Performance Modeling and Dynamic Scheduling of Make-to-order Supply Chains

N. R. Srinivasa Raghavan* N. Viswanadham†

Abstract

In this paper, we originate innovative models of the integrated supply chain network (SCN) and propose certain dynamic scheduling policies applicable to the same. Specifically, we use fluctuation smoothing policies which are special cases of least slack policies. We treat the SCN as encompassing activities at various facilities of the enterprise, each of which could be modeled as a single/multi server queue. Thus the entire SCN could be modeled as a fork-join queueing network. Using this model, and by treating the orders for various products as jobs waiting in separate buffers in front of the facilities, we schedule the orders on to the facilities to minimize the mean and variance of cycle times and lateness of these orders. We demonstrate the effectiveness of these scheduling policies by conducting simulation experiments.

1 Introduction

In recent years, companies have begun understanding the importance of integrating their supply chains – which links every disparate function from raw material purchasing through customer order delivery to fulfill customer needs – as a primary means of improving customer satisfaction. Effective management of the supply chain is essential for meeting customers' needs, retaining their loyalty, and for

*Department of Computer Science and Automation, Indian Institute of Science, Bangalore – 560 012, India. email: raghavan@csa.iisc.ernet.in

†Department of Mechanical & Production Engineering, National University of Singapore, Singapore – 119260. email: mpenv@nus.edu.sg

profitability of all the stakeholders of the supply chain. Conversely, a less-than-optimally functioning supply chain can undermine customer satisfaction and loyalty, thus cutting off avenues for profitable growth [1, 4, 8].

We consider the supply chain network (SCN) as an interconnection of facilities such as suppliers, OEMs, distribution, transportation etc. Each of these facilities itself could be a large dynamical system consisting of several subprocesses and facilities (machines, transporters, etc). But for the purpose of scheduling the SCN as a whole, we consider each of the facilities as single server queueing models of G/G/1 type manufacturing/transporting multiple products; (see [9] for an instance of treating a facility as a M/G/1 queue;) thus the entire SCN can be treated as a fork-join queueing network (FJQN). We are interested in minimizing the mean and variance of the end to end cycle time, i.e. from procurement to delivery.

We deal with a very important subject i.e. supply chain coordination through scheduling the orders on the supply chain network to minimize the network lead time. This assumes that mechanisms are in place for communicating information among the supply chain constituents [2]. Also we assume the presence of a process owner or an integrator for the supply chain who collects the global information and does the scheduling. This would also provide a basis for planning the resources at the manufacturing enterprise level. We would like to remark that very little literature exists for solving the scheduling problem for the entire SCN in a process oriented perspective.

The main focus of our study will be on optimal ways of scheduling customer orders in the supply chain, with special reference to Make-to-Order (MTO) SCNs. In Section 2, we discuss the issues involved in scheduling in SCNs. Section 3 presents congestion models for MTO SCNs using fork-join queueing networks. In Section 4, we provide a class of policies known as Fluctuation Smoothing (FS) policies, used first in the context of reentrant lines by Lu et al [6]. In Section 5, we briefly dwell upon the simulation results and their relevance in the real world and in Section 6 we conclude our work.

2 The scheduling problem

In this section, we present a new way of dealing with the integrated enterprise scheduling problem. As a combination of technological advances (like EDI, EFT, Internet, new material handling techniques using robotics, POS etc) and proliferating corporate partnerships (themselves the result of a focus on core competencies), as well as customer demand for more choice and convenience, have made

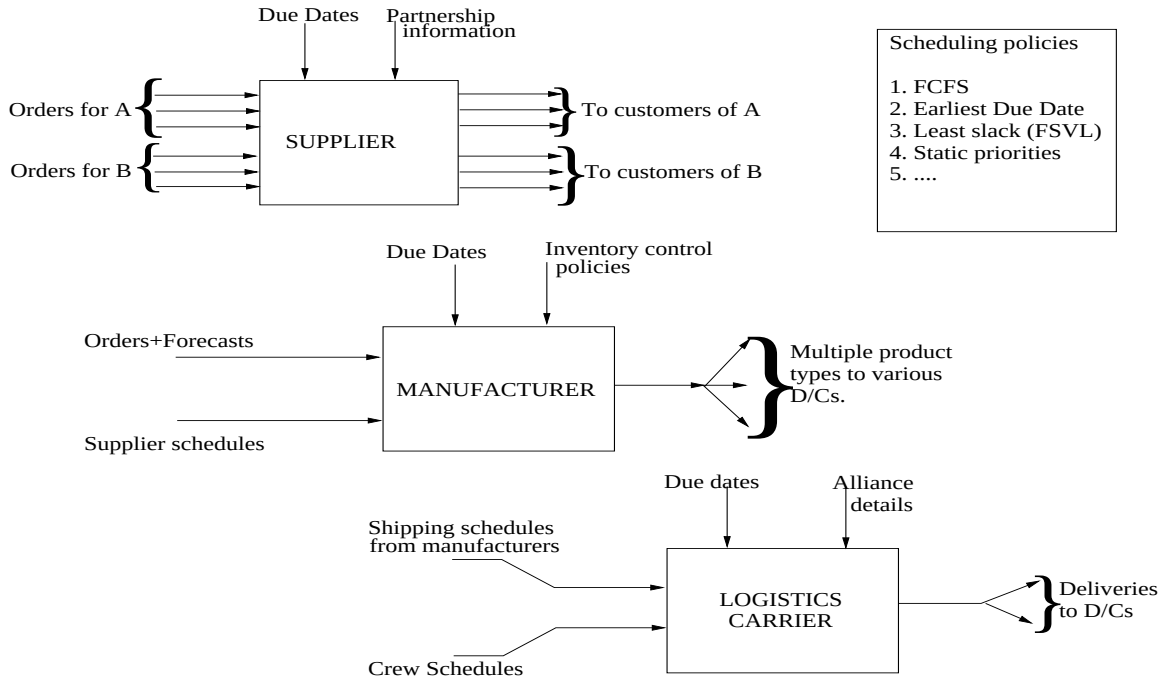

Figure 1: The scheduling problem in a supply chain network

mass customization possible, more and more importance goes to the scheduling of the supply chain since every sub-process in the supply chain adds value to the final product/service. In a number of industries like - computers, pharmaceuticals, chemicals, autos, health care, retail banking, and information services - the percentage of time spent by a customer order in logistics and other sub-processes has grown steadily over the past two decades, even as manufacturing lead time has been brought down to about 5-10% of the total lead time, thanks to new techniques of manufacturing like FMS, and new approaches to control like CNC.

As is clear from the physics of the system (see Figure 1),

- the supplier of raw materials to a manufacturer can be supplying to several other manufacturers
- the manufacturer on the other hand can be procuring from several suppliers like the ones above and sell the finished goods to several distributors
- the distributor can purchase finished goods from several manufacturers

Thus it is clear that any treatment of the scheduling problem in isolation will yield sub-optimal results. Given such an environment, the scheduling problem that we

wish to solve has a special feature bearing clear distinction from classical literature on scheduling. The latter have typically, the following features:

1. The solutions proposed cater only to the manufacturing plant which is treated in isolation from all other members of the supply chain.

2. The static (both deterministic and stochastic) scheduling methods which attempt at obtaining a near optimal solution using techniques like Lagrangian Relaxation, assume that the machine capacities are fixed and don't allow for incorporating dynamic constraints like break downs.

3. The dynamic scheduling (both deterministic and stochastic) are either too hard to solve, or the approximations proposed are only for specific configurations of the manufacturing plant.

We have enhanced the application of the scheduling methods in two ways: firstly, we apply them in the broader context of the supply chain, and secondly, we don't restrict the model to single product cases, which is common in literature [6]. We have studied the feasibility of using fluctuation smoothing policies in the supply chain context and have found encouraging results.

3 The physics of a multi-class make-to-order supply chain network

Here, our aim is to illustrate using examples that QN models can capture the interactions that occur in a supply chain network, and can be used to determine the performance measures of interest. Consider a supply chain network consisting of two product lines, A and B, which are assembled in one final assembly plant FAP, and distributed by dedicated fleet of logistics (See Figure 2). The Bill-of-materials (BOM) for each batch of the products is: A needs one each of A1 and A2; B, one each of A3 and A4. Suppliers S1 and S2 provide sub-assemblies A1 and A2 respectively; and S3 and S4 supply A3 and A4 respectively. For instance, the two products could be different varieties of desk-jet printers, with the sub-assemblies representing the cartridges and a base product. Let us assume that there are two types each of cartridges (A1 and A3) and the base product (A2 and A4). Thus the end products A and B are assembled from, respectively, (A1, A2) and (A3, A4). Other combinations are possible, although we do not consider them presently. We assume that the organization has worked out the relationships with the suppliers

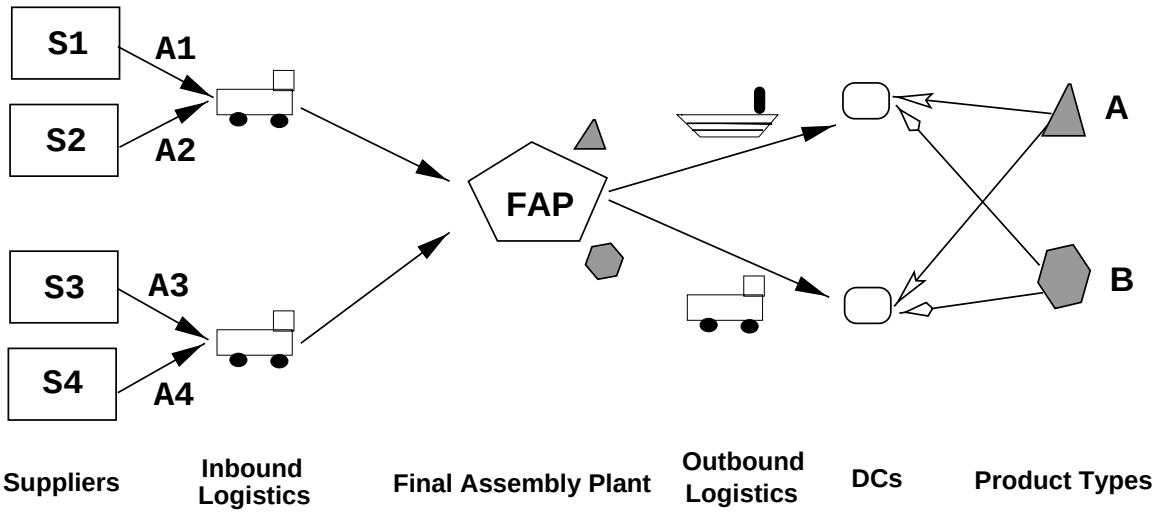

Figure 2: The supply chain network considered for analysis

to supply the components. The distribution is handled by two separate fleet of transporters (probably third party): one for the sub-assemblies and the other for the outbound finished goods. Batch processing is assumed everywhere. We are not concerned here with optimal batch sizes and the like. For our purposes, a job represents a batch of sub-assemblies or finished products.

3.1 A multi-class FJQN model for the supply chain network considered

We describe how a MTO SCN can be modeled by a Generalized Queueing Network. The flow of material and information is assumed to be in the same direction, *viz,* from left to right (see Figure 4).

We model the facilities of the SCN by 'servers' in the FJQN. We would like to emphasize here that traditional research has treated a manufacturing system as a Jackson network. We extend the scope of the same and model the entire supply chain as a generalized queueing network, with the manufacturing system as a single node of the network, interconnected with various other facilities. Two issues need attention: modeling the interfaces between the supply chain elements and the production planning and control (PPC) technique. The latter we fixed as make-to-order (MTO) and we model the interfaces as follows. Suppose a supplier 'S' makes two different products A and B and supplies to two manufacturers M1 and M2. Then S has four orders to serve: those for products A and B from M1 and

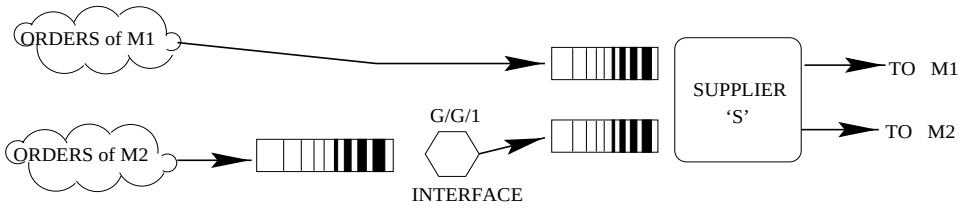

Figure 3: Modeling the interface between supplier and manufacturers

from M2. The relationship between M1, M2 and S determines how quickly S supplies the orders, or, what priority S gives to the orders while scheduling facilities that S owns. For example, if M1 has an alliance with S (one of the Vendor Managed Inventory strategies), then the due dates for M1's orders for A and B are set depending on the stocks at M1. This reflects in the queueing model as the absence of any interface server between S and M1 (see Figure 3). On the other hand if M2 treats S as just another reliable supplier, then scheduling for M2's orders at S commences only after firm orders have been received by S. In the queueing network, this is modeled as an input scheduler queue which is G/G/1 type *preceding* S as in Figure 3. Alternatively, this can be reflected in assignment of the manufactured product, i.e., as a delay in the delivery logistics, *succeeding* S. We follow the latter method while modeling interfaces (see Figure 4). Some of the interfaces could be procuring the bill of lading if the component sub-assemblies are shipped from abroad, quality checks on incoming goods, etc. Thus, an important feature of our FJQN model is that we not only model the value adding processes (like production and logistics functions), but also the various interfaces that usually contribute to increased delays in the SCN, and also are non-value adding. Also, the modeling is done at an *aggregated* level.

We now define as a *job* in the MTO GQN, the customer order which gets processed at the various stages. We don't distinguish the *nature* of the order from one stage to another. What we mean is that, though the customer order takes on various forms at each of the stages of the GQN, we consider the service time of the order at a stage as essentially, value added to the end product order at that stage.

The model has two classes of jobs, A and B. This model is a fork-join queueing network [10, 3]. Forking occurs, for e.g. when a demand for a batch of class A products simultaneously generates two orders, one each for a batch of sub-assemblies A1 and A2. Joining represents synchronization, occurring for e.g. before transporting sub-assemblies A1 and A2 to the final assembly plant (FAP).

The model has nine nodes, corresponding to the four suppliers, two logistics carriers with interfaces included, one final assembly plant, and the dedicated fleet

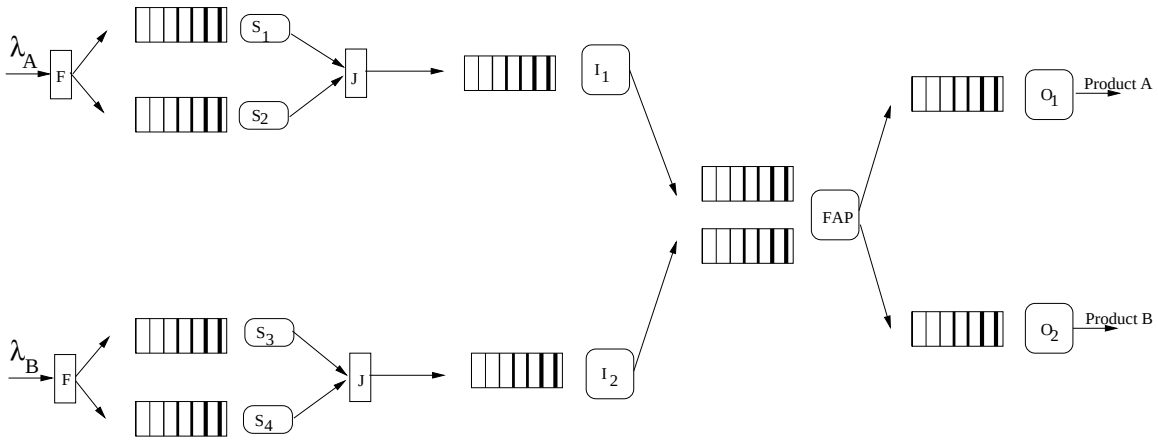

Figure 4: An open, two class, fork-join queueing network model of the supply chain network considered in Figure 2

of logistics for inbound and outbound material flow. We note that in this model, the interface effects have been clubbed with the logistics server itself; we have modeled more detailed supply chains with interfaces as individual servers, which is not presented in this paper, to make exposition easy. Although can be done easily, we are not modeling the distributors (or the retailers) and the process of moving the products to *end-customers*. To simplify matters, we assume infinite waiting capacity for all the queues. There are queues in front of all the facilities of the supply chain. For instance, a queue at supplier S1 will contain backorders for sub-assemblies A1 required to produce A, while the queue at supplier S3 contains backorders for sub-assemblies A3 required to produce B.

As shown in the Figure 4, there are two end products A and B. In-bound logistics I_1 and I_2 are *manufacturer* managed. The manufacturer FAP has the option of switching production between product types A and B. Finally, the end products are distributed to the customers by individual out-bound logistics carriers O_1 and O_2, which are managed by the retailer.

We have to specify a sequencing policy (refer to section 4 for details) to decide which of the orders will be taken up next for processing by the all the nodes.

The model is an open queueing network model, with two external arrival processes corresponding to the orders for product lines A and B. Let λ_A and λ_B be the arrival rates of these processes. We investigated a variety of random variables for the inter arrival time distributions. We also considered bulk arrivals with each arriving order carrying demand for a random number of products. It may be observed that each arriving order at the suppliers triggers a batch of jobs to be present in their

respective buffers. Each batch of jobs comes with a specific due date, which we generate using uniform random numbers in a given range. We assume that all jobs of a batch are processed together, before the server switches to another batch for processing. The service process at all nodes are modeled as normal random variables. In our experiments, we model the FAP as a multi-server node, with two servers, each dedicated for A and B respectively. Alternately, we could have a generic manufacturing facility at the FAP, which is an FMS that can process any type of job. This situation is an easy extension to our model, though not considered by us. There are two buffers in front of the FAP. The first buffer contains batches of sub-assemblies A1, A2 for producing A; the second contains batches of A3 and A4 for producing B. Here too, the server at FAP uses an appropriate scheduling policy to select the next batch to be processed, and serves a batch exhaustively before switching. Once finished goods are available at the FAP, dedicated fleet of transport facilities move the goods in batches from the FAP to all the DCs where the orders were received. The above FJQN model is analytically intractable for many reasons such as non-product form structure, non-exponential processing times, forking, joining, etc [3]. However it is amenable for rapid experimentation using simulation. We reiterate here that our attempt is strategically different from all current approaches at modeling manufacturing systems, in that, we model the entire supply chain (and not a single manufacturing plant) as an FJQN.

4 Fluctuation smoothing policies for scheduling supply chains

Fluctuation smoothing policies [7] are a special case of *Least Slack* scheduling policies, in which, for every part π (an order in our case) that enters the network, there is associated a real number $\beta(\pi)$. Also to each buffer of interest (where scheduling is of significance, for e.g. in the buffer of a bottle neck facility), b_i, there is associated a real number γ_i. If a lot is located in buffer b_i, the slack $s(\pi)$, is defined by,

$$s(\pi) := \beta(\pi) - \gamma_i$$

A least slack scheduling policy gives highest priority to the part π for which the slack is minimum. Whenever the server is to choose the next part after a service completion, it picks up a part with the least slack. Now a particular choice of $\beta(\pi)$ and γ_i will give the particular least slack policy a unique capability. See [7] for a good overview of fluctuation smoothing (FS) policies.

1. **Reducing the variance of lateness (FSVL):** Suppose each part π has a due-date $d(\pi)$ associated with it. Let γ_i be the expected remaining time in the network for buffer i. Then the slack is given as [7]:

   $s(\pi) := d(\pi) - \gamma_i$

   A least slack policy based on the above definition for slack will reduce the lateness of some parts at the cost of increasing the lateness of others. In other words, this policy will push the parts which are lagging behind schedule, pulling those that are ahead of schedule, minimizing the variance in the lateness of the parts.

2. **Reducing the variance of cycle time (FSVCT):** In the above policy, if we interpret the due-date differently, *viz* the time of entry of the part into the network, $\alpha(\pi)$, then all the parts are always late (w.r.t the due-date). Thus the slack is redefined as in [7]:

   $s(\pi) := \alpha(\pi) - \gamma_i$

3. **Reducing the mean cycle time (FSMCT):** This is done in an ingenious way by the authors of [7]. They use approximations for the GI/GI/1 queue, to say that if we control the burstiness of arrivals into a buffer, we in essence reduce the variance of the *arrival process*, which has a significant effect on the cycle time by the formulas available. Hence, in this case, we set the due-date of all entering parts as equal to: n/λ, where n is the lot number of the arriving part and λ is the mean arrival rate. Once this is done, we use this value for calculating the slack, as follows:

   $s(\pi) := n/\lambda - \gamma_i$

5 Simulation results

We conducted detailed simulation experiments for the supply chain network shown in Figure 4. We assume that the arrival process to this FJQN is compound Poisson with each arrival meant for end product A carrying a Normally distributed number of items, N(10,4), and that of B, N(20,4). We tag each arriving order with a due date which is a uniform random variable.

Although the scheduling problem can be studied at each of the nodes of this FJQN, we considered the bottleneck server to be the manufacturing plant FAP. We also had set up costs at the FAP, as also break downs occurring at this node, with appropriate random variables. We ran experiments for five scheduling policies: FIFO, EDD, FSVL, FSVCT, and FSMCT, for exponential releases, with equal

arrival rates for both the end products. The effect of increased utilization at FAP on the four major performance measures of interest, *viz*, the mean and the variance of cycle time, and the mean and the variance of delay of both the end products A and B were studied. Our scheduling policies unanimously yielded better results when compared to existing techniques. We present in the form of graphs (which show trend lines, since we consider only certain values for arrival rates) the performance of these scheduling policies, for A and B. The utilization of FAP was increased gradually, by increasing the arrival rates of orders for A, or/and B, keeping the service rate constant at the node. Refer Figures 5–8. We observe that in all these graphs, at low arrival rates the performance of all the scheduling policies are comparable, while the effect of the fluctuation smoothing policies is clearly felt at high arrival rates of the end product orders (consequently high utilizations at FAP).

From these results, we infer the following insights for supply chain process operations managers:

- Electronic data interchange (EDI) can be used to transmit the information on the mean remaining delay for entering orders from a particular node. This can be used in the fluctuation smoothing policies that we have used, for reducing the mean and variance of supply chain lead times.
- Integrated scheduling of the supply chain, rather than myopic scheduling at a facility ignoring delays at other facilities, can provide better performance in terms of lead times and delays.

Though our experiments were performed on small scale supply chains, we conjecture that using them for large scale supply chains using make-to-order policy, will yield even better insights and results.

6 Conclusions

We have developed a new method of modeling the dynamics of a MTO supply chain using generalized FJQNs and have presented a class of least slack policies for scheduling orders. Our scheme is different from the one in [5], in the sense that we have investigated the fluctuation smoothing policies for *multi class* MTO supply chains. For a wide range of problem parameters, the fluctuation smoothing policies of scheduling are shown to outperform existing scheduling methods using

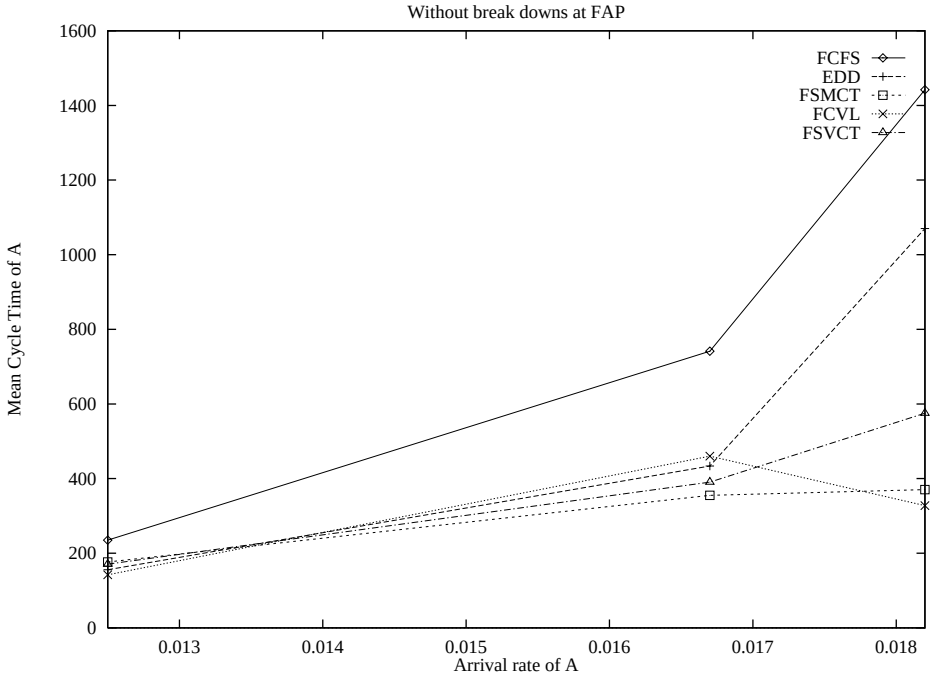

Figure 5: Mean cycle time of orders for A, with no break downs allowed at FAP

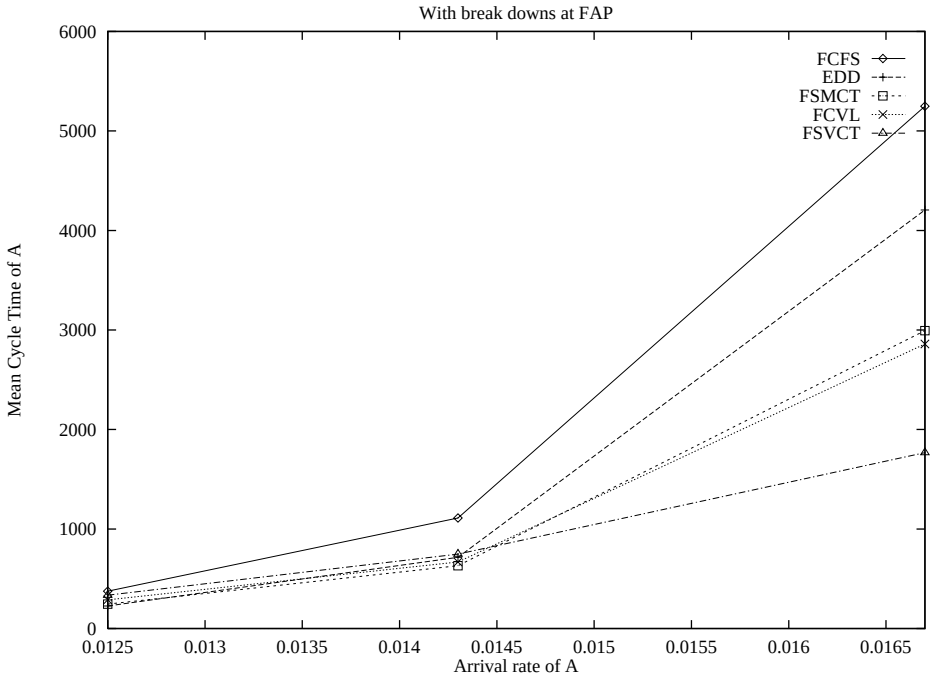

Figure 6: Mean cycle time of orders for A, with break downs at FAP

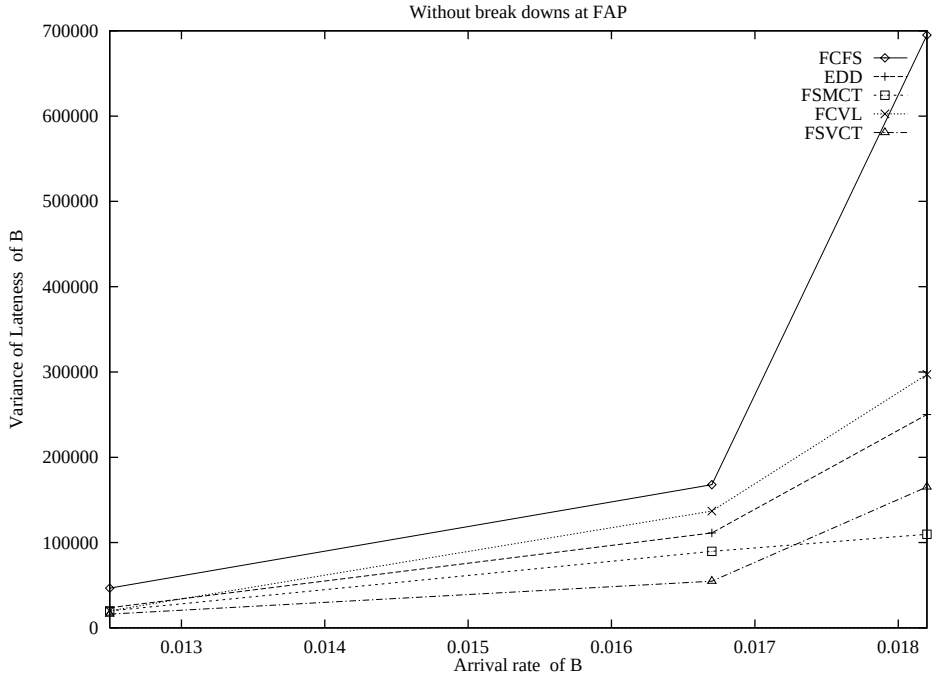

Figure 7: Variance of lateness of orders for B, with no break downs allowed at FAP

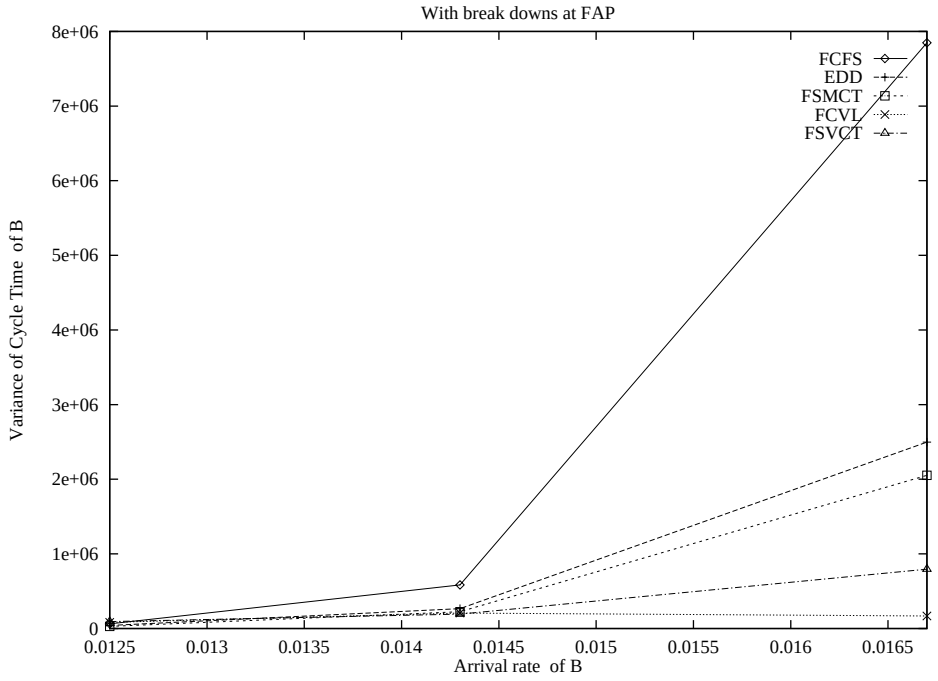

Figure 8: Variance of cycle time of orders for B, with break downs at FAP

simulation. Interesting conclusions on the efficient operations of a supply chain are also derived from these experiments.

References

[1] D. L. Anderson. Supply chain management. *Sloan Management Rev.*, 38(4):5–5, SUM 1997.

[2] D. J. Bowersox and D. J. Closs. *Logistical Management*. The McGraw-Hill Co. Inc., 1996.

[3] F. Bacelli, W. A. Massey and D. Towsley. Acyclic fork-join queueing systems. *Journal of the ACM*, 36:615–642, 1989.

[4] Sanjeev Gupta. Supply chain management in complex manufacturing. *IIE Solutions*, pages 18–23, March 1997.

[5] P. R. Kumar. Scheduling queueing networks. In F. P. Kelly and R. J. Williams, editors, *The IMA Volumes in Mathematics and its Applications*, pages 21–70, 1995.

[6] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Transactions on Automatic Control*, 36(12):1406–1416, December 1991.

[7] S. H. Lu, Deepa Ramaswamy, and P. R. Kumar. Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. *IEEE Transactions on Semiconductor Manufacturing*, 7(3):374–388, August 1994.

[8] N. Viswanadham, Y. Narahari, and N. R. S. Raghavan. Design/analysis of manufacturing enterprises: A business process approach. In N. C. Suresh and J. M. Kay, editors, *Special Edition on Group Technology and Cellular Manufacturing*. Kluwer Academic Publishers, 1998.

[9] U. S. Karmarkar, S. Kekre, and S. Kekre. Multi-item batching heuristics for minimization of queueing delays. *Eur. Jl. Opns. Res.*, 58:99–111, 1992.

[10] N. Viswanadham and Y. Narahari. *Performance Modeling of Automated Manufacturing Systems*. Prentice Hall, Englewood Cliffs, NJ, 1992.