

---

## **Throughput and inventory estimation of a pull-based supply system**

---

**Jishnu Hazra**

Indian Institute of Management, Bangalore 560 076, India  
E-mail: hazra@iimb.ernet.in

**Paul Schweitzer and Avi Seidmann**

University of Rochester, NY 14627, USA

\*\*\*\*\***AUTHOR, please supply e-mail addresses**\*\*\*\*\*

**Abstract:** We analyse an assembly system fed by components produced by suppliers and operating under a pull system with random processing times. Under certain assumptions we present a performance evaluation algorithm which is based on physical decomposition of the network and use of an aggregation/disaggregation algorithm. We consider two important performance measures: system throughput and work-in-process inventory. Performance evaluation of these systems is difficult because of the mating of parts and a kanban-like operating mechanism. Our algorithm was tested using simulation and found to be accurate.

**Keywords:** Kanban; pull systems; assembly networks; CONWIP; performance evaluation.

**Reference** to this paper should be made as follows: Hazra, J., Schweitzer, P. and Seidmann, A. (xxxx) 'Throughput and inventory estimation of a pull-based supply system', *Int. J. Manufacturing Technology and Management*, Vol. X, No. Y, pp.000–000.

**Biographical notes:**

[AUTHOR(S) – PLEASE SUPPLY BRIEF CAREER HISTORY – APPROX. 100 WORDS PER AUTHOR]

## 1 Introduction

In this paper we develop an algorithm to analyse the performance of a production system employing the pull technique, via the use of production authorisation signals or kanbans. The system under study consists of a main assembly line and is fed by assembled components manufactured by suppliers. These assembled components or sub-assemblies are sub-systems of the main product with multiple options. For example, seat assemblies in a car of the same model have different fabric colours as options or steering wheels with different options like cruise control, air bags and left and right hand drive. Similarly, many car companies like Toyota have over 100 different bumper assemblies to meet the requirements of different customer segments. The Nissan Motor Company has developed a pull system with the Tachikawa Spring Company (TSC) which makes car seats [1]. Nissan provides real time information to TSC about the kind of seats required in their assembly plant. These seats come in different colours and fabrics. TSC assembles the seat only when it receives the pull signal from Nissan and delivers these seats in the exact sequence required by Nissan. The Toyota Motor Manufacturing, Inc. (TMM), based in Georgetown, Kentucky, also uses a similar pull system with its seat supplier, Johnson Controls. A Harvard Business School Case describes the operation of the pull system at this Toyota plant [2].

In the next section we present a literature review and the model is discussed in Section 3. The algorithm is developed in Section 4, Section 5 contains the numerical results and finally we conclude in Section 6.

## 2 Literature review

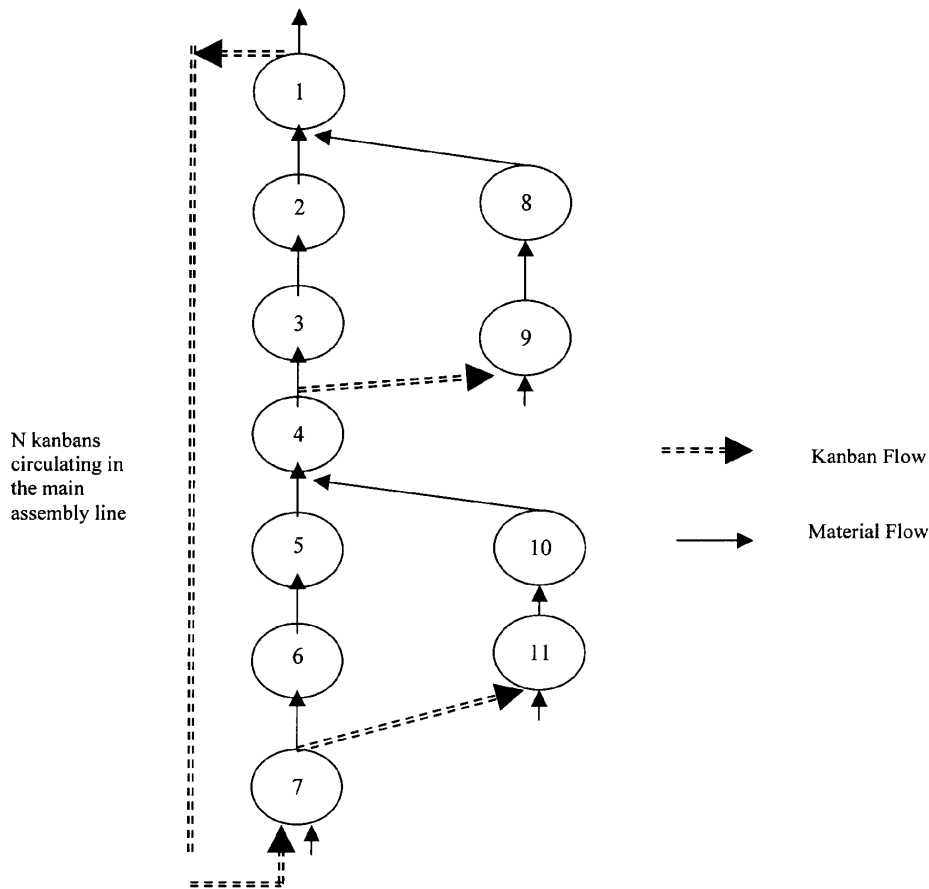
Most of the analyses on pull-based (kanban) production systems are based on simulation because of the inherent difficulty in mathematically modelling them, so there are few analytical models in this area. A model to determine the number of cards in a deterministic kanban assembly system is shown in [3] whilst a serial kanban system using a Markov chain is modelled in [4]. An approximate analytical algorithm, which can solve reasonably large problems, to evaluate the performance of a serial kanban system with machines having exponential processing time distribution is analysed in [5]. The technique is based on the machine by machine decomposition method and solving a set of fixed point equations. A general purpose analytical method based on a decomposition technique for evaluating the performance of multi-stage kanban controlled production systems is presented in [6]. The kanban system is modelled as a queuing network with synchronisation mechanisms. An iterative procedure is used to solve each decomposed sub-system.

In [7] a manufacturing system controlled by a release mechanism called CONWIP (an acronym for CONstant WIP because work-in-process inventory remains constant) is introduced and the paper discusses a serial production network. The above model is extended to an assembly system where  $n$  serial lines merge into a single assembly machine [8]. In [9] an analytical model for the performance evaluation of fabrication/assembly systems with kanban controls is presented. [10] discusses a problem with a similar control mechanism but with a general network topology with multiple assembly machines. We refer the readers to [11,12] and [10] for a more comprehensive literature survey on pull-based assembly systems.

### 3 The model

Our work extends the material flow control framework of [8] and [10]. Unlike in these two papers we explicitly model the supplier assembly system with real time information flow (the pull signal) between the main assembler and the supplier. From a queuing perspective the above papers deal with a closed queuing network, whereas our model in this paper is a combination of a closed and open queuing network.

**Figure 1** A pull-based production assembly network with two supply lines



Consider the production system shown in Figure 1, where circles denote machines. Machines 7 down to 1 constitute the main assembly line; machine 7 is the first stage machine whilst machine 1 is the final stage machine. Machines 11 and 10 and machines 9 and 8 are the two supply lines. For example, supply line 1 (machines 11 and 10) could be making different types of bumpers for cars being assembled in the main line. Supply line 2 could be making the seat assembly. In both these supply lines there are different varieties and the assembly of these options starts only when it receives the signal from the main assembly line. This signal is termed ‘broadcast’ in the industry circle and

readers should refer to [2] and [13] for the finer details of the operations. Referring to Figure 1, machines 1 and 4 carry out the assembling operations for which it requires inputs from two different machines. Machine 1, for example, requires input from both machines 2 and 8 before it can assemble the two parts.

The main assembly line is controlled by a CONWIP control mechanism with  $N$  parts circulating in the loop. We have used the CONWIP control structure because of its many inherent performance advantages as revealed by simulation studies in [14]. CONWIP lines are also easier to specify compared to conventional kanban systems. A single parameter,  $N$ , is required for CONWIP systems whilst the number of kanbans for each machine has to be specified in the conventional kanban system. Moreover, mathematically the conventional kanban system becomes intractable.

The supply line operates only when it gets an authorisation from a certain stage in the main line. For example in the system in Figure 1, supply line 1 starts producing the component for the  $n$ th job only when this job completes processing on machine 7 in the main assembly line. This is shown by the information flow line from machine 7 to machine 11 in Figure 1. Similarly, supply line 2 can produce the component for the  $n$ th job when this job completes processing on machine 4 in the main assembly line. The signal is, therefore, sent from machine 4 to machine 9. In most cases the supply line is not located in the factory compound of the main assembly line and the signal is therefore an electronic signal rather than a physical card as described in many textbooks. The pull signal, here, is the production authorisation signal. In other words, it is only after receiving the signal that a supplier starts assembling the component for the corresponding job in the main assembly line.

The CONWIP system ensures that the total work-in-process (WIP) inventory in the main assembly line is equal to  $N$ . We assume that buffer capacity is unlimited (or equal to at least  $N$ ) and therefore machines are never blocked. The buffer connecting machines  $i$  and  $j$ , where  $i$  is the downstream machine and  $j$  is the upstream machine, is denoted by  $(i,j)$ , and the number of parts in buffer  $(i,j)$  is represented by the random variable  $B_{ij}$ . The number of parts in buffer  $B_{ij}$  also includes the part being currently processed on machine  $i$ . The actual value of random variable  $B_{ij}$  is given by  $b_{ij}$ . If machine  $i$  is the first stage machine in the main line or the supply line (for example machines 7, 9 and 11 in Figure 1) then we use  $(i,0)$  to denote the input buffer of  $i$ . It is possible to show that the following relationships are true:

$$\left. \begin{aligned} B_{45} + B_{56} + B_{67} &= B_{4,10} + B_{10,11} + B_{11,0} \\ B_{12} + B_{23} + B_{34} &= B_{18} + B_{89} + B_{9,0} \\ B_{12} + B_{23} + B_{34} + B_{45} + B_{56} + B_{67} + B_{7,0} &= N \end{aligned} \right\} \quad (1)$$

The third equation is true because we are using CONWIP control, with  $N$  parts circulating in the main assembly line.

We represent the network topology by  $(M, N, \{j_1, i_1, k_1\}, \dots, \{j_p, i_p, k_p\})$ . Here  $M$  denotes the number of machines in the main assembly line (machines are numbered sequentially and machine numbered  $M$  is the first stage machine and machine 1 is the final stage machine).  $N$  is the number of parts circulating in the main assembly line. The supply line is represented by a set of three parameters. The  $p^{\text{th}}$  supply line (in Figure 1, there are two supply lines) is represented by  $\{j_p, i_p, k_p\}$ . Here  $j_p$  is machine  $j$  on the main assembly line where the assembly of  $p^{\text{th}}$  supply component takes place. The production authorisation

signal is sent when a job just enters the queue for machine  $i$  in the main assembly line. Finally,  $k_p$  is the number of machines in this supply line. Thus we can represent the system in Figure 1 as  $(7, N, \{1, 3, 2\}, \{4, 6, 2\})$  and the system in Figure 4 as  $(7, N, \{1, 3, 3\})$ .

#### 4 An algorithm to compute performance measures

In this section we present an approximate algorithm to compute performance measures like throughput rate (system capacity) and mean queue lengths (work-in-process inventory). Analytical models for performance evaluation of manufacturing systems are useful at a system design stage to estimate capacity and inventory. Practical applications of analytical models for performance evaluation can be found in [15] and [12].

The assumptions in the model are enumerated below:

- 1 The processing time of each machine has an exponential distribution. We denote the processing rate of machine  $i$  by  $\mu_i$ .
- 2 The machines are reliable.
- 3 When machine 1 finishes processing a part, that part leaves the system, and a new part is released to the input buffer of machine  $M$  instantaneously. This is the CONWIP order release mechanism.
- 4 Machine  $i$  can operate only when all the input buffers of machine  $i$  have at least one part; a machine is starved when any of its input buffers is empty.
- 5 The total number of parts in the loop connecting machines 1 and  $M$  is equal to  $N$ .
- 6 The structure of the system is given by  $(M, N, \{j1, i1, k1\}, \dots, \{jp, ip, kp\})$  and the following conditions must hold: for every pair of supply line  $p$  and  $q$ , if  $jp < jq$  then  $ip < iq$ . This condition is required to ensure that network decomposition can be done in the way it is described later.

The assumption of exponential distribution for processing times is made to make the problem amenable for computational purposes. This assumption is common in published literature and justification is given in similar studies such as [7, 16, 5, 8]. Moreover, assumption of exponential distribution gives the worst case analysis for the performance measures under study and therefore provides a bound.

The algorithm is based on network decomposition. Each sub-network, which results from the decomposition, is then solved using an aggregation-disaggregation algorithm. It is an efficient algorithm and can be used to solve reasonably large assembly-like queuing networks. Whilst the system under study is Markovian the state space is very large and not amenable to solution by exact methods. For example, the number of states for  $N=12$  in Figure 1 is over 35 million and for  $N=30$  in Figure 4 the number of states is nearly 20 billion. The only practical method to estimate the performance measures of problems of such a size is by approximate algorithms and in Section 5 of this paper we have presented numerical results for problems of similar magnitudes.

The basic idea behind this algorithm is to partition the network physically so that each sub-network is easy to solve. The reader should note that the original network is a combination of an open and closed queuing system. The main assembly line in Figure 1 consisting of machines 1 to 7, and controlled by CONWIP, is a closed queuing system

because the number of jobs is constant and is equal to  $N$ . The supply lines are, however, open queuing systems because at any point in time the number of jobs can vary from 0 to  $N$ .

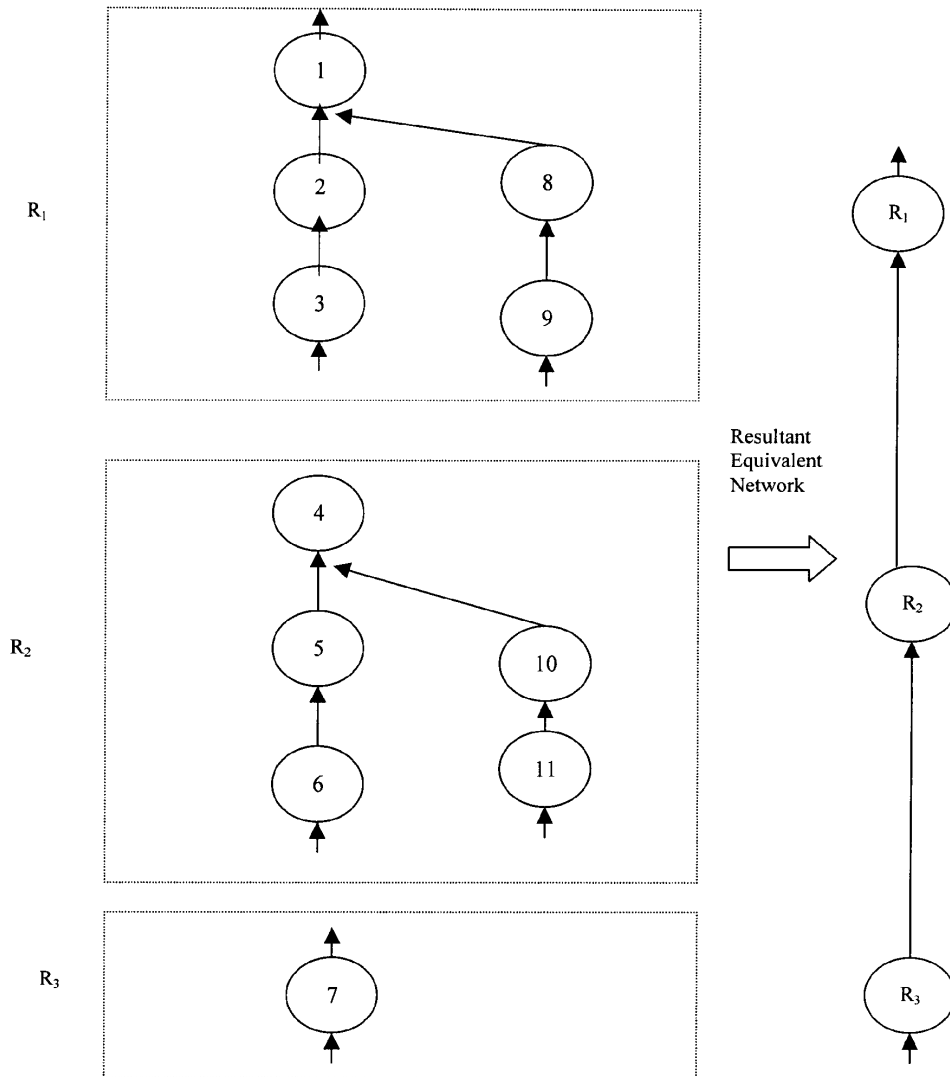
The partitioning is done in such a way that the number of jobs in the partitioned sub-network in the main assembly is always equal to the number of jobs in the supply line of the same partitioned network. For the assembly network in Figure 1 we partition it into three sub-networks  $R_1$ ,  $R_2$  and  $R_3$ , where  $R_1$  consists of machines 1, 2, 3, 8, and 9;  $R_2$  consists of machines 4, 5, 6, 10, and 11; and,  $R_3$  consists of machines 7 only. For example, in the sub-network  $R_2$  from the first equation in equation (1) we get  $B_{45} + B_{56} + B_{67} = B_{4,10} + B_{10,11} + B_{11,0}$ . Thus  $R_2$  is a valid partitioning. Each one of the above three sub-networks is replaced by a single machine with state-dependent processing rate. Therefore the resulting network, in this example, is a closed queuing network with three serial machines and state dependent processing rates. The justification of similar approximations in queuing networks was first proposed in [17].

The number of parts, in each of the sub-networks, remains constant until a certain transition takes place which results in one sub-network gaining a part and another losing a part. The number of parts in each branch of any sub-network can vary from 0 to  $N$  (we count the number of parts in one branch only as the other branch – the supply line, will have the same number of parts from equation 1). The sub-network behaves like a closed queuing network with an assembly structure.

For example, in sub-network  $R_1$  there will be  $n$  parts in each branch, where  $0 \leq n \leq N$ . Suppose,  $0 < n < N$ , then when machine 1 completes processing a part then the number of parts in sub-network  $R_1$  reduces by 1 to  $n-1$ ; on the other hand if machine 4 (in sub-network  $R_2$ ) finishes processing a part before machine 1 then the number of parts in  $R_1$  goes up by 1 to  $n+1$ . If  $n = 0$  then the number of parts goes up by 1 on completion of processing by machine 4. If  $n = N$ , then  $R_2$  and  $R_3$  will have no parts and on completion by machine will result in number of parts in  $R_1$  going down to  $N-1$  and  $R_3$  will gain a part.

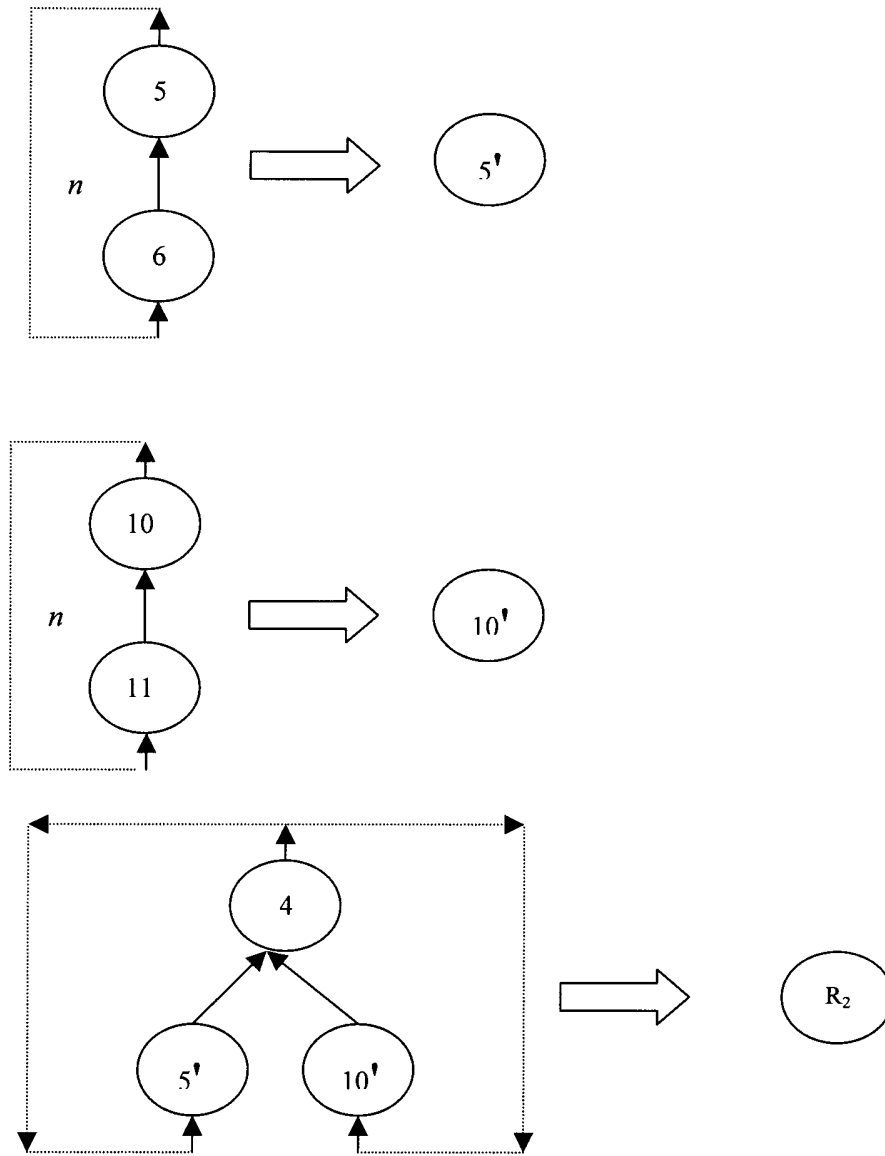
We replace the sub-network by a single equivalent machine that has a state dependent processing rate: the processing rate depends on the number of parts in the sub-network. If  $\mu_i(n)$  is the processing rate of the equivalent machine representing sub-network  $R_i$  with  $n$  parts in this sub-network then we have  $\mu_i(n) = TH_i(n)$ ,  $n = 1, \dots, N$ , where  $TH_i(n)$  is the throughput rate of the sub-network  $R_i$  with  $n$  parts in it. We model the sub-network as a closed queuing network with  $n$  jobs where  $1 \leq n \leq N$ . Having determined these state dependent processing rates the transformed system is now a closed serial system with  $r$  nodes where  $r$  is the number of partitions. The throughput of this system is approximately equal to the throughput of the original system. For the network in Figure 1 the partitioning and the resultant equivalent network is depicted in Figure 2. This reduced system is now easy to solve, and there are various algorithms available to compute the throughput and the queue lengths. The marginal queue length distribution of the original network is then obtained through a process of disaggregation.

**Figure 2** Decomposition of the network in Figure 1 and the resultant new network



Consider the sub-network  $R_3$ . It consists of a single machine and therefore  $TH_3(n) = \mu_7$ , for  $n > 0$  and  $TH_3(n) = 0$  when  $n = 0$ .

**Figure 3** Aggregation of sub-network  $R_2$  into an equivalent single machine



For sub-network  $R_2$  the equivalent processing rates are derived as follows:

Machines 5 and 6 are first replaced by machine  $5'$  where processing rate of this machine is given by the expression  $\theta_{5'}(n) = TH(\mu_5, \mu_6, n)$ . We define  $TH(\mu_5, \mu_6, n)$  as the throughput of a two stage closed serial system with  $n$  jobs circulating as shown in Figure 3. Similarly machines 10 and 11 are replaced by a single machine,  $10'$ . The processing rate of this machine is given by  $\theta_{10'}(n) = TH(\mu_{10}, \mu_{11}, n)$ . We now solve a

closed two-stage assembly system with assembly machine 4 fed by machines 5' and 10' as shown in Figure 3. Here feeder machines 5' and 10' have state dependent processing rates as given by expressions above. The flow balance equation from the theory of continuous time Markov chain is given by:

$$\begin{aligned} & \{\mu_4 I(b_{4,5'} > 0, b_{4,10'} > 0) + \mu_5 I(b_{4,5'} < 0) + \mu_{10'} I(b_{4,10'} < 0)\} \pi^n(b_{4,5'}, b_{4,10'}) = \\ & \mu_4 I(b_{4,5'} < n, b_{4,10'} < n) \pi^n(b_{4,5'} + 1, b_{4,10'} + 1) + \mu_5 I(b_{4,5'} > 0) \pi^n(b_{4,5'} - 1, b_{4,10'}) + \\ & \mu_{10'} I(b_{4,10'} > 0) \pi^n(b_{4,5'}, b_{4,10'} - 1), \quad 0 \leq b_{36}, b_{37} \leq n, \text{ and } 1 \leq n \leq N. \end{aligned} \quad (2)$$

In equation (2)  $I(\cdot)$  stands for the indicator function and takes a value 1 if the condition within the bracket is true and 0 otherwise.  $\pi^n(x, y)$  is the steady state probability of state  $(x, y)$  given that there are  $n$  jobs in the sub-network. From equation (2) the probabilities can be computed.

The processing rate of the equivalent machine of sub-network  $R_2$  is given by  $\mu_4 \sum_{x>0, y>0} \pi^n(x, y)$ ,  $n = 1, \dots, N$ . Similarly, the processing rate of the equivalent machine of sub-network  $R_1$  is derived.

The original network in Figure 1 is now replaced by the network at the right hand side of Figure 2. This new network, which is a closed queuing network with state-dependent processing rate, can now be solved using the balance equation of a continuous time Markov chain.

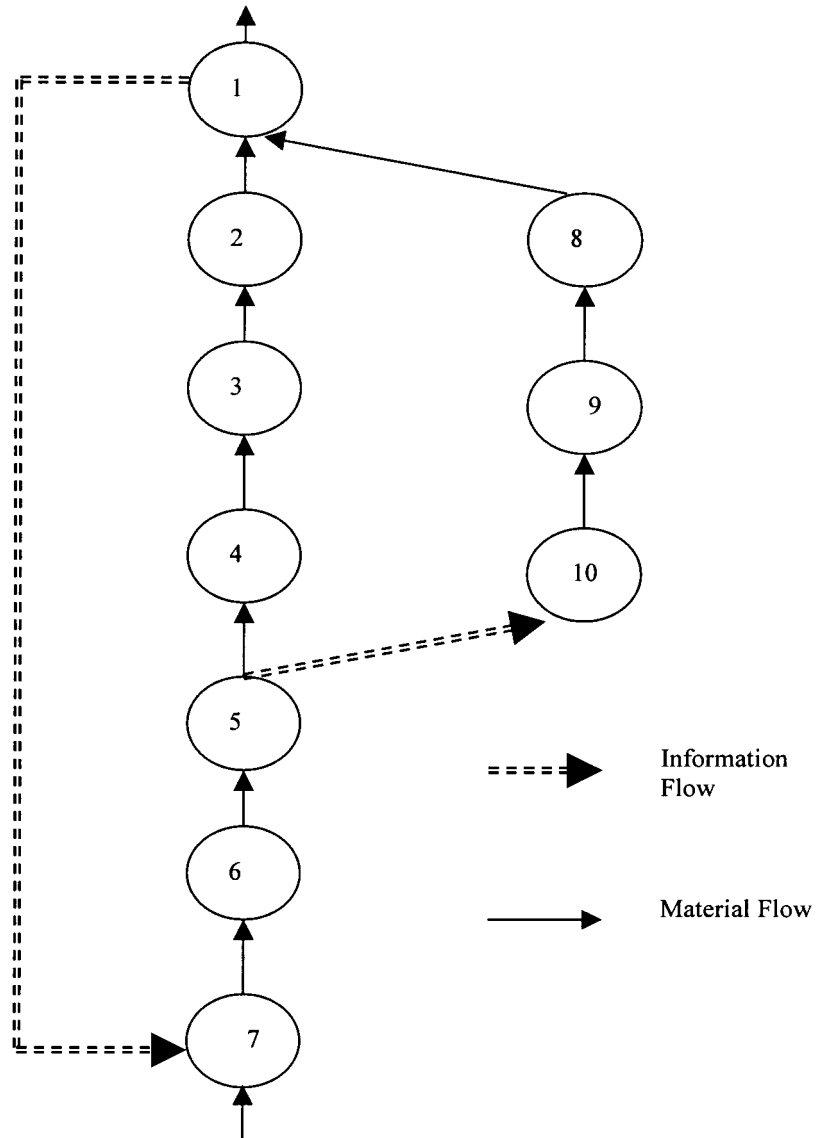
The mean queue length (or work-in-process inventory) can also be derived because we obtain an estimate of the marginal queue length distribution. For example, the marginal queue length distribution for buffer  $B_{ij}^p$ , where buffer  $(i, j)$  is in sub-network  $p$ , is  $\Pr(B_{ij}^p = n_1) = \sum \Pr(B_{ij}^p = n_1 / R_p = n) \Pr(R_p = n)$ .

The term  $R_p = n$  means that sub-network  $p$  has  $n$  parts in each branch. The first term on the right side is obtained by solving the sub-network  $p$  with  $n$  parts circulating in the loop. The second term on the right side is obtained by solving the closed serial system of the equivalent network.

## 5 Numerical results

In this section we present numerical results to test the accuracy of our algorithm. We have used two network topologies, which are shown in Figures 1 and 4. Two sets of machine processing rate data have been used for each topology, and for each set we have used three values of  $N$ .

**Figure 4** A pull-based production assembly network with one supply line



For the network topology in *Figure 1* we have used the following data sets:

*Data set 1:*  $\mu_i = 4$  units per time for all machines  $i$ . This is homogeneous system where all machines have identical processing rates.

*Data set 2:*  $\mu_1 = 9, \mu_2 = 4, \mu_3 = 6, \mu_4 = 8, \mu_5 = 5, \mu_6 = 5, \mu_7 = 6, \mu_8 = 5, \mu_9 = 5, \mu_{10} = 5$  and  $\mu_{11} = 6$ .

For both these data sets we have taken  $N$  to be 4, 8 and 12. In both Tables 1 and 2, we denote  $\theta$  as the throughput rate of the system measured as units per time period.

**Table 1** Error analysis for system throughput and expected WIP inventory for Data set 1 and Figure 1

$N$	4			8			12		
	<i>algo</i>	<i>Sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>
$\theta$	1.37	1.42	-3.31	2.07	2.12	-2.46	2.47	2.52	-1.83
E(B <sub>12</sub> )	0.83	0.80	3.50	1.59	1.55	2.25	2.33	2.34	-0.21
E(B <sub>18</sub> )	0.83	0.81	2.60	1.59	1.57	1.34	2.33	2.32	0.39
E(B <sub>23</sub> )	0.47	0.49	-3.88	0.97	0.98	-0.82	1.48	1.47	0.34
E(B <sub>89</sub> )	0.47	0.48	-2.48	0.97	0.97	-0.10	1.48	1.48	-0.14
E(B <sub>34</sub> )	0.47	0.48	-2.48	0.97	0.97	-0.10	1.48	1.47	0.75
E(B <sub>90</sub> )	0.47	0.48	-1.88	0.97	0.96	0.93	1.48	1.47	0.75
E(B <sub>45</sub> )	0.83	0.80	3.89	1.59	1.57	1.15	2.33	2.29	1.74
E(B <sub>4,10</sub> )	0.83	0.80	3.89	1.59	1.56	1.60	2.33	2.32	0.69
E(B <sub>56</sub> )	0.47	0.48	2.48	0.97	0.98	-0.82	1.48	1.48	-0.07
E(B <sub>10,11</sub> )	0.47	0.48	-1.26	0.97	0.99	-1.52	1.48	1.48	-0.27
E(B <sub>67</sub> )	0.47	0.47	-0.42	0.97	0.98	-0.82	1.48	1.49	-0.74
E(B <sub>11,0</sub> )	0.47	0.48	-1.88	0.97	0.98	-0.51	1.48	1.46	1.09
E(B <sub>70</sub> )	0.46	0.48	-4.17	0.94	0.97	-3.49	1.42	1.47	-3.33

**Table 2** Error analysis for system throughput and expected WIP inventory for Data set 2 and Figure 1

$N$	4			8			12		
	<i>algo</i>	<i>sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>
$\theta$	1.91	2.00	-4.20	2.83	2.92	-3.05	3.32	3.39	-2.15
E(B <sub>12</sub> )	0.65	0.61	6.36	1.09	1.07	2.34	1.37	1.37	0.22
E(B <sub>18</sub> )	0.74	0.70	5.12	1.46	1.42	2.89	2.35	2.38	-1.14
E(B <sub>23</sub> )	0.76	0.77	1.94	1.85	1.86	-0.16	3.34	3.37	-1.01
E(B <sub>89</sub> )	0.55	0.57	-2.82	1.16	1.17	-0.85	1.77	1.79	-1.18
E(B <sub>34</sub> )	0.43	0.43	-1.37	0.84	0.84	-0.24	1.18	1.17	0.26
E(B <sub>90</sub> )	0.55	0.56	-1.61	1.16	1.17	-1.11	1.77	1.75	1.09
E(B <sub>45</sub> )	0.64	0.60	6.30	1.10	1.05	3.99	1.44	1.39	3.60
E(B <sub>4,10</sub> )	0.76	0.72	5.58	1.40	1.38	1.37	1.99	1.98	0.30
E(B <sub>56</sub> )	0.55	0.56	-2.49	1.16	1.17	-1.03	1.77	1.78	-0.73
E(B <sub>10,11</sub> )	0.55	0.56	-2.13	1.17	1.16	0.60	1.80	1.78	1.29
E(B <sub>67</sub> )	0.55	0.56	-2.49	1.16	1.16	-0.17	1.77	1.76	0.57
E(B <sub>11,0</sub> )	0.43	0.44	-2.71	0.84	0.84	-0.36	1.19	1.17	1.71
E(B <sub>70</sub> )	0.42	0.43	-3.00	0.81	0.84	-4.15	1.14	1.17	-2.40

For the network topology in *Figure 4* we have used the following data sets:

*Data set 1:*  $\mu_i = 4$  units per time for all machines  $i$ . This is homogeneous system where all machines have identical processing rates.

*Data set 2:*  $\mu_1 = 4, \mu_2 = 4, \mu_3 = 5, \mu_4 = 4, \mu_5 = 3, \mu_6 = 4, \mu_7 = 5, \mu_8 = 5, \mu_9 = 4,$  and  $\mu_{10} = 4$ .

Here we have taken the values of  $N$  to be 10, 20 and 30.

To check the accuracy of the algorithm we have used simulation. For this purpose we have used ARENA for simulating. The analytical algorithm was coded in Pascal. The error in estimation of performance measure  $\alpha$  is defined as:

$$\frac{\alpha(\text{algorithm}) - \alpha(\text{simulation})}{\alpha(\text{simulation})}$$

Whilst no general conclusions can be reached as to how the accuracy of the algorithm is affected by various factors such as value of  $N$ , number of machines and the processing rates (that is, if it is a homogeneous or non-homogeneous system) the estimation of throughput rate is quite accurate. The accuracy of queue length estimation varies and can be off by 20% especially for large values of  $N$ . This is because there is a high correlation between different queue lengths for a particular topology; a large error for one particular queue length results in a large error in some other queue lengths. The correlation is a result of a queuing network that is partially ‘closed’.

From Equation (1) we see that the work-in-process in the supplier system is related to work-in-process of the main assembly system. Moreover, queue lengths, unlike throughput, is a more detailed level performance measures and therefore is expected to have lower accuracy when using approximate algorithm.

The running time for the analytical algorithm is small. For example, for the data shown in Table 1 the running time was 0.05, 0.5, and 2.3 seconds for  $N$  equal to 4, 8 and 12 respectively on a Pentium III PC.

Using the algorithm and simulation it can be verified that the WIP inventory is increasing and convex in  $N$ . This property is observed in both the analytical algorithm and the simulation. This behaviour was also observed in other numerical results not presented in this paper. The monotone concavity property of throughput with respect to  $N$  is also preserved by the analytical algorithm.

Going back to Figure 4 for analysis we compare the effect of sending the production authorisation signal from the main assembly line to the supplier line. When the signal is sent to the supplier line on completion by machine 1, then the release mechanism at the supply system is also CONWIP, as in the main assembly line. Here the WIP inventory in the supplier line becomes equal to the WIP inventory in the main assembly line. The total system inventory is therefore  $2N$ . When the production authorisation signal is sent after a completion by machine 7 then the inventory in the supplier line will vary from 0 to  $N$  and expected inventory in the supplier line is likely to be less than  $N$ . The total system inventory will therefore be less than  $2N$ . Therefore in terms of inventory performance the supplier is better off getting the production authorisation signal at a later stage. Thus there exists a trade-off in deciding whether to send a signal early or late. Sending the signal at an earlier stage to the supply line will imply higher system throughput rate and higher inventory, whilst the opposite effect will take place when signals are sent at a later stage. The assembler will benefit with higher throughput (assembler’s inventory always remains constant at  $N$ ) whilst the supplier will benefit by having lower inventory.

The assembler and the supplier would therefore be required to come to some compromise. The optimum information transfer structure for the system will depend on the cost and revenue structure of the two players. The model has shown the importance of information sharing between the assembler and the supplier and its implications for performance measure. Information sharing between the manufacturer and the supplier becomes more important with increased product variety, shorter product life cycle and high demand variability.

## 6 Conclusion

In this paper we presented a model of a production system having an assembly structure that uses a pull signal for order release mechanism. The production system under study produces a product where the components or sub-assemblies are made by a supplier and later assembled into the product in the main assembly line. Our model is based on the premise that suppliers have a flexible production system (low set-up time) and a tightly coordinated system with the assembler. The supplier can delay the production of components until it receives the production authorisation signal from the assembler. This makes the entire production system very flexible, which is an important competitive tool in today's market. As mentioned earlier, such pull systems are used in the car industry.

Analytical models to study this specific pull system are, to the best of our knowledge, not found in published literature. We develop an algorithm that is used to obtain important performance measures like system throughput and work-in-process inventory at each machine centre. Machine utilisation and mean waiting times can thus be computed. The algorithm is based on physical decomposition of the network followed by an aggregation/disaggregation algorithm. Using simulation the algorithm was found efficient in terms of accuracy and speed.

A drawback of the model is that production set-ups have not been considered. We have specifically assumed assembly-like operations which have generally negligible or no set-ups. We have also assumed that availability of raw material is not an issue. Another limitation of the model is that transportation time from the supplier to the assembler has not been incorporated in the model. A pure delay function (modelled as an infinite server queuing system) is difficult; moreover, the units transferred from the supplier to the assembler are typically done in small batches.

## References

- 1 Hall, R.W. (1983) *Zero Inventories*, Dow Jones-Irwin.
- 2 Mishina, K. (1992) 'Toyota Motor Manufacturing, USA Inc.', *Harvard Business School Case # N1-693-019*.
- 3 Bitran, G.R. and Chang, L. (1987) 'A mathematical programming approach to a deterministic Kanban system', *Management Science*, Vol. 33, pp.427-441.
- 4 Deleersnyder, J.L., Hodgson, T.J., Muller, H. and O' Grady, P.J. (1989) 'Kanban controlled pull systems: an analytic approach', *Management Science*, Vol. 35, pp.1079-1091.
- 5 Mitra, D. and Mitrani, I. (1990) 'Analysis of Kanban discipline for cell coordination in production lines, Part 1', *Management Science*, Vol. 36, pp.1548-1566.

- 6 DiMascolo, M., Frein, Y. and Dallery, Y. (1991) 'An analytical method for performance evaluation of Kanban controlled production systems', *Technical Report*, Institut National Polytechnique de Grenoble, Grenoble, France.
- 7 Spearman, M.L., Hopp, W.J. and Woodruff, D.L. (1990) 'CONWIP: a pull alternative to Kanban', *International Journal of Production Research*, Vol. 28, pp.879–894.
- 8 Duenyas, I. and Hopp, W.J. (1993) 'Estimating the throughput of an exponential CONWIP assembly system', *Queueing Systems*, Vol. 14, pp.133–157.
- 9 Rao, P.C. and Suri, R. (2000) 'Performance analysis of an assembly station with input from multiple fabrication lines', *Production and Operations Management*, Vol. 9, pp.283–302.
- 10 Hazra, J. and Seidmann, A. (1996) 'Performance evaluation of closed tree-structured assembly systems', *IIE Transactions*, Vol. 28, pp.591–599.
- 11 Dallery, Y. and Gershwin, S.B. (1992) 'Manufacturing flow line systems: a review of models and analytical results', *Queueing Systems*, Vol. 12, pp.3–94.
- 12 Suri, R., Sanders, J.L. and Kamath, M. (1993) 'Performance evaluation of production networks', in S.C. Graves *et al.* (Eds.) *Handbooks in OR & MS*, Elsevier Science Publishers, Vol. 4, Ch. 5.
- 13 Ohno, T. (1988) *Toyota Production System: Beyond Large-Scale Production*, Productivity Press.
- 14 Yucesan, E. and de Groote, X. (2000) 'Lead times, order release mechanisms and customer service', *European Journal of Operational Research*, Vol. 120, pp.118–130.
- 15 Burman, M; Gershwin, S.B. and Suyematsu, C. (1998) 'Hewlett-Packard uses operations research to improve the design of a printer production line', *Interfaces*, Jan/Feb., Vol. 28, Issue 1, pp.24–36.
- 16 Spearman, M.L. and Zazanis, M.A. (1992) 'Push and pull production systems: issues and comparisons', *Operations Research*, Vol. 40, Issue 3, pp.521–532.
- 17 Chandy, K.M., Herzog, U. and Woo, L. (1975) 'Parametric analysis of queueing networks', *IBM Journal of Research and Development*, Vol 19, pp.36–42.

**Table 3** Error analysis for system throughput and expected WIP inventory for Data set 1 and Figure 4

<i>N</i>	<i>10</i>			<i>20</i>			<i>30</i>		
	<i>algo</i>	<i>sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>
$\theta$	2.36	2.37	-0.42	2.98	3.00	-0.67	3.27	3.30	-0.91
E(B12)	2.30	2.38	-3.36	4.41	4.07	8.35	6.55	6.09	7.55
E(B23)	1.31	1.27	3.15	2.64	2.65	-0.38	3.95	3.66	7.92
E(B34)	1.31	1.26	3.97	2.64	2.53	4.35	3.95	3.57	10.64
E(B45)	1.31	1.32	-0.76	2.64	2.61	1.15	3.95	4.45	-11.24
E(B56)	1.26	1.24	1.61	2.55	2.65	-3.77	3.87	4.14	-6.52
E(B67)	1.26	1.24	1.61	2.55	2.89	-11.76	3.87	3.97	-2.52
E(B70)	1.26	1.30	-3.08	2.55	2.61	-2.30	3.87	4.12	-6.07
E(B18)	2.30	2.26	1.77	4.41	4.64	-4.96	6.55	5.56	17.81
E(B89)	1.31	1.28	2.34	2.64	2.43	8.64	3.95	4.38	-9.82
E(B9,10)	1.31	1.29	1.55	2.64	2.26	16.81	3.95	4.01	-1.50
E(B10,0)	1.31	1.40	-6.43	2.64	2.52	4.76	3.95	3.81	3.67

Author, please indicate where, in the text, these two Tables should appear.

Thank you

**Table 4** Error analysis for system throughput and expected WIP inventory for Data set 2 and Figure 4

$N$	10			20			30		
	<i>algo</i>	<i>sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>	<i>algo</i>	<i>sim</i>	% <i>error</i>
$\theta$	2.33	2.34	0.00	2.83	2.86	-1.05	2.96	2.93	1.02
E(B12)	2.17	1.93	12.40	3.68	3.81	-3.41	4.50	4.30	6.74
E(B23)	1.28	1.32	-3.03	2.31	2.24	3.12	2.85	2.46	15.86
E(B34)	0.84	0.83	1.20	1.29	1.31	-1.53	1.46	1.36	7.35
E(B45)	1.28	1.25	2.40	2.31	2.13	8.45	2.85	2.95	-3.39
E(B56)	2.37	2.58	-8.14	6.94	7.05	1.56	14.14	14.92	-5.23
E(B67)	1.24	1.28	-3.13	2.22	2.17	2.30	2.76	2.66	3.76
E(B70)	0.82	0.81	1.23	1.26	1.29	-2.33	1.44	1.35	6.67
E(B18)	2.17	2.05	5.85	3.68	3.41	7.92	4.50	4.47	0.67
E(B89)	0.84	0.86	-2.33	1.29	1.35	-4.44	1.46	1.46	0.00
E(B9,10)	1.28	1.25	2.40	2.31	2.34	-1.28	2.85	2.69	5.95
E(B10,0)	1.28	1.17	9.40	2.31	2.39	-3.35	2.85	2.45	16.33