



**HAL**  
open science

# Simulation Optimization of Pull Control Policies for Serial Manufacturing Lines and Assembly Manufacturing Systems with Genetic Algorithms

Dimitrios E Koulouriotis, Alexander S Xanthopoulos, Vasilios D Tourassis

► **To cite this version:**

Dimitrios E Koulouriotis, Alexander S Xanthopoulos, Vasilios D Tourassis. Simulation Optimization of Pull Control Policies for Serial Manufacturing Lines and Assembly Manufacturing Systems with Genetic Algorithms. *International Journal of Production Research*, 2010, 48 (10), pp.2887-2912. 10.1080/00207540802603759 . hal-00580104

**HAL Id: hal-00580104**

**<https://hal.science/hal-00580104v1>**

Submitted on 26 Mar 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**Simulation Optimization of Pull Control Policies for Serial Manufacturing Lines and Assembly Manufacturing Systems with Genetic Algorithms**

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2008-IJPR-0488.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	29-Sep-2008
Complete List of Authors:	Koulouriotis, Dimitrios; Democritus University of Thrace, Production and Management Engineering Xanthopoulos, Alexander; Democritus University of Thrace, Production and Management Engineering Dep. Tourassis, Vasilios; Democritus University of Thrace, Production and Management Engineering Dep.
Keywords:	ASSEMBLY SYSTEMS, GENETIC ALGORITHMS
Keywords (user):	ASSEMBLY SYSTEMS, GENETIC ALGORITHMS



# Simulation Optimization of Pull Control Policies for Serial Manufacturing Lines and Assembly Manufacturing Systems with Genetic Algorithms

Koulouriotis D.E., Xanthopoulos A.S. and Tourassis V.D.

## Author's Affiliation and Contact Data:

Koulouriotis E. Dimitrios : Assistant Professor [Corresponding Author]  
Production and Management Engineering Dep.,  
Democritus University of Thrace  
University Campus, Xanthi, 67100, Greece

e-mail : [jimk@pme.duth.gr](mailto:jimk@pme.duth.gr)

Tel-fax : +30 25410-79358

Xanthopoulos S. Alexander: e-mail : [axanthop@pme.duth.gr](mailto:axanthop@pme.duth.gr)

Tel-fax : +30 25410-79358

Cand. Phd

Production and Management Engineering Dep.,  
Democritus University of Thrace  
University Campus, Xanthi, 67100, Greece

Tourassis D. Vasilios:

e-mail : [vtourassi@pme.duth.gr](mailto:vtourassi@pme.duth.gr)

Tel-fax : +30 25410-79341

Professor

Production and Management Engineering Dep.,  
Democritus University of Thrace  
University Campus, Xanthi, 67100, Greece

# Simulation Optimization of Pull Control Policies for Serial Manufacturing Lines and Assembly Manufacturing Systems with Genetic Algorithms

Koulouriotis D.E., Xanthopoulos A.S. and Tourassis V.D.

Department of Production and Management Engineering, Democritus University of Thrace  
University Campus, Xanthi

Several efficient pull production control policies for serial lines that implement the lean/JIT manufacturing philosophy can be found in the production management literature. A recent development that is less well-studied than the serial line case is the application of pull type policies to assembly systems where manufacturing operations take place both sequentially and in parallel. Systems of this type contain assembly stations where two or more parts from lower hierarchically manufacturing stations merge in order to produce a single part of the subsequent stage. In this paper we extend the application of the Base Stock, Kanban, CONWIP, CONWIP/Kanban Hybrid and finally the Extended Kanban production control policies to assembly systems that produce final products of a single type. Discrete-event simulation is utilized in order to evaluate the performance of serial lines and assembly systems. It is essential to determine the best control parameters for each policy when operating in the same environment. The approach that we propose and probe for the problem of control parameter selection is that of a genetic algorithm with resampling, a technique used for the optimization of stochastic objective functions. Finally, we report our findings from numerical experiments conducted for two serial line simulation scenarios and two assembly system simulation scenarios.

*Keywords:* JIT manufacturing systems; pull production control policies; genetic algorithm.

## 1. Introduction

Effective production control, which translates to the control of the material flow that takes place within a manufacturing system in a way that resolves the trade-off between a high service rate and a low holding costs level plays a key role to the competitiveness of any manufacturing system. The recognition of the importance of effective production control by the research community gave rise to a significant number of sometimes quite disparate production control mechanisms. However, many production control mechanisms for flow-shop systems fall under two fundamental categories: *push* type and *pull* type. Production control schemes that utilize future demand forecasts to control the manufacturing process are known as push type production control mechanisms. On the other hand, pull type control policies react to actual occurrences of demand only. Various definitions of push and pull (as well as a clarification of the meaning of these two terms) can be found in Pyke and Cohen (1990). Pull type control policies implement the JIT manufacturing philosophy and have attracted considerable attention over the past years as they are widely considered to outperform push type mechanisms. According to the JIT manufacturing philosophy a manufacturing system should maintain the minimum levels of safety stocks that are required in order to meet the target service level while in the same time has the ability to react rapidly to incoming orders. Two fundamental pull control policies are Kanban and Base Stock (Buzacott and Shanthikumar 1993). Generalized Kanban (see Buzacott and Shanthikumar (1992) for example) and Extended Kanban (Dallery Y. and Liberopoulos G. 2000) are two policies that were developed in an attempt to combine the merits of Kanban and Base Stock. Spearman et al. (1990) introduced CONWIP, a policy that can be seen as a special case of the Kanban control policy. Applying CONWIP control in the manufacturing system as a whole and Kanban control in all of the system's manufacturing stages except the last one resulted in a new hybrid control policy known as CONWIP/Kanban Hybrid (see Paternina-Arboleda and Das 2001).

The bibliography regarding serial manufacturing lines controlled by the host of the above mentioned policies is quite voluminous. However, many important pull type control policies can be also applied to manufacturing systems containing assembly stations. An assembly station is a manufacturing cell where two or more parts from upstream manufacturing stations merge in order to produce a single part of the subsequent stage. Up to now the research carried out for systems of this type is relatively limited. Two variants of the

1  
2 Kanban control policy for assembly systems are analyzed in Di Mascolo and Dallery (1996) and Matta et al.  
3 (2005). The Extended Kanban for assembly systems was presented in Chaouiya et al. (2000). Takahashi et al.  
4 (2005) study assembly systems controlled by Kanban, CONWIP and a variation called Synchronized CONWIP  
5 using simulation. In this paper we extend the application of the Base Stock, Kanban, CONWIP,  
6 CONWIP/Kanban Hybrid and finally the Extended Kanban production control policies to assembly systems  
7 that produce final products of a single type. The release of component parts in an assembly station is  
8 *simultaneous*. In this case, a component part that it is stored in its corresponding output buffer and has the  
9 controller authorization to enter the assembly station will have to wait until all the other components become  
10 available too. This required synchronization intensifies the queuing effects that appear between consecutive  
11 manufacturing/assembly stations.  
12

13 The system performance of serial manufacturing lines and assembly manufacturing systems can be  
14 evaluated by discrete-event simulation (Law and Kelton 1991) or via queuing-model techniques (for example  
15 see Manitz (2007) and Matta et al. (2005)). In this paper we use simulation to evaluate the major performance  
16 measures of the system such as average throughput, average WIP and average level of backordered demands.  
17 Pull production control policies are heuristics that are characterized by a small number of control parameters  
18 that assume integer values. The performance of a system operating under a certain pull type control policy is  
19 significantly affected by parameter selection. Therefore, it is obvious that control parameter selection is a task  
20 of grave importance when it comes to the design of a pull type manufacturing system. It is reasonable that we  
21 would like to set the control parameters of each policy when operating in the same environment to their best  
22 values in order to gain insight regarding their behavior and proceed to performance comparisons. The approach  
23 that we propose and probe for the problem of control parameter optimization is that of *simulation optimization*,  
24 a practice that is rapidly becoming one of the most wide-spread tools in the field of complex systems  
25 design/analysis. Indicatively, we refer the reader to Swisher et al. (2000) for a survey on simulation  
26 optimization bibliography. In this paper, we interface the simulation model of the system we wish to examine  
27 with a genetic algorithm. Related work can be found in Dengiz and Alabas (2000) as well as in Spinellis and  
28 Papadopoulos (2000) where simulation is used in conjunction with tabu search and simulated annealing  
29 respectively, in order to determine the optimum design parameters of manufacturing/inventory systems. Finke  
30 *et al.* (2007) also use tabu search to solve a variation of the classic flow shop system where machines are  
31 organized in tiers. The choice of a genetic algorithm as the optimization method was made on the basis of the  
32 prevalent, but not as well-studied in the literature as one might expect conception, that genetic algorithms are  
33 inherently well-suited for “noisy” environments. Due to the fact that the objective function that we wish to  
34 maximize is stochastic as its values are generated by the simulation model we use resampling, i.e., performing  
35 multiple evaluations of the same parameter vector and using the mean of these evaluations as the fitness  
36 measurement of this individual, a practice discussed by Fitzpatrick and Grefenstette (1988) and Hammel and  
37 Bäck (1994).  
38  
39

40 The remaining material of this paper is structured as follows. Sections 2 -2.1 give a brief description of six  
41 important pull production control policies for serial manufacturing lines. Sections 3-3.5 are devoted to the  
42 presentation of the Kanban, Base Stock, CONWIP, CONWIP/Kanban Hybrid and Extended Kanban control  
43 policies for assembly manufacturing systems. In sections 4-4.1 we discuss the main aspects of the simulation  
44 optimization methodology that we followed and namely, the construction of the simulation model, the formal  
45 definition of the parameter optimization problem and issues concerning the genetic algorithm that was used.  
46 We report our findings from the simulation experiments that we conducted for two serial line and two assembly  
47 system cases in sections 5-5.2. Finally, in section 6 we state our concluding remarks and point to possible  
48 directions for future research.  
49

## 50 51 **2. System description of a serial manufacturing line**

52  
53 A typical manufacturing line produces a single product type and consists of a number of manufacturing stations  
54 with intermediate buffers. Raw materials undergo operations such as machining, forming etc., and are gradually  
55 converted into final products to be delivered to customers. Raw materials are assumed to be available to the  
56 system at any time, so the first machine is never starved. Customer demands that arrive to the system request  
57 the release of one finished part from the finished goods buffer. Whenever a demand arrives to the system it is  
58 immediately satisfied from the finished parts inventory. If there are no parts available in the last buffer the  
59 demand is backordered. Manufacturing facilities have the ability to work on only one part at a time during a  
60 manufacturing cycle. As soon as a stage  $i$  part is manufactured, it is placed in the output buffer of that stage. A  
control policy determines when a stage  $i$  finished part is to be released from the output buffer of that stage to

the next manufacturing facility. The finished stage  $i$  part is released to the next stage provided that the stage  $i + 1$  manufacturing facility is available at that time. There is no delay in material handling between stages. All machines have random production time, time between failures and repair time. Time intervals between demand arrivals are also stochastic. Each pull production control policy coordinates the release of parts from one stage to another in a different way. In other words, the timing with which the information of a demand arrival is transmitted from the last production stage to all of the upstream stages is policy – dependent. The following sections briefly explain the way that the Kanban, Base Stock, CONWIP, CONWIP/Kanban Hybrid, Extended Kanban and Generalized Kanban control policies for serial lines operate. A serial line consisting of three manufacturing stations is depicted in Figure 1.

*Figure 1 should be placed about here*  
**Figure 1.** Three station manufacturing line

## 2.1 Base Stock control policy

A Base Stock manufacturing line is completely described by  $N$  parameters, the base stock levels  $B_i$  of each production station,  $i = 1, 2, \dots, N$ , where  $N$  is the number of the system's manufacturing stations. The  $B_i$  parameters correspond to the number of parts that exist in the system's buffers at the time the system is in its initial state that is before any demands have arrived to the system. This control policy operates as follows. When a demand arrives to the system it is immediately transmitted to every manufacturing station, authorizing it to start working on a new stage  $i$  part. Base Stock has the advantage of reacting rapidly to incoming demand, with the drawbacks of providing no upper bounds for the system's buffer levels, as well as the relatively loose coordination of the manufacturing stages.

## 2.2 Kanban control policy

A Kanban manufacturing line's control parameters are the production authorizations  $K_i$  of each station,  $i = 1, 2, \dots, N$ . The Kanban policy offers very tight synchronization between the various production stations of the system. The  $K_i$  parameter corresponds to the maximum number of parts that are allowed in station  $i$  (manufacturing facility – output buffer). Production station  $i$  is authorized to start working on a new part as soon as a finished station  $i$  part is released from its output buffer. The information of a demand arrival is transmitted from the last manufacturing station to the first one station – by – station. If there is a point where a part is not available this transmission is interrupted. Due to this mechanism of back-propagating the demand arrival information, Kanban systems have relatively long response times to customer orders.

## 2.3 CONWIP control policy

CONWIP is an abbreviation for CONstand Work In Process. According to this policy the total number of parts that exist in the system (Work In Process) can never exceed a certain level, which is the single control parameter of the CONWIP policy. All machines in a CONWIP line are authorized to produce whenever they have this ability (they are operational and have a raw part to work on) except the first one. The first machine of the system is authorized to start working on a new part as soon as a unit from the finished parts buffer is released to a customer.

## 2.4 Generalized Kanban and Extended Kanban control policy

These two control policies combine the merits of Base Stock and Kanban as they react rapidly to the arrival of demands and effectively control the WIP at the same time. They are described by two parameters per stage, the base stocks  $B_i$  and the production authorizations  $K_i$ , ( $K_i > B_i$ ), that are borrowed from the Base Stock and Kanban policies respectively. The finite number of production authorizations guarantees that the system's inventories will not exceed the pre – defined levels, but the stage coordination here is not as tight as in Kanban. A station can be granted a production authorization even if a part is not released from its output buffer. For a



detailed description of the way Generalized Kanban and Extended Kanban operate the reader is referred to Liberopoulos and Dallery (2000), Dallery and Liberopoulos (2000) and Buzacott and Shanthikumar (1992).

## 2.5 CONWIP/Kanban Hybrid

A CONWIP/Kanban Hybrid system, as implied, operates under a combination of the CONWIP and Kanban control policies. Departure of a finished part from the system authorizes the first stage to allow a new raw part to enter the system. All stages except the last one have a finite number of production authorizations  $K_i, i = 1, 2, \dots, N - 1$ . Stage production authorizations  $K_i$  and the total WIP that is allowed in the system are CONWIP/Kanban Hybrid's control parameters.

## 3. System description of a manufacturing system with assembly stations

In this research paper we examine manufacturing systems with assembly stations that produce a single type of final products. A manufacturing station has only one immediate upstream system station and in its turn, feeds only one subsequent manufacturing or assembly station. On the other hand, an assembly station has two or more immediate upstream stations that supply it with the component parts that will merge in order to produce a single part. An assembly station also gives its finished parts to only one immediate downstream station. We limit ourselves to the case where component parts are released in an assembly station *simultaneously* (see Chaouiya et al. (2000) and Di Mascolo and Dallery (1996) for another variant). This implies that a component part that is stored in output buffer  $i$  and has the authorization to enter an immediately downstream assembly station will have to wait until all the other components become available too. Since this part has not exited buffer  $i$  it is incorporated in the estimation of buffer's inventory level. An unlimited supply of raw materials is assumed for this type of system too. Customer demands are satisfied instantaneously from the finished goods stock. If there are no parts available in the last buffer the demand is backordered. Manufacturing facilities and assembly stations have the ability to work on only one part at a time during a manufacturing cycle. As soon as a stage  $i$  part is finished, it is placed in the output buffer of that stage. A control policy determines when a station  $i$  finished part is authorized to be released from the output buffer of that station to the next manufacturing/assembly facility. All stations in the system have random operation time, time between failures and repair time. Time intervals between demand arrivals are also stochastic. The following sections describe the Kanban, Base Stock, CONWIP, CONWIP/Kanban Hybrid and Extended Kanban control policies for assembly manufacturing systems. The systems are depicted as queuing networks with synchronization stations, a notion borrowed by Liberopoulos and Dallery (2000). In our opinion this is the most illustrative and easy to comprehend modeling framework. A synchronization station with four input queues and three outgoing customers is shown in Figure 2.

*Figure 2 should be placed about here*

**Figure 2.** Synchronization station with four input queues and three outgoing customers

The perpendicular line in Figure 2 symbolizes a server with instant service times that is fed by queues a, b, c and d. As soon as there is at least one customer in each of the input queues, these customers enter the server and instantaneously, one customer of type e, one of type f and one of type g exit the server. For additional details the reader is referred to Liberopoulos and Dallery (2000). For simplicity reasons, all systems in the following sections consist of two manufacturing stations and one assembly station, however everything which holds for the dynamics of each control policy for this small system can be easily extended for larger systems.

### 3.1 Base Stock control policy for assembly systems

Figure 3 represents the model of an assembly system controlled by the Base Stock policy with two manufacturing stations and one assembly station. In our model  $S_i, i = 1, 2, 3$ , are the systems manufacturing/assembly stations while queues  $O_i, i = 1, 2, 3$  represent the output buffers of the corresponding stations. By  $O_0$  and  $O_0'$  we denote the raw materials storing facilities. By assumption these two queues cannot be totally empty at any time. Queues  $D_i, i = 1, 2, 3$ , contain the demands for a unit of a station  $i$  finished part

whereas queue  $D_4$  contains the customer demands for final products. Finally,  $I_i, i = 1, 2, 3$ , stands for the input buffer of station  $i$ . *Input buffers are not physical spaces in our model*, they are merely used for illustrative reasons, and to be more specific, to flag a change in the state of finished station  $i-1$  parts. When a part "moves" from output buffer  $O_{i-1}$  to input buffer  $I_i$  it means that it has been authorized by the *control policy* to be processed in station  $i$  as soon as this becomes possible, i.e. the station might be occupied by another part or it could be under repair. Through the use of these virtual buffers we explicitly show the timing with which a station is given the order to start working on a new part regardless of the fact that this station could or could not carry out this order at that time. Material flow is symbolized by continuous lines while the flow of information is shown by dashed lines. Initially, all queues in the model are empty except of the raw material buffers  $O_0$  and  $O_0'$ , and the output buffers  $O_i, i = 1, 2, 3$  that contain  $B_i, i = 1, 2, 3$  stage  $i$  finished parts. The integers  $B_i, i = 1, 2, 3$  that will be hereafter called as the *base stocks* of the stations are the system's only control parameters.

*Figure 3 should be placed about here*

**Figure 3.** Base Stock assembly system with two manufacturing stations and one assembly station

As shown in Figure 3, when a demand arrives to the system it is immediately transmitted to every manufacturing/assembly station, authorizing it to start working on a new station  $i$  part. Note that there must be at least one part in each of the queues  $O_1, O_2$  and  $D_3$  in order for the assembly station  $S_3$  to start processing a new station 3 part (simultaneous assembly systems).

### 3.2 Kanban control policy for assembly systems

A Kanban system with two manufacturing stations in parallel that feed an assembly station is illustrated in Figure 4.  $S_i, i = 1, 2, 3$ , stands for the system's manufacturing/assembly station  $i$  while queues  $OK_i, i = 1, 2, 3$  represent the output buffers of the corresponding stations 1, 2, 3 and queues  $O_0$  and  $O_0'$  are the raw material buffers. Queues  $DK_i, i = 1, 2, 3$ , contain station  $i$  production authorizations (kanbans). Queue  $D_4$  contains the customer demands for final products and finally queues  $I_i, i = 1, 2, 3$ , represent the input buffers of stations  $i = 1, 2, 3$ . We remind that input buffers are not physical areas of storage (see section 3.1). When the system is in its initial state, that is before any demands have arrived to the system, all queues in the model are empty except of the raw material buffers  $O_0$  and  $O_0'$ , and the output buffers  $OK_i, i = 1, 2, 3$  that contain  $K_i, i = 1, 2, 3$  stage  $i$  finished parts. Each station in the system has a fixed number of production authorizations or kanbans which are the system's control parameters. In other words, the inventory in all of the system's output buffers  $OK_i, i = 1, 2, 3$  cannot exceed the predefined levels. The kanbans of the stations  $S_i, i = 1, 2, 3$  are equal to the numbers  $K_i, i = 1, 2, 3$  of parts that exist in the output buffers initially.

*Figure 4 should be placed about here*

**Figure 4.** Kanban assembly system with two manufacturing stations and one assembly station

The material and information flows are not independent, as they are in the Base Stock policy. The arrival of a demand to the system "pulls" a part from the output buffer of station  $S_3$  and sends a kanban to queue  $DK_3$ . The kanban in queue  $DK_3$  authorizes the simultaneous release of one part from buffers  $OK_1$  and  $OK_2$  while in the same time, one kanban is sent to queue  $DK_1$  and one to queue  $DK_2$ . This way the information of a customer arrival is propagated backwards through the release of kanbans. If at some point of the system, there is an empty output buffer, then this propagation is ceased. It should have become clear by now that the Kanban control policy for assembly systems offers very tight coordination between the system's manufacturing/assembly stations.



### 3.3 CONWIP control policy for assembly systems

Figure 5 presents the model of CONWIP assembly system with two manufacturing stations and one assembly station.  $S_i, i = 1, 2, 3$ , is the system's manufacturing/assembly station  $i$  while queues  $O_i, i = 1, 2, 3$  represent the output buffers of the corresponding stations. We denote the raw materials buffers that, by assumption, can never be depleted as  $O_0$  and  $O_0'$ . CONWIP type production authorizations are placed in queues  $DC_1$  and  $DC_2$ . Queue  $D_4$  contains the customer unsatisfied demands and  $I_i, i = 1, 2, 3$ , stands for the input buffer of station  $i$ . The initial state of the CONWIP assembly system is the following: the output buffers contain  $B_i, i = 1, 2, 3$ , finished station  $i$  parts (base stocks) and all of the remaining queues in the system are empty. The system's control parameters are the maximum number  $C$  of parts allowed in the finished products buffer  $O_3$  (CONWIP type production authorizations), and the base stocks  $B_i, i = 1, 2, 3$ .

*Figure 5 should be placed about here*

**Figure 5.** CONWIP assembly system with two manufacturing stations and one assembly station

All stations in a CONWIP assembly system are authorized to produce whenever they have this ability except of the ones that are fed by the raw materials buffers. In Figure 5 these stations are  $S_i, i = 1, 2$ . These stations are allowed to start processing a new part by the moment that they are granted a CONWIP type production authorization (queues  $DC_1$  and  $DC_2$ ). Each one of the queues  $DC_1$  and  $DC_2$  is sent one CONWIP type production authorization as soon as one product exits buffer  $O_3$ . The WIP in assembly systems controlled by the CONWIP policy tends to accumulate in the finished products buffer.

### 3.4 CONWIP/Kanban Hybrid for assembly systems

A CONWIP/Kanban system with two manufacturing stations in parallel that feed an assembly station is illustrated in Figure 6. In this Figure, the  $S_i, i = 1, 2, 3$ , are the systems manufacturing/assembly stations, queues  $OK_i, i = 1, 2$ , represent the output buffers of the corresponding stations 1, 2, and queues  $O_0$  and  $O_0'$  are the raw material buffers. Queues  $DK_i, i = 1, 2$ , contain station  $i$  (Kanban type) production authorizations, while queues  $DC_i, i = 1, 2$ , contain CONWIP type production authorizations. Queue  $D_4$  contains the customer demands for final products, queue  $O_3$  the finished products stock and finally queues  $I_i, i = 1, 2, 3$ , represent the input buffers of stations  $i = 1, 2, 3$ . Initially, queues  $OK_i, i = 1, 2$ , contain  $K_i, i = 1, 2$ , finished parts and the finished goods buffer  $O_3$  contains  $B_3$  final products. All the other queues except  $O_0$  and  $O_0'$  (raw material buffers) are empty. As we have mentioned in former sections the system has an unlimited supply of raw materials.

*Figure 6 should be placed about here*

**Figure 6.** CONWIP/Kanban Hybrid assembly system with two manufacturing stations and one assembly station

In a CONWIP/Kanban Hybrid assembly system all stations *except the last one* have a fixed number  $K_i$  of station  $i, i = 1, 2, \dots, n-1$ , production authorizations ensuring that the inventory in these stations along with the corresponding buffers will not grow unboundedly. The system's last station has the authorization to produce whenever it can. The stations which are fed by the raw materials buffers are allowed to start processing a new part by the moment that they are granted a CONWIP type production authorization (queues  $DC_1$  and  $DC_2$ ) in addition to a kanban type production authorization (queues  $DK_1$  and  $DK_2$ ). Queues  $DC_1$  and  $DC_2$  receive a CONWIP type production authorization as soon as a finished product exits buffer  $O_3$ . The control parameters of

the system are the station  $i$  production authorizations  $K_i$ ,  $i=1,2,\dots,n-1$ , the number  $C$  of CONWIP type production authorizations plus the base stock  $B_3$  of the last station.

### 3.5 Extended Kanban control policy for assembly systems

An Extended Kanban assembly system with simultaneous release of components (see Chaouyia et al. for another variant) in the assembly stations is depicted in Figure 7. Queues  $S_i$ ,  $i=1,2,3$ , are the systems manufacturing/assembly stations while queues  $OK_i$ ,  $i=1,2,3$  represent the output buffers of the corresponding stations 1,2,3 and queues  $O_0$  and  $O_0'$  are the raw material buffers. Queues  $FK_i$ ,  $i=1,2,3$ , contain station  $i$  production authorizations (kanbans). Queues  $D_i$ ,  $i=1,2,3$ , contain demands for station  $i$  finished products, queue  $D_4$  holds the customer demands for final products and finally queues  $I_i$ ,  $i=1,2,3$ , represent the input buffers of stations  $i=1,2,3$ . We remind that input buffers are not physical areas of storage (see section 3.1). Dashed lines symbolize information flow while solid lines represent material flow. (Here kanban authorizations/cards are treated as physical entities.). In the system's initial state,  $B_i$  station  $i$  parts are stored in queues  $OK_i$ ,  $i=1,2,3$  and raw materials are placed in queues  $O_0$  and  $O_0'$ . Queues  $FK_i$ ,  $i=1,2,3$ , contain  $K_i - B_i$ ,  $i=1,2,3$ , kanban authorizations and all other queues in the system are vacant. The control parameters of the Extended Kanban system are the total station  $i$  production authorizations  $K_i$  and the base stocks  $B_i$ ,  $i=1,2,3$ . As one may see, in the extreme case of  $K_i = B_i$ ,  $i=1,2,3$ , the system is reduced to simple Kanban assembly system.

*Figure 7 should be placed about here*

**Figure 7.** Extended Kanban assembly system with two manufacturing stations and one assembly station

The information of a customer arrival is transmitted instantly to all stations, a property borrowed from the Base Stock policy. The Extended Kanban control policy has a fixed number of kanbans in order to constrain the WIP, but in the same time offers more loose coordination between manufacturing/assembly stations than the original Kanban policy. By examining Figure 7, it becomes clear that a station can be granted an authorization to start working on a new part even if no parts have exited its output buffer.

## 4. Simulation optimization of serial manufacturing lines and assembly manufacturing systems

We use discrete-event simulation to evaluate the three major performance measures of serial manufacturing lines and assembly systems which are average throughput (number of final products delivered to customers per time unit), average WIP (WorkInProcess) and average level of backordered demands. Pull production control policies are simple heuristics that are characterized by a small number of control parameters that assume integer values. We seek to set the control parameters of each policy when operating in the same environment to the values that maximize a properly defined objective function in order to proceed to performance comparisons. To serve this purpose we make use of simulation optimization which, according to Hall and Bowden (1998), is "*... the practice of linking an optimization method with a simulation model to determine appropriate settings of certain input parameters so as to maximize the performance of the simulated system*". In this paper we interface the simulation model of the serial lines and assembly systems we wish to examine with a genetic algorithm.

### 4.1 Simulation of manufacturing serial lines and assembly systems

Discrete – event simulation, (Law and Kelton 1991), is the modeling tool that we used in order to simulate the operation of serial lines and assembly systems. According to this methodology, the simulation program is constructed on the basis of the discrete events that are possible to take place in a given system. In the case of an assembly system or a manufacturing serial line, we have the following events.

1. Customer demand arrival
2. Completion of a new station  $i$  part

3. Failure of station  $i$
4. Completion of a repair in station  $i$

Since we examine pull type production control policies the event that appears first in our model is a demand arrival. The discrete – event simulation algorithm of a manufacturing serial line or an assembly system that operates under a pull production control policy is presented in Listing 3.1.

Initialization. Manufacturing system's topology, service times, failure times, demand times etc., duration of the simulation  $T_{sim}$ ,  $t = 0$ ,  $t_b = 0$ , where  $t$  is the current simulation time and  $t_b$  the previous simulation event's time of appearance.

WHILE (  $t < T_{sim}$  )

Find minimum time of occurrence from the next events listing  
 Event execution. The system's state variables are adjusted with the use of its state equations  
 Controller call. Determine whether the station  $i$  will be operational or idle during the next simulation cycle.  
 Where necessary, update the next events time of occurrence

**Listing 3.1.** Manufacturing system discrete – event simulation algorithm

## 4.2 Optimization problem – Objective function

The mathematical formulation of the parameter optimization problem for serial lines and assembly systems controlled by pull production control policies is given below. Let  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$ ,  $x_i \in \mathbf{Z}$ , be the control parameter vector of some pull production control policy. For example,  $\mathbf{x}$  could be the station  $i$  production authorizations (kanbans) in a Kanban system, the initial buffer levels in a Base Stock system etc. The objective is to find the control parameter values  $\mathbf{x}$  that maximize the expected value of the stochastic objective function  $F(\mathbf{x}, \omega)$ , where  $F(\mathbf{x}, \omega)$  is the derived profit from the system's operation per time unit.

$$\text{Maximize: } E[F(\mathbf{x}, \omega)] \quad (1)$$

$$\text{subject to: } x_i \in \mathbf{Z}, i = 1, 2, \dots, n$$

$\omega$  is used to denote the stochastic nature of  $F$ . The evaluation of function  $F$  is the result of a simulation experiment. The value of  $F$  is calculated according to 2.

$$F = pTH - \sum_{i=1}^N h_i \bar{H}_i - b\bar{B} \quad (2)$$

$F$  is the derived profit from the system's operation per time unit,  $p$  the profit from a finished part's sale to a customer,  $TH$  the system's average throughput which is calculated by dividing the number of parts the system's last station produced by the simulation time  $T_{sim}$ .  $h_i$  stands for the cost of storing one part in output buffer  $i$  per time unit while  $\bar{H}_i$  is the average inventory level in output buffer  $i$ . The cost of one unsatisfied demand per time unit is denoted as  $b$  and finally,  $\bar{B}$  is the system's average number of backordered demands. Repeating a simulation for the same value of  $\mathbf{x}$  would result in different values of  $F$  due to the stochasticity of the simulation model that is denoted by  $\omega$ .

## 4.3 Genetic algorithm

Genetic algorithms are stochastic iterative search techniques that do not require any kind of information on the objective function's structural properties as the search direction is determined solely by means of objective function evaluation. A genetic algorithm generates a population of individuals-solutions at each iteration. The next population is created by using computations that involve stochastic selections. Due to the fact that the objective function that we wish to maximize is stochastic we use *resampling*, a practice that is discussed in Beyer (2000), Fitzpatrick and Grefenstette (1988) and Hammel and Bäck (1994). For a given

individual/parameter vector  $\mathbf{x}$  we measure  $F(\mathbf{x}, \omega)$   $m$  times and use the mean of these measures as the expected value of  $F(\mathbf{x}, \omega)$ .

$$E[F(\mathbf{x}, \omega)] = \frac{1}{m} \sum_{i=1}^m F(\mathbf{x}, \omega)_i, \quad \mathbf{x} = \text{const} \quad (3)$$

This technique is perhaps the most straight-forward when it comes to evaluation of stochastic functions but it can be very expensive computationally as the calls to the simulation model increase by a factor of  $m$ . On the other hand, the higher the value of  $m$  is, the more accurate the evaluation of  $E[F(\mathbf{x}, \omega)]$  becomes. Another factor that affects the quality of the estimate of  $E[F(\mathbf{x}, \omega)]$  is the simulation time  $T_{sim}$  for which we run the model. We determine the values of  $m$  and  $T_{sim}$  by trial and error in order to have a fairly accurate estimate of  $E[F(\mathbf{x}, \omega)]$  without exhaustively long simulation experiments. This is a very delicate point as estimates that are not sufficiently accurate may cause the algorithm to incorrectly interpret improvements caused by the stochasticity of the simulation model as meliorations caused by a change in the parameter set or fail to detect better solutions that cause minor improvements in the objective function. The fact that we are dealing with a stochastic objective function must be taken into consideration when setting the stopping criterion too. Our algorithm terminates when a number of  $g$  successive generations do not yield more than a  $p\%$  improvement on the objective function. It is essential to determine roughly through experimentation the range of  $E[F(\mathbf{x}, \omega)]$ s that a given  $\mathbf{x}$  will produce in order to set parameter  $p$  appropriately. This way we ensure that the algorithm will not be trapped in an endless loop and eventually terminate at some point. In cases where we suspect that a more thorough search of the decision space is needed, as a complementary stopping criterion, we prevent the algorithm from stopping before it has produced a minimum number of generations. The remaining parameters of the genetic algorithm are the standard ones of population size  $s$ , crossover probability  $p_{cross}$  and mutation probability  $p_{mut}$ .

## 5. Results – Discussion

### 5.1 Serial manufacturing line results

We examined a four – stage manufacturing line with equal operation times. Machines operate with service rates which are normally distributed random variables with mean 1.0 parts/time unit and s.d. 0.1 ( $R_p \sim N(1.0, 0.1)$ ). Repair to failure times are exponentially distributed with mean 1000.0 time units. Failures are operation dependent. Repair times are also assumed exponential with a MTTR of 10.0 time units. We focused our attention on two cases regarding the demand arrival process; one where the system has to satisfy a moderate demand for finished goods and one where the demand arrival rate is higher and close to the maximum throughput of the given system. In the first case, times between two successive customer arrivals are exponential random variables with mean 1.66 time units (0.6 customers per time unit), while in the second case the mean time between customer arrivals is equal to 1.25 time units (0.8 customers per time unit). Simulation time is set to 30000.0 time units. The sale of one finished part to the customers results in a 120,0 units profit. Inventory costs for storing one part per time unit in buffer  $i$  are shown in Table 1.

*Table 1 should be placed about here*

Linear backorder costs are incurred at a rate of 10.0 whenever there are backorders. The authors conducted a series of experiments in an attempt to come up with the most suitable genetic algorithm parameter set for this specific problem. The genetic algorithm's parameters that appeared to yield the best results were: population size = 40, crossover probability = 0.5, mutation probability 0.05, parameter  $p = 5\%$  and  $g = 40$  (see section 4.3). Each individual-parameter vector was evaluated 30 times, ( $m=30$ ). In some cases we used the complementary stopping criterion in order to achieve a more thorough search of the solution space with the minimum number of generations to be produced by the genetic algorithm set to 100. The best parameters of each policy for the first simulation case are presented in Table 2. We denote base stock in buffer  $i$  by writing  $B_i$  and station  $i$  production authorizations by  $K_i$ . CONWIP type production authorizations are represented by the symbol  $C$ . Figure 8 shows the fitness of the best solution found by the genetic algorithm in each iteration for every control policy versus the number of generations.



*Figure 8 should be placed about here*

**Figure 8.** GA max  $E[F]$  plots - serial manufacturing lines – demand rate = 0.6

Table 3 presents the average WIP distribution through out the system and the backordered demands average together with the corresponding expected values of  $F$ . These figures were calculated by averaging the output of 30 executions of each policy's simulation model, where every model was allowed to run for 300,000.0 time units, that is ten times the simulation time used for fitness evaluation in the genetic algorithm.

*Table 2 should be placed about here*

*Table 3 should be placed about here*

By reading the results presented in Table 2 we observe that the genetic algorithm converged to relatively low values for the control parameters of all pull production control policies. Especially for the case of the Kanban control policy, the  $K_i$ s were set to the lowest possible values these parameters can assume (if a station in a Kanban system has zero production authorizations then this station would never produce a single part). Note that by increasing the number of kanbans, base stocks or CONWIP type authorizations, the average throughput of the serial line increases but so does the WIP along with the related costs. The major cost factors in our serial line simulation scenario are the storage costs, and primarily the inventory level in the system's last buffer, as the average of backordered demands is weighted moderately. Given the production rates of the machines and the rate of incoming orders, we could infer the system can easily handle customer arrivals without having to maintain large safety stocks and this is depicted in the parameter sets that the algorithm converged to. The characteristic attribute of the Kanban policy which is tight coordination between manufacturing stations is evident in the first row of Table 3. We observe that the WIP is distributed practically evenly between buffers 1 to 4. However the average WIP in buffers 1, 2 and 3 are the highest among all control policies. The Base Stock control policy achieves a significantly leaner operation in expense of the lower service level. In a system that operates under the CONWIP policy all the machines except the first one have the perpetual authorization to produce whenever they can. There is a constant flow of parts through the system that ends in the finished goods buffer where parts are accumulated, and that is why the average finished products inventory is the highest of all control policies. As a consequence, parts are not stalled in intermediate buffers due to e.g. lack of production authorization which in turn results in very low average WIP levels in the first three buffers as it is evident when observing Table 3. The existence of additional station  $i$  production authorizations in the CONWIP/Kanban Hybrid policy does not alter the system's performance in a dramatic way, as one may observe in Table 3. The application of the Generalized Kanban control policy resulted in the highest profit rate in our simulation cases, while Extended Kanban ranked second. Both Generalized Kanban and Extended Kanban offer effective WIP control and maintenance of a high service rate. These two control policies have a finite number of production authorizations in order to prevent the system's Work In Process inventories from growing unboundedly but the stage coordination is not as tight as in Kanban. This is a result of the  $K_i - B_i, i = 1, 2, \dots, n$ , initially free kanban cards that Generalized Kanban and Extended Kanban make use of. Examination of the best parameters table (Table 2) shows that this technique allows the allocation of a greater number of production authorizations than simple Kanban that speed up the system's reaction to demand occurrences, while the base stocks approximate those used in the Base Stock system.

The best parameters of each policy found by the genetic algorithm for the second simulation case, (demand arrival rate=0.8 arrivals/time unit), are presented in Table 4. Figure 9 shows the fitness of the best solution found by the genetic algorithm in each iteration for every control policy versus the number of generations.

*Figure 9 should be placed about here*

**Figure 9.** GA max  $E[F]$  plots - serial manufacturing lines - demand rate = 0.8

Table 5 presents the average WIP distribution through out the system and the backordered demands average together with the corresponding expected values of  $F$ . Again, these figures were calculated by averaging the output of 30 executions of each policy's simulation model, where every model was allowed to run for 300,000.0 time units.

*Table 4 should be placed about here*



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

*Table 5 should be placed about here*

As we expected, the genetic algorithm selected higher values for the control policies' parameters, due to the need for increased safety stocks of intermediate and final products, caused by the higher demand rate. What we are primarily interested in, is to determine whether the increase of the demand rate modifies the ranking of the six production coordination mechanisms reported for the first simulation scenario. The Base Stock policy still manages to outperform the Kanban policy. An interesting point is that this happens for entirely different reasons in the two simulation cases. In the first, low demand rate case, the Base Stock policy outperforms Kanban due to its lower average inventories caused by the zero installation stock in the first three buffers while in the second, high demand rate case, the same happens but this time due to its approximately 43% lower level of backorders. A possible explanation for this is that the selection of the objective function's weights (holding cost of one stage  $i$  part per time unit, cost of one backorder per time unit) favors the Base Stock policy over the Kanban policy. The Generalized and Extended Kanban systems rank in the first two positions in both simulation cases giving strong indications that they can excel in various demand rate environments. Perhaps the most interesting observation that stems from this second series of optimization experiments is the striking improvement in the performance of the CONWIP and CONWIP/Kanban Hybrid control policies. More specifically, the CONWIP policy ranked last in the low demand simulation case, while in the high demand case, it exhibits a performance comparable to those of the two-parameter-per stage policies. The inherent characteristic of the CONWIP policy, which is uninterrupted flow of materials through the system and WIP accumulation in the finished goods buffer, is what renders it highly sensitive to demand fluctuations for these two simulation cases. However, notice that the CONWIP and the Hybrid policies would probably not perform that well in the high demand rate case if the serial line was not balanced, as it is in our study. For example, consider a situation where the bottleneck of the system is located, i.e. in the second machine due to fact that this machine is more susceptible to failure than the other machines. In that case, the WIP would tend to accumulate in the first buffer, leading to excessive storage costs.

## 5.2 Assembly system results

We examined an assembly system that consists of four manufacturing stations and three assembly stations. The topology of the system is shown in Figure 10.

*Figure 10 should be placed about here*

**Figure 10.** Assembly system topology

All stations operate with service rates which are normally distributed random variables with mean 1.0 parts/time unit and s.d. 0.01 ( $R_p \sim N(1.0,0.01)$ ). Repair to failure times are exponentially distributed with mean 1000.0 time units. Failures are operation dependent. Repair times are also assumed exponential with a MTTR of 10.0 time units. Similarly to the serial line case, we examined two scenarios regarding the demand; in the first scenario times between two successive customer arrivals are exponential random variables with mean 1.66 time units (0.6 customers per time unit), while in the second scenario the mean time between customer arrivals is equal to 1.25 time units (0.8 customers per time unit). Simulation time is set to 40000.0 time units. The sale of one finished part to the customers results in a 70.0 units profit. Inventory costs for storing one part per time unit in buffer  $i$  are shown in Table 6.

*Table 6 should be placed about here*

Linear backorder costs are incurred at a rate of 10.0 whenever there are backorders. Significant emphasis is given to customer service level and this is done through the inventory/backordered demands costs scheme. The cost per unsatisfied demand has a much larger impact on the objective function's value, than the cost of a part stored in buffer  $i$ . The parameters of the genetic algorithm were: population size = 30, crossover probability = 0.5, mutation probability 0.05, parameter  $p = 5\%$  and  $g = 40$  (see section 4.3). Each individual-parameter vector was evaluated 30 times, ( $m=30$ ). In some cases we used the complementary stopping criterion in order to achieve a more thorough search of the solution space with the minimum number of generations to be produced by the genetic algorithm set to 100. The best parameters for each policy for the low demand rate case are presented in Table 7. We denote base stock in buffer  $i$  by writing  $B_i$  and station  $i$  production authorizations by

$K_i$ . CONWIP type production authorizations are represented by the symbol  $C$ . Figure 11 shows the fitness of the best solution in each iteration found by the genetic algorithm for every control policy versus the number of generations.

*Figure 11 should be placed about here*

**Figure 11.** GA max  $E[F]$  plots - assembly systems – demand rate = 0.6

Table 8 presents the average WIP distribution through out the assembly system and the backordered demands average together with the corresponding expected values of  $F$ . These performance measures were calculated by averaging the output of 30 executions of each policy's simulation model, where every model was allowed to run for 300,000.0 time units.

*Table 7 should be placed about here*

*Table 8 should be placed about here*

In the assembly system of our simulation scenario component parts are released in an assembly station simultaneously (see sections 3.1-3.5). This means that a component part that is stored in its corresponding output buffer and has the authorization to enter the assembly station will have to wait until all the other components become available too. The simultaneous release of component parts in the assembly station can be treated as a part of the control policy or as a limitation posed due to technical reasons, i.e., the assembly operation cannot start until all components are gathered. Whichever the case may be, this required synchronization intensifies the queuing effects that appear between consecutive manufacturing/assembly stations. The Kanban and the Base Stock control policies rank in the two last places by the expected profit  $E[F]$  criterion. The average number of backordered demands that the Kanban policy achieves is the highest among all competing policies. Evidently, the mechanism of authorizing production in the system's workstations through the release of kanban cards has a negative effect on the system's response time to incoming customer orders. On the other hand, the Base Stock policy yields the lower average of backorders, in the expense of relatively poor WIP control. CONWIP is one of the simplest control policies but this does not prevent it from excelling in the task of controlling an assembly system. WIP is constrained effectively in all storage facilities except the last one, something which characterizes this policy, while in the same time, incoming orders are satisfied almost instantly (average of backorders=0.86). The CONWIP/Kanban control policy for assembly systems performs worse than the CONWIP policy regarding the WIP in the output buffers of the manufacturing stations 1, 2, 3 and 4. Far better results are obtained for output buffers 5 and 6, where the WIP is kept at the lowest levels among all competing policies throughout the simulation period. The Extended Kanban control policy yielded the highest expected profit rate in our assembly system simulation scenario, however given the stochasticity of the objective function we could argue that its performance is practically the same with that of the CONWIP mechanism. The statistics of this policy's performance displayed in Table 8 indicate that the existence of  $K_i - B_i, i = 1, 2, \dots, n$ , initially free kanban cards in the Extended Kanban assembly system allow the unhampered flow of information within the system's bounds which gives it the ability to react rapidly to demand occurrences without the need for excessive safety stocks.

The best parameters for each policy for the second simulation case, (demand rate =0.8) are presented in Table 9. Figure 12 shows the fitness of the best solution in each iteration found by the genetic algorithm for every control policy versus the number of generations.

*Figure 12 should be placed about here*

**Figure 12.** GA max  $E[F]$  plots - assembly systems – demand rate = 0.8

Table 10 presents the average WIP distribution through out the assembly system and the backordered demands average together with the corresponding expected values of  $F$ . These performance measures were calculated by averaging the output of 30 executions of each policy's simulation model, where every model was allowed to run for 300,000.0 time units.

*Table 9 should be placed about here*

*Table 10 should be placed about here*

1  
2 Unlike the serial line system examined in section 5.1 the ranking of the control policies is not altered  
3 significantly with the exception of the CONWIP and CONWIP/Kanban hybrid control policies which swap  
4 positions. However, the difference in the values of the objective function is marginal and could probably be  
5 attributed to the simulation itself. The reasons that we did not observe a change in the ranking of the policies  
6 can be found in the different inventory/backorder cost schemes which were used for the serial line and the  
7 assembly system. We remind that in the serial line case the major cost factor was the average inventory level of  
8 the last buffer while in the assembly system the most heavily weighted cost factor was the average number of  
9 backorders. It is interesting to highlight the differences between the parameter sets that the genetic algorithm  
10 selected for the two demand rate cases. With the exception of the CONWIP/Kanban hybrid policy, in all  
11 policies, the base stock in the last buffer is significantly lower in the moderate demand case than in the high  
12 demand case. The differences between the base stock levels of the two demand rate cases are substantially  
13 smaller as we move upstream. This indicates that the throughput, and as a consequence the number of  
14 backorders, is primarily affected by the initial inventory of the last buffer and that the impact of inventory in the  
15 system's throughput diminishes in upstream buffers.  
16  
17

## 18 **6. Conclusion and future research**

20 Effective production control is a topic that has attracted the research community's interest since many years as  
21 it is of great importance to any manufacturing system's competitiveness. In this paper we briefly present six  
22 important pull type production control policies that are used in manufacturing serial lines. Then we extend the  
23 use of five of them and namely Kanban, Base Stock, CONWIP, CONWIP/Kanban Hybrid and Extended  
24 Kanban to assembly systems with simultaneous release of components in the assembly stations. Each one of  
25 these control policies is completely described by a number of control parameters. Pull type control policies are  
26 simple heuristics so there are no significant difficulties in their implementation except that of the control  
27 parameter selection, a multi – dimensional optimization problem. As a measure of both the serial lines' and the  
28 assembly systems' performance we used a linear function of the system's average throughput, the average WIP  
29 and the average level of backorders. Simulation optimization is our approach to the parameter optimization  
30 problem. The optimization method that we used was a genetic algorithm that uses resampling, a choice that was  
31 dictated by the fact that the objective function we wanted to maximize was stochastic. We could argue that a  
32 genetic algorithm linked to a simulation model proves to be a valuable tool for the design of complex systems  
33 like the ones treated in this paper despite the fact that a considerable amount of fine-tuning is needed in order to  
34 determine appropriate settings for the genetic algorithm. We examined two serial manufacturing line simulation  
35 scenarios and two assembly system simulation scenarios. In all cases we used stochastic production times,  
36 times between failures, times to repair and demand. Based on the control parameter sets that we obtained using  
37 the genetic algorithm, we presented performance comparisons between the pull type control policies we  
38 presented. We could argue that the main qualitative attributes of all of the pull type control policies for serial  
39 lines seem to apply to systems that contain assembly stations too. The two-parameter-per stage policies,  
40 (Generalized and Extended Kanban), seem to be the more efficient ones regardless of the cost function weights,  
41 the demand rate and the existence or not of parallel machines in the system. The uncoupling of the information  
42 flow from the flow of parts in the system, offers the ability to opt for less base stock than the Kanban policy  
43 and achieves better response to demand occurrences. Our simulated data indicate that, when fine-tuned, the  
44 Generalized and Extended Kanban mechanisms can outperform less sophisticated mechanisms like the Kanban  
45 and the Base Stock. However, the increased number of parameters that need to be tuned is an important  
46 disadvantage that cannot be overlooked. A real-world manufacturing system can easily consist of, e.g. 20  
47 machines, which translates to 40 parameters that need to be optimized. This poses a significant computational  
48 burden especially if high precision results are required. The CONWIP and CONWIP/Kanban Hybrid policies  
49 can approximate the performance of the Generalized and Extended Kanban mechanisms for certain settings as  
50 shown in sections 5.1 and 5.2, while being computationally cheaper. On the other hand, these two policies can  
51 perform poorly in situations where the demand rate is moderate and the cost of final product inventory is high.  
52 Pull type control policies as efficient as they can be, still remain heuristic, sub – optimal control mechanisms  
53 that do not receive any feedback in terms of some kind of reward signal on the produce/do not produce  
54 decisions they make. Here is where the Artificial Intelligence field comes into scope. Development of new and  
55 efficient, AI based production controllers is a challenging but most promising research topic, and some work  
56 has already been done to this direction. Another possible alternative would be the analysis and simulation of  
57 different kinds of manufacturing systems such as a multi – product manufacturing lines, pull control  
58 mechanisms that utilize Advance Demand Information etc.  
59  
60

## References

- Beyer, H.G., 2000. Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanics and Engineering*, 186, 239-267
- Bowden, R.O. and Hall, J.D., 1998. Simulation optimization research and development. *Proceedings of the 1998 Winter Simulation Conference*, 1693-1698
- Buzacott, J.A. and Shanthikumar G.J., 1993. *Stochastic Models of Manufacturing Systems*. Pren. Hall.
- Buzacott, J.A. and Shanthikumar, G.J., 1992. A general approach for coordinating production in multiple cell manufacturing systems. *Production and Operations Management*, 1(1), 34-52
- Chaouiya, C., Liberopoulos, G. and Dallery, Y., 2000. The extended kanban control system for production coordination of assembly manufacturing systems. *IEE Transactions*, 32, 999-1012
- Dallery, Y. and Liberopoulos, G., 2000. Extended kanban control system: combining kanban and base stock. *IIE Transactions*, 32(4), 369 – 386
- Dengiz, B. and Alabas, C., 2000. Simulation optimization using tabu search. *Proceedings of the 2000 Winter Simulation Conference*, 805-810
- Finke, D.A., Medeiros, D.J. and Traband, M.T., 2007. Multiple machine JIT scheduling: a tabu search approach. *International Journal of Production Research*, 45(21), 4899-4915
- Fitzpatrick, J.M. and Grefenstette, J.J., 1988. Genetic algorithms in noisy environments. *Machine Learning: Special issue on genetic algorithms*, 3, 101-120
- Hammel, U. and Bäck, T., 1994. Evolution strategies on noisy functions. How to improve convergence properties. *Parallel Problem Solving from Nature*, 3, 159-168.
- Law, A. and Kelton, D., 1991. *Simulation Modelling and Analysis*. New York: McGraw – Hill
- Liberopoulos, G. and Dallery, Y., 2000. A unified framework for pull control mechanisms in multi – stage manufacturing systems. *Annals of Operations Research*, 93, 325 – 355.
- Di Mascolo, M. and Dallery, Y., 1996. Performance evaluation of kanban controlled assembly systems. *Symposium on Discrete Events and Manufacturing Systems of the Multiconference IMACS-IEEE/SMC CESA'96*
- Matta, A., Dallery, Y. and Di Mascolo, M., 2005. Analysis of assembly systems controlled with kanbans. *European Journal of Operational Research*, 166, 310-336
- Manitz, M., 2007. Queueing-model based analysis of assembly lines with finite buffers and general service times. *Computers and Operations Research*, doi: 10.1016/j.cor.2006.12.016
- Paternina – Arboleda, C. D. and Das, T. K., 2001. Intelligent dynamic control policies for serial production lines. *IIE Transactions*, 33, 65 – 77.
- Pyke, D.F. and Cohen, M.A., 1990. Push and pull in manufacturing and distributions systems. *Journal of Operations Management*, 9(1), 24-43
- Spearman, M.L., Woodruff, D.L. and Hopp, W.J., 1990. CONWIP: a pull alternative to Kanban. *International Journal of Production Research*, 28, 879 – 894.

1  
2 Spinellis, D.D. and Papadopoulos, C.T., 2000. A simulated annealing approach for buffer allocation in  
3 reliable production lines. *Annals of Operations Research*, 93, 373-384  
4

5 Swisher, J.R., Hyden, P.D., Jacobson, S.H. and Schruben, L.W., 2000. A survey of simulation optimization  
6 techniques and procedures. *Proceedings of the 2000 Winter simulation conference*, 119-128  
7

8 Takahashi, K., Myreshka and Hirotsu D., 2005. Comparing CONWIP, synchronized CONWIP and  
9 Kanban in complex supply chains. *International Journal of Production Economics*, 93-94, 25-40  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

For Peer Review Only



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

Figures

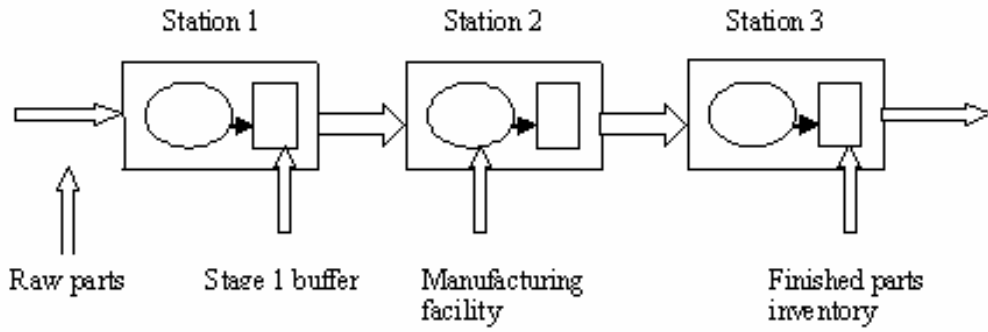


Figure 1. Three station manufacturing line

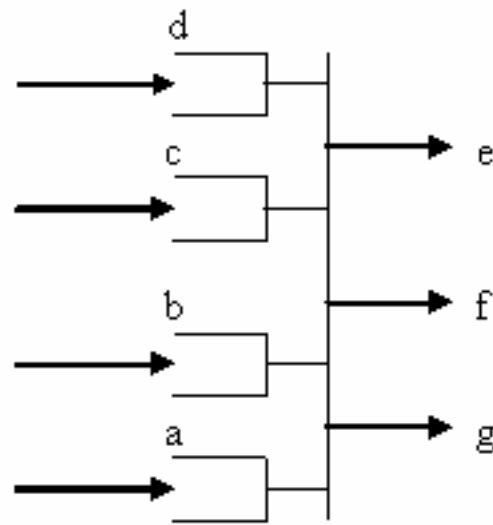


Figure 2. Synchronization station with four input queues and three outgoing customers

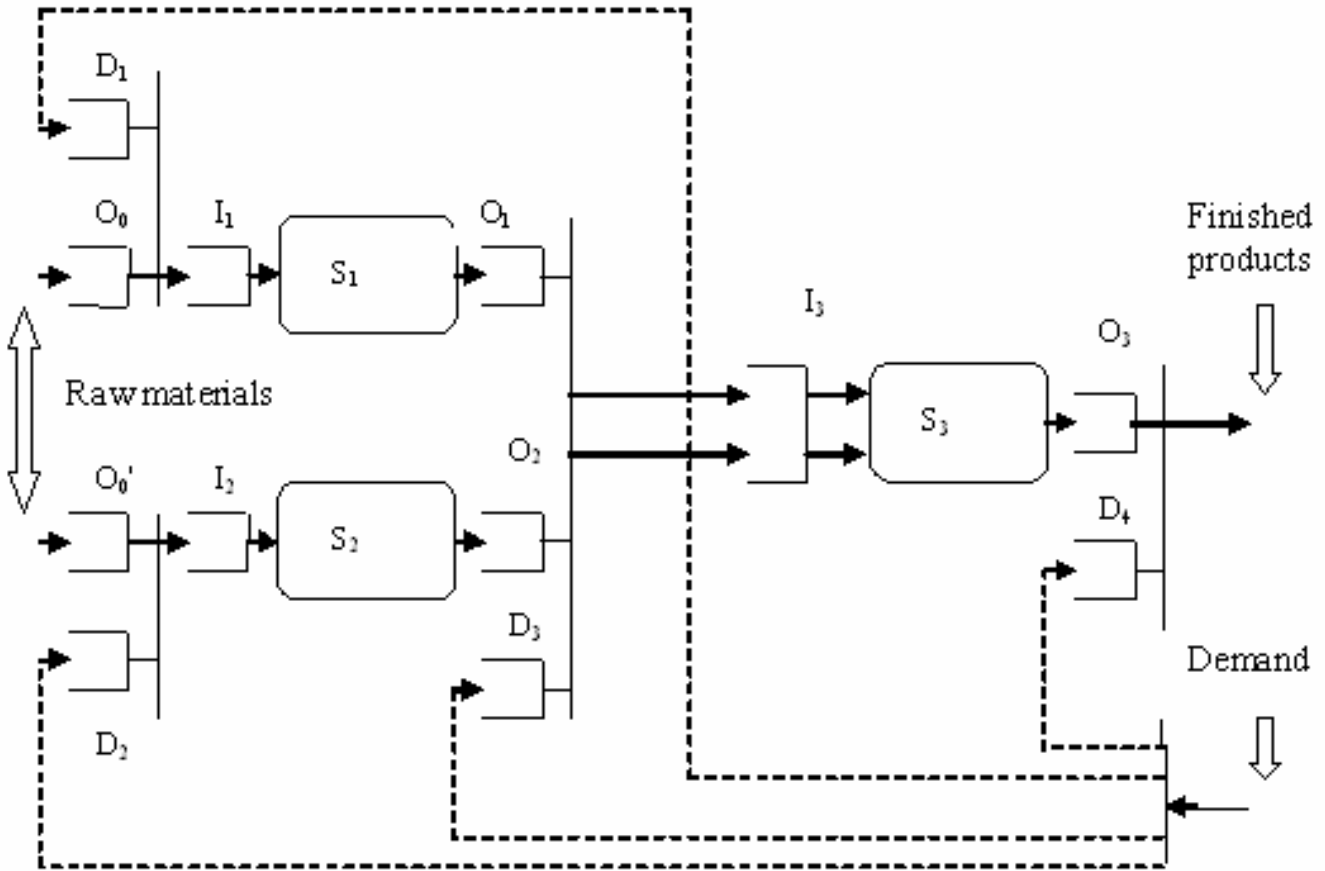


Figure 3. Base Stock assembly system with two manufacturing stations and one assembly station

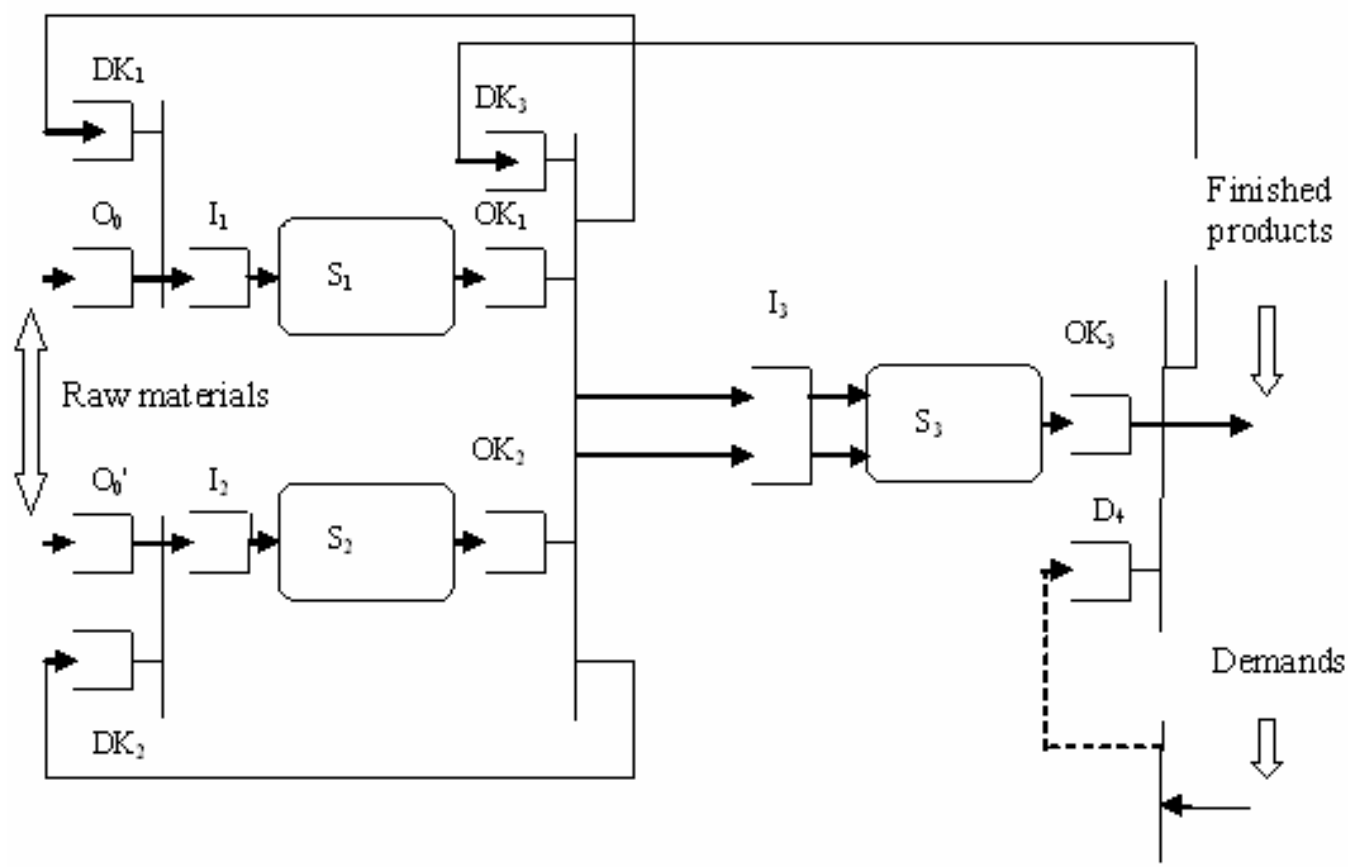


Figure 4. Kanban assembly system with two manufacturing stations and one assembly station

View Only

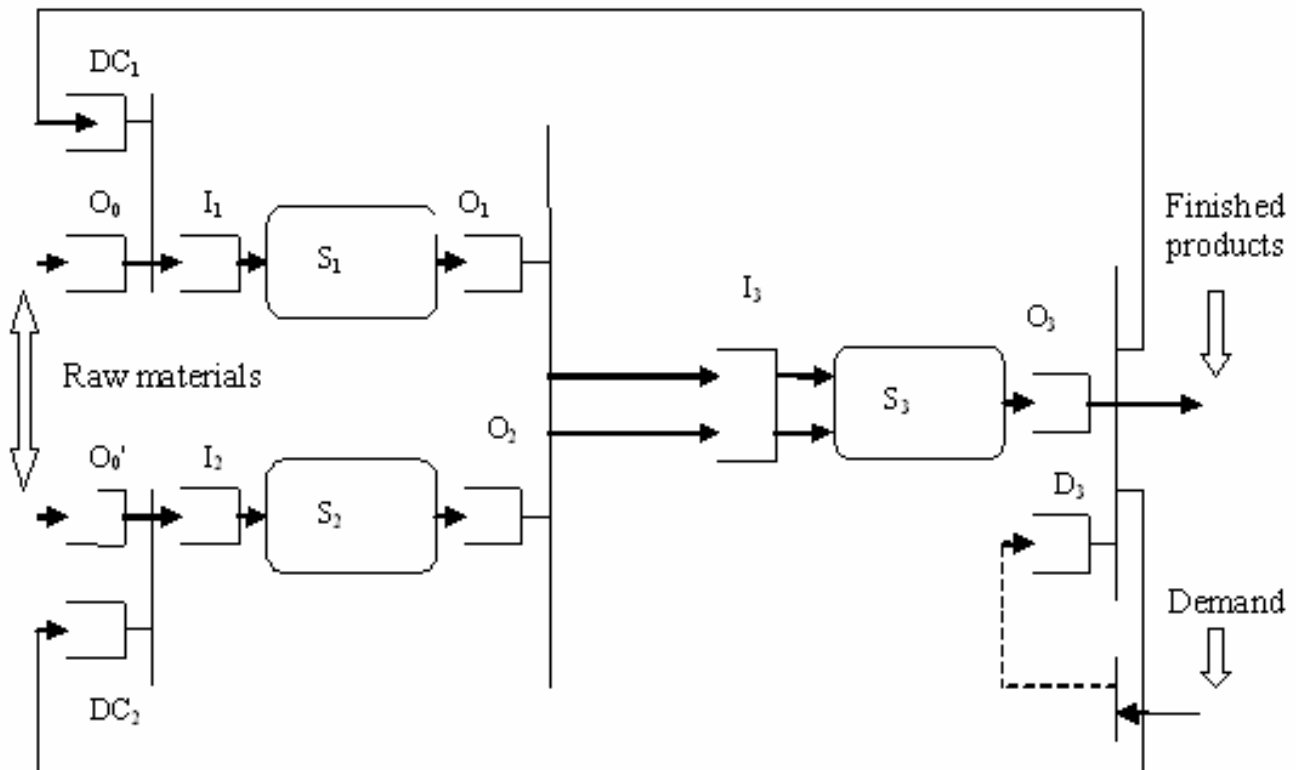


Figure 5. CONWIP assembly system with two manufacturing stations and one assembly station

Review Only

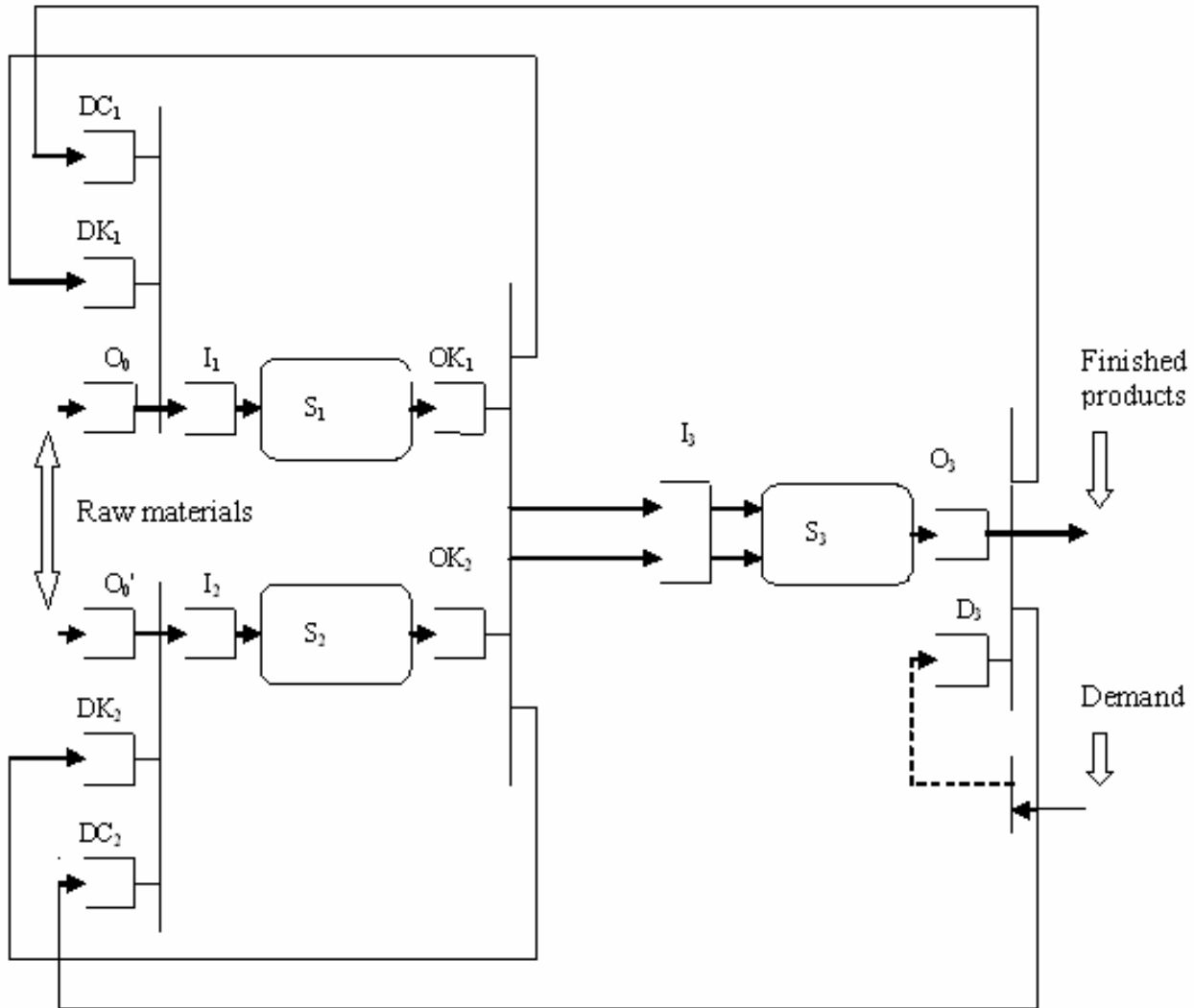


Figure 6. CONWIP/Kanban Hybrid assembly system with two manufacturing stations and one assembly station



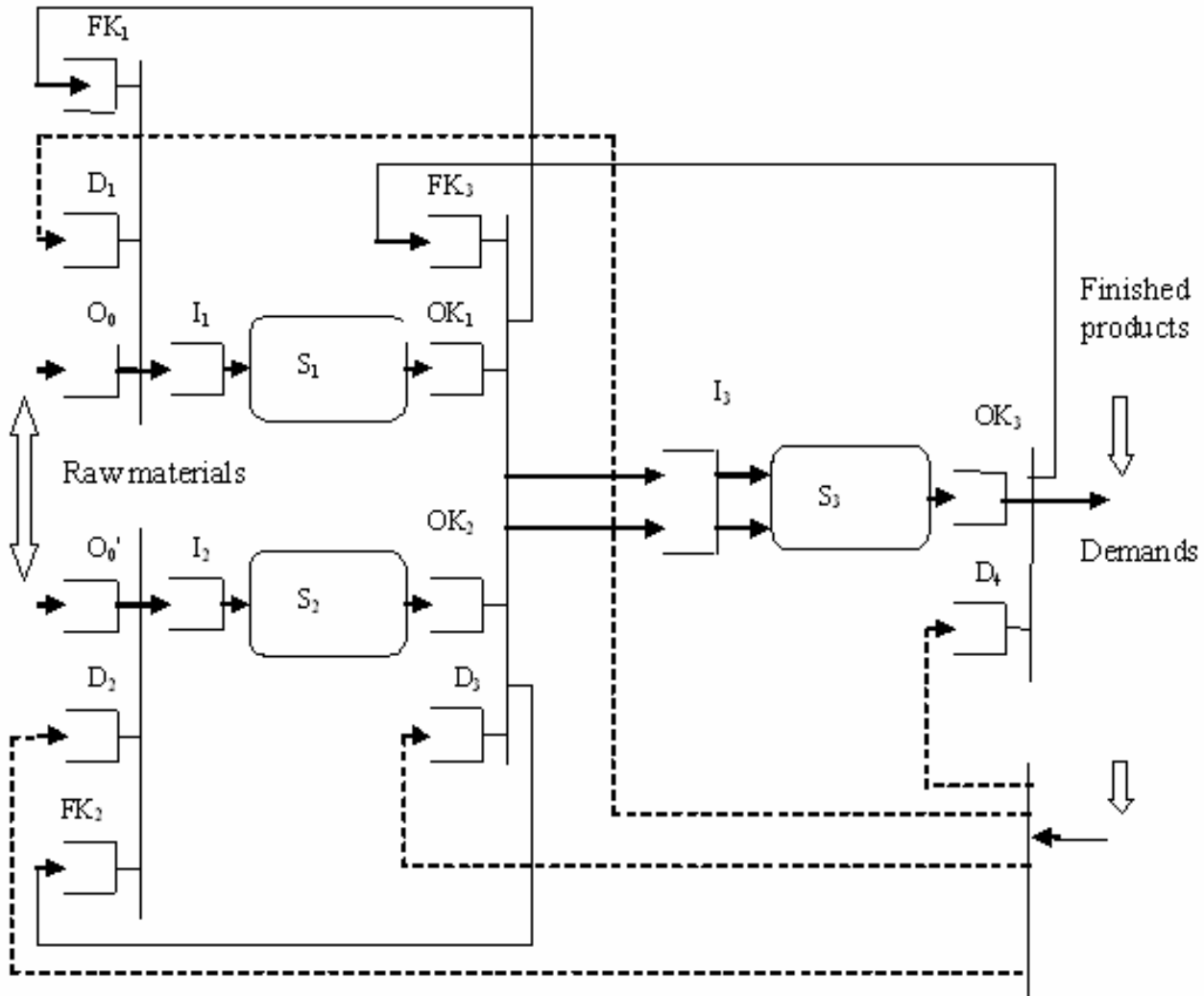


Figure 7. Extended Kanban assembly system with two manufacturing stations and one assembly station

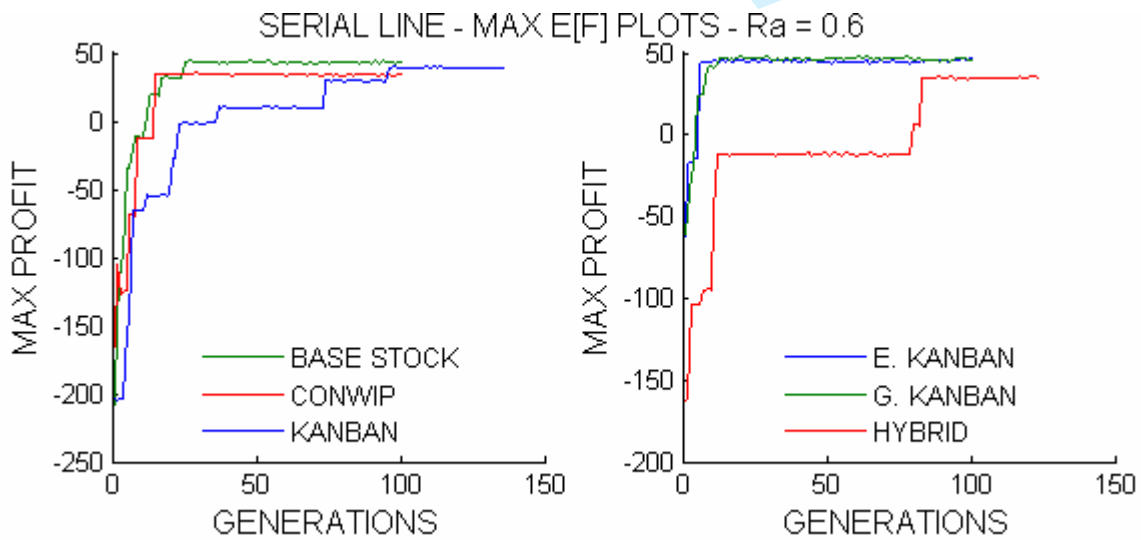


Figure 8. GA max  $E[F]$  plots - serial manufacturing lines – demand rate = 0.6

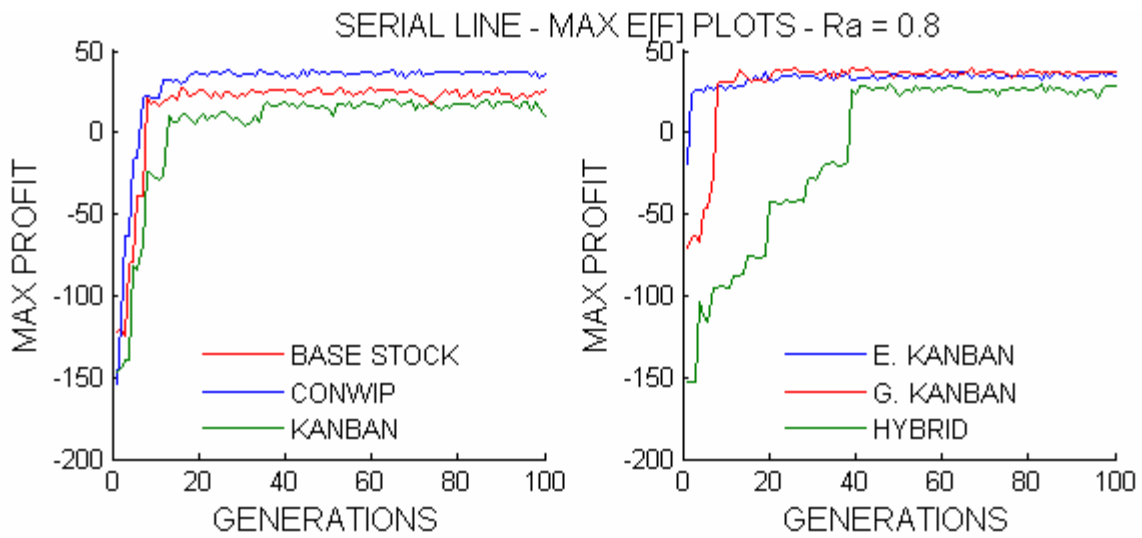


Figure 9. GA max  $E[F]$  plots - serial manufacturing lines - demand rate = 0.8

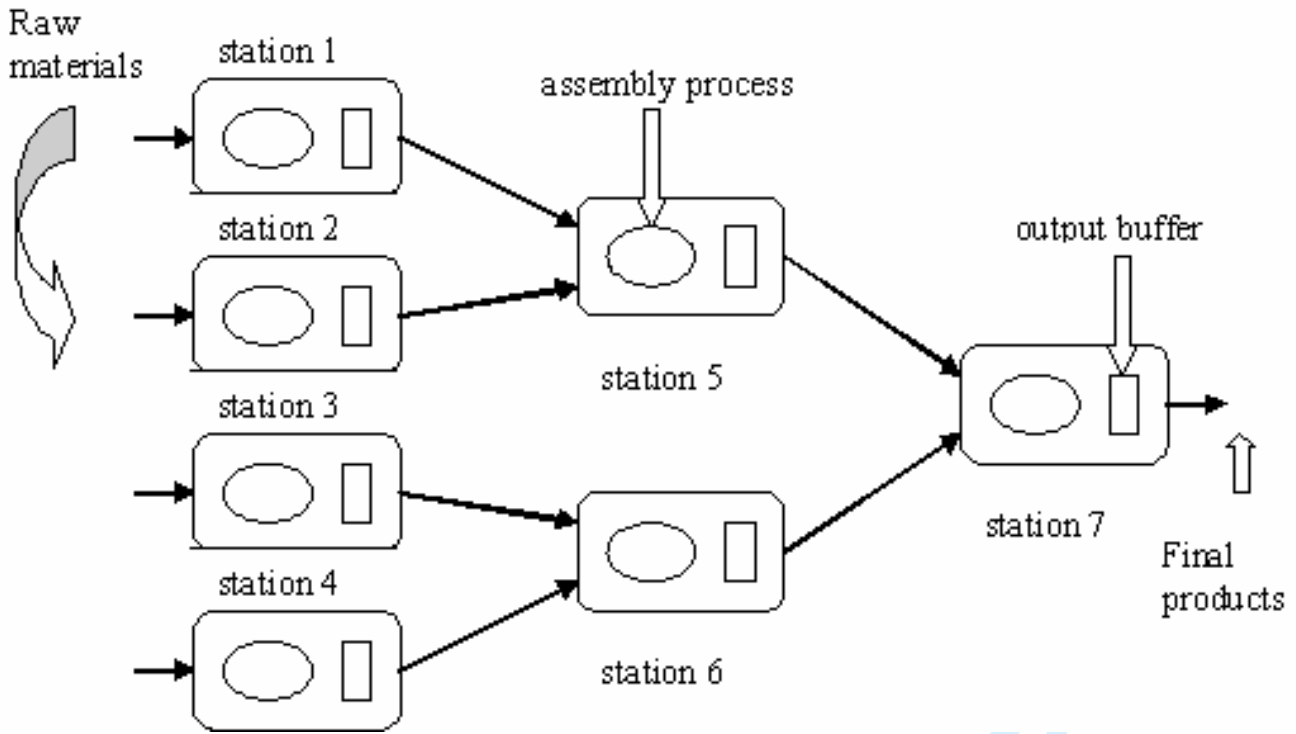


Figure 10. Assembly system topology

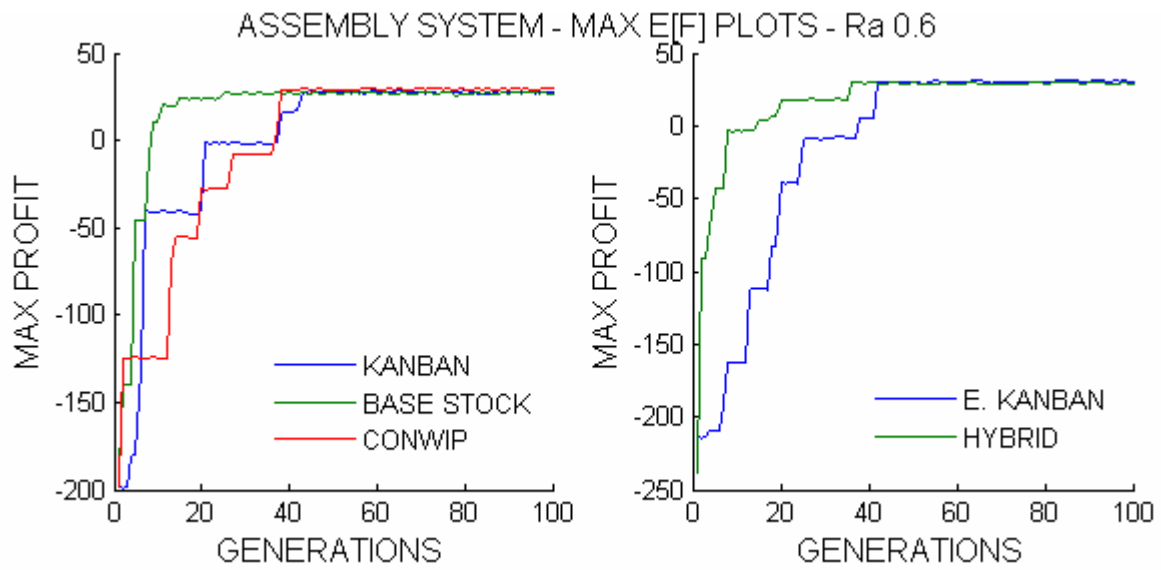


Figure 11. GA max  $E[F]$  plots - assembly systems – demand rate = 0.6

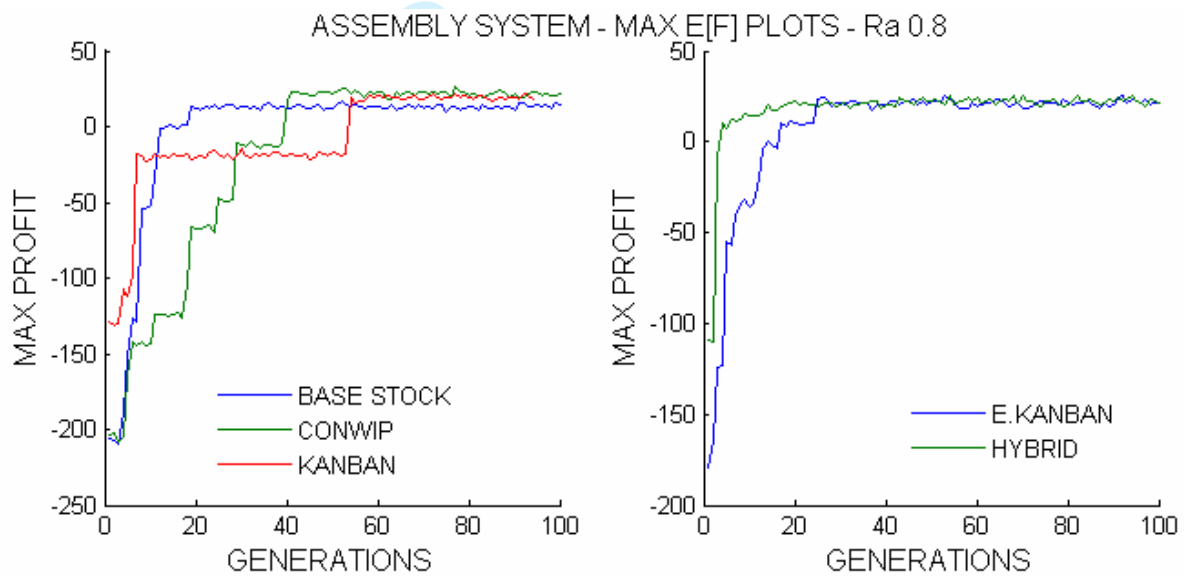


Figure 12. GA max  $E[F]$  plots - assembly systems – demand rate = 0.8

## Tables

**Table 1.** Inventory costs – serial line simulation cases 1-2.

	Buffer 1	Buffer 2	Buffer 3	Buffer 4
Inventory cost	10.0	10.0	15.0	20.0

**Table 2.** Best control parameters – Serial line simulation case 1. (demand arrival rate = 0.6)

Policy	$B_1/K_1$	$B_2/K_2$	$B_3/K_3$	$B_4/K_4$	$C$
Kanban	1	1	1	1	-
	1	1	1	1	
Base Stock	0	0	0	2	-
	$\infty$	$\infty$	$\infty$	$\infty$	
CONWIP	1	1	2	0	4
	$\infty$	$\infty$	$\infty$	$\infty$	
CONWIP/Kanban Hybrid	2	1	1	0	4
	2	1	1	$\infty$	
Extended Kanban	0	0	1	1	-
	2	2	3	10	
Generalized Kanban	0	0	1	1	-
	1	2	5	14	

**Table 3.** Average inventories –backorders average – expected profit – serial line simulation case 1. (demand arrival rate = 0.6). The cells contain the mean of these performance measures and the standard deviation in parenthesis.

Policy	Buffer 1	Buffer 2	Buffer 3	Buffer 4	Backorders average	$E[F]$
Kanban	0.39 (0.0)	0.37 (0.0)	0.36 (0.0)	0.34 (0.0)	1.14 (0.06)	40.87 (0.64)
Base Stock	0.17 (0.02)	0.16 (0.02)	0.16 (0.02)	0.26 (0.0)	1.79 (0.04)	43.13 (0.83)
CONWIP	0.05 (0.0)	0.04 (0.0)	0.04 (0.0)	1.25 (0.01)	0.9 (0.05)	36.33 (0.46)
CONWIP/ Kanban Hybrid	0.09 (0.0)	0.04 (0.0)	0.02 (0.0)	1.24 (0.01)	0.94 (0.07)	36.12 (0.61)
Extended Kanban	0.1 (0.0)	0.08 (0.0)	0.17 (0.0)	0.17 (0.0)	1.81 (0.04)	46.01 (0.45)
Generalize d Kanban	0.03 (0.0)	0.04 (0.0)	0.16 (0.0)	0.17 (0.0)	1.9 (0.05)	46.55 (0.48)

**Table 4.** Best control parameters – Serial line simulation case 2. (demand arrival rate = 0.8 )

Policy	$B_1/K_1$	$B_2/K_2$	$B_3/K_3$	$B_4/K_4$	$C$
Kanban	1	2	1	1	-
	1	2	1	1	
Base Stock	0	2	1	2	-
	$\infty$	$\infty$	$\infty$	$\infty$	
CONWIP	0	1	2	2	5
	$\infty$	$\infty$	$\infty$	$\infty$	
CONWIP/Kanban Hybrid	1	2	2	1	6
	1	2	2	$\infty$	
Extended Kanban	0	2	1	1	-
	2	2	2	6	
Generalized Kanban	0	1	1	2	-
	6	5	6	5	

**Table 5.** Average inventories –backorders average – expected profit – serial line simulation case 2. (demand arrival rate = 0.8). The cells contain the mean of the performance measures and the standard deviation in parenthesis.

Policy	Buffer 1	Buffer 2	Buffer 3	Buffer 4	Backorders average	$E[F]$
Kanban	0.18 (0.0)	0.88 (0.01)	0.16 (0.01)	0.12 (0.0)	5.9 (0.32)	21.67 (2.91)
Base Stock	0.71 (0.06)	0.97 (0.06)	0.80 (0.07)	0.45 (0.0)	3.38 (0.15)	24.23 (2.75)
CONWIP	0.19 (0.0)	0.17 (0.00)	0.16 (0.0)	0.81 (0.01)	3.83 (0.28)	35.35 (2.67)
CONWIP/ Kanban Hybrid	0.06 (0.0)	0.11 (0.0)	0.11 (0.0)	1.16 (0.02)	4.11 (0.27)	28.28 (2.14)
Extended Kanban	0.30 (0.01)	0.51 (0.0)	0.31 (0.0)	0.13 (0.0)	4.48 (0.30)	35.74 (2.93)
Generalize d Kanban	0.3 (0.01)	0.34 (0.01)	0.34 (0.01)	0.32 (0.0)	4.18 (0.22)	36.22 (2.32)

**Table 6.** Inventory costs – assembly system simulation cases 1-2.

	Buffer 1	Buffer 2	Buffer 3	Buffer 4	Buffer 5	Buffer 6	Buffer 7
Inventory cost	0.5	0.5	0.5	0.5	1.0	1.0	2.0

**Table 7.** Best control parameters – Assembly system simulation case 1. (demand arrival rate = 0.6).

Policy	$B_1/K_1$	$B_2/K_2$	$B_3/K_3$	$B_4/K_4$	$B_5/K_5$	$B_6/K_6$	$B_7/K_7$	C
Kanban	1	1	1	1	1	1	2	-
	1	1	1	1	1	1	2	
Base stock	0	0	0	0	2	2	5	-
	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
Conwip	2	2	2	2	0	0	2	4
	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
Hybrid	4	4	4	4	1	1	0	5
	4	4	4	4	1	1	$\infty$	
Extended Kanban	0	0	0	0	0	0	4	-
	2	2	2	2	4	4	4	

**Table 8.** Average inventories –backorders average – expected profit – assembly system simulation case 1. (demand arrival rate = 0.6). The cells contain the mean of the performance measures and the standard deviation in parenthesis.

Policy	Buffer1	Buffer 2	Buffer 3	Buffer 4	Buffer 5	Buffer 6	Buffer 7	Backorders average	$E[F]$
Kanban	0.39 (0.0)	0.39 (0.0)	0.39 (0.0)	0.39 (0.0)	0.38 (0.0)	0.38 (0.0)	1.0 (0.0)	0.94 (0.06)	29.0 (0.6)
Base Stock	0.24 (0.03)	0.24 (0.03)	0.24 (0.03)	0.24 (0.02)	1.19 (0.04)	1.18 (0.04)	3.64 (0.02)	0.45 (0.03)	27.38 (0.42)
CONWIP	0.06 (0.0)	0.06 (0.0)	0.07 (0.0)	0.06 (0.0)	0.12 (0.0)	0.12 (0.01)	1.71 (0.01)	0.86 (0.06)	29.56 (0.6)
Hybrid	0.24 (0.01)	0.24 (0.01)	0.24 (0.01)	0.24 (0.01)	0.03 (0.0)	0.03 (0.0)	2.53 (0.01)	0.69 (0.04)	29.48 (0.38)
Extended Kanban	0.11 (0.0)	0.1 (0.0)	0.1 (0.0)	0.1 (0.0)	0.19 (0.03)	0.19 (0.04)	1.71 (0.02)	0.81 (0.06)	29.74 (0.51)



**Table 9.** Best control parameters – Assembly system simulation case 2. (demand arrival rate = 0.8).

Policy	$B_1/K_1$	$B_2/K_2$	$B_3/K_3$	$B_4/K_4$	$B_5/K_5$	$B_6/K_6$	$B_7/K_7$	$C$
Kanban	2	2	2	2	4	4	9	-
	2	2	2	2	4	4	9	
Base stock	1	1	1	1	6	6	15	-
	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
Conwip	4	4	4	4	1	1	8	13
	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
Hybrid	5	5	5	5	8	8	0	13
	5	5	5	5	8	8	$\infty$	
Extended Kanban	0	0	0	0	1	1	10	-
	5	5	5	5	9	9	11	

**Table 10.** Average inventories –backorders average – expected profit – assembly system simulation case 2. (demand arrival rate = 0.8). The cells contain the mean of the performance measures and the standard deviation in parenthesis.

Policy	Buffer 1	Buffer 2	Buffer 3	Buffer 4	Buffer 5	Buffer 6	Buffer 7	Backorders average	$E[F]$
Kanban	1.13 (0.0)	1.13 (0.0)	1.13 (0.0)	1.13 (0.0)	2.81 (0.02)	2.81 (0.02)	5.5 (0.06)	1.8 (0.2)	19.12 (1.92)
Base Stock	1.11 (0.1)	1.12 (0.1)	1.1 (0.09)	1.1 (0.08)	5.81 (0.1)	5.83 (0.1)	11.2 (0.06)	0.59 (0.06)	13.82 (0.76)
CONWIP	0.64 (0.04)	0.63 (0.04)	0.63 (0.05)	0.63 (0.05)	1.14 (0.06)	1.16 (0.06)	7.2 (0.08)	1.56 (0.14)	22.42 (1.42)
Hybrid	0.59 (0.03)	0.59 (0.02)	0.6 (0.02)	0.6 (0.03)	0.93 (0.05)	0.94 (0.05)	7.16 (0.07)	1.61 (0.19)	22.52 (1.9)
Extended Kanban	0.61 (0.04)	0.61 (0.04)	0.61 (0.02)	0.61 (0.04)	1.08 (0.06)	1.08 (0.05)	5.46 (0.06)	1.86 (0.28)	22.86 (1.89)