2005

Allocating work in process in a multiple-product CONWIP system with lost sales

Sarah M. Ryan
Iowa State University, smryan@iastate.edu

Jumpol Vorasayan
Iowa State University

Allocating work in process in a multiple-product CONWIP system with lost sales

Abstract
To operate a multiple-product manufacturing system under a CONWIP control policy, one must decide how to assign kanbans to products. With a fixed total number of kanbans in a competitive environment, the goal is to determine their allocation to product types in order to minimize lost sales equitably. In particular, we consider systems in which the products may make multiple visits to the same station with a different processing time distribution on each repeat visit. With a fixed number of kanbans dedicated to each product, the system is modeled as a multiple-chain multiple-class closed queuing network. A nonlinear program simultaneously provides an approximate performance evaluation and optimizes the allocation of kanbans to product types. In numerical examples, the allocations identified are similar to those obtained by exhaustive enumeration with simulation, but frequently differ significantly from a naÃ¨ve allocation according to demand rates. A variant of the model that minimizes the total work-in-process to achieve specified throughput targets yields results similar to a previous heuristic method.

Keywords
CONWIP, lost sales, multiclass closed queuing network, nonlinear program

Disciplines
Industrial Engineering | Systems Engineering

Comments
The Version of Record of this manuscript has been published and is available in International Journal of Production Research 2005, http://www.tandfonline.com/10.1080/00207540420002688 75. Posted with permission.

Allocating work in process in a multiple-product CONWIP system with lost sales

Sarah M. Ryan* and Jumpol Vorasayan
Department of Industrial & Manufacturing Systems Engineering
Iowa State University

*Corresponding author:
Associate Professor
Department of Industrial & Manufacturing Systems Engineering
2019 Black Engineering Building
Iowa State University
Ames, IA 50011-2164

Email: smryan@iastate.edu
Phone: 515-294-4347
Fax: 515-294-3524

July 2004

For publication in *International Journal of Production Research*

Allocating work in process in a multiple-product CONWIP system with lost sales

Abstract

To operate a multiple product manufacturing system under a CONWIP control policy, one must decide how to assign kanbans to products. With a fixed total number of kanbans in a competitive environment, the goal is to determine their allocation to product types in order to minimize lost sales equitably. In particular, we consider systems in which the products may make multiple visits to the same station with a different processing time distribution on each repeat visit. With a fixed number of kanbans dedicated to each product, the system is modeled as a multiple chain multiple class closed queuing network. A nonlinear program simultaneously provides an approximate performance evaluation and optimizes the allocation of kanbans to product types. In numerical examples, the allocations identified are similar to those obtained by exhaustive enumeration with simulation but frequently differ significantly from a naïve allocation according to demand rates. A variant of the model that minimizes the total work-in-process to achieve specified throughput targets yields results similar to a previous heuristic method.

Keywords: CONWIP, lost sales, nonlinear program, multiclass closed queuing network

1. Introduction

In the effort to produce increasing varieties of products while fully utilizing resources, many manufacturers are producing different types of products by using different process plans on a common set of machines. A product type may be defined by its routing among the machines and its processing time on each. In a large competitive market, the demands for the various product types are independent, and those demands that cannot be filled immediately by

finished products are lost (Andersson and Melchiors 2001). The goal is to meet the demands for the entire suite of products without excessive inventory and without favoring one product type over the others.

One approach for controlling the inventory in the system is to adopt a Constant Work in Process (CONWIP) control policy (Spearman, Woodruff and Hopp 1990). We assume that the production process is operated as a make-to-stock CONWIP system. Therefore, there is a fixed limit on the total inventory (both work in process and finished goods) in the system, which is controlled by the number of kanbans placed in circulation. For a multiproduct CONWIP system, recent articles have offered conflicting advice on whether this fixed inventory limit should apply to the entire system or whether each product type should have its own individual limit. Equivalently, should the kanbans be generic to all product types, or dedicated to specific product types? In the next section, we briefly review the literature and conclude that when the processing requirements for the different product types are dissimilar it may be preferable to use dedicated kanbans. For all the examples studied in Section 4, simulation results show that dedicated kanbans do indeed yield lower lost sales rates. Though more research needs to be done to definitively resolve this issue, in this paper we assume that each product type has a fixed number of kanbans dedicated to it.

To maintain our focus on the allocation of kanbans to product types, we rely on easily implementable sequencing policies such as first-come-first-served or random selection. We also assume that the policy is non-idling, i.e. a machine is not permitted to idle when there are any jobs waiting to be processed. More sophisticated and information-intensive sequencing rules have been developed for a multiproduct CONWIP environment to achieve high throughput (Chevalier and Wein 1993), minimize a combination of lost sales and setup costs (Kim and Van

Oyen 1998), or achieve a high service level with low inventory in a make-to-stock flow shop (Framinan, Ruiz-Usano and Leisten 2000). Further, we assume that the total number of kanbans has already been determined. The problem is to find an allocation of kanbans that minimizes the maximum rate of lost sales over the product types. The numerical examples illustrate how our model can be used with different total numbers of kanbans to achieve a desired balance between inventory and lost sales.

We model the production system as a multiple chain multiple class closed queuing network. The single-server stations in the network represent the production resources, and the customers are the kanbans. The kanbans dedicated to a specific product type form a set of customer classes. To model the lost sales, we also include a 'finished goods' station for each product type, where the processing time is the interarrival time between demands for that product (note that Kim and Van Oyen (1998) used the same modeling device). The model allows the possibility of probabilistic routing, different processing times for different customer classes, and multiple visits by a class to the same station, with possibly a different processing time distribution on each repeat visit. Repeat visits to a station could occur either for unplanned rework or as part of the process plan; for example, two-tone paint may be applied in separate visits to the painting station, with curing in between. Repeat visits to the same station are also common in semiconductor manufacturing. Since product-form methods or decomposition algorithms cannot be used to analyse the performance of such a network, a new performance evaluation method is needed to compare alternative kanban allocations.

To simultaneously evaluate the network performance and optimize the kanban allocation, we adapt the nonlinear programming (NLP) approach used by Ryan and Choobineh (2003). In contrast to the reference paper, we include tighter constraints that result from the fixed number of

customers in each class, and we optimize the allocation directly in the nonlinear program. In numerical examples, comparisons with simulation show that the allocation found by the model results in nearly the lowest maximum rate of lost sales for a variety of demand rate and processing time relationships. The results show that with a limited amount of inventory, in the case that the total demand rate is close to the maximum system throughput, kanban allocations can significantly affect the lost sales. Moreover, the best kanban allocations cannot necessarily be guessed from the demand proportions.

The remainder of this paper is organized as follows: Section 2 contains a literature review of multiple-product kanban/CONWIP systems, their closed queueing network models and previous approaches to controlling lost sales. Section 3 provides the mathematical model and Section 4 shows numerical examples that illustrate the implementation of the model and its results. Conclusions are presented in Section 5. An Appendix shows that the results of a variant of our model are consistent with those found by a heuristic control chart approach (Hopp and Roof 1998).

2. Literature review

The superiority of CONWIP systems over push or pure kanban systems is argued in several papers (Spearman et al. 1990; Spearman and Zazanis 1992). It also has been shown to perform well compared with other types of pull systems ((Spearman 1992) and (Bonvik, Couch and Gershwin 1997)). However, Gstettner and Kuhn (1996) opposed the findings of Spearman (1992) by showing that the distribution of kanbans among the stations had a significant effect on the performance of the system. When the distribution of the kanbans is chosen correctly as a control parameter, the kanban system will reach a given production rate with less work in process (WIP) than in a CONWIP system.

In a multi-product CONWIP system, kanbans can be either dedicated to specific product types or shared among them. Early research on multi-product systems with limited inventory essentially assumed that kanbans would be shared. For example, Harrison and Wein (1990) used a Brownian approximation model to obtain a sequencing policy to maximize the total throughput of a two-station closed queueing network in heavy traffic. The type of job that enters the system is selected deterministically by a backlog list or probabilistically according to the product mix. This method was extended to multiple product types by Chevalier and Wein (1993). In this approach, the WIP mix is not controlled directly. Instead, it varies over time in response to stochastic processing times and queuing interactions.

Some simulation studies have indicated that dedicated kanbans provide superior performance. When kanbans are dedicated to each product, individual product WIP levels are controlled and the system can be modeled as a multiple chain closed queuing network. When a job has completed processing, it is replaced by a new job of the same type. Buzacott and Shanthikumar (1993) obtained the throughput of a multiple machine multiple product system in certain special cases by evaluating the performance of a multiple chain multiple class closed queuing network. Duenyas (1994) used a simulation model to show that, in some specific cases with diverse product routings, the dedicated kanban policy can have advantages over the shared kanban policy. The simulation experiments by Framinan et al. (2000) also showed that dedicated kanbans achieved better results than shared kanbans in a make-to-stock flowshop.

Nevertheless, the call for using dedicated kanbans is not unanimous. Simulation studies of various push and pull policies were presented by Krishnamurthy, Suri and Vernon (2004) for a serial system. They found that in a balanced system, where a product's processing time is positively related to its demand rate, dedicated kanban (pull-CONWIP) and shared kanban

(Hybrid-CONWIP) policies could achieve a specified throughput with lower inventory levels than would be possible in either pure pull or pure push systems. However, in the case of unbalanced systems, in which a product's proportion of demand is inversely related to its processing time, pull-CONWIP performed the worst among the different policies. The methods for determining the numbers of kanbans for each product type or adjusting them for different demand rates were not explained. Baynat, Buzacott and Dallery (2002) argued that for a single stage kanban control system, such as a CONWIP system, shared and dedicated sytems perform identically. However, their argument depended crucially on an assumption of separate input queues for the different products, which contradicts the conventional assumption, as in (Duenyas 1994) or (Framinan et al. 2000), that incoming products would be ordered in a single queue according to a backlog list.

The previous research has suggested that dedicating kanbans to specific product types is preferable when routings differ significantly. However, one major problem is to find how many kanbans to allocate to each product type for the best system performance. Hopp and Roof (1998) developed a simple adaptive production control method to minimize WIP levels while achieving production targets under the CONWIP protocol. Their method, termed Statistical Throughput Control (STC), used real-time data to automatically adjust WIP levels (via kanban counts) to achieve the target throughput in dynamic environment. They illustrated the performance of the STC method in multi-product systems by considering two examples: a single line with multiple products and multiple lines with shared machines. In both examples, the appropriate WIP levels adjusted by STC achieved the target throughput. In some cases these WIP levels varied cyclically over time as one kanban was repeatedly added and taken away. In the Appendix, we modify our approach to find the minimum WIP necessary to achieve throughput targets and see

that it finds similar kanban counts. Srinivasan, Ebbing and Swearingen (2003) used an iterative heuristic with approximate mean value analysis to identify WIP levels that match throughputs to projected demand rates in a high-variety, low-volume manufacturing plant.

Ryan, Baynat and Choobineh (2000) developed a heuristic allocation procedure in a CONWIP controlled job shop in which multiple products with distinct routings compete for the same set of resources. The objective was to determine the minimum total WIP and WIP mix to satisfy a uniform service level across product types, as measured by the proportion of arriving orders that wait to be fulfilled. They modeled the job shop as a closed queuing network and used an approximate performance evaluation procedure to determine a WIP total and mix that would achieve a high throughput consistent with the product mix. When tested with randomly arriving demands following the product mix, the WIP mix did provide balanced customer service. Numerical examples showed that the most effective WIP mix was often quite different from the specified product mix. Ryan and Choobineh (2003) developed a nonlinear programming model to set the constant level of work in process for each product type in a job shop operating under CONWIP control with a fixed number of kanbans dedicated to each product type. Based on a model with shared kanbans, they identified the minimum total WIP that was guaranteed to yield throughput near the maximum possible for the specified product mix and then set individual WIP levels by multiplying the optimal WIP mix proportions by the minimum total WIP. This paper explicitly accounts for demand rates by minimizing lost sales and improves accuracy with a smaller feasible region for the NLP.

Focusing on lost sales rather than throughput, Kim and Van Oyen (1998) developed a heuristic scheduling policy for a single machine to minimize a combination of lost sales costs for two products and the setup cost of switching between products. They modeled a single server

with two job classes in a CONWIP release policy as a two-class tandem queueing network with a fixed amount of WIP for each job type. Altiok and Shiue (1995) and Andersson and Melchiors (2001) also considered lost sales in a multiple-product shared resource environment but assumed a base stock policy.

To identify the allocation of kanbans to minimize lost sales we need an optimization method together with a reliable performance evaluation method. Performance evaluation is difficult when products revisit the same stations. Though simulation could be used to evaluate the performance, efficient multidimensional simulation-based optimization methods are not yet available. Such routines would be complicated in this case by the dual objectives of minimizing lost sales and equalizing them across the product types. The procedure developed in this paper combines optimization and performance evaluation while constraining each product's lost sales to be equal.

3. The model

The multiproduct system with separate kanbans for each product type is modeled as a multiple class multiple chain closed queueing network. Each processing station has a single server with exponentially distributed service times, which may have a different mean for each product type. In addition, a product may visit the same station more than once with a different service rate at each visit. For tracking purposes, at each station the model includes a separate queue for each type of product on each visit to the station. However, we assume that actual sequencing is first-come-first-served or random, irrespective of the product type. In (Ryan and Choobineh 2003), a nonlinear program was developed to evaluate the throughput performance of single chain queuing network, corresponding to a system in which the kanbans are shared among the product types. Here we propose additional constraints to specify fixed numbers of kanbans

dedicated to each type of product. In addition, we measure the performance of the system by the rate of lost sales for each product rather than throughput, and optimize the system allocation directly rather than by the two stage process used previously.

3.1. Queueing model of multiproduct system with lost sales

Using an approach also taken by Kim and Van Oyen (1998), lost sales are measured at a separate finished goods station for each product type. These stations represent the point at which finished products are matched with randomly arriving customer demands. At each finished goods station, the server is always available, and the times between (real or virtual) service completions correspond to the interarrival times of demands for that product type. However, there may or may not be finished products available when a demand arrives. If the finished goods buffer contains at least one product when a service completion occurs, then this is a real service completion and the corresponding demand is satisfied. On the other hand, while the finished goods buffer is empty, all service completions are virtual and correspond to arrivals of demands that are not satisfied, i.e. lost sales. Figure 1 illustrates how the finished goods station represents these two situations.

[insert figure 1 about here]

Under the non-idling policy, the server must be busy performing real service completions whenever its buffer is not empty: Virtual service completions occur only when the server is idle because it is starved. Therefore, the rate of lost sales of a particular product type equals its demand rate multiplied by the proportion of time its finished goods server is idle. We assume that unlimited raw materials are available, so that the kanban released when the customer obtains a product is immediately attached to raw material and thus re-enters the system as a new

customer of the same type. Therefore the number of customers of each type in the queueing network remains constant.

3.2. Nonlinear Programming Model

The notations used in the model are defined in four categories: input parameters, stochastic processes, decision variables, and output values.

The input parameters:

P = Number of distinct product types to be manufactured

R = Number of customer classes, $R \geq P$. A class $r = 1, \ldots, P$ consists of kanbans for type r products in their initial stages of processing. In case a product type visits a station more than once, the customer (kanban) changes to a different class $k > P$ after each visit to the station

$C(r)$ = Set of customer classes that correspond to product type r, $r = 1, \ldots, P$

N = Total number of items in the system (number of kanbans)

S = Number of single server processing stations, including the finished goods station for each product (denoted as f_r for product type r).

$R(s)$ = Set of classes that visit station s.

L = The number of buffers ((k,s) pairs). Pairs (k,s) and (j,v) are defined if customer class k visits station s and customer class j visits station v. We say that $(j,v) \geq (k,s)$ if either $j \geq k$ or $j = k$ and $v \geq s$.

q_{ksjv} = Probability that a class k customer, having completed processing at station s, joins the queue at station v as a class j customer. Assume $q_{ksks} = 0$. If the product type r corresponding to class k has a single process plan (route) that does not revisit

station s, then $q_{kskv} = 1$, where v is the next station after s in that product's routing. If product r will revisit station s, then $q_{ksk'v} = 1$, where v is the next station after s in the product's routing and $k' \in C(r)$.

μ_{ks} = Mean processing rate for class k customers at station s. Processing times are exponentially distributed.

λ_r = Mean demand rate for type r product. Times between demand arrivals follow an exponential distribution.

Stochastic processes:

$X_{ks}(t)$ = The number of class k customers at station s (in queue or in service) at time t.

$W_{ks}(t)$ = 1 if the station s server is busy with a class k customer at time t and 0 otherwise.

Decision variables:

ρ_{ks} = $E[W_{ks}(t)]$ in steady state, i.e. the utilization of station s for class k, or the proportion of time station s is busy with class k customers.

ρ_{rf_r} = Utilization of the finished goods station f_r for product type r.

z_{ksjv} = $E[W_{ks}(t)X_{jv}(t)]$ for any t (in steady state).

Output values:

$KB(r)$ = The number of kanbans for classes that correspond to product type r.

$PR(r)$ = Proportion of all kanbans allocated to product type r: $PR(r) = KB(r)/N$.

$LS(r)$ = Lost sales of product type r at finished goods station r: $LS(r) = \lambda_r * (1 - \rho_{rf_r})$.

In independent developments, Kumar and Kumar (1994) and Bertsimas, Paschalidis and Tsitsiklis (1994) derived a set of linear equalities that govern the expected values represented by our decision variables. These relationships follow from uniformizing the continuous time Markov chain for $\{X_{ks}(t), \forall(k,s)\}$ and assuming that the first, second and cross moments $E[X_{ks}(t)]$, $E[X_{ks}^{2}(t)]$ and $E[X_{ks}(t)X_{jv}(t)]$ are stationary. Additional equalities, as given below, follow from the fixed total number of customers in the system, the non-idling assumption, and the requirement that server utilizations not exceed one. These equalities can be used as linear programming constraints. For example, Kumar and Kumar (1994) found upper and lower bounds, respectively, on the system throughput by maximizing and minimizing the sum of $\mu_{ks}\rho_{ks}$ over an appropriate set of buffers (*k*, *s*) subject to these constraints. However, the distance between the bounds was rather wide. Ryan and Choobineh (2003) reduced the gap by including additional nonlinear constraints motivated by mean value analysis under first-come-first-served or random sequencing. In this paper, we include tighter constraints that follow from the fixed number of products of each type and formulate an objective function designed to minimize and balance lost sales across the product types. Also, whereas Ryan and Choobineh used the nonlinear program's solution only as a guide for the allocation of kanbans to product types, this paper optimizes the allocation directly in the nonlinear program.

We assume a non-idling policy; that is, if any customers are present at station *s*, the station must not be idle. Mathematically, this assumption is expressed as $X_{ks}(t) > 0 \Rightarrow \sum_{j\in R(s)} W_{js}(t) = 1$. Then $X_{ks}(t) = \sum_{j\in R(s)} W_{js}(t)X_{ks}(t)$ and we can write $E[X_{ks}(t)] = \sum_{j\in R(s)} E[W_{js}(t)X_{ks}(t)] = \sum_{j\in R(s)} z_{jsks}$. Also note that the expected total population

of type r jobs is given by $\sum_{k\in C(r)}\sum_{s=1}^{S}\sum_{j\in R(s)} z_{jsks} = KB(r)$, where the sum over k is over those customer classes that correspond to product type r.

The constraints defined by Ryan and Choobineh (2003) include:

The system population:

- The constraint for the total amount of work in process

$$\sum_{r=1}^{R} KB(r) = N \tag{1}$$

- The kanbans dedicated to each product type r

$$\sum_{k\in C(r)}\sum_{s=1}^{S}\sum_{j\in R(s)} z_{jsks} = KB(r),\ \forall r \tag{2}$$

Sampling equalities:

$$\sum_{(j,v)} z_{ksjv} - N\rho_{ks} = 0, \forall (k,s) \tag{3}$$

Non-idling:

$$\sum_{j\in R(v)} z_{jvks} - \sum_{j\in R(s)} z_{jsks} \le 0, \forall (k,s), v \ne s\ . \tag{4}$$

Stationary first moments of queue lengths:

$$-\mu_{ks}\rho_{ks} + \sum_{(j,v)} q_{jvks}\mu_{jv}\rho_{jv} = 0, \forall (k,s)\ . \tag{5}$$

Stationary second moments of queue lengths:

$$\begin{aligned}&\sum_{(l,y)} q_{lyks}\mu_{ly}z_{lyjv} + \sum_{(l,y)} q_{lyjv}\mu_{ly}z_{lyks} - \mu_{ks}z_{ksjv} - \mu_{jv}z_{jvks} + q_{ksjv}\mu_{ks}\rho_{ks} + q_{jvks}\mu_{jv}\rho_{jv}\\ &+2\delta_{ksjv}\mu_{ks}\rho_{ks} = 0,\ \forall (k,s), \forall (j,v) \ge (k,s),\end{aligned} \tag{6}$$

where $\delta_{ksjv} = 1$ if $(k,s) = (j,v)$, and 0 otherwise.

Sojourn time constraints:

$$\sum_{j \in R(s)} z_{jsks} \le \rho_{ks} + \mu_{ks}\rho_{ks} \sum_{j \in R(s)} \frac{1}{\mu_{js}} \sum_{l \in R(s)} z_{lsjs}, \forall (k,s)$$
$$\sum_{j \in R(s)} z_{jsks} \ge \mu_{ks}\rho_{ks} \sum_{j \in R(s)} \frac{1}{\mu_{js}} \sum_{l \in R(s)} z_{lsjs}, \forall (k,s) \quad (7)$$

Utilization constraint:

$$\sum_{k} \rho_{ks} \le 1, \forall s\,. \quad (8)$$

In addition, all decision variables were assumed nonnegative.

Constraints (1) - (6) are linear in the decision variables, z and ρ. Including the nonlinear (specifically, bilinear) constraints (7) complicates the solution but significantly reduces the feasible region.

In this paper, we add constraints that follow from having a fixed number of kanbans dedicated to each product type. Note that these are nonlinear because $KB(r)$ and ρ_{ks} are both (functions of) decision variables.

Class specific sampling equalities:

$$\sum_{j \in C(r)} \sum_{v} z_{ksjv} - KB(r)\rho_{ks} = 0, \forall (k,s), \forall r\,. \quad (9)$$

The overall goal of the allocation of kanbans to product types is to minimize the rate of lost sales. Although it might be possible to achieve a low total rate of lost sales by favoring customers for one product type at the expense of others, we assume that equity across product types is desired. Therefore, we wish to minimize the maximum lost sales across product types. The direct approach to formulating this minimax objective would be to minimize *x* subject to the additional constraints that $x \ge \lambda_r * (1 - \rho_{rf_r})$ for each *r*. We tested the results by fixing the values of *KB*(*r*) and comparing the maximum lost sales obtained in the model with those found by simulation or any available analytical methods. The allocation that the model identified with the

lowest maximum lost sales differed significantly from what was found in simulation in some cases. Better accuracy was achieved by an indirect approach. We added constraints to guarantee equality of lost sales among product types:

$$\lambda_r \left(1-\rho_{rf_r}\right) = \lambda_{r+1}\left(1-\rho_{r+1,f_{r+1}}\right),\ r = 1,\ \ldots,\ R\text{ - }1. \tag{10}$$

Then, since higher levels of finished goods inventory should reduce lost sales, we maximized the total number of kanbans in finished good station buffers. Therefore, the objective is to

$$\text{Maximize } \sum_r z_{rf_r rf_r}\ . \tag{11}$$

4. Numerical examples

The model may be applied to systems with arbitrary routings and expected processing times. We present three examples, each with two products to facilitate examination of the WIP mix in terms of $PR(1)$, the proportion of WIP that is allocated to type 1 products. However, the model can accommodate any number of product types. The first example supports the use of dedicated rather than shared kanbans. We use simulation to compare the lost sales of dedicated vs. shared kanbans and to examine the sensitivity of the model's results to the processing time distribution. All simulations were conducted using 30 replications, each 1000 time units long after a 300 time unit warm up period.

The second example consists of a single service station plus the two finished goods stations. We designed this model to be simple in order to easily check the mathematical program and study the results of changing processing times and demand rates. In this case, when the processing times for the two products are identically distributed, the nonlinear program can be compared with the results of a multi-class mean value analysis.

The third example features more complicated routings with repeat visits to the same station and a different processing time distribution for each visit. In this case, since no analytical

methods for evaluating performance currently exist, we compare the lost sales predicted by the NLP with simulation results as the WIP allocation is varied.

In Examples 2 and 3 the optimal kanban allocation strongly depends on whether processing capacity exceeds the combined demand rates or vice versa. Though the actual bottleneck of the overall system, including the finished goods stations, may vary from time to time and be influenced by queuing interactions, we broadly label the system as having 'no bottleneck' if $\lambda_r < \min \mu_{ks}$ for each product type *r*, each processing station *s* and each class $k \in C(r)$. On the other hand, the system does have a production bottleneck if at least one product type has a processing rate at some station that is less than or equal to its demand rate.

4.1 *Example 1*. This system, similar to what has been mentioned in Duenyas (1994), has 2 product types. Product type one is processed at station 1, 2, and 3 while product type two is processed at station 3 and 4. After processing, each product type is stocked in its own finished good station, stations 5 and 6, respectively. The demand rates for the two product types are equal, $\lambda_1 = \lambda_2 = 50$. All service stations also have processing rates equal to 50. At station 2, we also examine the case of $\mu_{12} = 20$. When $\mu_{12} = 50$ the system throughput is equal to the demand rate while when $\mu_{12} = 20$ the production system has a bottleneck at station 2 that limits the production rate to less than demand rate.

First, shared and dedicated kanban policies are compared with the number of kanbans fixed at 10 cards. For the shared kanban policy, we introduce an artificial station 0 that merely assigns a free kanban to product type 1 with a probability that varies from 0.0 to 1.0, and otherwise to product type 2 (see figure 2). In the dedicated kanban policy, shown in figure 3, each type of kanban has its own separate route. The fixed number of kanbans for each type 1 is varied from 0 to 10.

[insert figure 2 about here]

[insert figure 3 about here]

The results in figure 4 show the superiority of dedicated kanbans over shared kanbans in both the bottleneck and no bottleneck cases. The maximum lost sales across product types is sensitive to the WIP mix in the system. The appropriate allocation will significantly reduce the maximum lost sales particularly in the no bottleneck case. In either case, the best WIP proportions in the shared kanban policy equal the demand proportions. But under the dedicated kanban policy it is interesting to see that the optimal WIP allocation does not equal the demand ratio even when the processing rates equal the demand rates. This result implies that the optimal allocation in dedicated kanban policy is more sensitive to processing rates and system configuration than that of the shared kanban policy.

[insert figure 4 about here]

The exponential (EXPO) processing time distribution has standard deviation equal to its mean, which may be more variable than actual processing times, and also lacks symmetry. To see the effect of different processing time distributions, we also simulated normally and uniformly distributed processing times with standard deviation approximately 10% of the mean in stations 1 through 4, while the demands are still assumed to follow a Poisson process.

As seen in figures 5 and 6, the maximum lost sales from the uniform (UNIF) and normal (NORM) processing times are virtually identical. Moreover, the maximum lost sales from the exponential processing time distribution differ just slightly from those of the other processing time distributions in all cases, and are minimized by the same kanban allocation for all

processing time distributions. These observations indicate that the assumption of exponential processing times in the nonlinear program, which allows tractability, does not compromise accuracy.

[insert figure 5 about here]

[insert figure 6 about here]

4.2. Example 2. Two products are processed at station 3 and go separately to their finished goods stations, which are station 1 for product 1 and station 2 for product 2, respectively, as shown in figure 7.

[insert figure 7 about here]

For this example, the full NLP is:

Maximize $z_{1111} + z_{2222}$

Subject to:

$$z_{1111} + z_{1313} + z_{2313} + z_{2222} + z_{1323} + z_{2323} = N \quad (1.1)$$

$$z_{1111} + z_{1313} + z_{2313} = KB(1) \quad (2.1)$$

$$z_{1323} + z_{2222} + z_{2323} = KB(2) \quad (2.2)$$

$$z_{1111} + z_{1113} + z_{1122} + z_{1123} - N\rho_{11} = 0 \quad (3.1)$$

$$z_{1311} + z_{1313} + z_{1322} + z_{1323} - N\rho_{13} = 0 \quad (3.2)$$

$$z_{2211} + z_{2213} + z_{2222} + z_{2223} - N\rho_{22} = 0 \quad (3.3)$$

$$z_{2311} + z_{2313} + z_{2322} + z_{2323} - N\rho_{23} = 0 \quad (3.4)$$

$$z_{2211} - z_{1111} \le 0 \quad (4.1)$$

$$z_{1311} + z_{2311} - z_{1111} \le 0 \quad (4.2)$$

$$z_{1113} - z_{1313} - z_{2313} \le 0 \quad (4.3)$$

$$z_{2213} - z_{1313} - z_{2313} \le 0 \quad (4.4)$$

$$z_{1122} - z_{2222} \le 0 \quad (4.5)$$

$$z_{1322} + z_{2322} - z_{2222} \le 0 \quad (4.6)$$

$$z_{1123} - z_{1323} - z_{2323} \le 0 \quad (4.7)$$

$$z_{2223} - z_{1323} - z_{2323} \le 0 \quad (4.8)$$

$$\lambda_1\rho_{11} - \mu_{13}\rho_{13} = 0 \quad (5.1)$$

$$\lambda_2\rho_{22} - \mu_{23}\rho_{23} = 0 \tag{5.2}$$

$$-\lambda_1 z_{1111} + \mu_{13} z_{1311} + \lambda_1\rho_{11} = 0 \tag{6.1}$$

$$\lambda_1 z_{1113} - \mu_{13} z_{1313} + \mu_{13}\rho_{13} = 0 \tag{6.2}$$

$$-\lambda_2 z_{2222} + \mu_{23} z_{2322} + \lambda_2\rho_{22} = 0 \tag{6.3}$$

$$\lambda_2 z_{2223} - \mu_{23} z_{2323} + \mu_{23}\rho_{23} = 0 \tag{6.4}$$

$$\lambda_1 z_{1111} - \lambda_1 z_{1113} - \mu_{13} z_{1311} + \mu_{13} z_{1313} - \lambda_1\rho_{11} - \mu_{13}\rho_{13} = 0 \tag{6.5}$$

$$-\lambda_1 z_{1122} + \mu_{13} z_{1322} - \lambda_2 z_{2211} + \mu_{23} z_{2311} = 0 \tag{6.6}$$

$$-\lambda_1 z_{1123} + \mu_{13} z_{1323} + \lambda_2 z_{2211} - \mu_{23} z_{2311} = 0 \tag{6.7}$$

$$\lambda_1 z_{1122} - \mu_{13} z_{1322} - \lambda_2 z_{2213} + \mu_{23} z_{2313} = 0 \tag{6.8}$$

$$\lambda_1 z_{1123} - \mu_{13} z_{1323} + \lambda_2 z_{2213} - \mu_{23} z_{2313} = 0 \tag{6.9}$$

$$\lambda_2 z_{2222} - \lambda_2 z_{2223} - \mu_{23} z_{2322} + \mu_{23} z_{2323} - \lambda_2\rho_{22} - \mu_{23}\rho_{23} = 0 \tag{6.10}$$

$$z_{1111} - \rho_{11} - \rho_{11} z_{1111} \le 0 \tag{7.1}$$

$$z_{1313} + z_{2313} - \rho_{13} - \rho_{13} z_{1313} - \frac{\mu_{13}}{\mu_{23}}\rho_{13} z_{1323} - \rho_{13} z_{2313} - \frac{\mu_{13}}{\mu_{23}}\rho_{13} z_{2323} \le 0 \tag{7.2}$$

$$z_{2222} - \rho_{22} - \rho_{22} z_{2222} \le 0 \tag{7.3}$$

$$z_{1323} + z_{2323} - \rho_{23} - \frac{\mu_{23}}{\mu_{13}}\rho_{23} z_{1313} - \rho_{23} z_{1323} - \frac{\mu_{23}}{\mu_{13}}\rho_{23} z_{2313} - \rho_{23} z_{2323} \le 0 \tag{7.4}$$

$$z_{1111} - \rho_{11} z_{1111} \ge 0 \tag{7.5}$$

$$z_{1313} + z_{2313} - \rho_{13} z_{1313} - \frac{\mu_{13}}{\mu_{23}}\rho_{13} z_{1323} - \rho_{13} z_{2313} - \frac{\mu_{13}}{\mu_{23}}\rho_{13} z_{2323} \le 0 \tag{7.6}$$

$$z_{2222} - \rho_{22} z_{2222} \ge 0 \tag{7.7}$$

$$z_{1323} + z_{2323} - \frac{\mu_{23}}{\mu_{13}}\rho_{23} z_{1313} - \rho_{23} z_{1323} - \frac{\mu_{23}}{\mu_{13}}\rho_{23} z_{2313} - \rho_{23} z_{2323} \le 0 \tag{7.8}$$

$$\rho_{11} \le 1 \tag{8.1}$$

$$\rho_{22} \le 1 \tag{8.2}$$

$$\rho_{13} + \rho_{23} \le 1 \tag{8.3}$$

$$z_{1111} + z_{1113} - \rho_{11} KB(1) = 0 \tag{9.1}$$

$$z_{1311} + z_{1313} - \rho_{13} KB(1) = 0 \tag{9.2}$$

$$z_{2211} + z_{2213} - \rho_{22} KB(1) = 0 \tag{9.3}$$

$$z_{2311} + z_{2313} - \rho_{23} KB(1) = 0 \tag{9.4}$$

$$z_{1122} + z_{1123} - \rho_{11} KB(2) = 0 \tag{9.5}$$

$$z_{1322} + z_{1323} - \rho_{13} KB(2) = 0 \tag{9.6}$$

$$z_{2222} + z_{2223} - \rho_{22} KB(2) = 0 \tag{9.7}$$

$$z_{2322} + z_{2323} - \rho_{23} KB(2) = 0 \tag{9.8}$$

$$\lambda_1\rho_{11} - \lambda_2\rho_{22} = \lambda_1 - \lambda_2 \tag{10.1}$$

We studied the example in four cases, each with the total number of kanbans fixed at ten but with different demand rates and expected processing times. In order to expose the effect of WIP allocation on the system, the parameters were set so that maximum throughput of each part would nearly equal its demand rate.

- Case 2.1. Balanced demand rates whose total matches the processing rate:

  $\lambda_1 = \lambda_2 = 50$, $\mu_{13} = \mu_{23} = 100$.

- Case 2.2. Unbalanced demand with total that matches the processing rate:

  $\lambda_1 = 70$, $\lambda_2 = 30$, $\mu_{13} = \mu_{23} = 100$.

- Case 2.3. Balanced demand and unbalanced processing rates:

  $\lambda_1 = \lambda_2 = 50$, $\mu_{13} = 150, \mu_{23} = 75$

- Case 2.4. Unbalanced demand and unbalanced processing rates:

  $\lambda_1 = 70$, $\lambda_2 = 30$, $\mu_{13} = 150, \mu_{23} = 75$.

The NLP was first solved without fixing the allocation of kanbans. Therefore, the numbers of kanbans allocated to each product type were decision variables, so that constraints (9.1) – (9.8) were nonlinear (in addition to the nonlinear constraints (7.1) – (7.8)).

Furthermore, we wished to verify these solutions against known results and test the sensitivity of the objective value to the allocation. We fixed the number of type 1 kanbans at values from 0 to 10 in order to see its effect on the maximum lost sales across product types. With $KB(1)$ and $KB(2)$ fixed, we solved the NLP and extracted the lost sales rates $\lambda_r \left(1 - \rho_{rf_r}\right)$ from the solutions, then recorded the maximum lost sales over $r = 1, 2$ as $\max\left(LS(1), LS(2)\right)$.

In cases 2.1 and 2.2, with identical processing rates for both product types, these lost sales rates can also be determined according to the M/M/1 multiclass lost demand FCFS model

of Buzacott and Shanthikumar (1993). The lost sales from this analysis can be computed by $\max(\lambda_1(1-SL_1), \lambda_2(1-SL_2))$, where SL_r is the service level of product type *r*. In cases 2.3 and 2.4 having unbalanced processing rates, we used simulation to find the lost sales from different allocations instead. The lost sales were observed at both finished goods stations.

Figures 8 and 9 compare the maximum lost sales predicted by the NLP against those obtained from the M/M/1 analysis and figures 10 and 11 compare the NLP against the simulation. The proportions of kanbans for type 1 are marked on the horizontal axis and the curves connect discrete points found for whole numbers of kanbans. The 'X' marked on the horizontal axis indicates the proportion of kanbans allocated to product type 1 by solving the NLP without fixing $KB(r)$. This yields the minimum of the maximum lost sales of product types.

[insert figure 8 about here]

[insert figure 9 about here]

[insert figure 10 about here]

[insert figure 11 about here]

Without fixing the kanban allocation, in the balanced demand cases 2.1 and 2.3, the virtually equal allocation of kanbans was selected, but in cases 2.2 and 2.4, type 1 products received a greater share of the WIP in order to meet higher demand rates. The proportion of kanbans allocated to each product type was close but not exactly equal to that product's share of the total demand. In addition, in cases 2.2, 2.3, and 2.4, where the NLP's suggested allocation differs from the one identified by the M/M/1 analysis or simulation, the true minimax lost sales

found by the M/M/1 analysis or simulation was relatively flat in an interval that contained the allocation found by the NLP.

Though the maximum lost sales predicted by the NLP is not completely accurate, this approach identifies an allocation point in a single step. In contrast, using the exact analytical method (where available) or simulation to evaluate lost sales across more than two product types would require overlaying a multidimensional search procedure to identify the best allocation. Note that, although there is no guarantee that the solution to the NLP is a global optimum, these results confirm that the NLP solution does identify the allocation that minimizes the maximum lost sales as estimated by the NLP. The larger differences in the NLP's maximum lost sales between the balanced (cases 2.1 and 2.3) and unbalanced demand rates (cases 2.2 and 2.4) implies that the NLP is rather sensitive to the differences in demand rates.

In the NLP, there is no integer restriction on $KB(r)$, so that $KB(r)$ can be any real number from 0 to N. To apply the result in the real-world situation where the number of kanbans is an integer, one could either choose the closest integer to the value of $KB(r)$ as the allocation of kanbans for product type r or allow this number of kanbans to alternate between the two closest integers over time.

4.3 Example 3.

The third example features repeat visits by the same products to the same stations, so that competition occurs not only between the product types but also between products of the same type at different stages of processing. Among the four processing stations, product type 1 visits both stations 2 and 4 twice, while product 2 revisits machines 3 and 4. The same product is assigned to a different customer class for the repeat visits. Therefore, this system, shown in

figure 12, has two products and four classes. The processing rates at stations 1 and 4 of product 1 were set to be twice those of product 2.

[insert figure 12 about here]

We studied the problem in six cases with different demand rates and WIP totals.

- Case 3.1. Balanced demand with low WIP: $\lambda_1 = \lambda_2 = 50$, $N = 10$.
- Case 3.2. Unbalanced demand with low WIP: $\lambda_1 = 70, \lambda_2 = 30$, $N = 10$.
- Case 3.3. Unbalanced high demand with low WIP: $\lambda_1 = 50, \lambda_2 = 100$, $N = 10$.
- Case 3.4. Balanced demand with high WIP: $\lambda_1 = \lambda_2 = 50$, $N = 20$.
- Case 3.5. Unbalanced demand with high WIP: $\lambda_1 = 70, \lambda_2 = 30$, $N = 20$.
- Case 3.6. Unbalanced high demand with high WIP: $\lambda_1 = 50, \lambda_2 = 100$, $N = 20$.

The processing rates were chosen to be high enough relative to the demand rates to avoid a bottleneck occurring at any processing station in cases 3.1, 3.2, 3.4, and 3.5 (i.e. the finished goods stations are the bottlenecks for the whole system). Cases 3.3 and 3.6 represent situations in which the combined demand rate exceeds the capacity of the system. The bottleneck of the system will occur at stations 1 and 3.

With the revisiting routes, there are no exact analytical methods for evaluating the performance of the queuing network. Again, we compared the NLP with simulation results. In addition to exponential processing times, we also simulated normal and uniform processing times, each with standard deviation equal to 10% of the mean. Figures 13 - 18 show that the maximum lost sales as a function of the kanban allocation varies considerably between the cases but the nonlinear program captures the effect quite accurately. As a result, the optimal allocation identified by the NLP is quite close to what would be selected based on the simulation results. In

cases 3.1, 3.2, 3.4 and 3.5, with relatively high demand, the optimal kanban allocation closely matches the demand ratio. In the cases 3.3 and 3.6 when the processing rates limit the overall throughput, the optimal kanban allocation is quite different. In case 3.3, the percent increase in lost sales from choosing demand-proportional rather than the optimal kanban allocation (rounded to whole numbers) is 7% for exponential, 11% for uniform, and 12% for normally distributed processing times. In case 3.6, these percent increases are 20%, 19% and 17%, respectively. Increasing the total number of kanbans from 10 to 20 decreases lost sales by up to 52% depending on their allocation, with the more dramatic decreases occurring close to the optimal allocations.

[insert figure 13 about here]

[insert figure 14 about here]

[insert figure 15 about here]

[insert figure 16 about here]

[insert figure 17 about here]

[insert figure 18 about here]

5. Conclusion

Though the benefits of limiting work in process inventory are well documented, the design of a multi-product CONWIP system poses the question of how to allocate the inventory among the product types. Managers may want to optimize the overall performance of the system, but they recognize that satisfying the demand for one product type while neglecting the customers of another product type is unsatisfactory. Testing the performance of alternative allocations is made more difficult by the complexity of evaluating the performance of a

multiclass queuing system, particularly one in which the same stations are visited repeatedly by the same customer for different types of service.

This paper presented a new approach for simultaneously evaluating the system performance and optimizing the kanban allocation. We extended the model obtained of (Ryan and Choobineh 2003) by adding finished goods stations to measure lost sales and formulating additional constraints to improve the accuracy of the NLP performance estimate. The additional constraints restrict the utilization of machines to be governed by a fixed number of type specific kanbans instead of the total amount of work in process. The NLP model performs well in example systems with both balanced and unbalanced demand rates and with different relationships between processing and demand rates. Although the examples presented had only two types of products for ease of presentation, the model can accommodate any number of product types.

Investing in more work in process results in better utilization and flexibility in the system; however, it will incur higher holding costs, which increase as the products progress to downstream operations. Future research will incorporate additional considerations such as the holding cost, processing cost and cost of lost sales and study the optimal total amount of work in process, in addition to its allocation among product types.

Acknowledgment: This work was supported by the National Science Foundation under grant DMI-9996373.

The Appendix

The NLP model with an adapted objective function has been verified by comparing with a heuristic control chart approach by Hopp and Roof (1998), which used an adaptive production

control method called Statistical Throughput Control (STC) to adjust WIP levels in order to satisfy a target throughput. The STC procedure starts by establishing the target throughput rate for each product. Then it decreases or increases the initial WIP by one card at a time to maintain the throughput of the system within upper and lower limits around the target throughput.

For less competitive environments where throughput is of more concern than lost sales, the NLP model can also be used to achieve specific throughput targets TH_r with minimal total inventory. For this purpose, we set the objective as

$$\text{Minimize } N. \tag{A1}$$

The lost sales equalities (10) are then replaced by a throughput constraint for each product, where the throughput is measured at some station s_r dedicated to that product. If there is a product for which no such station exists, a fictitious station with a very high processing rate could be included in the model.

$$\mu_{rs_r}\rho_{rs_r} \geq TH_r, \forall r \tag{A2}$$

In this system, two products are processed at seven stations, only one of which is shared by both products. The routings, shown in figure 19, are the same as in (Hopp and Roof 1998) and very similar to the extended model from (Kim and Van Oyen 1998). We used the parameters of (Hopp and Roof 1998) but assumed the processing time is exponentially distributed rather than deterministic. In order to compare with the STC method, we modified the NLP to minimize the number of kanbans necessary to achieve specified throughput targets, and incorporated the target throughput constraints (A2). Therefore N, $KB(1)$ and $KB(2)$ are all decision variables. The solution can be compared to the ending card counts found by the STC method.

[Insert figure 19 about here]

The comparison of the ending card counts is shown in table 1. For the STC method, the ending card counts are the values obtained from a deterministic simulation in (Hopp and Roof 1998) starting from the initial number of cards shown. In some cases, the ending card count does not stabilize but instead cycles between two numbers over time. The optimal numbers of cards from the NLP are similar to the ending numbers of cards found by STC. The differences possibly are due to our assumption of stochastic rather than deterministic processing times. However, the results suggest that the impact of this assumption is relatively minor.

[Insert table 1 about here]

References

ALTIOK, T. and SHIUE, G. A., 1995, Single-stage, multi-product production/inventory systems with lost sales. *Naval Research Logistics*, **42**, 889-913.

ANDERSSON, J. and MELCHIORS, P., 2001, A two-echelon inventory model with lost sales. *International Journal of Production Economics*, **69**, 307-315.

BAYNAT, B., BUZACOTT, J. A. and DALLERY, Y., 2002, Multiproduct kanban-like control systems. *International Journal of Production Research*, **40**, 4225-4255.

BERTSIMAS, D., PASCHALIDIS, I. C. and TSITSIKLIS, J. N., 1994, Optimization of multiclass queuing networks: polyhedral and nonlinear characterizations of achievable performance. *The Annals of Applied Probability*, **4**, 43-75.

BONVIK, A. M., COUCH, C. E. and GERSHWIN, S. B., 1997, A comparison of production-line control mechanisms. *International Journal of Production Research*, **35**, 789-804.

BUZACOTT, J. A. and SHANTHIKUMAR, G. J., 1993, *Stochastic Models of Manufacturing Systems* (Englewood Cliffs, NJ: Prentice Hall).

CHEVALIER, P. B. and WEIN, L. M., 1993, Scheduling networks of queues: heavy traffic analysis of a multistation closed network. *Operations Research*, **41**, 743-757.

DUENYAS, I., 1994, A simple release policy for networks of queues with controllable inputs. *Operations Research*, **42**, 1162-1171.

FRAMINAN, J. M., RUIZ-USANO, R. and LEISTEN, R., 2000, Input control and dispatching rules in a dynamic CONWIP flow-shop. *International Journal of Production Research*, **38**, 4589-4598.

GSTETTNER, S. and KUHN, H., 1996, Analysis of production control systems kanban and CONWIP. *International Journal of Production Research*, **34**, 3253-3273.

HARRISON, J. M. and WEIN, L. M., 1990, Scheduling networks of queues: heavy traffic analysis of a two-station closed network. *Operations Research*, **38**, 1052-1064.

HOPP, W. J. and ROOF, M. L., 1998, Setting WIP levels with statistical throughput control (STC) in CONWIP production lines. *International Journal of Production Research*, **36**, 867-882.

KIM, E. and VAN OYEN, M. P., 1998, Dynamic scheduling to minimize lost sales subject to set-up costs. *Queuing Systems*, **29**, 193-229.

KRISHNAMURTHY, A., SURI, R. and VERNON, M. K., 2004, Re-examining the performance of MRP and kanban material control strategies for multi-product flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, to appear.

KUMAR, S. and KUMAR, P. R., 1994, Performance bounds for queuing networks and scheduling policies. *IEEE Transactions on Automatic Control*, **39**, 1600-1611.

RYAN, S. M., BAYNAT, B. and CHOOBINEH, F. F., 2000, Determining inventory levels in a CONWIP controlled job shop. *IIE Transactions*, **32**, 105-114.

RYAN, S. M. and CHOOBINEH, F. F., 2003, Total WIP and WIP mix for a CONWIP controlled job shop. *IIE Transactions*, **35**, 405-418.

SPEARMAN, M. L., 1992, Customer service in pull production systems. *Operations Research*, **40**, 948-958.

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.

SPEARMAN, M. L. and ZAZANIS, M. A., 1992, Push and pull production systems: issues and comparisons. *Operations Research*, **40**, 521-532.

SRINIVASAN, M. M., EBBING, S. J. and SWEARINGEN, A. T., 2003, Woodward Aircraft Engine Systems sets work-in-process levels for high-variety, low-volume products. *Interfaces*, **33**, 61-69.

FIGURES

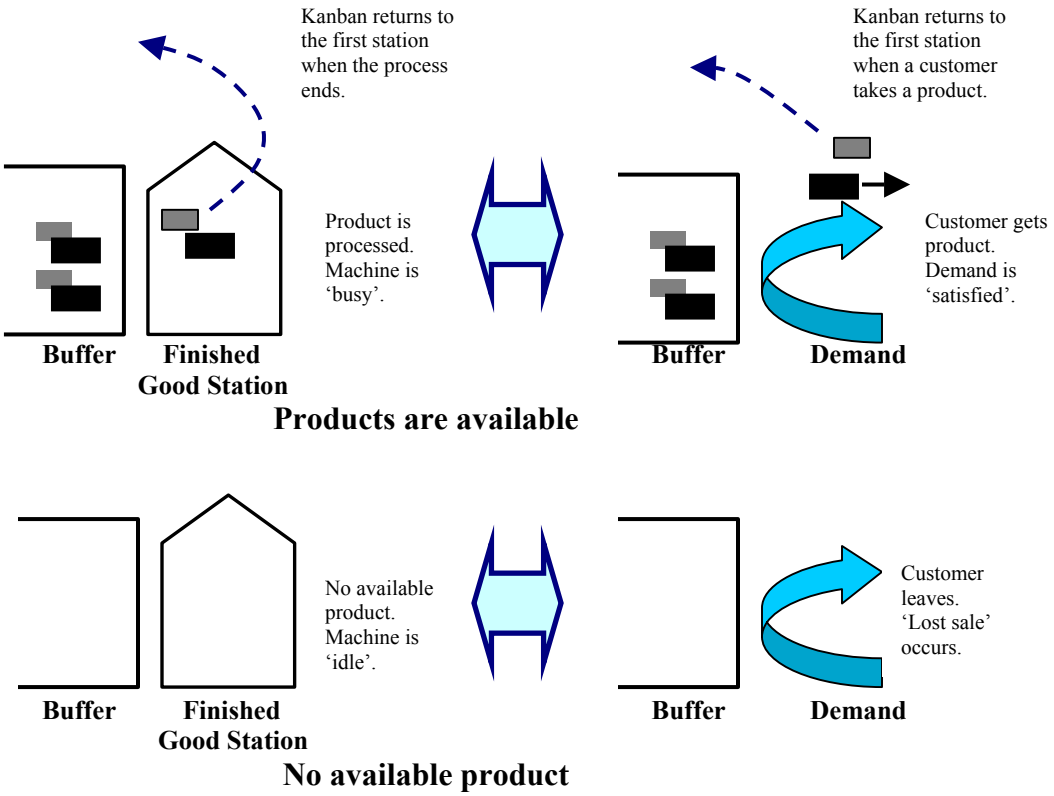

Figure 1. Satisfied demand vs. lost sales at finished goods station

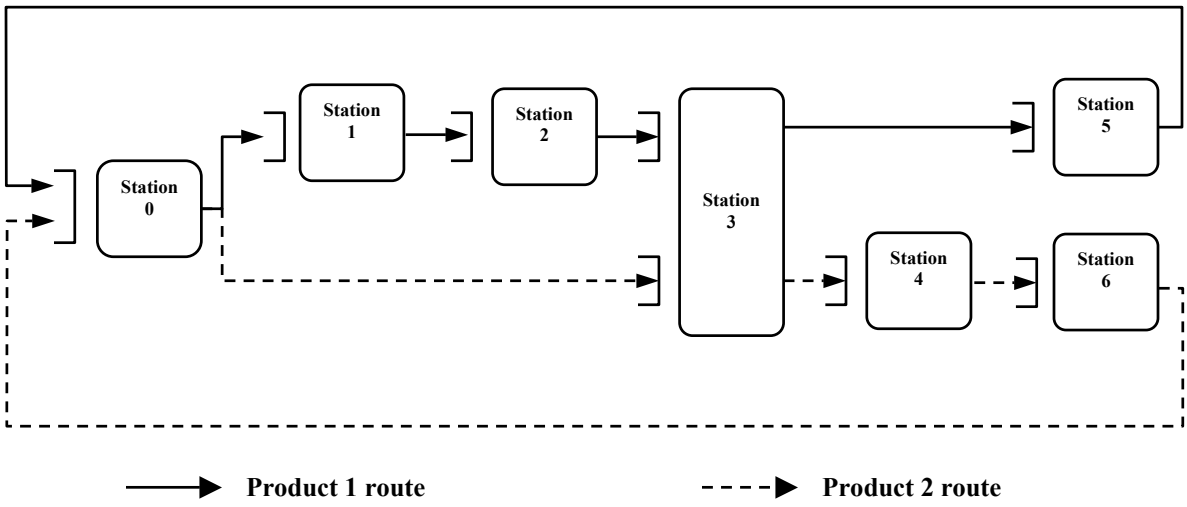

Figure 2. Shared kanban system

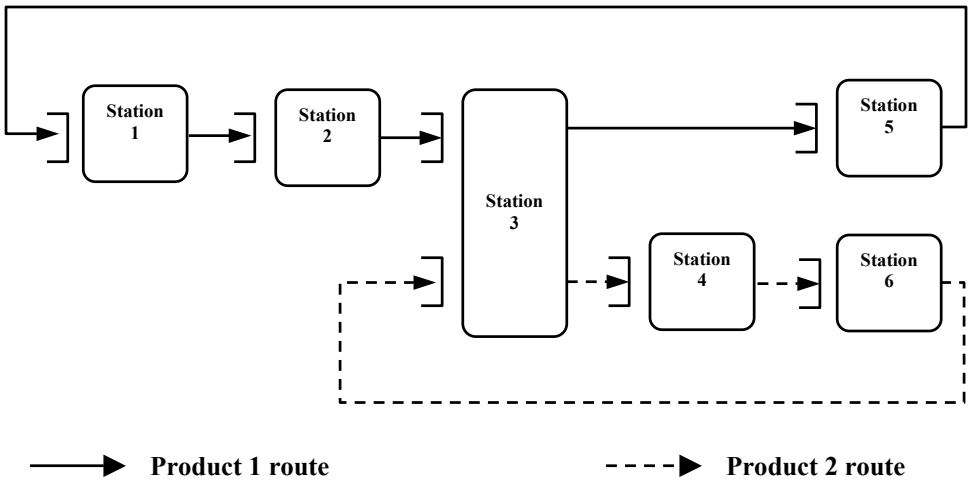

Figure 3. Dedicated kanban system

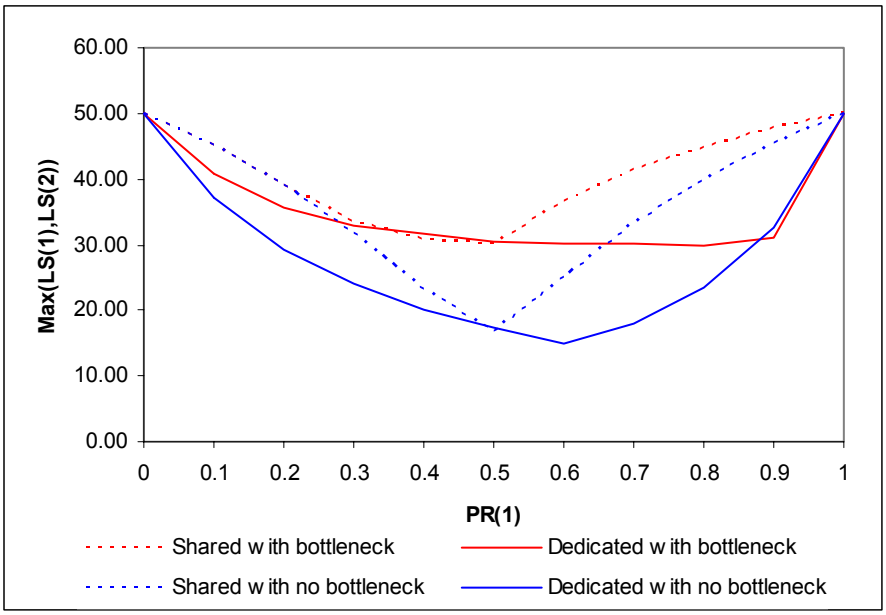

Figure 4. Maximum lost sales from simulation for Example 1.

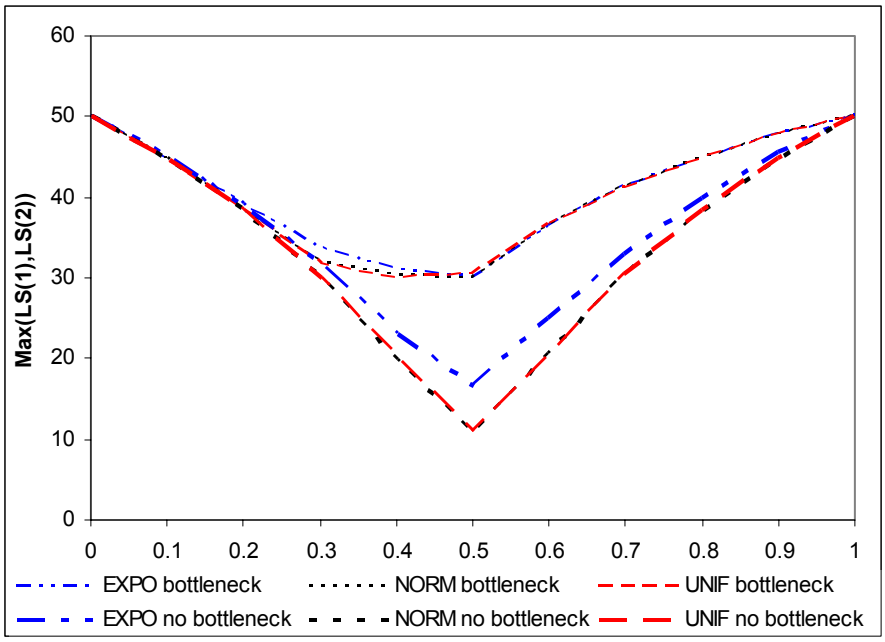

Figure 5. Maximum lost sales of different processing time distributions for Example 1 with shared kanbans.

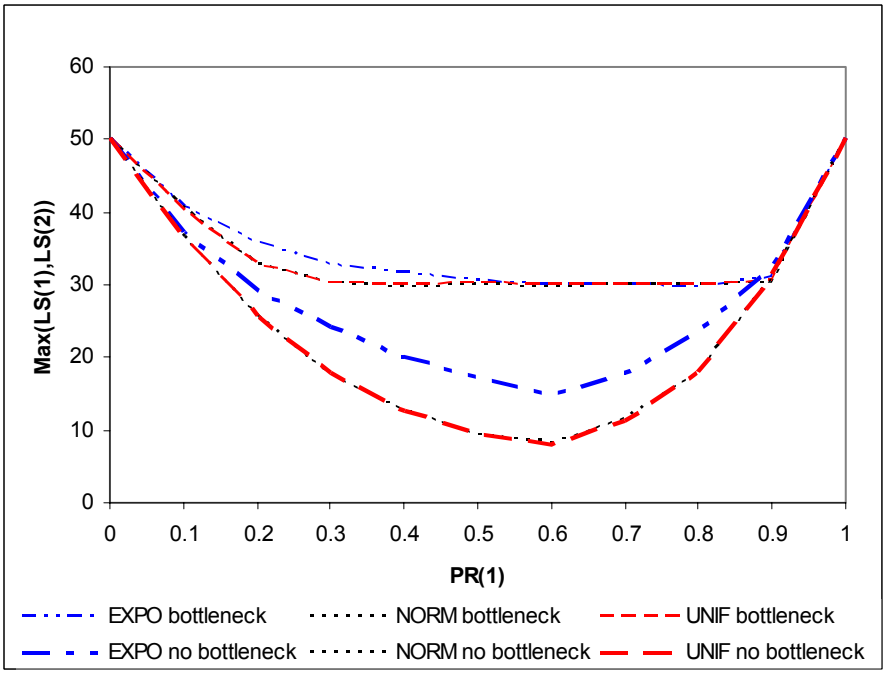

Figure 6. Maximum lost sales of different processing time distributions for Example 1 with dedicated kanbans.

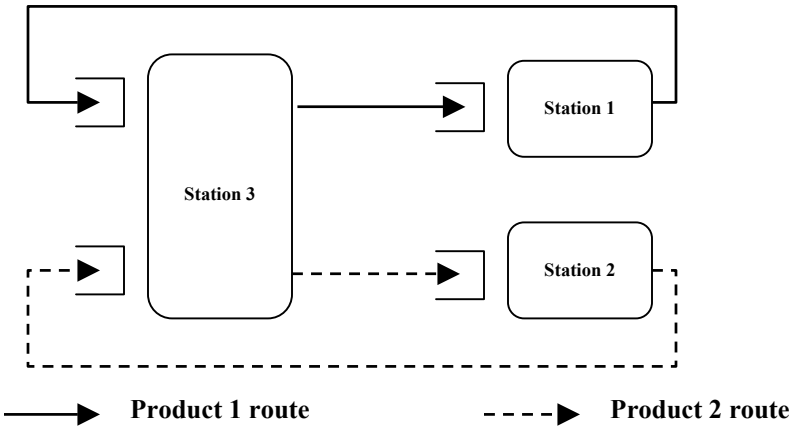

Figure 7. The routings for Example 2.

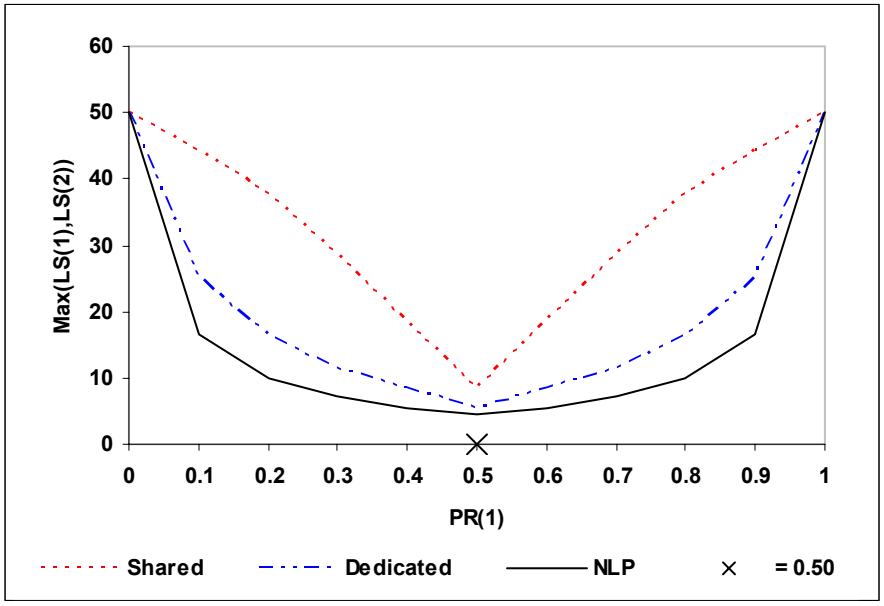

Figure 8. Maximum lost sales from M/M/1 and NLP model with different PR(1) for case 2.1

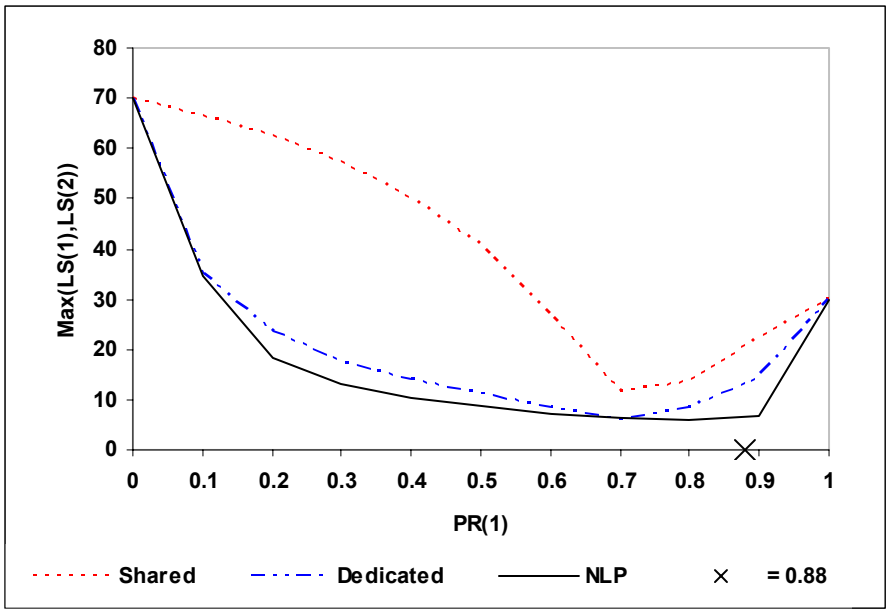

Figure 9. Maximum lost sales from M/M/1 and NLP model with different PR(1) for case 2.2

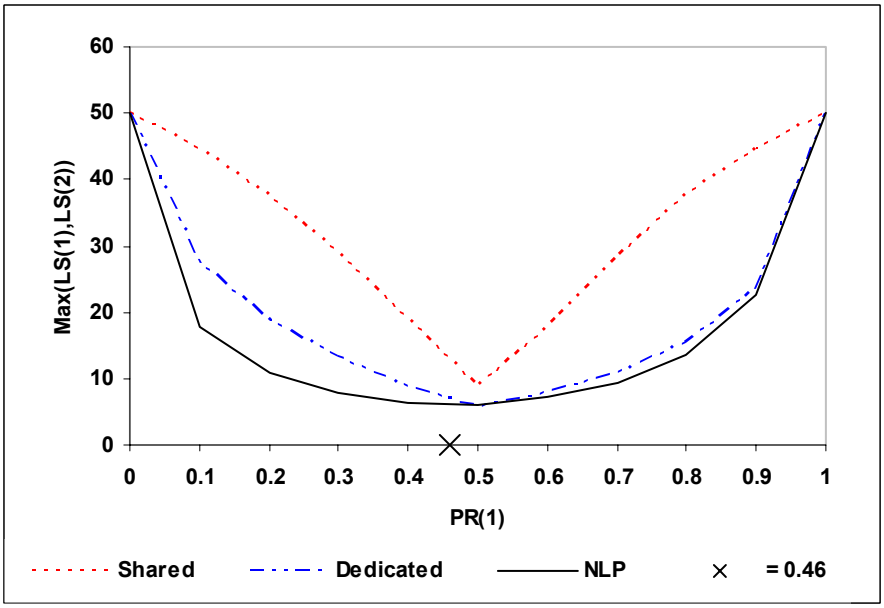

Figure 10. Maximum lost sales from simulation and NLP model with different PR(1) for case 2.3

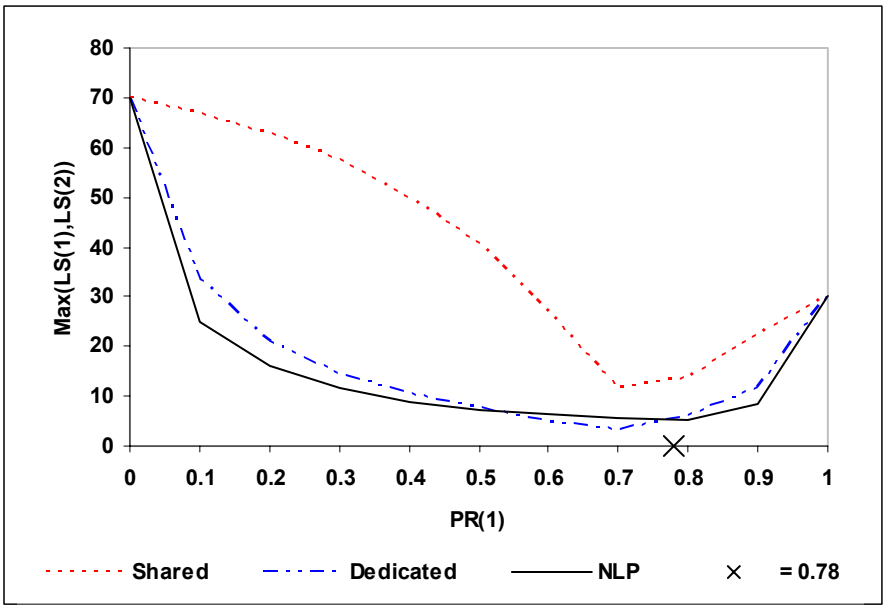

Figure 11. Maximum lost sales from simulation and NLP model with different PR(1) for case 2.4

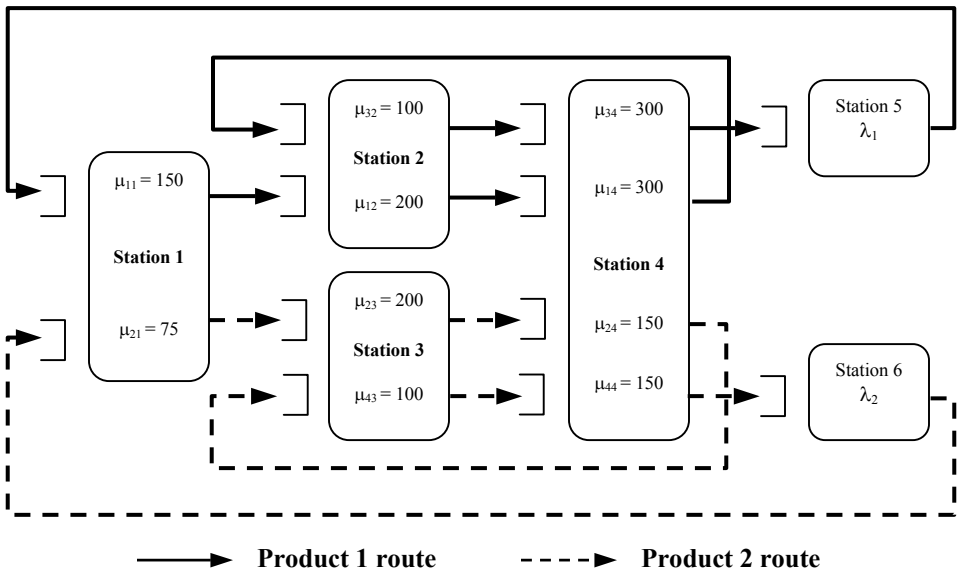

Figure 12. System for example 3.

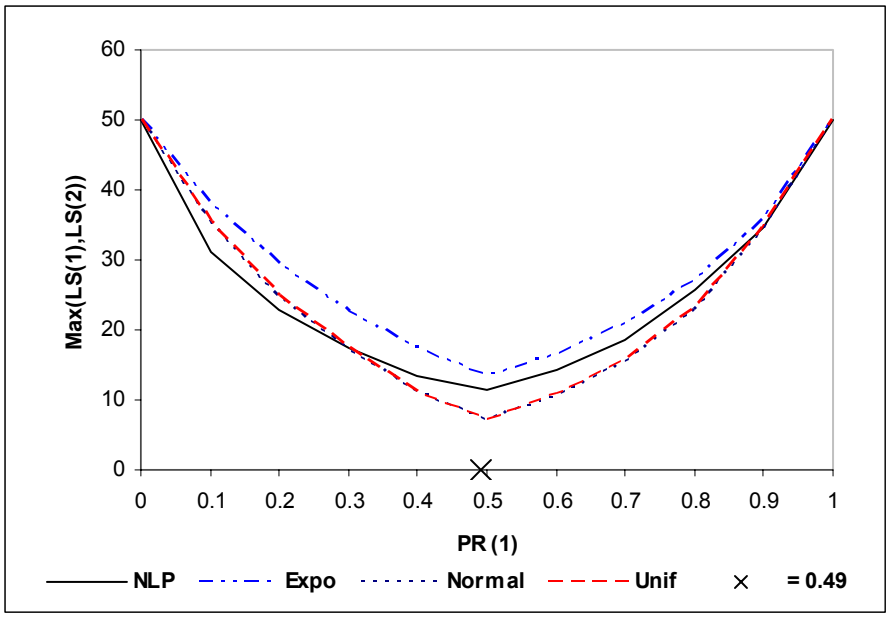

Figure 13. Maximum lost sales from simulation and NLP with different PR(1) for case 3.1

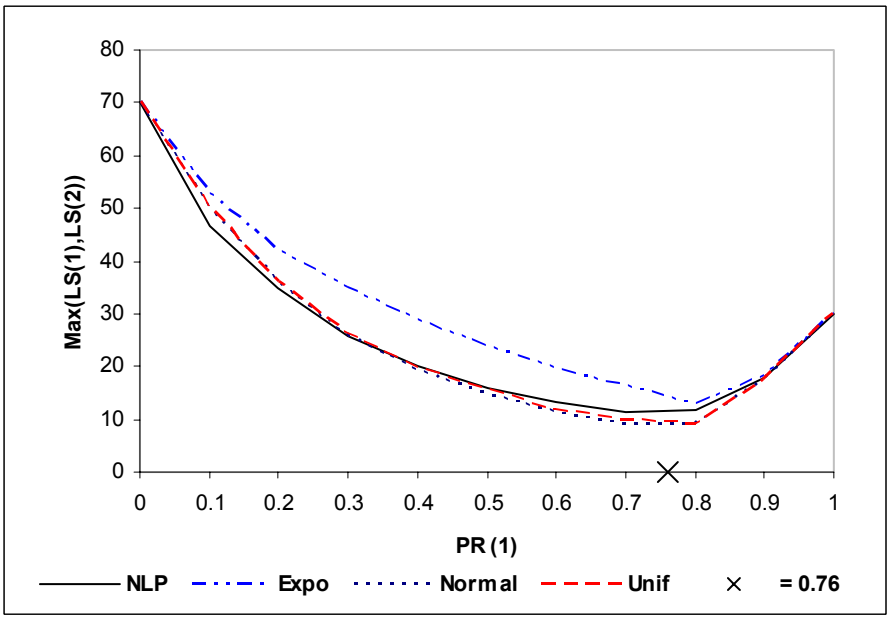

Figure 14. Maximum lost sales from simulation and NLP with different PR(1) for case 3.2

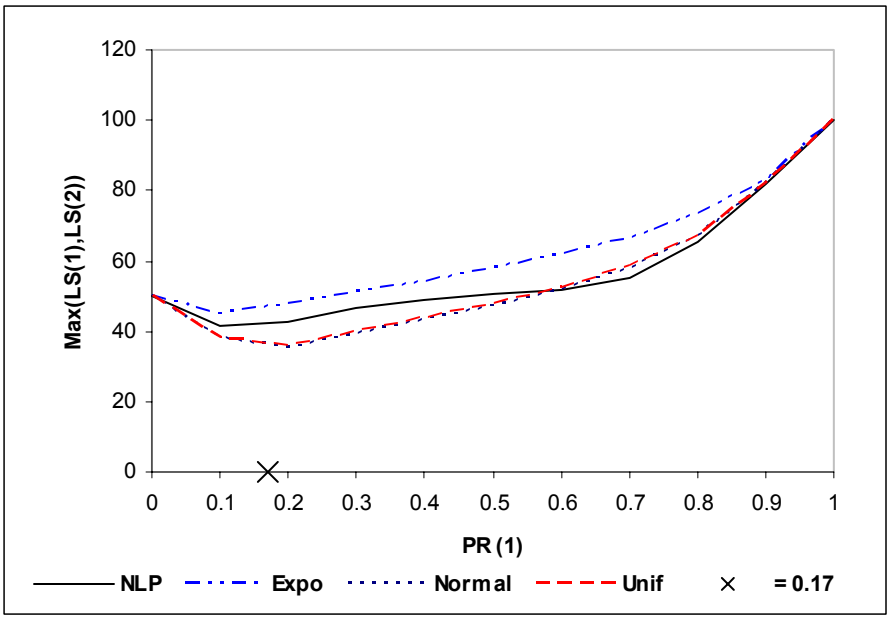

Figure 15. Maximum lost sales from simulation and NLP with different PR(1) for case 3.3

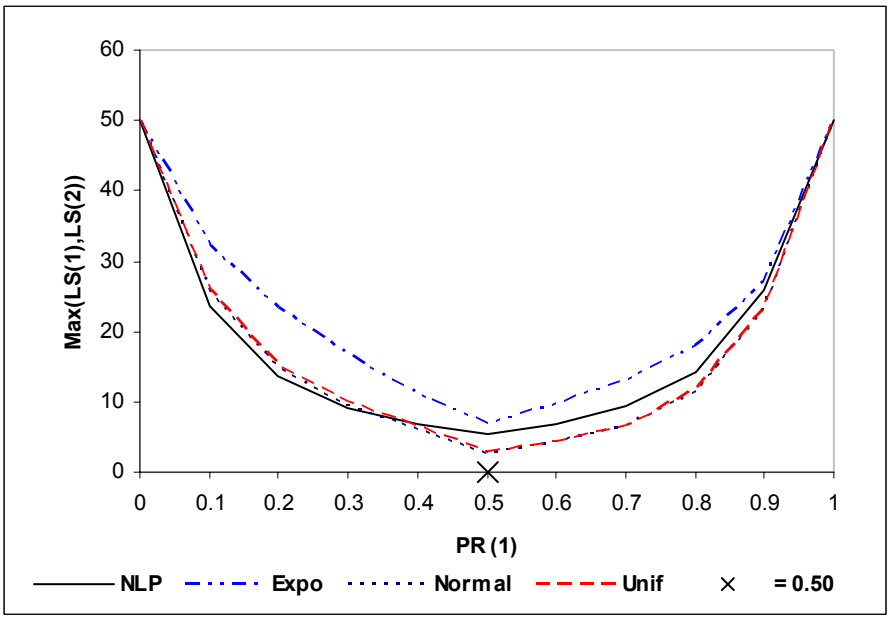

Figure 16. Maximum lost sales from simulation and NLP with different PR(1) for case 3.4

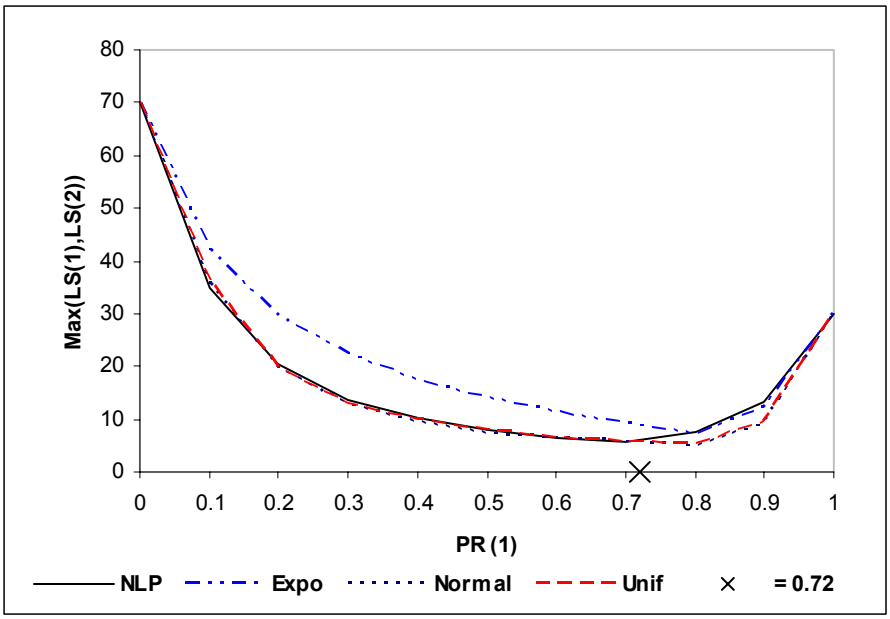

Figure 17. Maximum lost sales from simulation and NLP with different PR(1) for case 3.5

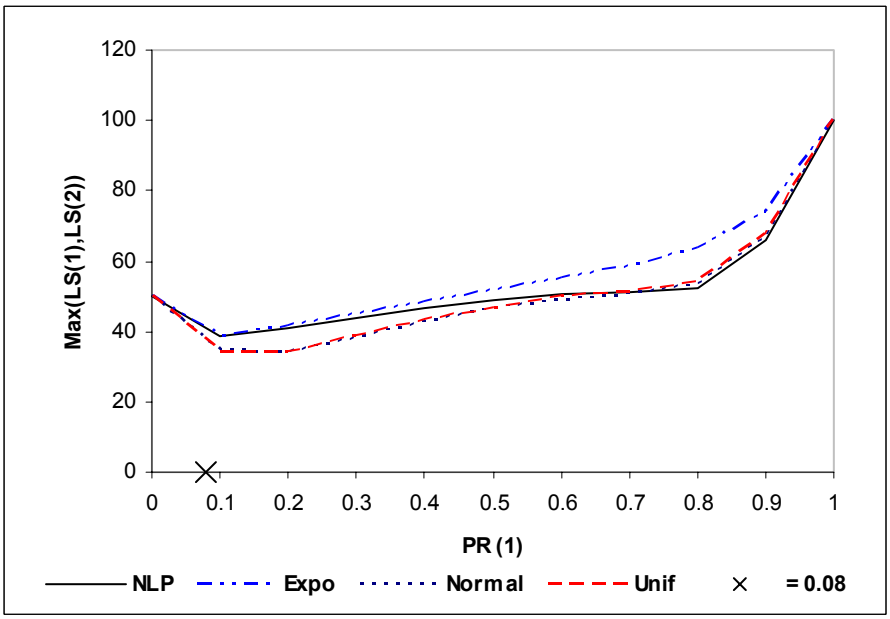

Figure 18. Maximum lost sales from simulation and NLP with different PR(1) for case 3.6

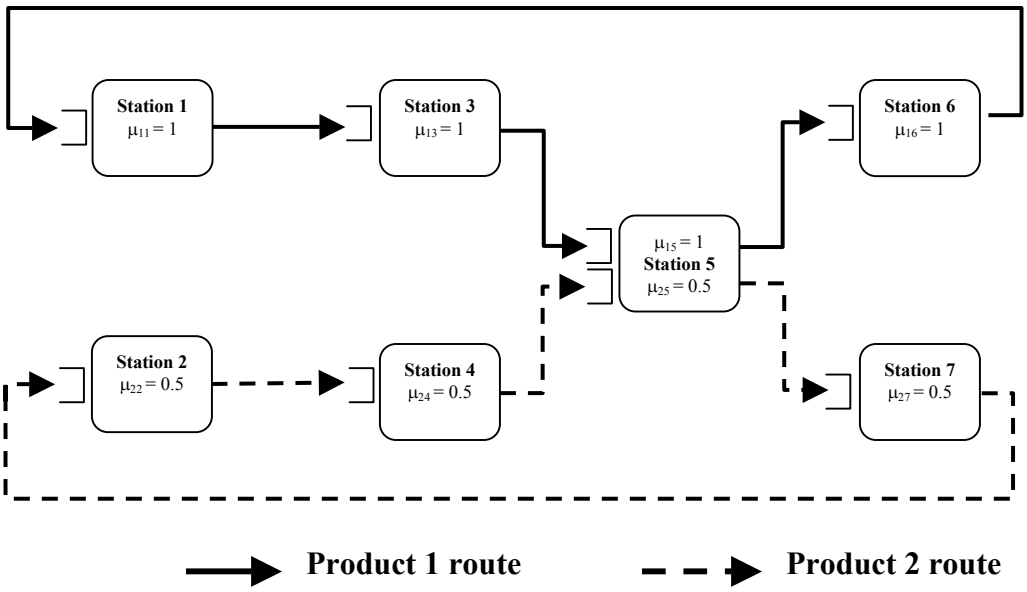

Figure 19. System for Example 4

TABLE

Case	Target TH		Initial card count		Ending card count		Optimal number of kanbans	
					Hopp and Roof (1998)		This paper	
	Product 1	Product 2	Product 1	Product 2	Product 1	Product 2	Product 1	Product 2
Case 1	0.400	0.250	4	4	2-3	2	1.87	2.00
Case 2	0.400	0.250	1	1	2-3	2	1.87	2.00
Case 3	0.400	0.250	4	1	2-3	2	1.87	2.00
Case 4	0.400	0.250	1	4	2-3	2	1.87	2.00
Case 5	0.250	0.250	5	4	1-2	2	1.08	2.00
Case 6	0.600	0.125	5	4	3	1-2	2.55	1.00
Case 7	0.200	0.375	5	4	1-2	3	1.00	3.00

Table 1. Initial and ending card count from Hopp and Roof (1998) compared with the NLP's minimum number of kanbans to achieve the target throughputs