAN for parallel servers has been given by Reiser [18], but its code is not publically available.) At present, many efficient CQN models described in the literature (e.g. [6–9]) can handle only single server stations and state-dependent approximations can require significant computational efforts for large multiclass networks (e.g. [32]). Therefore, three alternative approaches to

Table 2
Process requirements: configuration I

Station (m)	Part type (r) $S(1, m)$	$\pi(1, m)$	$S(2, m)$	$\pi(2, m)$	$S(3, m)$	$\pi(3, m)$
A	20	1.0	30	1.0	25	1.0
B	15	1.0	15	1.0	18	1.0
C	–	0.0	–	0.0	25	1.0
INS	35	0.1	45	0.1	35	0.1
L/UL	7	1.0	8	1.0	7	1.0
TRANS	2	3.1	2	3.1	2	4.1

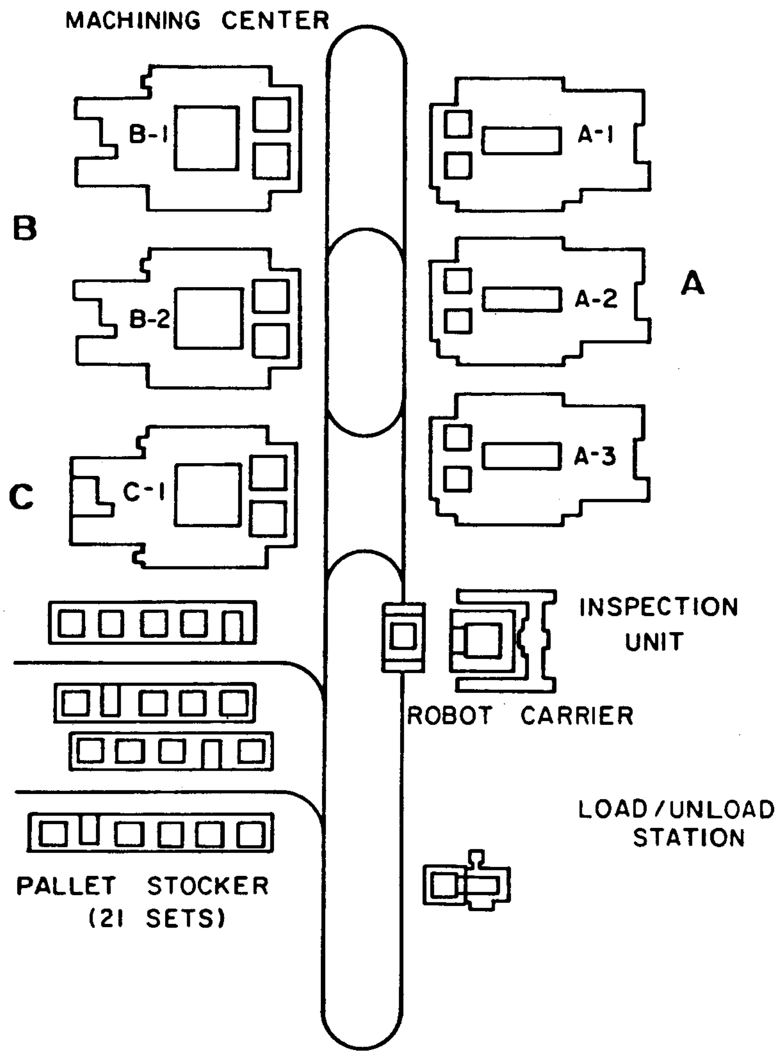

Fig. 2. The FMS facility.

Table 3
Process requirements: Configuration Ia

	Part type (r)					
Station (m)	$S(1, m)$	$\pi(1, m)$	$S(2, m)$	$\pi(2, m)$	$S(3, m)$	$\pi(3, m)$
A	25	1.0	25	1.0	25	1.0
B	16	1.0	16	1.0	16	1.0
C	–	0.0	–	0.0	25	1.0
INS	7.3	0.1	7.3	0.1	7.3	0.1
L/UL	38	1.0	38	1.0	38	1.0
TRANS	2	3.1	2	3.1	2	4.1

analyzing parallel server stations with single server models were investigated instead. *Configuration II* is identical to I except for the parallel server stations representation. This configuration has only *single server* stations; each stations m' in I having $J(m')>1$ is replaced in II by $J(m')$ parallel single server stations denoted as

$$m'_1, m'_2, \ldots, m'_{J(m')} .$$

Each of these $J(m')$ servers gets a fixed fraction $1/J(m')$ of the arriving load. These $J(m')$ stations operate with *distinct queues* satisfying

$$\pi(r, m'_i) = \pi(r, m')/J(m')$$

and

$$S(r, m'_i) = S(r, m') , \quad i = 1, 2, \ldots , J(m') .$$

Additionally, we investigated *Configuration IIa* which was derived from Ia in a similar fashion that Ia was derived from I. This specific configuration can be solved by all five models. It is expected that Configuration II will lead to pessimistic sojourn times estimates since some of the parallel servers may be idle while others have queues (no jockeying). Since the FMS is modeled as a closed network, these pessimistic estimations should lead to downward bias in the computed throughputs.

Next, in *Configuration III* each station in I having $J(m')$ parallel machines is replaced by a faster machine operating at $J(m')$ times the rate.

Finally, *Configuration IV* is also similar to I except for the parallel server stations representation; each station m' in I having $J(m')>1$ is replaced in IV by two stations in a tandem arrangement (see Fig. 3). The first station (m'_1) is a single server FCFS and the second (m'_2) operates as AS. The parameters of IV are:

$$\pi(r, m'_1) = \pi(r, m'_2) = \pi(r, m') ,$$

$$S(r, m'_1) = S(r, m')/J(m') ,$$

$$S(r, m'_2) = S(r, m')(J(m') - 1)/J(m') ,$$

$$SD(m'_1) = FCFS, SD(m'_2) = AS .$$

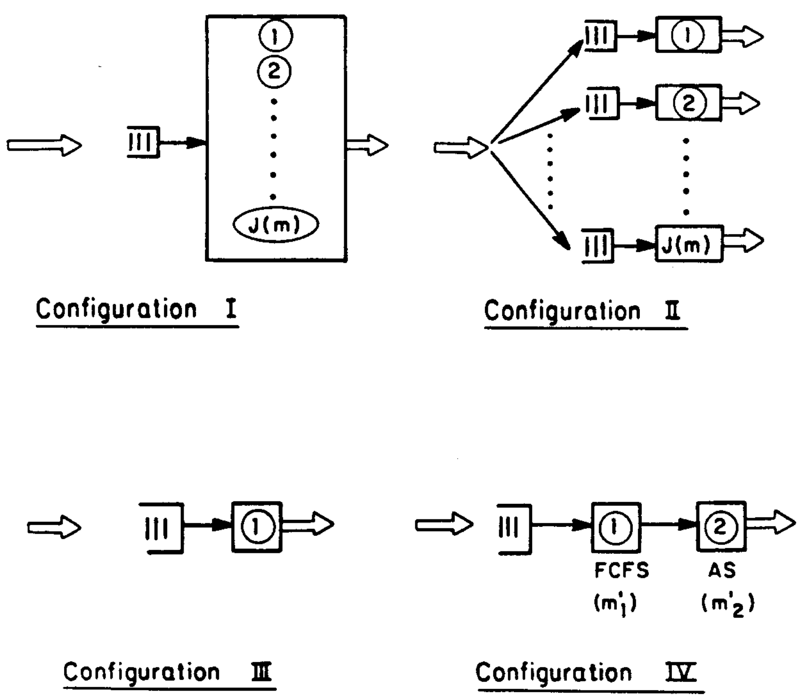

Fig. 3. Parallel stations representation in Configurations I, II, III and IV.

The structure of IV also attempts to overcome the difficulties of the single server models. It is based on the heuristic assumption that under medium to heavy utilization, $J(m')$ servers will clear a queue $J(m')$ times as fast as a single server. Hence, the expected waiting time on queue for (m_1') is close to that for station (m') with $J(m')$ parallel servers. The second station in IV, (m_2'), is added to rectify the mean-time-in-station estimates by assuring that the total service remains the same. For further discussion of this approach see [22]. Our conjecture was that configuration IV will result in more accurate predictions than configurations II and III.

3.2. Non-central transporters

Following this explanation, there is another subtle point that should be treated. Using m_1' and m_2' in IV, rather than m' in I, means that the transporter has to cater for an *additional* leg within this pair. Of course, the transfer time for this dummy leg is zero. The CAN-Q model, on the other hand, assumes central transporter with *equal* transfer times. Using CAN-Q in IV we cancelled the central transporter (by inserting zero transfer times as inputs to the program), and we augmented the model with another "station". The mean service times at the new "station" is set equal to the mean transfer times in I and its routing proportion is set to the $\pi(r, \text{TRANS})$ of the central transporter in I.

Next, we focus on two key issues in performance evaluation of FMS with computer models of CQN. The first issue is *the degree of correspondence between the predictions of the various models for the same configuration*. Here Configurations I and Ia were analyzed with CAN-Q, PMVA and PSIM (Tables 4 and 5) and Configurations II and IIa were analyzed with all the five models (Tables 6 and 7). The second issue investigated here is *the value of modeling an FMS, operating with several parallel server stations, using single server models* (i.e. by Configurations II, III and IV). This issue was analyzed by comparing the results of CAN-Q and PMVA over Configurations I, II, III and IV (Table 8). The numerical results and their interpretations are presented next.

4. Assessment of results

Tables 4–8 display a selected summary of various performance measures computed by the five models. These measures are defined in Section 2. In Tables 4–6 the performance measures are compared with the PSIM results since there is no publicly available code for computing the exact results for these configurations. The performance

Table 4
Configuration I (PMVA, CAN-Q vs. PSIM)

	$RO(m)$						$W(r, A)$		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	0.731	0.785	0.410	0.833	0.853	0.888	37.79	36.38	37.31
PSIM	0.720	0.790	0.405	0.870	0.864	0.899	37.10	34.70	35.44
CAN-Q	0.728	0.782	0.409	0.827	0.850	0.884			
Δ(PMVA)%	+2	−1	+1	−4	−1	−1	+2	+5	+5
Δ(CAN-Q)%	+1	−1	+1	−5	−2	−2			
	$Q(m)$						$GS(r) \times 10^4$		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	1.67	2.31	0.27	2.38	3.27	3.98	333	356	381
PSIM	1.66	2.46	0.34	2.07	3.40	3.87	342	358	382
CAN-Q	1.71	2.32	0.27	3.01	3.06	3.53	331	352	384
Δ(PMVA)%	+1	−6	−21	+14	−3.8	+2.8	−3	−1	-
Δ(CAN-Q)%	+3	−6	−21	+45	−10	−9	−3	−2	−1

Table 5
Configuration Ia (PMVA, CAN-Q vs. PSIM)

	RO(m)						*W(r, A)*		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	0.732	0.751	0.408	0.836	0.858	0.894	40.2	39.2	39.2
PSIM	0.732	0.763	0.437	0.880	0.867	0.908	39.2	36.7	36.9
CAN-Q[a]	0.729	0.749	0.406	0.829	0.855	0.891			
Δ(PMVA)%	–	−2	−7	−5	−1	−2	+3	+7	+6
Δ(CAN-Q)%	–	−2	−7	−6	−1	−2			
	Q(m)						$GS(r) \times 10^4$		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	1.69	1.87	0.26	2.42	3.39	4.24	335	369	369
PSIM	1.75	2.02	0.36	2.14	3.35	4.12	349	372	372
CAN-Q[a]	1.73	1.93	0.27	3.06	3.19	3.73	332	369	369
Δ(PMVA)%	−3	−7	−28	+13	+1	+3	−4	−1	−1
Δ(CAN-Q)%	−1	−4	−25	+43	−5	−9	−5	−1	−1

[a] Denotes *exact* analytic solution.

Table 6
Configuration II (PMVA, CAN-Q, MVAN, MVHEUR vs. PSIM)

	RO(m)						*W(r, A)*		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	0.642	0.684	0.357	0.787	0.748	0.775	72.23	70.03	71.82
PSIM	0.622	0.671	0.354	0.788	0.753	0.775	80.64	72.05	75.67
CAN-Q	0.639	0.680	0.355	0.790	0.744	0.770	–	–	–
MVAN[a]	0.648	0.689	0.359	0.806	0.754	0.781	72.89	67.75	71.76
MVHEUR	0.642	0.684	0.357	0.787	0.748	0.775	72.31	70.11	71.90
Δ(PMVA)%	+3	+2	+1	–	−1	–	−10	−3	−5
Δ(CAN-Q)%	+3	+1	–	–	−1	−1			
Δ(MVAN)%	+4	+3	+1	+2	–	+1	−10	−6	−5
Δ(MVHEUR)%	+3	+2	+1	–	−1	–	−10	−3	−5
	Q(m)						$GS(r) \times 10^4$		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	1.01	1.28	0.19	1.83	1.87	2.22	315	302	318
PSIM	1.00	1.25	0.19	1.41	1.93	2.34	321	298	307
CAN-Q	1.03	1.28	0.19	2.33	1.81	2.09	316	297	316
MVAN[a]	1.07	1.33	0.20	1.55	1.92	2.22	322	301	317
MVHEUR	1.01	1.28	0.19	1.83	1.87	2.22	315	302	318
Δ(PMVA)%	+1	+2	–	+29	−3	−5	−2	+1	+3
Δ(CAN-Q)%	+3	+2	–	+65	−6	−11	−2	–	+3
Δ(MVAN)%	+7	+6	+5	+10	–	−5	–	+1	+3
Δ(MVHEUR)%	+1	+2	–	+29	−3	−5	−2	+1	+3

[a] Denotes *heuristic* recursive solution for distinct processing times.

Table 7
Configuration IIa (PMVA, PSIM, CAN-Q, MVHEUR vs. MVAN)

	$RO(m)$						$W(r, A)$		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	0.645	0.657	0.357	0.791	0.751	0.782	73.1	71.8	71.8
PSIM	0.645	0.665	0.376	0.795	0.756	0.797	73.5	69.6	68.8
CAN-Q	0.640	0.652	0.354	0.791	0.745	0.775	–	–	–
MVHEUR	0.645	0.657	0.357	0.791	0.751	0.782	73.1	71.8	71.8
MVAN[a]	0.651	0.662	0.359	0.811	0.756	0.787	73.8	70.9	70.9
Δ(PMVA)%	−1	−1	−1	−2	−1	−1	−1	+1	+1
Δ(PSIM)%	−1	–	+5	−2	–	+1	–	−2	−3
Δ(CAN-Q)%	−2	−2	−1	−2	−1	−2	–	–	–
Δ(MVHEUR)%	−1	−1	−1	−2	−1	−1	−1	+1	+1
	$Q(m)$						$GS(r) \times 10^4$		
	TRANS	L/UL	INS	C	B	A	3	2	1
PMVA	1.03	1.09	0.183	1.87	3.75	6.78	316	311	311
PSIM	1.10	1.21	0.190	1.54	3.84	6.73	327	312	313
CAN-Q	1.03	1.10	0.188	2.35	3.62	6.44	317	307	307
MVHEUR	1.03	1.09	0.183	1.87	3.74	6.78	316	311	311
MVAN[a]	1.09	1.15	0.196	1.59	3.83	6.79	324	310	310
Δ(PMVA)%	−6	−5	−7	+17	−2	–	−2	–	–
Δ(PSIM)%	+1	+5	−3	−3	–	−1	+1	+1	+1
Δ(CAN-Q)%	−6	−4	−4	+47	−5	−5	−2	−1	−1
Δ(MVHEUR)%	−6	−5	−7	+17	−2	–	−2	–	–

[a] Denotes an *exact* analytic solution.

measures in Table 7 are compared against the *exact* results of the MVAN model. These tables present empty entries for the $W(r, A)$ for CAN-Q since CAN-Q does not provide waiting times statistics per product type at each station.

The CAN-Q model requires predetermination of the desired output mix. In the experiments reported here, we first ran the PMVA models and later used their *computed* production mixes as inputs to CAN-Q, rounding them to the closest integer percentage. The reader is reminded that CAN-Q is qualitatively different from the four other models; the PMVA, MNHEUR, PSIM and MVAN are based on multiclass CQN models, whereas CAN-Q is based on a single class. Its results are presented in order to examine the potential deviations of key performance measures (i.e. $RO(m)$ and $Q(m)$) as a result of using these two distinct modeling perspectives.

Observing Tables 4–7 indicates that, for the same configuration, the differences in the part type throughput predictions ($GS(r)$) and the machines' and transporters' utilizations ($RO(m)$) are less than 7% in all cases; the differences in the queue length and mean queueing times estimates in most cases (except for stations C and INS) are less than 10%. This is a well-known phenomenon that using closed queueing network models the throughputs and utilizations are typically predicted with greater accuracy than queue lengths and sojourn times [16]. As can be seen, the differences in predicting $Q(c)$ and $Q(\text{INS})$ are as large as 30–45%. We conjecture that the reason for these deviations stems from the fact that only a small fraction of the parts population traveled through stations INS and C. This resulted in significantly larger estimation variability and, hence, wider simulated sampling confidence intervals for these stations parameters.

Table 7 shows the average deviations of the four models' results from the MVAN predictions. These are fairly close except for the $Q(c)$ estimates; these deviations may result from the fact that since only one part type ($r = 3$) visits station C the estimation errors for $Q(c)$ are not 'aver-

Table 8
Comparative analysis (Configurations II, III and IV vs. I)

			$RO(m)$						$W(r, A)$		
Configuration			TRANS	L/UL	INS	C	B	A	3	2	1
	I		0.731	0.785	0.410	0.833	0.853	0.888	37.79	36.38	37.3
PMVA	II		0.642	0.684	0.357	0.787	0.748	0.775	70.03	70.03	71.8
	III		0.740	0.796	0.416	0.838	0.864	0.902	51.15	49.30	50.5
	IV		0.719	0.772	0.404	0.828	0.840	0.874	40.46	40.46	41.4
		Δ(II–I)/I%	−12	−13	−13	−6	−12	−13	+85	+92	+92
		Δ(III–I)/I%	+1	+1	+1	+1	+1	+2	+35	+36	+35
		Δ(IV–I)/I%	−2	−2	−1	−1	−2	−2	+7	+11	+11
Configuration											
	I		0.728	0.782	0.409	0.827	0.850	0.884			
CAN-Q	II		0.639	0.680	0.355	0.790	0.744	0.770			
	III		0.737	0.792	0.414	0.837	0.860	0.894			
	IV		0.720	0.774	0.404	0.818	0.841	0.874			
		Δ(II–I)/I%	−12	−13	−13	−4	−12	−13			
		Δ(III–I)/I%	+1	+1	+1	+1	+1	+1			
		Δ(IV–I)/I%	−1	−1	−1	−1	−1	−1			
			$Q(m)$						$GS(r)$		
Configuration			TRANS	L/UL	INS	C	B	A	3	2	1
	I		1.67	2.31	0.27	2.38	3.27	3.98	333	356	381
PMVA	II		1.01	1.28	0.19	1.83	1.87	2.22	315	302	318
	III		1.76	2.48	0.28	2.45	4.01	5.45	335	370	379
	IV		1.56	2.13	0.25	2.31	3.37	4.34	331	349	373
		Δ(II–I)/I%	−40	−45	−30	−23	−43	−44	−5	−15	−17
		Δ(III–I)/I%	+5	+7	+4	+3	+22	+37	+1	+4	−1
		Δ(IV–I)/I%	−7	−8	−7	−3	+3	+9	−1	−2	−2
Configuration											
	I		1.71	2.32	0.27	3.01	3.06	3.53	331	352	384
CAN-Q	II		1.03	1.28	0.19	2.33	1.81	2.09	316	297	316
	III		1.83	2.51	0.28	3.29	3.80	4.73	333	355	388
	IV		1.62	2.18	0.27	2.81	3.20	3.91	327	348	380
		Δ(II–I)/I%	−40	−45	−30	−23	−41	−41	−5	−16	−18
		Δ(III–I)/I%	+7	+8	+4	+9	+24	+34	+1	+1	+1
		Δ(IV–I)/I%	−5	−6	–	−7	+5	+11	−1	−1	−1

aged' over several product types; on the other hand, the transporters, which handled all product types, present small deviations in most cases.

It is interesting to note that there are only minor deviations between most of the MVHEUR and the MVAN approximations for Configurations II and IIa (Tables 6 and 7). The MVAN approximations for Configuration II (Table 6) are produced by the *heuristic* method presented by Hildebrant [13]. Moreover, the results in Table 6 seem to indicate the superiority of the MVHEUR (and PMVA) over the MVAN heuristic recursion. These findings suggest that when a product form does not hold, MVHEUR is preferred to the heuristic version of MVAN since both maintain the same accuracy level while the former is parsimonious in terms of its time and memory requirements. This observation is consistent with earlier observation by Suri and Hildebrant [31]. Based on the Table 7 results, the predictions of PMVA and MVHEUR are – for most measures – within a few percent of the MVAN exact results. The prediction error of these models is caused by using Schweitzer's approximated correction term to accelerate the solution times [21].

Configurations II, III and IV are proposed as a substitute for Configuration I. Table 8 presents the relative deviations of several measures in Configurations II, III and IV with respect to I. As anticipated, using Configuration II results in *pessimistic* estimates for the mean machine utilizations, mean waiting time and the throughputs; for example, the bias is −6% to −13% for the $RO(m)$'s and −5% to −17% for the $GS(r)$'s. In addition, Configuration II leads to *optimistic* queue length estimates since common queues in front of the parallel server stations are replaced by several parallel queues; the downward bias for $Q(m)$ is between −23% and −45%.

Configuration III replaces the $J(m)$ parallel machines at station m by a single machine which is $J(m)$ times as fast; thus, parts visiting the parallel machines stations spend less time in actual processing and more time waiting on queues. Indeed, the research results indicate that there is a significant increase in the $Q(m)$'s and in the $W(r, A)$'s estimates which lead to slight increases in the computed machine utilizations and FMS throughputs estimates. Also, note that in this configuration the largest deviations in the queue length estimates occur at the two parallel machine stations A and B.

Observing Configuration IV clearly indicates better performance than II or III – resulting from smaller deviations with respect to the values of I. From Table 8 it is evident that the deviations of the throughputs and machine utilization estimates are rather small ranging between −1% and −2%. Considering the mean queue length and the mean waiting time estimates, which are directly related, we note deviations of 11% or less.

Although a complete analysis of the issues involved in approximate modeling of parallel server stations with single server models was not presented here, these results indicate that an approach similar to the one realized by Configuration IV should be preferred to the 'crude' approach proposed by II and III. Better yet, when feasible, is to employ a model initially designed to treat parallel server stations [31, 32].

The computational effort

The computational efforts involved in using these five models is a function of two major factors. One is the *underlying mathematical models* and the other is the efficiency of the *software tools* developed to implement them. In the cases of CAN-Q and MVAN, the solution procedure is recursive and *finite*. The user cannot affect the solution effort. By contrast, the solution procedure of MVHEUR and PMVA is iterative and infinite. The solution effort depends on the *initial guess* and on the *convergence criteria*. Since both models use successive substitutions the convergence criteria significantly impacts the required number of iterations.

Experimenting with the models described here did not reveal any significant differences in the core memory and computer time requirements between models in which the computational effort depends on the population size (CAN-Q, MVAN), and those in which it does not (MVHEUR, PMVA). This observed similarity in computer times may be partly attributed to the smallness of the test populations. It may also be partly attributed to variances in programming styles and memory allocation procedures.

As expected, the PSIM program stands out as using relatively long computer times. These are determined by the desired confidence level for the key variables. In this study the PSIM simulations in all cases were terminated by using a sequential stopping rule that employed Batch Means Analysis to estimate the confidence interval for the estimates of the FMS throughputs. Once the half-width of the confidence interval was less than 5% of the estimated means, the simulation was stopped [15]. It should be noted, however, that such a stopping rule may lead to overly optimistic estimates of the properties of the confidence interval. A PSIM average run length is several CPU minutes as compared with a few seconds in all other analytic models. This relative difference of computational effectiveness provides a significant advantage for analytic models in interactive design sessions.

5. Summary and conclusions

The purpose of this paper is to investigate the relative modeling capabilities, the accuracy of estimates and the solution procedures of five well-known computer network models used for complex FMS design evaluations. Two of the models (MVAN, MVHEUR) can handle only single server FCFS or AS stations. Thus, special attention is given here to the problem of overcoming this limitation when necessary. Three alternate solutions to that problem are proposed and tested empirically.

Explicit representation of the transporters exists only in CAN-Q, PMVA and PSIM. We propose a way to model the transporters in generic CQN models. Priority scheduling is also explicit in PSIM and PMVA and impossible in all of the other models. Only CAN-Q can model a system with a *predetermined output mix*, but is incapable of modeling a system in which a *fixed number of parts* per product type journey through it.

Numerical experimentations reported here indicate that the accuracy level of the analytic models (CAN-Q, MVAN, MVHEUR and PMVA) is rather close in most cases. Deviations in the throughput and utilization estimates were significantly smaller than in the queue length and waiting time estimates. Of course, the values used in our examples were chosen arbitrarily, simply as illustrative case, so the tables we present cannot be taken as generally applicable. The method of research, however, and the main lessons discussed about the issues involved in using CQN models for FMS along with the attributes and the relative capabilities of the models considered, have general implications for manufacturing systems designers.

Acknowledgements

The authors wish to thank Professor J.J. Solberg, from Purdue University, for providing the CAN-Q program and Mr. K. Kindle, from the University of Rochester, for the MVAN and MVHEUR computer programs.

Appendix: Modeling the transporters with a generic CQN model

A.1. A central transporter

A system with a central transporter satisfies:

$$G(r, \text{TRAN}) = \sum_{m=1}^{M} G(r, m)\,, \tag{A.1}$$

since each visit to a station requires a transportation service. The mean transport time for a type r part is denoted by $S(r, \text{TRAN})$. From Little's law we get:

$$G(r, m) = \frac{\pi(r, m)K(r)}{\left(\begin{array}{l}\sum_{j=1}^{M} \pi(r, j)[W(r, j) + S(r, j) \\ + W(r, \text{TRAN}) + S(r, \text{TRAN})]\end{array}\right)}, \quad m = 1, 2, \ldots, M,\ r = 1, 2, \ldots, R\,. \tag{A.2}$$

Substituting (A.2) into (A.1) leads to:

$$\begin{aligned} G(r, \text{TRAN}) &= \sum_{m=1}^{M} \frac{\pi(r, m)K(r)}{\left(\begin{array}{l}\sum_{j=1}^{M} \pi(r, j)[W(r, j) + S(r, j) \\ + W(r, \text{TRAN}) + S(r, \text{TRAN})]\end{array}\right)} \\ &= \frac{K(r) \sum_{m=1}^{M} \pi(r, m)}{\left(\begin{array}{l}\sum_{j=1}^{M} \pi(r, j)[W(r, j) + S(r, j)] \\ + [W(r, \text{TRAN}) + S(r, \text{TRAN})] \sum_{j=1}^{M} \pi(r, j)\end{array}\right)}. \end{aligned} \tag{A.3}$$

Defining the sum of the $\pi(r, m)$'s as

$$\pi(r, \text{TRAN}) = \sum_{j=1}^{M} \pi(r, j)\,. \tag{A.4}$$

leads to the reformulation of Little's law for the network by adding to it the TRANS station as the $(M+1)$-st station:

$$G(r, m) = \frac{K(r)\pi(r, m)}{\sum_{j=1}^{M+1} \pi(r, j)[W(r, j) + S(r, j)]}\,, \quad m = 1, 2, \ldots M, M+1\,. \tag{A.5}$$

Hence, in the case of the central transporter, we end up with defining the transporter as an additional station whose visit ratio, $\pi(r, \mathrm{TRAN})$, is defined by (A.4).

A.2. Several transporters

Suppose that the FMS uses N transportation systems which do not interfere with the motions of each other. Each system $n (n = 1, 2, \ldots, N)$ is responsible for a set of $\{A_n\}$ of the transportation legs in the system. In this case we get for transporter n:

$$\pi(r, M+n) = \sum_{m=1}^{M} \pi(r, m) \sum_{j=1}^{M} P_r(m, j), \quad n = 1, 2, \ldots, N, \qquad \text{(A.6)}$$

$\{m \to j\} \in A$, i.e. the transfer from m to j belongs to set A_n, and P_r is the $(M \times M)$ routing matrix of type r between the stations in the system. The reader can easily verify that (A.4) is a special case of (A.6) for $N = 1$.

References

[1] M.M. Barash, Computerized manufacturing systems for discrete products, in: G. Salvendy (Ed.), The Handbook of Industrial Engineering (Wiley, New York, 1981) ch. VII-9.

[2] R.A. Berdine, Caterpillar's DNC system two years later, Proc. 15th Numerical Control Society Annual Meeting, Chicago (1978).

[3] R.M. Bryant, A.E. Krzesinski and P. Teunissen, The MVA preempt resume priority approximations, Proc. ACM SIGMETRICS Conference on Measurement and Modelling of Computer Systems (1983).

[4] J.A. Buzacott and D.D. Yao, Flexible manufacturing systems: Review of Models, Management Sci. 32 (7) (1986) 890–905.

[5] J.P. Buzen, Computational algorithms for closed queueing networks with exponential servers, C. ACM 16 (9) (1973) 527–537.

[6] J.B. Cavaille and D. Dubois, Heuristic methods based on mean value analysis for flexible manufacturing systems performance evaluation Proc. IEEE Conference on Decision and Control, Orlando, Florida (1982).

[7] K.M. Chandy and D. Neuse, Linearizer: A heuristic algorithm for queueing network models of computer systems, Comm. ACM 25 (2) (1982) 126–134.

[8] K.M. Chandy, and M.S. Lakshmi, An approximation technique for queueing networks with preemptive priority queues, Working Paper, Department of Computer Sciences, The University of Texas at Austin, 1984.

[9] W.M. Chow, Approximations for large scale closed queueing networks, Performance Evaluation, 3 (1) (1983) 1–11.

[10] A. Cobham, Priority assignment in waiting line problems, Operations Research, 2 (2) (1954) 70–76.

[11] W. Eversheim and P. Hermann, Recent trends in flexible automated manufacturing, Journal of Manufacturing Systems, 1 (2) (1982) 139–148.

[12] S.B. Gershwin, R.R. Hildebrant, R. Suri and S.K. Mitter, A control theorist's perspective on recent trends in manufacturing systems:, Proc. IEEE Conference on Decision and Control, Las Vegas, Nevada, December, 1984.

[13] R.R. Hildebrant, Scheduling flexible machining systems using mean value analysis, Proc. IEEE Conference on Decision and Control, Albuquerque, New Mexico, 1980.

[14] L. Kleinrock, Queueing Systems, Volume II: Computer Applications, Wiley, New York, 1976.

[15] A. Law, and W.D. Kelton, Simulation Modeling and Analysis, McGraw Hill, New York, 1982.

[16] E.D. Lazowska, J. Zahorjan, G.S. Graham, and K.C. Sevcik, Quantitative System Performance, Prentice Hall, Englewood Cliffs, New Jersey, 1984.

[17] K. Rathmill, N. Greenwood and M. Houshmand, Computer simulation of FMS, Proc. of the 2nd International Conference on FMS, London, U.K., (1983) 251–280.

[18] M. Reiser, Mean value analysis and convolution method for queue-dependent servers in closed queueing networks, Performance Evaluation, 1 (1) (1981) 7–18.

[19] M. Reiser and S.S. Lavenberg, Mean value analysis of closed multichain queueing networks, J. ACM 2, 4 (1980) 313–323.

[20] P. Schweitzer, Approximate analysis of multiclass closed networks of queues, International Conference on Stochastic Control and Optimization, Free University, Amsterdam, 1979.

[21] P.J. Schweitzer, A. Seidmann and S. Shalev-Oren, The correction terms in approximate mean value analysis, Operations Research Letter, 4 (5) (1986) 197–200.

[22] A. Seidmann, S. Shalev-Oren and P.J. Schweitzer, Mean value analysis of interactive computer systems with HOL priority and parallel processors, Working Paper QM87-17, Simon School, The University of Rochester.

[23] S. Shalev-Oren, A. Seidmann and P.J. Schweitzer, Analysis of flexible manufacturing systems with priority scheduling: PMVA, Annals of Operations Research, 6 (1985) 115–139.

[24] J.J. Solberg, A mathematical model of computerized manufacturing systems, Proc. of the 4th International Conference on Production Research, Tokyo, Japan, August 1977.

[25] J.J. Solberg and S.Y. Nof, Analysis of flow control in alternative manufacturing configurations, Journal of Dynamic Systems, Measurements and Control, 102 (9) (1980) 141–147.

[26] K.E. Stecke, Production Planning Problems for Flexible Manufacturing Systems, Ph.D. Dissertation, School of Industrial Engineering, Purdue University, W. Lafayette, Indiana, 1981.

[27] K.E. Stecke, Design, planning, scheduling, and control problems of flexible manufacturing systems, Proc. First ORSA/TIMS Special Interest Conference on FMS, Ann Arbor, 1984.

[28] R. Suri, New techniques for modelling and control of flexible automated manufacturing systems, IFAC Conference, Kyoto, Japan, 1981.

[29] R. Suri, Robustness of queueing network formulae, J. ACM, 30 (3) (1983) 564–594.

[30] R. Suri, A low-memory approximation for multiclass queueing networks with multiple server stations, Technical Report, Harvard University, Cambridge, Mass., 1983.

[31] R. Suri and R.R. Hildebrant, Modeling flexible manufacturing systems using mean-value analysis, J. of Manufacturing Systems. 1 (3) (1984).

[32] J. Zahorjan and E.D. Lazowska, Incorporating load dependent servers in approximate mean value analysis, ACM Performance Evaluation Review, 12 (3) (1984) 52–62.