Designing pull production control systems

Gaury, E.G.A.

Publication date:
2000

Document Version
Publisher's PDF, also known as Version of record

CentER

Designing Pull Production Control Systems: Customization and robustness

Eric Gaury

DESIGNING PULL PRODUCTION CONTROL SYSTEMS: CUSTOMIZATION AND ROBUSTNESS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Katholieke Universiteit Brabant, op gezag van de rector magnificus, prof. dr. F.A. van der Duyn Schouten, in het openbaar te verdedigen ten overstaan van een door het college voor promoties aangewezen commissie in de aula van de Universiteit op

vrijdag 10 maart 2000 om 14.15 uur

door

ERIC GEORGES ANTOINE GAURY

geboren op 29 juni 1973 te Dijon, Frankrijk

PROMOTORES: Prof. Dr. J.P.C. Kleijnen
Prof. Dr. H. Pierreval

This research was performed under the auspices of the Laboratory of Computer Science for Modeling and Optimization of Systems (LIMOS) at Blaise Pascal University (UBP), Clermont-Ferrand, France, and the Center for Economic Research (CentER), Tilburg University, the Netherlands.

N° d'ordre : 1196
EDSPIC : 212

UNIVERSITÉ BLAISE PASCAL – CLERMONT II

ECOLE DOCTORALE
SCIENCES POUR L'IGÉNIEUR DE CLERMONT-FERRAND

Thèse présentée par
ERIC GEORGES ANTOINE GAURY

pour obtenir le grade de
DOCTEUR D'UNIVERSITÉ
Spécialité : Informatique / Productique

DESIGNING PULL PRODUCTION CONTROL SYSTEMS: CUSTOMIZATION AND ROBUSTNESS

Soutenue publiquement à Tilburg, Pays-Bas, le vendredi 10 mars 2000
devant le jury composé de

Prof. dr. J. ASHAYERI
Prof. dr. Y. DALLERY
Prof. dr. D. den HERTOG
Dr. F. ROUBELLAT

Prof. dr. J. KLEIJNEN
Prof. dr. H. PIERREVAL

Contents

Table of figures

List of tables

Preface

This book is the outcome of a few years work and a rather complex process. It all started four years ago (1996) when I was given the opportunity to spend a full year abroad, as part of my engineering cursus at the French Institute of Advanced Mechanical Engineering (IFMA) in Clermont-Ferrand, France. As part of that year I spent six months at Tilburg University under the supervision of Professor Jack Kleijnen. My supervisor at IFMA, Professor Henri Pierreval, knew Jack Kleijnen through conferences and common interests in simulation and regression analysis. The goal of my stay was to identify research topics that could exploit both the competence of the IFMA research team in Operations Research (OR) issues in a manufacturing context and in computer science, and the skills of the Tilburg OR group in OR tools and methodologies. One area that we felt might lead to interesting issues, was pull production control. The six month research showed that this area was indeed promising. My Ph.D. research, however, started only a year later (1997), after my graduation at IFMA, as I took the opportunity to spend my military service as a researcher in Tilburg. The collaboration between the French and the Dutch teams became official through the cosupervision of my Ph.D. by Jack Kleijnen and Henri Pierreval.

Dealing with educational and administrative systems of two countries turned out not to be an easy task. Issues related to my military service further added to the complexity of the situation. Thus, the completion of my Ph.D. would never have been possible without the help and support (financial as well as administrative) of many people and institutions. Therefore, I would like to thank:

- The French Embassy in The Hague and particularly D. Pladys, scientific representative, for supporting my research during the military service and for partly funding the travel expenses of the French committee members.
- Cees Verhoeven, former Personnel Officer at Tilburg University for his help in finding a way to satisfy the financial constraints for my military service.
- Michel Schneider, Director of Graduate Studies in Computer Science, Operations Research, and Medical Imaging at Blaise Pascal University (UBP), Clermont-Ferrand, and Jeffrey James, former Director of Graduate Studies at the Center for Economic Research (CentER), Tilburg University, for accepting the idea of a cosupervised Ph.D. and for their help in dealing with the related paperwork.
- CentER and LIMOS (Laboratory of Computer Science for Modeling and Optimization of Systems) for their financial support since 1996.
- Claude Bonthoux, former Director of IFMA, for allowing me to perform part of my research at IFMA.
- The secretaries of all the institutions (UBP, IFMA, CentER, BIK) for their assistance in solving all kind of daily problems.

I am particularly indebted to my two supervisors, Jack Kleijnen and Henri Pierreval, who never gave up – despite the amount of administrative files to be completed and the

complexity of the situation; I hope they will not miss too much the last minute rushes and express mails between France and the Netherlands. I most appreciated their openness to each other's views and the broadness of their competencies. They are undoubtedly the core of my success.

I am grateful to all my colleagues in France and in the Netherlands for their support, and for their stimulating comments on the various papers we 'committed' and the many versions of my dissertation. I also thank the members of my committee for accepting to evaluate my work, despite the many constraints, and for their valuable comments. I particularly appreciated the extra efforts made by the French members to provide specific reports to UBP and to come to Tilburg for my defense.

Many thanks to numerous friends from all around the world, for helping me to keep a healthy balance between work and leisure. Of course, my frequent moving from one place to another caused many difficult good-byes, but the value of the souvenirs and shared experiences accumulated during more than four years is inestimable. I owe you a lot.

Last but not least, I especially thank my parents for their constant support and for not questioning what I was doing.

Eric Gaury
Paris, January 2000

Chapter 1
Introduction

1.1 Production control and inventory management

The objective of production control is to satisfy customer demands in terms of type of product, amount, and delivery time; that is, production control deals with three issues, namely, which type to produce, in which amount, and at what date. There are many ways of meeting this objective, but at different costs. Demand satisfaction may easily be achieved by keeping large amounts of finished products. This solution, however, may not be practical because keeping large amounts of inventory has many drawbacks (see section 1.2). Thus, inventory management is a necessary activity supporting production control. Without management tools such as production control and inventory management, the efficiency of manufacturing systems would be endangered.

Until the late 70s, many manufacturing systems in the US were controlled through MRP (MRPI: Material Requirements Planning, or MRPII: Manufacturing Resource Planning). Next, authors reported the results obtained through Just-In-Time (JIT) in Japan, especially at Toyota (Sugimori *et al.* 1977, Schonberger 1982, Monden 1993). JIT is Toyota's philosophy of minimizing waste. MRP and JIT are often opposed, which is emphasized by the reduction of MRP to push and JIT to pull (see section 2.1 for a further discussion of push vs. pull). In practice, however, most manufacturing systems are controlled through integrated MRP/JIT systems. A thorough discussion of integrated MRP/JIT issues is given in Benton and Shin (1998).

1.2 JIT and the benefits of inventory reduction

Many western industries have adopted JIT tools as best practice. This success in industry has raised a large interest in the research community. Thus, more than 800 articles related to JIT have been published (Golhar and Stamm, 1991). The reason for this tremendous interest in JIT is that waste reduction can be very fruitful: it is often acknowledged that the amount of time spent by a product in a factory is due for 10% only to value adding activities, and for 90% to handling, storage, quality control, etc. The concepts embodied in

JIT include total quality, continuous improvement (Kaizen), employee involvement, inventory reduction, and pull production.

JIT emphasizes inventory reduction. Indeed, JIT sees inventory as an evil in many aspects. First, inventory is an investment; it has a financial cost that affects price competitiveness. Second, it is used to hide problems such as defective quality of products, process inefficiencies (breakdowns, long set-up times, large batch sizes, etc.), difficulties in respecting due dates, etc. Third, inventory is a source of lack of quality: for instance, products kept in inventory become obsolete when new products are introduced; higher cost results in case of late detection of a defect. Inventory building-up characterizes "just-in-case management".

Often, people working on JIT use the metaphor of a company floating as a boat on a sea of inventory (see Figure 1). Lowering the inventory (sea) level creates difficulties (uncovers rocks), which lead to perturbations (on which the boat may crash). The idea emphasized by the JIT philosophy is that the sea level should be gradually lowered, and uncovered rocks should be removed. The JIT technique for lowering inventory is pull production, and the technique for removing rocks in continuous improvement.

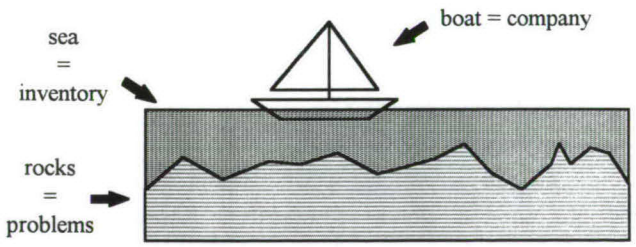


Figure 1. Metaphor of a company as a boat floating on a sea of inventory

1.3 Outline of the dissertation

This thesis is concerned with the design of pull control systems for single product flowlines. We further limit the scope of this research to Make-To-Stock systems. We shall make one main assumption: we consider internal production flows only, that is, we assume that the supply of raw materials and components is continuous and infinite.

The outline of this thesis is the following. In Chapter 2 we review pull control systems developed in the literature, and we propose a new classification. Two issues arise, namely, which type of pull systems should managers choose, and how should they set the various parameters of the chosen system. These design issues will be our main concern. We distinguish several formulations of our design problem depending on the assumptions we make when modeling the production environment (which we shall define more precisely in the Chapters 3 to 6: in Chapters 3 and 4 we consider the production environment as given,

whereas in Chapters 5 and 6 we study production environments that are not known with certainty and may show dynamic behavior). We conclude Chapter 2 by emphasizing the need for new design approaches.

In Chapter 3 we show that selecting a specific pull system among all possible pull systems is a complex problem, which has not been investigated in the literature. Our contribution is the design of a generic model, that is, a common representation of all the pull systems presented in Chapter 2. We propose a procedure based on evolutionary computation and simulation to configure the generic system for a given production system and production environment; we call this procedure *customization*. The result shows which pull system should be implemented. The benefits of customizing are illustrated for an example production system taken from the literature, for which the optimal configurations of several known pull systems have been determined in the past: we find a pull system that performs significantly better than the best system in the literature.

In Chapter 4 we gain more insight into customization and its benefits by applying our methodology to a variety of production lines. We review the pull literature to determine this variety, and use experimental design to generate a sample of twelve production line configurations. For each production line we apply the customization methodology proposed in Chapter 3. The results provide many conclusions concerning the best pull structures, their performance, and their complexity.

In Chapter 5 we identify three sources of uncertainty that may arise when designing pull systems through simulation: (i) stochastic uncertainty, which is due to the use of (pseudo)random numbers in our discrete-event simulation, (ii) subjective uncertainty, which results from our need to model stochastic behaviors through probability distributions based on either sampled data or expert opinions, and (iii) dynamic uncertainty, which is resulting from variations over time in the real production environment. Through simple examples we illustrate the possible effects of these three sources of uncertainty, and we emphasize the need for assessing and integrating the effects of these uncertainties in the design process. We contribute to this issue by proposing a novel procedure based on Uncertainty/Risk Analysis (URA) and Taguchi's robust design.

In Chapter 6 we apply our procedure to the design of pull systems under uncertainty. We specify two robustness criteria – one based on service and the other one based on Work In Process (WIP) – and we give a rigorous definition of the robust customization problem. Then, we consider the issue of comparing the robustness of pull systems. We study the relative performance of four pull systems using two comparison procedures, namely, stochastic dominance and confidence ellipsoids built through bootstrapping. We conclude that a control system can be selected only if managers specify their attitude towards risk and characterize their preferences. To support managers, we investigate the effects of the various parameters within their control (the card numbers, the type of probability distributions used in URA, and various parameters that specify the managers' attitude

towards risk and characterize their preference). We apply the complete robust customization procedure to the production system studied in Bonvik *et al.* (1997).

In Chapter 7 we summarize the main conclusions of this thesis, and give research perspectives.

Chapter 2
Pull Control Systems: Classification and Selection

Abstract

In this chapter we develop a new classification of the pull systems that have already been proposed in the literature. We identify three classes of pull systems: traditional, segmented, and joint. Traditional pull systems are Kanban, Conwip, and Base Stock. Segmented systems partition the production line into segments, each controlled through a traditional pull system. Joint systems combine several traditional pull systems on the same segment of the production line. Managers willing to implement pull control in their production systems, have to deal with two main issues: which type of pull systems to choose, and how to set the various parameters of the chosen system. These are the two design issues that we consider throughout the whole dissertation. We briefly review how these two design issues are treated in the literature: we study which performance measures are used, how parameters are set, and how pull systems are compared. We conclude by showing the need for new design approaches.

2.1 Typology of pull control systems in the literature

The principle of *pull control* is described as follows in Spearman and Zazanis (1992): "A pull system is characterized by the practice of downstream work centers pulling stock from previous operations, as needed. All operations then perform work only to replenish outgoing stock. Work is coordinated by using some sort of signal (or *Kanban*) represented by a card or a sign." We consider Make-to-Stock systems, so the pulling signal at the last stage is released when a finished product is delivered to customers (in Make-to-Order systems the release of a new order would occur as an order is completed). Many publications consider infinite demand, which results in order completion and product delivery occurring at the same time. In pull systems, a work center may be blocked either because it is starving – no parts in the input inventory – or because it is not allowed to produce – all cards are attached to products and they will be released only if the downstream work center uses one of these products.

Figure 2 shows a possible implementation of the pull principle in a production line. When machine $i + k$ pulls a product from its input inventory, it releases the card attached to the part and sends this card to an upstream machine i – not necessarily the immediately preceding machine (i and k positive integers). This card allows machine i to start production, that is, pull a product from its input inventory, attach the card to this product, and start processing. Therefore the number of cards that circulates in a specific control loop, remains constant over time. This number determines the maximum Work-In-Process (WIP) in the production line segment controlled by the loop.

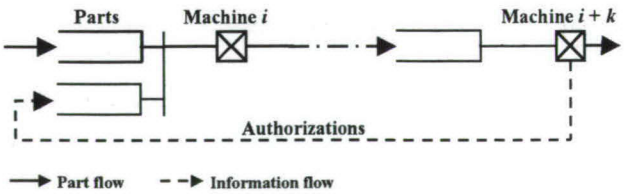


Figure 2. Pull principle illustrated through a simple line of queues

Pull control is often opposed to *push control*. There is no generally accepted definition of push. Many researchers, however, define push control as based on demand forecasts to schedule production. For instance, Spearman *et al.* (1990) defines push systems as those 'where production jobs are scheduled' and pull systems as those 'where the start of one job is triggered by the completion of another'. Ou and Jiang (1997) emphasize that '[push] controls throughput by establishing a master production schedule and keeps record of WIP to detect problems in meeting the schedule... [pull] controls WIP and adjusts throughput to match the required demand... [:] the work center works only to replace the number of items pulled'. For Amin and Altiok (1997), '[production in a push system] is triggered in an

upstream part of the system based on the demand forecasts'; in a pull system the market conditions directly control the production schedules. They conclude that push systems seem to emphasize throughput, whereas pull systems emphasize WIP inventories (JIT goal: zero inventory).

Many types of pull systems have been developed. The topic of the next sections is our own classification of these systems. Throughout this dissertation we focus on production lines processing a single part type. We identify three classes: traditional, segmented, and joint. Traditional pull systems are Kanban, Conwip, and Base Stock. Segmented systems partition the production line into segments, each controlled through a traditional pull system. Joint systems combine several traditional pull systems on the same segment of the production line.

2.1.1 Three traditional pull systems

A few pull control systems are widely used in industry. They were often used in practice, before appearing in the research literature, where they are known under the names Kanban, Conwip, and Base stock. Next, we describe these three traditional systems.

2.1.1.1 Kanban

The Kanban strategy was developed by Dr. Taichi Ohno, manager at the Toyota Motors company. The principle is to limit the inventory level at each stage of a process, by defining control loops between each pair of consecutive stages (Monden, 1993); see Figure 3. There are many implementation forms for Kanban. Berkley (1992) proposes a classification of Kanban models; he uses operational design criteria, such as the blocking mechanism, the withdrawal strategy, and the type of Kanban cards. Huang and Kusiak (1996) survey various Kanban implementations and alternative pull systems, and classify previous studies. Chu and Shih (1992) compare numerous simulation studies on Just-In-Time (JIT) production systems. Price *et al.* (1994) review optimization models of Kanban systems. Sing and Brar (1992)'s review considers several issues, such as design, modeling, scheduling, and comparisons with other control systems. Gaury *et al.* (1997a) survey the way modeling techniques and simulation are used to study Kanban systems.

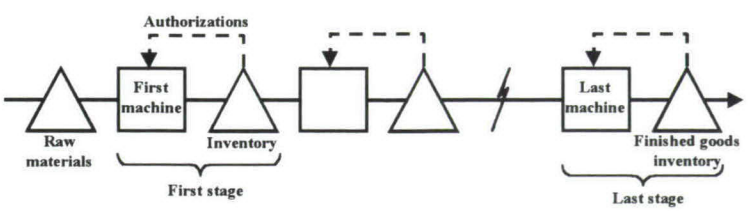


Figure 3. Kanban system

2.1.1.2 Conwip

Conwip stands for Constant Work In Progress. Spearman *et al.* (1990) proposed the name Conwip, but Bertrand (1983) and Lambrecht and Segaert (1990) proposed similar approaches under the names of workload control and long-pull systems respectively. These approaches can be considered as capacity-based order review/release (ORR) strategies; see Philipoom and Fry (1992) for a short review of ORR strategies.

The objective of Conwip is to combine the low inventory levels of Kanban with the high throughput of Push. To achieve this objective, Conwip uses a Push system that, however, has only a limited number of parts allowed into the production system: raw materials can be released into the system only when the last stage asks for it (Pull principle). This limitation is implemented through a single control loop that links the last stage to the first one. As explained in the introduction of section 2.1, we consider Make-to-Stock systems only, so card flows along the control loops are triggered by the delivery of products to customers. Though Conwip was originally designed for Make-to-Order systems, recent publications adapted it to Make-to-Stock systems (see for instance Bonvik *et al.*, 1997). Within the system, each stage produces as fast as it can (Push principle). Comparing Figure 3 and Figure 4 shows that Conwip's implementation is much simpler than Kanban's: there are fewer control loops. Thus, modeling and optimization are easier. Actually, a Conwip system can be viewed as a Kanban system with a single loop that controls the whole production line.

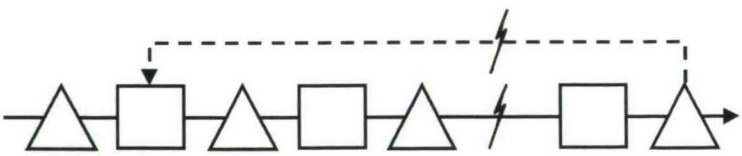


Figure 4. Conwip system

2.1.1.3 Base stock

There are several definitions of the base stock system. This system was developed in the 1950s, and has been extensively used by practitioners ever since. Bonvik *et al.* (1997) and Lee and Zipkin (1992) consider the base stock policy originally described by Kimball (1988); see Figure 5. The base stock level (say) S_i at stage i ($i = 1, \ldots, N$, with N total number of stages) refers to the echelon inventory at that stage, that is, the total amount of products produced by stage i after the inventory point of stage $i - 1$ that is in the line, either as production order or as inventory in all downstream production and inventory points. Thus the echelon inventory of stage i is included in the echelon inventory of stage i - 1. Any demand for finished products immediately triggers demands at each preceding stage: demand information is broadcast from the last stage to each stage in the production line.

Demands that cannot be filled from stock, are backordered. There is no upper bound for the inventory level at any stage. The base stock levels S_i are the only parameters of this system.

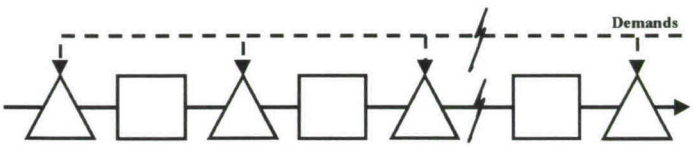


Figure 5. Base stock system

An advantage of base stock is its responsiveness to demand: as soon as a demand occurs, all the stages can start working simultaneously. A drawback, however, is that consecutive stages are not coordinated: if one stage fails, preceding stages do not stop working. A solution is to limit the amount of inventory, using authorizations as Kanban does. Then when a finished good is delivered, one card is sent to each stage of the production line, thereby allowing stages to produce. Such a system is called an *Integral Control System*; see Buzacott and Shanthikumar (1993).

2.1.2 Segmented systems

Segmented systems partition the production system into 'cells' (segments), and use a separate policy per cell. Figure 6 shows a variety of segmented systems, introduced in the literature. A large part of the literature on these systems focuses on Conwip. Another part looks at the combination of Kanban and MRP along a production flow line. Other types of segmented systems have been proposed but not studied.

Di Mascolo *et al.* (1996) emphasize that each stage of a Kanban system may consist of more than one machine. Indeed, a stage may be associated with a subpart of the production system; such a part may be a manufacturing flow line, a flexible manufacturing cell, etc. Such systems can be viewed as segmented Conwip systems. Di Mascolo *et al.* assume that the partitioning of the production system into cells is given; they do not tell how this partition should be done in practice. Tayur (1993) develops theoretical and qualitative results for partitioning of a production line into Kanban cells, and the allocation of cards to each cell. Ettl and Schwehm (1995) suggest that for systems of realistic size powerful heuristics are needed to solve the problems of partitioning the line, and allocating the cards. They propose a heuristic based on a general-purpose genetic algorithm and an analytical modeling method to simultaneously solve both problems.

Segmented push/pull systems have also been investigated. For instance, Cochran and Kim (1998) consider a production line that is controlled partly through MRP and partly through a pull strategy; they use simulated annealing to optimize inventory levels and the junction point that separates the production line into two subsystems. Olhager and Ostlund (1990) emphasize that the junction point can be the customer order point, a bottleneck resource, or a point derived from the product structure.

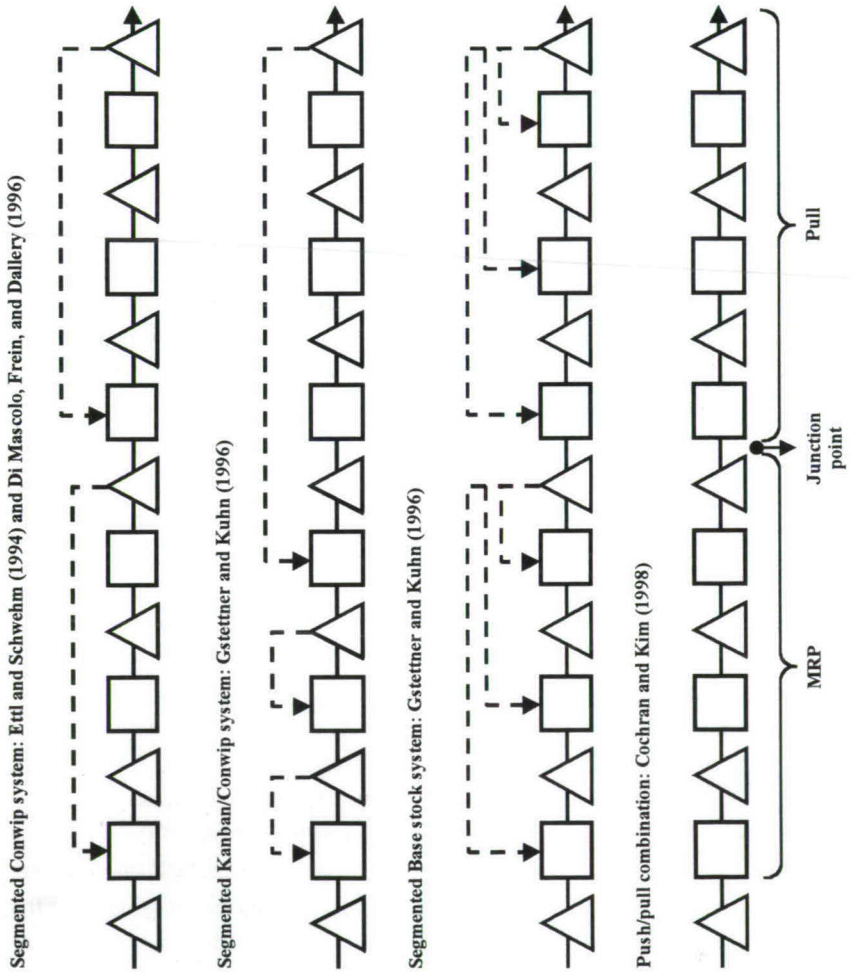


Figure 6. Segmented systems in the literature

Recent papers suggest new types of segmented pull systems that are not just limited to Kanban control. Gstettner and Kuhn (1996) propose segmented Kanban/Conwip and Base stock systems; however, they do not further study such systems, nor do they propose a design procedure. The issue is then not only to partition the line and allocate cards, but also to choose the type of pull control (Kanban, Conwip, Hybrid) for each subpart of the production system.

Much research remains to be done on segmented systems: there is no reason for limiting the pull control system per segment to Conwip. Hence the design problem is rather complex: *simultaneously* define manufacturing segments, select control policies per

segment, and configure the control policy per segment. To the best of our knowledge, such a general approach has not been proposed in the literature.

2.1.3 Joint systems

To control a specific part of a system control mechanisms can also be combined: superimpose several control systems; see Figure 7 for an example, namely a combination of Kanban and Conwip. Such combinations we call *joint systems*. The goal of our research on joint systems is to combine benefits of several systems. The main research issue is to evaluate the performance of joint systems, relatively to other control systems. Comparisons should not be limited to performance aspects, but should also include implementation complexity (Conwip is simplest). One aspect of complexity is the control policy's number of parameters (number of cards, number of base stock levels). Obviously this number should be compared with the number of stages of the production system N. Kanban and Base stock both have N parameters, whereas Conwip has only one parameter (so Conwip's complexity is independent of the number of stages).

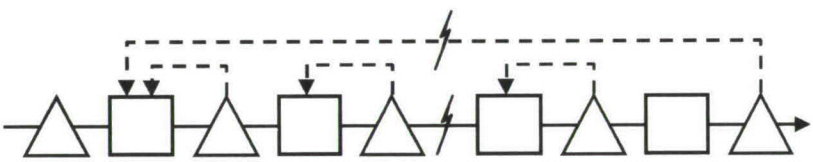


Figure 7. Joint Kanban/Conwip Hybrid

Bonvik *et al.* (1997) propose a control system called (*two-boundary*) *Hybrid* (see again Figure 7), which combines local control (one stage only) through Kanban, and integral control (whole line) through Conwip. Hybrid is easy to implement as a modification of Kanban; its number of parameters is N. An interesting characteristic of Hybrid is that production at the first stage is triggered by two signals: one from the second stage (Kanban pattern) and another one from the last stage (Conwip pattern). Production at stage 1 is allowed if *both* signals are present. In other words, the operator at stage 1 needs one card from stage 2 and one card from the last stage to start producing. *Both cards* are attached to the part; at stage 2 only the Kanban card is sent back to stage 1, and the Conwip card remains attached to the part until it reaches the finished good inventory and is delivered. We shall use the same mechanism whenever production is triggered by several signals.

Buzacott (1989) and Zipkin (1989) develop *Generalized Kanban*, which combines Base stock and Kanban. The objective is to combine Base stock's rapid reaction to demand and initial inventory levels, with Kanban's coordination between consecutive stages and local control of inventory; see Figure 8. Dallery and Liberopoulos (1995) propose a general approach to Base stock/Kanban joint systems, called *Extended Kanban*. Liberopoulos and Dallery (1997) propose variations of this approach for various production environments.

Extended Kanban has the same objective as Generalized Kanban, but it is claimed to be conceptually clearer and potentially easier to implement. Both systems, however, are rather complex, since each has $2N$ parameters.

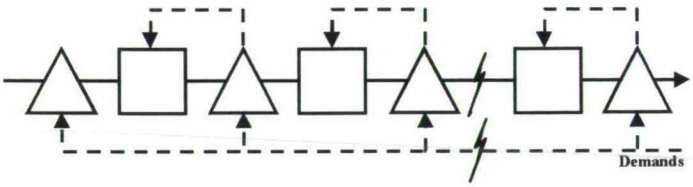


Figure 8. Joint Kanban/Base stock systems

2.2 Selection and configuration of pull systems

2.2.1 Two design issues, several formulations

There are two issues involved in the design of a pull system. The first issue is which type of pull system should be selected? More specifically, where should the control loops be placed for a given production system and production environment? Should we use a pattern of control loops as in one of the traditional pull systems, or should we select a segmented or a joint system? In the remainder of the dissertation we will call a specific pattern of control loops a *pull structure*. Selecting a pull structure includes defining a partitioning of the production system into segments. Once a structure is selected, the second design issue is to decide how many cards should be allocated to each control loop. In the remainder of the dissertation we will refer to this second issue as configuring a pull system: the *configuration* of a specific pull structure is the set of card numbers to be placed in the control loops. Figure 9 illustrates the goal of the two design issues.

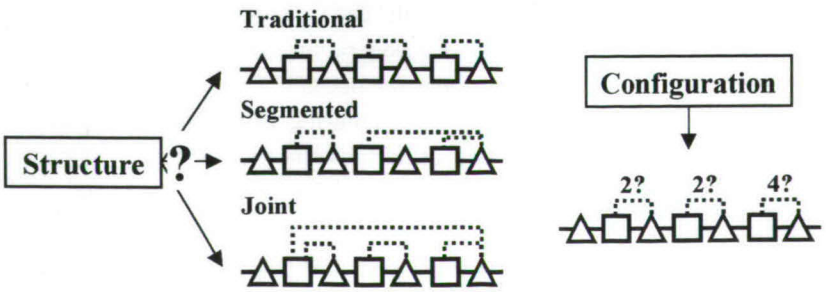


Figure 9. Two design issues: selecting a structure, and configuring the chosen structure

Throughout the dissertation we focus on these two design issues. However, we distinguish several formulations of our design problem depending on the assumptions we make when modeling the production environment. We define the production environment

as the set of factors that are not completely controlled by the designer or manager of the production system. This environment includes processing times, demand rate, time between failures, etc. (see section 5.2.1). In Chapter 3 and Chapter 4 we consider the production environment as given, whereas in Chapter 5 and Chapter 6 we study production environments that are not known with certainty and may have a dynamic behavior. Next, we discuss briefly the performance criteria used in the literature on pull production control.

2.2.2 Performance criteria

Many performance measures have been used in the pull literature. Chu and Shih (1992) classify these measures into three categories: overall, inventory related, and due-date related. Their review suggests that three criteria have been used frequently in the literature: facility utilization, throughput rate, and WIP. Facility utilization, however, should not be used as a performance measure, because the goal of a JIT manufacturing system is not to keep workers and machines busy (Goldratt and Fox, 1986). Thus, the remaining important criteria are (i) WIP and (ii) throughput rate.

(i) There are many ways of characterizing WIP. The most common approach is to consider overall WIP, that is, the difference between the number of parts that entered the system and the number of delivered finished goods. Overall WIP is suited for cases that have value more or less the same across the system; in most cases, however, inventory value increases through value-adding operations. Thus, in general finished goods are much more valuable than raw materials, and an objective might be to keep inventory at low levels in the final stages in order to minimize the financial investment. Then, WIP should be characterized through the sum of inventory value per stage with inventory value per product increasing as manufacturing operations are performed. We denote the inventory value at stage i by $V_{\mathrm{WIP}i}$. For both characterizations, the WIP performance is mainly measured through the mean (expected value).

(ii) Throughput rate should be measured relatively to demand rate: a system should not overproduce; it should meet demand very fast. Ideally, a manufacturing system should meet demand from stock: 100% service goal. Demands that are not met from stock may either be *backordered* or *lost*. Hence, the proportion of demand actually met from stock is a good indicator of system performance. We call this proportion *service level*, and we denote it by S. A 100% service goal, however, is unrealistic in many cases; managers might prefer a lower service goal as long as it is achievable by the system and acceptable from a customer viewpoint.

In conclusion, our goal when designing a pull system will be to achieve a predetermined service level with minimal overall WIP or minimal WIP value. We consider both measures so that comparison of our research with previous studies is possible. In Chapter 4 we also consider the impact of the WIP characterization on our results. The formulation of our goal implies the notion of WIP optimization under a constraint: the constraint is to achieve a

given service level. We denote this target level by τ. Thus the optimization problem is formulated as follows:

$$\begin{aligned} &\text{Min } WIP \text{ or Min } \textstyle\sum_{i=1}^{N} V_{\text{WIP}i} \\ &\text{s.t. } S \geq \tau \end{aligned} \tag{1}$$

The overall WIP can be seen as a WIP value with inventory value per part equal to 1.0 at all stages. Thus, in the remainder of this dissertation, the formulation of the optimization problem we use Min $\sum_{i=1}^{N} V_{\text{WIP}i}$ only, instead of Min WIP or Min $\sum_{i=1}^{N} V_{\text{WIP}i}$.

Another way of looking at the compromise between WIP and service uses a cost function. Then the goal is to minimize a cost function equal to the cost-weighted sum of WIP and the proportion of disservice: $a.WIP + b.(1 - S)$. Then optimization is simpler as the problem is formulated as follows:

$$\text{Min } [a.WIP + b.(1 - S)] \tag{2}$$

However, as some authors admit, the cost of disservice (shortage cost), namely b, is difficult to estimate in practice. Thus, many recent publications avoid the use of shortage costs. An example of discussion on service level versus shortage cost is Janssen (1998, p. 20). He emphasizes that when shortage costs incorporate the customers' loss of goodwill or when a service target is chosen, the optimization problem has a long-term perspective.

Next we review the techniques proposed in the literature for configuring a pull system.

2.2.3 Selecting the card numbers

One of the main foci of researchers in the field of pull control is the determination of the card numbers in a given pull structure, such that performance is optimal. The reason for this interest is that the card numbers have a major influence on the performance and they are the only means for balancing inventory and service performance. Gupta and Gupta (1989) and Huang *et al.* (1983) conclude that pull systems perform well, only when the numbers of cards are chosen optimally! Figure 10 illustrates that the performance of a Conwip system in terms of inventory is a linearly increasing function of the number of card. The performance in terms of service level is also linearly increasing but only for card numbers below a threshold value; above the threshold, it remains equal to 100%. The compromise issue can be illustrated by considering a cost function that aggregates inventory and service performance: for low card numbers, disservice costs have an important contribution to total cost, whereas for high card numbers costs are only due to excess inventory. Of course the respective contributions of inventory and disservice to the cost function depend on the choice of a and b in Equation (2).

Many techniques for selecting card numbers are presented in the literature: (i) empirical formulas, (ii) optimization based on analytical models, (iii) optimization based on

simulation models. Most techniques were developed for Kanban, but their extension to other pull structures has also been studied.

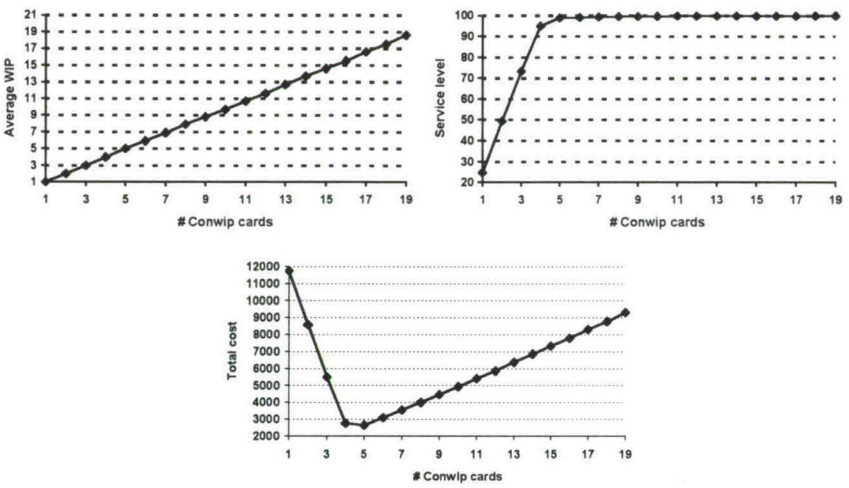


Figure 10. Performance as a function of the number of cards; $a = 500$ and $b = 15000$

(i) Sugimori *et al.* (1977) report that Japanese managers at Toyota use the following empirical inequality for computing the minimal number of cards to be used in control loop i:

$$y_i \geq D_i L_i (1 + \alpha_i)/a \tag{3}$$

where y_i is the number of Kanbans at stage i, D_i is the average demand per time unit at stage i, L_i is the average production lead-time at i, α_i is a variable for safety stock at i, and a is the container capacity (a single Kanban is attached to each container). A reason for using an inequality and a safety stock is that the card numbers are fine-tuned empirically as follows. Remove one card from the system and subsequently check for disruptions. If the system behavior is not satisfactory anymore, put the removed card back into the system, and try to identify and eliminate the causes of disruption (see section 1.2).

(ii) Analytical models can be either deterministic or stochastic. Price *et al.* (1994) gives a detailed review of deterministic optimization models of Kanban systems, their assumptions and results. Most stochastic approaches use queuing networks as models, and Markov chains and decomposition techniques for analyzing the modeled performance. Table 1 gives some references to queuing network analysis for various types of pull systems. An advantage of these analytical techniques is that performance evaluation may be quick and an exhaustive search can be performed on a limited domain to determine the card

numbers. Such exhaustive searches are performed in Duri (1996) and Wang and Wang (1990).

Table 1. Queuing network analysis of pull systems

Pull structure	Reference
Kanban	Di Mascolo *et al.* (1996)
Base stock	Buzacott *et al.* (1991)
Generalized Kanban	Frein *et al.* (1995)
Extended Kanban	Dallery and Liberopoulos (1995)

(iii) Simulation models use less restrictive assumptions than analytical models do, but require much more computational time for performance evaluation. Thus, exhaustive searches are rarely performed when determining the card numbers; instead, optimization techniques such as Response Surface Methodology (RSM) and evolutionary algorithms are preferred. Table 2 gives references to simulation-based optimization of pull systems.

Table 2. Simulation-based optimization in the pull literature

Reference	Optimization technique	Pull structure
Bonvik *et al.* (1996)	Exhaustive search	Kanban, Base stock, Conwip, Hybrid
Paris and Pierreval (1997)	Evolutionary algorithm	Kanban
Davis and Stubitz (1987)	RSM	Kanban
Chang and Yih (1994)	Simulated annealing	Kanban

Simulation can also be used to derive "metamodels" of pull systems. This metamodeling estimates a mathematical relationship between the simulation inputs (card numbers, for instance) and its outputs (such as performance measures) using (non-linear) regression analysis. Jothisankar and Wang (1993) use this technique for a two-stage Kanban system; they derive the card numbers as functions of demand rate. Aytug *et al.* (1996) determine a relationship between the card numbers and the average time to fill a customer order for a two-stage Kanban system. Hurion (1997) uses simulation to train a neural network metamodel, which he uses to estimate the optimal number of cards; finally, a local search is used to refine the solution.

Another problem considered in the literature is optimal allocation of cards along a production line, given a fixed number of cards for the whole line; see Gstettner and Kuhn (1996). We will not consider this type of problem in this dissertation.

This short review of the techniques for determining the number of cards in a control loop shows that a large variety of techniques are available depending on which degree of

precision to be achieved. Empirical formulas give instantaneous results, but assume a deterministic behavior of the production system. Analytical methods require a low computational cost, but use more restrictive assumptions and require analyst skills. Simulation allows stochastic modeling of the production system, but requires long computing times. This computational cost, however, is getting lower as computing power increases. Next we review comparisons among pull systems.

2.2.4 Review of comparisons among pull systems

Choosing among several pull systems is a complex problem. Indeed, it is hard to say whether a given pull structure is 'better' than another one. The fact that each pull structure can have many configurations complicates comparisons. In the literature the 'best' pull system among several structures is always found using the following procedure. For a given production environment, find the optimal configuration of each structure, and compare the performance of these optimal configurations. The structure that yields the best configuration is said to be the best structure. Whenever a new pull system is proposed, researchers make comparisons with existing pull systems and other inventory control techniques, such as order release strategies and reorder point systems. Thus the most complete comparison studies are also the most recent studies. We now review comparison studies, in the order of publication of pull systems in the literature; we start with one of the first implementations of the pull principle, namely Kanban.

As reports appeared on the benefits obtained in Japanese companies through the Kanban method, researchers tried to compare Kanban with classical methods such as MRP manufacturing and reorder point systems. A major publication is Krajewski *et al.* (1987): they report on a project to assess the expected performance of Kanban in typical U.S. manufacturing environments; they try to find which factors have the biggest impact on performance. They consulted a large panel of managers across the U.S., to formulate a list of factors that might affect performance, and to select low and high values for each factor. Then they build a sample of representative plants and simulate each plant. Their main conclusion is that Kanban is very efficient for some environments, but more traditional systems also perform well for these environments. In other environments, however, the Kanban method is much less efficient. Other studies emphasize that in a Western environment push systems perform better than pull systems. However, Gupta and Gupta (1989), Huang *et al.* (1983), and Schroer *et al.* (1985), conclude that high production rates can be realized, only when the number of kanbans is chosen optimally.

Veatch and Wein (1994) compare base stock to Kanban for two-stage systems. They show that base stock may or may not be better than Kanban, depending on the production environment. For instance, the position of the bottleneck has a major impact on the choice between the two systems.

To the best of our knowledge, Conwip has been compared with other strategies through simulation only. Roderick *et al.* (1994) and Roderick *et al.* (1992) compare Conwip to

MRP and three order release strategies. They conclude that Conwip gives the best performance measured in mean WIP, mean throughput, and proportion of tardy jobs. Thus, Roderick *et al.* (1992) recommends Conwip as a "strategy that should be seriously considered by practitioners for implementation in actual shop environments". Gstettner and Kuhn (1996), however, compare Conwip to Kanban for a five-stage line and conclude that Kanban is more flexible than Conwip when a specific performance level is to be achieved. They also show that Kanban reaches a given production rate with less inventory than Conwip does.

The literature on Hybrid is not large, because Hybrid appeared only recently. Bonvik *et al.* (1997) perform many optimizations using simulation and exhaustive search. They show that the advantage of Hybrid over Kanban increases, as the service target gets closer to 100%. They also consider Conwip, minimal blocking (a variant of Kanban), and base stock. They conclude that Hybrid is best in terms of average overall inventory for a given service target. Kanban and minimal blocking have similar performance; the same close relationship is observed for Conwip and base stock. The performance of Conwip and base stock falls between those of Kanban and Hybrid.

Duri (1996) compares the costs of the optimal configurations of base stock, Kanban, and Generalized Kanban. She determines these costs analytically using queuing network theory. If demands have to be satisfied immediately (no delay between demand and delivery), then the costs are the same, so Kanban would be preferred because of its simplicity. If a delay is allowed between demand and delivery (backordering), then Generalized Kanban and base stock give lower costs than Kanban for the same service level. Since base stock does not limit WIP, users might prefer Generalized Kanban.

Sometimes contradictory results can be found in the literature, particularly for Kanban, as we saw earlier in this section. An explanation is that comparisons are made for a given production environment, so a particular pull structure may be best in one environment, but may be outperformed in other environments. Only a very limited number of studies provide extensive information on the pull structure to choose for a particular production environment. Bonvik *et al* (1997) determine which pull structure – Kanban or Hybrid – to prefer when the service target is changed from 99 to 100%. For each target value they find the optimal configurations with minimal inventory of Kanban and Conwip, given the service target. They summarize their results through a plot showing the WIP performance as a function of the service level performance for the best configurations of Kanban and Hybrid: Hybrid is always better. These results, however, do not tell how the other pull structures perform in the same conditions, and if other environmental factors (such as the demand rate, the imbalance of processing times) affect the relative performance of pull systems. A more complete study is Karaesman and Dallery (1998), who compare Kanban, Base stock, Extended Kanban, and Generalized Kanban for a sample of 18 sets of parameter values. These parameters include production rates per stage and cost parameters. They conclude that Kanban performs well under certain conditions, and base stock

performs better under other conditions. Since Generalized Kanban is a combination of Kanban and Base stock, it performs well over a larger range of conditions – in the example under consideration Extended Kanban is a special case of Generalized Kanban. Their results, however, are limited to two-stage systems; they do not consider other pull structures, such as Hybrid.

2.3 Conclusion

The problem of designing a pull system is twofold: select a pull structure (decide where to place the control loops) and configure it (determine the number of cards to be used in each control loop selected). In section 2.1 we saw that many pull structures have been proposed in the literature. For most structures we have only partial knowledge of how well they perform. The only selection technique proposed in the literature consists in finding the optimal configuration of each pull structure under consideration, and comparing these optima. Applications of this technique, however, suffer from two main limitations: (i) the optimal configuration of a pull structure may be hard to find; it gets harder as the number of production stages increases, and (ii) optimization has to be repeated for each pull structure. Extension to bigger numbers of stages and pull structures would have a high computational cost. In fact, we shall see in the next chapter that the number of possible pull structures is increasing rapidly, as the number of stages in the production system increases. Thus, the technique comparing the optimal configuration of each pull structure is not attractive if we want to choose among all possible pull structures: there is a need for a new selection technique.

Another limitation of the literature is that many studies do not study the influence of the environment on the choice of a pull system. And when the issue is raised, the investigation is often limited to a few factors only; an exception is Krajewski *et al.* (1987), but the only pull structure they consider is Kanban. Thus a more complete study of how the production environment impacts the choice of a pull system is required. We perform such a study in Chapter 4.

Chapter 3
Generic Model and Customized Pull Systems: Methodology

Abstract

In Chapter 2 we proposed a typology of known pull systems. We also saw that choosing among these known systems is difficult; for a given production system and production environment it is not possible to say a priori which pull structure should be implemented. Furthermore, many other structures can be created that do not match any known pull structures: the choice is not limited to traditional pull control systems and their combinations. Selecting a specific pull system among all possible pull systems is a complex problem that has not been investigated in the literature. Our contribution to solving this problem is presented in this chapter: we design a generic model that is a common representation of all the pull systems presented in Chapter 2. For a given production system and production environment, the optimization of the generic system yields not only the pull structure (which control loops should be implemented), but also the optimal card numbers for each loop actually implemented. The result of this approach may be one of the traditional systems, but it may also be one of the following three new types. (1) The total line may be decomposed into several segments, each with its own traditional control system. (2) The total line or its segments may combine different traditional systems. (3) The line may be controlled through a new type of control system. Thus, the generic model does not only help choosing among known pull structures; it extends the concept of pull control. We call this extended approach customization. The benefits of this approach are shown for an example production system taken from the literature, for which the optimal configurations of several known pull systems have been determined in that literature: we find a novel pull system that performs significantly better than the best system in the literature, namely Kanban/Conwip Hybrid.

3.1 Introduction

The subject of this chapter is *customized* control systems that may replace traditional control systems such as Kanban, Conwip, and Base stock. In Chapter 2 we saw that many structures of pull system already exist, and that many others can be added. Our objective is to consider many more structures including traditional, segmented, and joint systems, as well as structures that do not match the typology of known systems presented in section 2.1. In section 3.2 we shall see that this objective leads to a problem of high complexity because the number of possible structures grows exponentially with the number of stages.

In Chapter 2 we also suggested that we need an alternative to the technique proposed in the literature for choosing among pull systems. Indeed, this technique consists in optimizing each structure type individually. Thus the computational cost of the technique grows proportionally to the number of considered structures. To achieve customization we propose a generic optimization model that may represent all possible pull control systems. This model connects each stage of a production line with each preceding stage (section 3.3). Customization consists of a *single optimization* of the corresponding model to determine which control loops actually need to be implemented. The advantage of this approach in terms of computational costs is increased by deriving structural properties of the generic model (section 3.4). In section 3.5 we design an evolutionary algorithm to perform the optimization; we use discrete-event simulation to evaluate the performance of the generic model. In section 3.6 we show the benefits of our customization approach through an example taken from the literature, for which the performance of several known pull systems have been determined in that literature.

3.2 Extending the traditional pull systems

We saw in our typology (Chapter 2) that many pull structures already exist or have been considered. Yet, we do not see any reason for limiting research to these systems only. Huang and Kusiak (1998) criticize traditional strategies, for not considering the specific characteristics of manufacturing systems. They propose an algorithm based on decision rules for choosing which strategy – push or pull – should be adopted at each stage of a manufacturing system. Even though they consider only local control (no combinations), they are among the first researchers to investigate what we call *customized* control systems. Each production system has its own specificity, and requires a special control system; that is, predefined systems such as Kanban, Conwip, and Base stock might not be good enough.

Many systems can be created that do not match the typology of known pull systems; a simple example is shown in Figure 11. In fact, for an N-stage serial line there are $N(N+1)/2$ possible control loops connecting a stage to a preceding stage. Each of these possible loops can be implemented or not. The choice of which loop to implement defines a pull structure. For instance, if we decide to implement loops only between consecutive

stages, then we have a Kanban system. The number of all possible pull structures is equal to $2^{N(N+1)/2}$. Table 3 gives the value of this expression for several values of N and shows that the problem of choosing a pull structure among all possible structures is a complex problem even for small numbers of stages. To the best of our knowledge this problem has not been considered in the literature before.

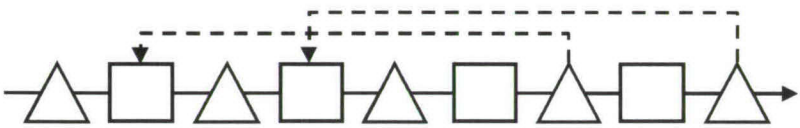


Figure 11. A pull structure that does not match the typology of known systems

Table 3. Number of possible control loops and pull structures as functions of the number of stages

# stages	# possible loops	# possible structures
2	3	8
4	10	1024
10	55	3.6×10^{16}

Our objective is to extend the choice of a specific pull system to all $2^{N(N+1)/2}$ structures, instead of focusing on a few known structures. In this perspective the technique used so far in the literature is not adapted. Indeed, optimizing each possible pull structure would not be possible: for a production line with four stages only, we would have to perform 1024 optimizations in order to select the best values for one to ten parameters. Instead we propose a generic optimization approach based on Gaury *et al.* (1997b) which was limited to Kanban, Conwip, and Hybrid only.

3.3 Customization through a generic pull control system

3.3.1 Generic optimization model for Kanban, Conwip, and Hybrid

In Gaury *et al.* (1997b) we propose a new methodology for choosing among Kanban, Conwip, and Hybrid. The idea is to optimize a *generic* system that combines the information flows of the three systems; see Figure 12. This generic system is an optimization model that can represent a Kanban, Conwip, or Hybrid system, depending on the choice of the card numbers in each control loop. The key concept of the optimization model is that a control loop with an infinite number of authorizations does not impose any constraint on the flow of parts (WIP). Indeed an infinite number of cards means that a machine is always allowed to produce. Thus the authorization information is not necessary and the corresponding control loop does not affect the performance of the production line.

Therefore it can be removed from the generic system; it does not need to be implemented. We denote an infinite number of cards by the usual symbol ∞. Table 4 shows how to set the parameters of the optimization model in order to get a Kanban, Conwip, or Hybrid system; c denotes a finite number of Conwip cards and k_i a finite number of kanbans.

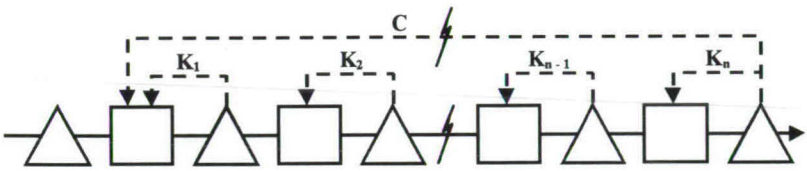


Figure 12. The generic model for Kanban, Conwip and Hybrid in Gaury *et al.* (1997b)

For a given production system, the optimal configuration of the generic system not only shows which type of pull strategy is preferred, but also which values should be selected for the various numbers of cards. For example, suppose that optimization of the generic model for a specific four-stage production line gives $k_1 = 2$, $k_2 = 3$, $k_3 = 5$, $k_4 = 4$, and $c = \infty$. According to Table 4, this configuration of the generic model is equivalent to a Kanban system. Thus, Kanban provides the best performance; its optimal configuration is $k_1 = 2$, $k_2 = 3$, $k_3 = 5$, and $k_4 = 4$.

Table 4. Generic system for three pull production control systems

	C	K_1	K_2	...	K_{n-1}	K_n
Kanban	∞	k_1	k_2	...	k_{n-1}	k_n
Conwip	c	∞	∞	...	∞	∞
Hybrid	c	k_1	k_2	...	k_{n-1}	k_n

We give a different example. In Gaury *et al.* (1998) we optimized the generic model using an evolutionary algorithm (see section 3.5) for a specific production line with four stages, inspired by a Toyota factory (we give more details about this production line in section 3.6). We found the following estimated card numbers: $k_1 = 6$, $k_2 = 3$, $k_3 = \infty$, $k_4 = \infty$, and $c = 14$. This system cannot be classified as Kanban, Conwip, or Hybrid; it may be considered to be a simplified Hybrid system. In general, the best solution found through the optimization of the generic model, does not necessarily correspond to any traditional control system (such as Kanban).

3.3.2 Generic model for customization

Our objective is to design a pull control system for a given line, without *a priori* limiting the type of control to *traditional* pull systems. Instead, we search for a control system in the

set of pull systems that consists of Kanban, Conwip, and its Hybrid, local and integral control, segmented and joint systems. To achieve this goal, *we link – through control loops – each stage to all its preceding stages.* More specifically, we link each stock point (inventory) to each preceding resource (machine). So we design a new generic system that accounts for all possible types of pull control. Figure 13 gives an example for a line with four stages; $k_{i;j}$ denotes the number of authorizations that circulate in the loop linking stage i to stage j, M_i and I_i denote the resource (Machine) and the stock point (Inventory) at stage i respectively. We number the stages in increasing order following the flow of parts; that is, the raw material enters at stage 1 and finished goods leave at stage N; we also have $j \leq i$. In Figure 13 there are 10 (potential) loops ($10 = 4(4 + 1)/2$). However, we do not implement control loops with infinite card numbers; also see the former generic model (Gaury *et al.*, 1997b), which, however, was limited to Kanban, Conwip, and Hybrid.

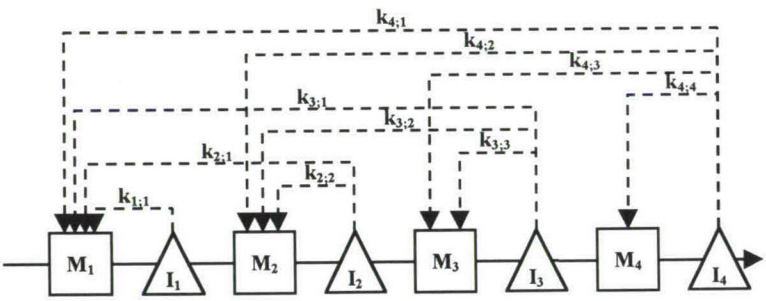


Figure 13. Generic system accounting for all possible pull patterns

Next we consider the general case of a production line with N stages controlled by a generic system (Figure 13 was the special case of $N = 4$). If we want the generic control system to be Kanban, then we select the numbers of authorizations as follows: $k_{i;j} = \infty$, $\forall (i, j) \in \{1,...,N\}^2 / i \neq j$ ($k_{i;j}$ is infinite for all i and j such that $i \neq j$), and $k_{i;i} << \infty$, $\forall i \in \{1,...,N\}$ ($k_{i;i}$ has a finite value for all i). Similarly, we obtain Conwip by choosing $k_{N;1} << \infty$, and $k_{i;j} = \infty$, $\forall (i, j) \in \{1,...,N\}^2 / (i, j) \neq (N, 1)$. The generic model cannot represent the Base stock policy of section 2.1.1.3 because that strategy uses information about demand occurrences, whereas the generic model focuses on information about actual deliveries. However, the Integral Control variant of Base stock (see again section 2.1.1.3) is a possible instantiation of the generic model. The generic model can also represent control systems that have not been investigated in the literature, for instance, a system with control loops that link each machine to the first machine (to release raw materials, this system requires authorizations from all machines to machine 1).

To find the best customized pull system for a given line, we optimize the generic system. If this optimization gives a solution with some card numbers being infinite, then the

corresponding loops are not implemented. The optimization problem can be stated by completing statement (1) (see section 2.2.2) as follows:

$$\begin{aligned} &\text{Min } \textstyle\sum_{i=1}^{N} V_{\text{WIP}i}(\{k_{i;j}\}) \\ &\text{s.t. } S \geq \tau \\ &\quad k_{i;j} \in \mathbf{N}^* \cup \{\infty\},\ \forall (i,j) \in \{1,\ldots,N\}^2, \\ &\quad \text{where } \mathbf{N}^* \text{ is the set of natural integers, zero excluded.} \end{aligned} \tag{4}$$

A practical problem is the complexity of this optimization: the generic system has $N(N+1)/2$ parameters, namely, the card numbers in all potential control loops (for example, for ten machines, the generic model has 55 control loops). This optimization concerns a non-linear model with integer variables. If in the example with ten machines we restrict the various card numbers to $\{1\ldots20\} \cup \{\infty\}$, then the search space still includes $21^{55} = 5.27 \times 10^{72}$ configurations of the generic system, which is a rather large search space. Our approach, however, is much less complex than when optimizing each possible pull structure: for ten machines we would need to perform 3.6×10^{16} optimizations, each involving a search space size ranging from 21 to 21^{55} configurations!

In the next section we study structural properties of the generic model. The objective is to find ways of limiting the search space without loss of generality for our customization approach, that is, without discarding any pull structure during the search.

3.4 Structural properties

In the previous section we explained that a control loop with an infinite number of cards does not need to be implemented because it does not add any constraint on the flow of products, that is, cards are always available to authorize production. Another formulation is that a control loop that does not constraint the flow of products can be replaced by a control loop with an infinite number of cards: cards are always available to authorize production, so adding more cards does not change anything. The object of this section is to identify the cases for which a given control loop does not constraint the flow of products.

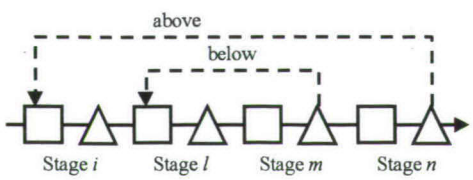


Figure 14. Illustration of the above/below relationship among control loops

Let $\{1 \ldots K_{i;j}\} \cup \{\infty\}$ be the search domain for the card number $k_{i;j}$ in the control loop linking stage j to stage i. We say that a control loop linking stage m to stage l is *below* the control loop linking stage j to stage i if we have: $i \leq l \leq m \leq j$. Then, we say that the control loop linking stage j to stage i is *above* the control loop linking stage m to stage l; see Figure 14.

Property 1

Any control loop, below a given control loop with c cards, must have less than c cards to be a constraint on the flow of products. Otherwise, this control loop does not need to be implemented and its number of cards can be set to an infinite value.

Let $k_{i;j} = c$, with $i \neq j$, then $(k_{l;m} \geq c$, with $i \leq l \leq m \leq j) \Rightarrow k_{l;m} = \infty$.

Thus, if $c \neq 1$, $k_{l;m} \in \{1, \ldots, c - 1\} \cup \{\infty\}$, else $k_{l;m} = \infty$.

Indeed, setting the number of cards to c in a given control loop (say CL_1) means that only c products at most can be present simultaneously in any portion of the system within the control loop. Thus, selecting a number of cards superior or equal to c in a control loop below CL_1 means that there will always be free cards so production will always be allowed for parts that entered this portion of the system. Figure 15 gives a simple illustration of property 1 for a two-stage system.

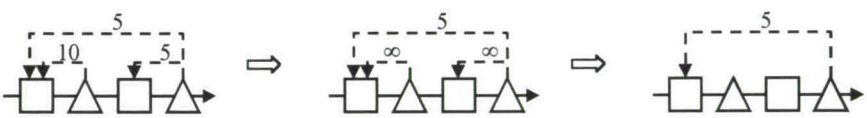


Figure 15. Simple illustration of property 1

To derive a second property, we define a *sequence of non-overlapping control loops* (say ω) as a set of control loops such that each machine or inventory is controlled by one and only one loop of the set. In Figure 16, such a sequence for stages 1 to 3 can be $\omega = (CL_{1;1}, CL_{3;2})$; we denote by $CL_{j;i}$ the control loop going from stage j to stage i. We define the number of cards in the sequence ω as $k(\omega) = k_{1;1} + k_{3;2}$. The sequence $(CL_{1;1}, CL_{3;1})$ has overlapping control loops because stage 1 is controlled by both $CL_{1;1}$ and $CL_{3;1}$. We denote

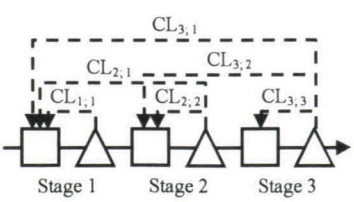


Figure 16. Possible control loops in a three-stage line

by $\Omega_{i;j}$ the set of all sequences of non-overlapping control loops below $CL_{j;i}$, with $i < j$. In Figure 16, we have $\Omega_{1;3} = \{(CL_{1;1}, CL_{3;2}), (CL_{1;1}, CL_{2;2}, CL_{3;3}), (CL_{2;1}, CL_{3;3})\}$.

Property 2

A control loop should have fewer cards than the total number of cards in any sequence of non-overlapping control loops on the same section of the line. Otherwise, this control loop does not need to be implemented and its number of cards can be set to an infinite value.

Let $k_{i;j} = c$, with $i \neq j$, then $(c \geq \text{Min } k(\omega)$, with $\omega \in \Omega_{i;j}) \Rightarrow k_{i;j} = \infty$.

Thus, $k_{i;j} \in \{1, \ldots, \text{Min}_{\omega \in \Omega_{i;j}} k(\omega)\} \cup \{\infty\}$.

A demonstration of this property is given in Appendix 1. That demonstration is based on a recursive formulation of the maximal number of parts allowed to enter a given portion of the system. This recursive formulation is particularly useful for implementation in a computer program.

Detailed example

In this example we show how properties 1 and 2 can be used to simplify a customized control system and to limit the number of possible values for the number of cards in a given control loop. Figure 17 gives an example of such a simplification procedure.

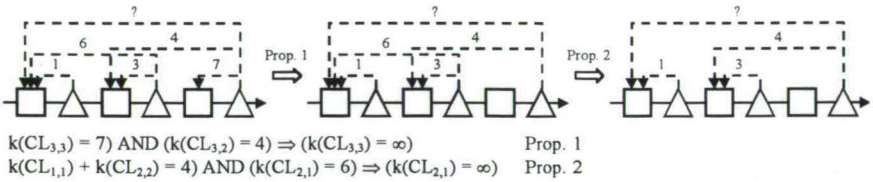


Figure 17. Example of simplification procedure

In order to determine the possible values for $CL_{3;1}$, we use property 2 and look at the number of cards in all the sequences of non-overlapping control loops on the corresponding portion of the line.

$k(CL_{1;1}, CL_{2;2}, CL_{3;3}) = 1 + 3 + \infty = \infty$,

$k(CL_{1;1}, CL_{3;2}) = 1 + 4 = 5$,

$k(CL_{2;1}, CL_{3;3}) = \infty + \infty = \infty$.

So, $\text{Min}_{\omega \in \Omega_{i;j}} k(\omega) = 5$, and $k_{3;1} \in \{1, 2, 3, 4, 5\} \cup \{\infty\}$. In this example a value of $k_{3;1}$ strictly bigger than five is equivalent to an infinite value.

We note that the order in which properties 1 and 2 are used, is not important. Indeed, applying property 2 directly on the non-simplified system in Figure 17 yields:

$k(CL_{1;1}, CL_{2;2}, CL_{3;3}) = 1 + 3 + 7 = 11$,

$k(CL_{1;1}, CL_{3;2}) = 1 + 4 = 5$,

$k(CL_{2;1}, CL_{3;3}) = 6 + 7 = 13$.

So the result is unchanged ($k_{3;1} \in \{1, 2, 3, 4, 5\} \cup \{\infty\}$).

Using the two properties, we can avoid evaluating equivalent solutions, which is particularly interesting during optimization. In the next section we detail the optimization technique used for customizing the pull control system.

3.5 Customization through simulation and evolutionary algorithms

3.5.1 Introduction

Evolutionary Algorithms (EAs) include a variety of algorithms that can be used to tackle our optimization problem. Three main classes of EAs have been identified in the literature: Evolutionary Programming (Fogel *et al.*, 1966), Evolution Strategies (Rechenberg, 1965), and Genetic Algorithms (Holland, 1975). Further, Bäck and Schwefel (1993) give and overview of the similarities and differences among these three classes. For more details about EAs, we refer to Bäck (1996) and Michalewicz (1992). Applications to simulation optimization can be found in Pierreval and Tautou (1997).

The algorithm we use for our experiments works with a set of potential solutions (pull structures); this set is called a *population*. Each iteration (*generation*) of the algorithm consists in a reproduction-evaluation cycle. Solutions in the population are selected for reproduction purposes - the best adapted solutions have a higher chance of being selected - and their offspring is submitted to recombination and mutation operators (which we describe in section 3.5.3.4). Recombination mixes parental information, while passing it on to the offspring. Mutation introduces innovation into the population. Next, the fitness of the new solutions is evaluated. The main steps of our algorithm are as follows (Spears *et al.*, 1993 and Bäck, 1996):

Step 0. Start with the generation counter equal to zero.
Step 1. Initialize a population of potential solutions.
Step 2. Evaluate the fitness of all solutions in the initial population.
Step 3. Increase the generation counter.
Step 4. Select a sub-population for reproduction (selection).
Step 5. Recombine selected parents (recombination).
Step 6. Perturb the mated population stochastically (mutation).
Step 7. Evaluate the fitness of the mated population (evaluation).
Step 8. Test the termination criterion, and stop or return to step 3.

Five main choices have to be made in order to implement this algorithm: encoding of solutions, fitness, selection mechanism, evolutionary operators, and parameters of the algorithm. Next we review the literature on these specific implementation issues.

3.5.2 Implementation issues of the EA algorithm

3.5.2.1 Solution encoding

In a computer program individuals can be implemented as a data structure. Often the structure is a vector, with components that are the optimization parameters themselves or representations. Recent literature, however, investigates the possibility of using tree structures instead of vectors (Pierreval and Tautou, 1997). These structures have components that may be binary, real, or integer values; they may also be qualitative variables (for instance, design options such as conveyor, automated guided vehicle, or forklift truck).

3.5.2.2 Fitness

Fitness is a value assigned to an individual that reflects how well this individual solves the optimization problem. Thus, the fitness is often an objective function value. Depending on the optimization problem, the objective may be based on a single criterion or a collection of several criteria including constraints. In the latter case, the objective function may be expressed as a weighted sum of several performance measures (equivalent to the single criterion case); it may also include a penalty function that penalizes individuals only if constraints are violated (see Michalewicz, 1992).

3.5.2.3 Selection

One of the most popular selection systems is the *roulette wheel* (Goldberg, 1989). In that system the decision whether to select an individual is made according to a probability assigned to each individual. That probability is based on the fitness of the individual, such that the one with the best fitness has the highest chance of surviving. The literature provides many other selection techniques, some of which may be combined with the roulette wheel:

- *Sigma-scaling* also accounts for the standard deviation of the individual fitness (Forrest, 1985);
- *Elitism* preserves a number of the best individuals, from one generation to another (De Jong, 1975);
- *Boltzmann selection* uses the principle of "crystallization" that is also used in simulated annealing (Goldberg, 1990, De la Maza and Tidor, 1993);
- *Rank selection* maintains the pressure of selection, even when the fitness of individuals gets very close to each other (Baker, 1985);
- *Tournament selection* makes individuals compete against each other (Goldberg and Deb, 1991).

3.5.2.4 Evolutionary operators: recombination and mutation

Recombination consists in mixing the information contained in a pair of individuals and creating a new pair. Several mixing strategies can be found in the literature. The simplest one is the single-point *crossover* (Goldberg, 1989), which replaces with probability p_{cross} two parents X^i and X^j by their offspring $X^{i'}$ and $X^{j'}$, as follows. An integer *pos* represents the point at which the solutions X^i and X^j are cut; *pos* is selected randomly between 1 and q - 1 where q is the number of components in the data structure. This random selection may be based on various probability distributions, uniform being the most common one. The inversion of the two parts of each individual leads to a new pair of individuals. This recombination process is shown in Figure 18. More complex mixing strategies are multiple-point crossovers, which require the definition of several crossover points (see Goldberg, 1989).

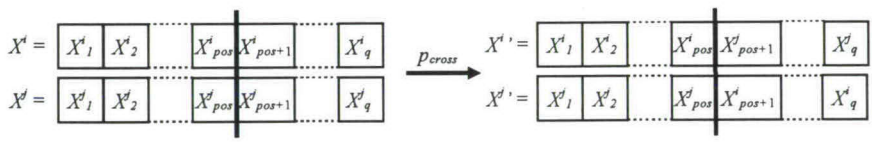


Figure 18. Recombination operator

Mutation creates new individuals by making small alterations of the data components. Each component has a chance p_{mut} of mutating. The new component value is chosen randomly among possible values (search domain). The following strategies can be found in the literature (Michalewicz, 1992, and Pierreval and Tautou, 1997): selection of the bounds of the domain, use of a uniform, a triangular, or a Gaussian probability distribution, etc.

3.5.2.5 Parameters of the EA algorithm

The last implementation issue concerns the choice of values for the various parameters in the EA. De Jong (1975) performs many experiments with Genetic Algorithms (GAs) in order to investigate the influence of parameter values on the EA performance. He concludes that the best population size is 50 to 100 individuals, the best single-point crossover rate is approximately 0.6 per pair of parents, and the best mutation rate is 0.001 per bit (GAs have binary-valued data components). Obviously these parameter values depend on his experimental conditions; for instance, a population of 50 to 100 individuals takes too much computer time when fitness is estimated through stochastic simulation. More generally, Mitchell (1996, p. 175-177) suggests that crossover, mutation, and selection should be balanced, depending on both the fitness function and the encoding. Therefore, she recommends choosing the parameter values according to a trial and error strategy.

3.5.3 EA algorithm implementation for customizing pull systems

3.5.3.1 Vector of card numbers

We choose to represent our generic system through a vector with components that are the various card numbers: $(k_{1;1}, k_{2;2}, k_{2;1}, k_{3;3}, k_{3;2}, \ldots, k_{N;1})$, where $k_{1;1}, \ldots, k_{N;1}$ are the numbers of cards shown in Figure 13 (also see section 3.3.2). The order in which the card numbers are listed in the vector, follows the order in which a part encounters the corresponding control loops from stage to stage: $k_{1;1}$ at stage 1, $k_{2;2}$ and $k_{2;1}$ at stage 2, etc. We are interested in vector components with a domain of the type $D \cup \{\infty\}$, where D is a finite set of integer values.

3.5.3.2 Fitness

Our goal is to achieve a predetermined service level, while minimizing the total WIP value or the overall WIP (see section 2.2.2). In section 3.5.2.2 we saw that there are EAs for such optimization problems with a constraint (such as service level above a specific value). The most widely used technique penalizes solutions that do not respect the constraint, artificially either decreasing or increasing the fitness of these solutions, according to the objective (Michalewicz, 1992). If the objective is to minimize the fitness value, then penalizing a solution that does not respect the constraint implies increasing its fitness value. In our case, the optimization problem stated in (4) becomes:

$$\begin{aligned}
&\text{Min } f(\{k_{i;j}\}),\\
&\text{where } f(\{k_{i;j}\}) = \begin{cases} \sum_{i=1}^{N} V_{\text{WIP}i}(\{k_{i;j}\}) \text{ if } S \geq \tau \\ \sum_{i=1}^{N} V_{\text{WIP}i}(\{k_{i;j}\}) + k \text{ otherwise,} \end{cases} \qquad (5)\\
&k > 0 \text{ is the penalty,}\\
&\text{s.t. } k_{i;j} \in \mathbf{N}^* \cup \{\infty\}, \forall (i,j) \in \{1,\ldots,N\}^2,\\
&\quad \text{where } \mathbf{N}^* \text{ is the set of natural integers, zero excluded.}
\end{aligned}$$

So when a solution does not meet the service constraint, the objective function f is increased by a penalty value equal to k. when the service constraint is respected, the optimum is the solution with the lowest total inventory value.

Michalewicz *et al.* (1996) mention that the main difficulty is choosing the right level of penalty: if the penalty is too low, the final solution might not respect the constraint; if the penalty is too high, the search might be confined to a too small part of the search space and converge to a local optimum. Therefore, we tried several ways of implementing the penalization. Figure 19 gives plots of the following fitness functions (all three plots have $k = 100$):

① $\begin{cases} \text{WIP if Service} \geq 99.9\% \\ \text{WIP} + k(99.9 - \text{Service}) \text{ otherwise,} \end{cases}$

② $\begin{cases} \text{WIP if Service} \geq 99.9\% \\ k\text{WIP}(99.9 - \text{Service}) \text{ otherwise,} \end{cases}$

③ $\begin{cases} \text{WIP if Service} \geq 99.9\% \\ \text{WIP}_0 + k(99.9 - \text{Service}) \text{ if Service} < 99.9\% \text{ and WIP} < \text{WIP}_0, \\ \text{WIP} + k(99.9 - \text{Service}) \text{ otherwise} \end{cases}$

where WIP_0 is a constant.

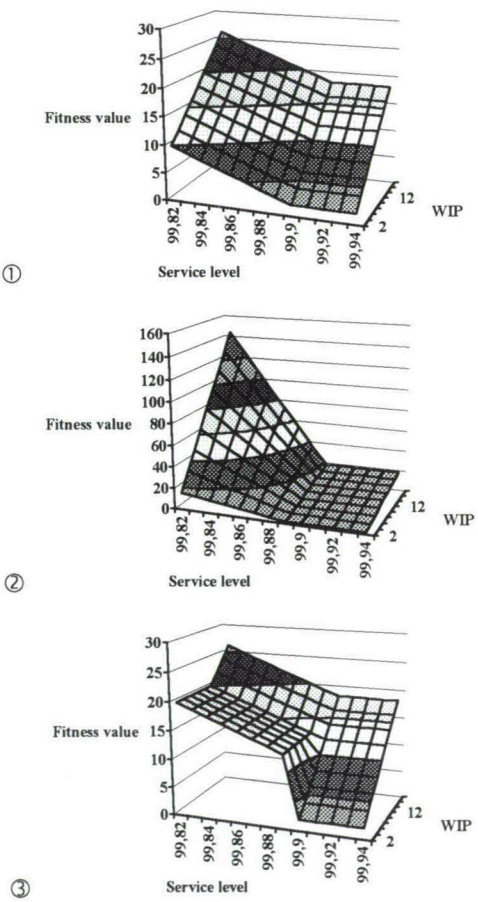


Figure 19. Fitness for optimization with service target constraint of 99.9%

Of course, this plot representation is not correct since certain WIP-Service combinations cannot occur in practice. For instance, it may not be possible to achieve high service level with low WIP. Nevertheless, such plots illustrate soft versus hard constraints. Indeed, an

acceptable solution may slightly violate a soft constraint, whereas it should not violate a hard constraint. For the sake of comparison with previous research, we choose to implement the service level constraint as a strong constraint. This means that the fitness function cannot be continuous. Otherwise, as Figure 19 illustrates, solutions may be chosen that are close to the plot's optimum but do not respect the constraint. This issue becomes critical if the solutions around the plot's optimum and respecting the constraint do not correspond to any existing pull system. Plot ③ is a compromise between continuous and non-continuous functions. If WIP is above the value WIP_0 (12 in the figure), chosen high enough, or if Service is above target, then the plot is the same as plot ①; otherwise, as soon as the service target is not respected, the fitness value jumps to WIP_0 augmented with a penalty, which is a function of the deviation from service target. We shall use such a fitness function for most of our optimizations.

We evaluate the fitness of an individual through discrete-event simulation. As a simulation language we use SIMAN (Pegden *et al.*, 1991). The EA sends the vector of parameters corresponding to the individual as input to the simulation model, which returns a fitness estimation (see Figure 20). The fitness is a function of the simulation output variables. The reason for choosing simulation is mainly that its assumptions are less restrictive; the price is long computing times. To estimate the fitness in case of stochastic simulation, we can use either several replications or a single long simulation run. In Chapter 5 we shall discuss in more detail issues related to uncertainties in simulation and ways of dealing with them.

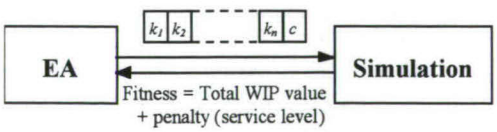


Figure 20. Simulation-optimization for pull control customization

3.5.3.3 Selection

We choose to implement the principles of elitism and the roulette wheel. So, part of the new population is an exact copy of the best solutions in the previous generation (elitism), whereas another part is selected randomly from the previous population (roulette wheel) and is changed by evolutionary operators. The roulette wheel selection is performed such that there are as few identical individuals in the new population as possible (Michalewicz *et al.*, 1996). The idea is to maintain a high degree of variety in the population, from generation to generation.

3.5.3.4 Evolutionary operators

The combination of EA and simulation is rather time-consuming. In order to save computer time, uninteresting solutions should be avoided. We use the two properties described in

section 3.4, whenever we need to choose a number of cards, that is, when we create the initial population, and whenever we use the mutation operator (alteration of data components) and recombination operator (partial repair of the offspring).

As a recombination operator we use a single-point crossover. To improve computing efficiency, the offspring may be partially 'repaired', avoiding uninteresting solutions. A simple reparation consists in making sure that there is an upper-bound to the overall WIP level. This can be done through Property 2, by searching for the minimal number of cards in all sequences of non-overlapping loops that cover the whole production line, including the "Conwip loop": Min k(ω), with $\omega \in \Omega_{1;N} \cup \{CL_{1;N}\}$. If this number is equal to infinity, then there is no upper-bound and we remedy this problem by selecting randomly a value for $k_{1;N}$ in the set {1, .., Min k(ω) - 1}.

For our customization we define the following mutation operator. The value of a mutated vector component $k_{i;j}$ must be chosen within the domain $\{1 \ldots c_{max}(k_{i;j})\} \cup \{\infty\}$, where $c_{max}(k_{i;j})$ is determined using the two properties of section 3.4:

$$c_{max}(k_{i;j}) = \text{Min}(c_{max1}, c_{max2}),$$

where c_{max1} and c_{max2} are determined using Properties 1 and 2 respectively. For this purpose we develop simple algorithms: for Property 1 we search for the smallest card number $k_{l;m}$, such that $l \geq i$ and $m \leq j$; for Property 2 we translate the recursive formula proposed in Appendix 1 into a recursive program.

The probability distribution for the selection of a vector component value within its domain is chosen as follows: ∞ is selected with a given probability denoted as p_∞, and any integer value of $\{1 \ldots c_{max}\}$ with *constant* probability $(1 - p_\infty)/\text{Card}(\{1 \ldots c_{max}\})$, where $\text{Card}(\{1 \ldots c_{max}\})$ is the number of integer values contained in the set $\{1 \ldots c_{max}\}$. The same probability distribution is used to define the initial population (step 1 in the algorithm).

This mutation operator can randomly generate any solution (or simplified equivalent) of the search space in the initial population. Furthermore, any solution of the search space can be reached from any other solution, using a finite sequence of mutations.

Next, we illustrate the benefits of customization. As an example we use a production system taken from the literature, for which the optimal configurations of several known pull systems have been determined by that literature.

3.6 Benefits of customization: an example

3.6.1 Bonvik et al. (1997)

The example in Bonvik *et al.* (1997) is a production line with four machines, inspired by the Toyota Motor Company; this line makes components for an automobile assembly line. Those authors perform extensive simulation experiments to study several control systems, namely Kanban, minimal blocking, Base stock, Conwip, and Hybrid Kanban/Conwip.

Their objective in terms of performance is to achieve a given service level with minimal inventory level. In these terms they show that the best Hybrid configuration outperforms the best Kanban configuration, and that this advantage grows as the demand rate increases. Kanban and minimal blocking perform similarly; the same close relationship is observed for Conwip and Base stock. Conwip and Base stock perform between Kanban and Hybrid.

To see whether considering new types of pull systems is of practical interest, we use the same production line as Bonvik *et al.* (1997). So we built a simulation model with the same assumptions. They assume that the delivery of raw materials is continuous and infinite, and that movements of products and cards are instantaneous. Moreover, inventory value is constant over the production line. The production system has the following other characteristics. Processing times at each station follow a lognormal distribution with a mean of 0.98 time units (minutes) and a standard deviation of 0.02 time units. Demand interarrival time is a constant, namely one time unit. (The system is feeding an assembly line that is modeled as a deterministic demand process consuming one part per minute.) If no finished product is available, the assembly line stops and demand is lost. (Actually, in Toyota plants, lost demands are prevented by working longer hours, until the production plan for the day is met.) Thus, it is essential to have a service level close to 100%; as Bonvik *et al.* (1997) do, we set the service target at 99.9%. Machines have times between failures and repair times that are exponentially distributed with means of 1000 and 3 time units respectively.

In accordance with Bonvik *et al.* (1997) we select a run length of 240,000 time units; we discard results collected during the transient period estimated to last 9,600 time units. We verify our simulation model by comparing its simulated output with results in Bonvik *et al.* (1997).

3.6.2 EA's convergence

We want to customize the generic model for the example production line described in the previous section. So we optimize the card numbers of the ten possible control loops. The result should show which control loops should be implemented. We use the evolutionary algorithm presented in section 3.5; we search in the integer set {1, .., 20} for each card number. In order to improve the search efficiency we introduce the best Conwip system (15 cards, as found by Bonvik *et al.*, 1997) in the initial population.

An important issue is whether the evolutionary algorithm converges to a same solution and at the same speed, independently of its parameter values. We study this convergence by optimizing the generic model for various mutation and recombination probabilities. The results are shown in Table 5, which consists of three parts: (i) the mutation and recombination probabilities (p_{mut} and p_{rec}), (ii) the resulting card numbers in each of the ten control loops of the best solution found by the algorithm (a shaded cell corresponds to an infinite number of cards), and (iii) the WIP and service performance of this best solution.

Table 5. Optimization results, given a 99.9% service target, for various EA parameter values (shaded: $k = \infty$)

p_{rec}	p_{mut}	k_{11}	k_{22}	k_{21}	k_{33}	k_{32}	k_{31}	k_{44}	k_{43}	k_{42}	k_{41}	WIP	Service
0.9	0.1			5					13		15	14.21	99.920
0.8	0.2			6			7		13		15	14.22	99.947
0.7	0.3			6							15	14.29	99.914
0.6	0.4						4		13		15	13.41	99.920
0.5	0.5						4		13		15	13.41	99.920
0.4	0.6			6	2						15	14.17	99.919
0.3	0.7						7		13		15	14.11	99.918
0.2	0.8						4		13		15	13.41	99.920
0.1	0.9			5		5					15	14.05	99.928

The nine solutions (nine lines in Table 5) have similar performance. The solutions for recombination probabilities equal to 0.2, 0.5, and 0.6 even have exactly the same performance: low WIP level compared with the other solutions, with a service level above the 99.9% target. The nine solutions also show similar structures: some control loops are never implemented (see columns labeled k_{11}, k_{22}, k_{44}, and k_{42}), others are implemented only once (k_{33} and k_{32}), and the remaining ones have finite card numbers in most cases. These remaining loops define a common structure for most solutions. This common structure is shown in Figure 21. Thus the algorithm does converge to a same structure of pull system. However, the card numbers in the implemented loops depend on the chosen mutation and recombination probabilities. So a single optimization does not yield a global optimum, but it may provide a good idea of the best structure.

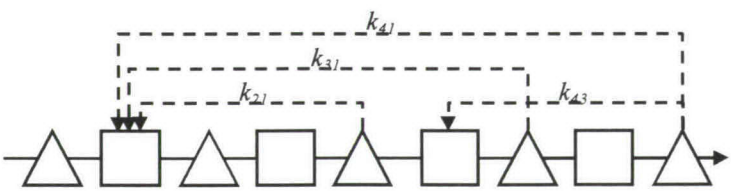


Figure 21. Common structure to most optimization results

The convergence speed can be studied by looking at the fitness of the best solution as a function of the generation number; see Figure 22. For reasons of readability we show these convergence plots for only four combinations of mutation and recombination probabilities. It is interesting to point out that convergence to a same solution (fitness $\cong$ 13.4) requires twelve generations in one case, and 21 in another case. In some other cases the optimization stops prematurely with higher fitness than in the best cases. This is due to our choice of the

stopping criterion: after a given number of generations without improvement of the best solution, the algorithm stops. Increasing that number may improve the convergence in terms of fitness, but at a cost of many more computations.

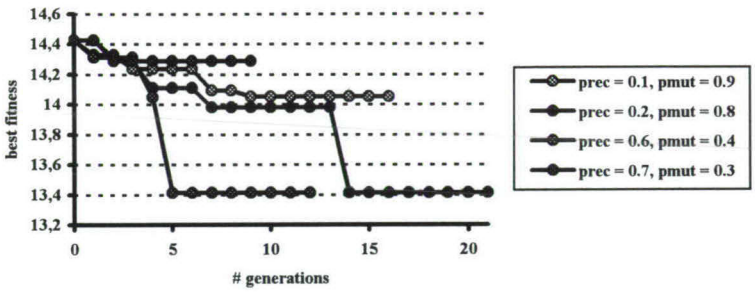


Figure 22. Convergence speed for various mutation and recombination probabilities

Our nine optimizations provide information on the best customized structure for the production line studied in Bonvik *el al.* (1997). However, they only indicate possible values for the best configuration of this customized structure: EAs are known to be efficient in finding good regions in the search space, but they may be less suitable for the exploration of these good regions. Therefore the literature recommends combining EAs with a local search technique, in a two-step optimization approach (Syrjakow and Szczerbicka, 1994). To perform this local search, we use a technique called Response Surface Methodology (RSM).

3.6.3 Fine-tuning through Response Surface Methodology

RSM is a heuristic sequential optimization technique based on regression (meta)modeling, design of experiments (DOE), and steepest ascent; see Kleijnen (1998). The principle of RSM is to build a set of regression (meta)models of the relationship among the simulation's input and output variables. An RSM algorithm is given in the following:

Step 1. Select a starting area in the search space, either randomly or using prior knowledge about the system to be optimized.

Step 2. Within the selected area, build a first-order regression (meta)model to get an approximation of the system's local input/output transformation.

If the metamodel is valid, then

Step 3. Use the regression model to estimate the gradient vector, showing the direction of the steepest ascent path.

Step 4. Select a starting point within the area defined in Step 1. Move from this point, along the steepest ascent path, into the direction that

improves the system's performance, until no further improvement is obtained.

Then, select a new area. Go to Step 2.

Else,

Step 5. Build a second-order regression model, within the selected area.

Step 6. Use the model of Step 5 to find analytically the input combination(s) that leads to an optimum.

To apply RSM to our specific customization problem two main adaptations have to be made. First, we need to build metamodels for both the overall WIP level and the service level. Second, since we already have an idea of the location of the optimal solution from the results presented in the previous section, we skip the first phase of RSM. Thus, we focus on the second RSM phase (Steps 5 and 6 of the algorithm), namely, estimate second-order polynomial approximations of the performance measures, and solve the resulting optimization problem analytically.

In section 3.6.2 (see Table 5), we identified four important simulation inputs (card numbers): k_{21}, k_{31}, k_{43}, and k_{41}. Our objective is to find the combination of these inputs that yields the best performance, that is, lowest inventory, given the 99.9% service level constraint. Hence, we want to build the following second-order polynomial approximations of overall WIP ($\overline{\text{WIP}}$) and Service level (S):

$$\overline{\text{WIP}} = \alpha_0 + \sum_{i,j}(\alpha_{i;j}k_{i;j} + \beta_{i;j}k_{i;j}k_{i;j}) + e$$

$$S = \gamma_0 + \sum_{i,j}(\gamma_{i;j}k_{i;j} + \chi_{i;j}k_{i;j}k_{i;j}) + e$$

where (i, j) takes value in the set {(2, 1), (3, 1), (4, 3), (4, 1)}, $\alpha_{i;j}$, $\beta_{i;j}$, $\gamma_{i;j}$, and $\chi_{i;j}$ are the regression parameters, and e is the additive random error. The regression parameters are estimated from the simulation input/output data: we simulate combinations of input values selected through design of experiments (DOE). We use a *central composite design*, which is a (fractional) factorial design (+1 and -1 is standardized notation) augmented with a one-factor-at-a-time design with two values per factor (-a, +a) (axial points giving star design), and the central point (0) (see Appendix 2). Using Table 5, we select the (non-standardized) input values for k_{21}, k_{31}, k_{43}, and k_{41} that are shown in Table 6. More details about regression parameters estimation for the customization problem can be found in Gaury *et al.* (1998).

Table 6. Input values for central composite design

	-a	-1	0	+1	+a
k_{21}	3	4	5	6	7
k_{31}	3	4	5	6	7
k_{43}	11	12	13	14	15
k_{41}	13	14	15	16	17

Once the regression parameters are estimated, we can solve the constrained optimization problem analytically through the technique of the *Lagrangean* multiplier (say) λ:

$$\text{Min}[\overline{\text{WIP}}\,(k_{i;j}) + \lambda S(k_{i;j}) - 99.9\lambda]$$

So we set the five partial derivatives $\partial/\partial k_{21}$, $\partial/\partial k_{31}$, $\partial/\partial k_{43}$, $\partial/\partial k_{41}$, and $\partial/\partial\lambda$ to zero, which gives five equations. Solving this system of equations gives several real-valued solutions, some of which are not acceptable (negative values for card numbers). Searching among the closest integer-valued solutions yields the following result: $(k_{21}, k_{31}, k_{43}, k_{41}) = (3, 6, 12, \infty)$. Next, we compare our customized system to the best result obtained so far for this specific production line, namely, Bonvik's Kanban/Conwip Hybrid.

3.6.4 Discussion of customizing

Bonvik *et al.* (1997) perform an exhaustive search for the best configuration for each of their production control systems. They simulate all configurations with card numbers less than 5 for stages 1 to 3, less than 25 for the last stage, and less than 25 for the Conwip loop. They conclude that Hybrid is the best system, and that its best configuration is $c = 15$, $k_1 = 2$, $k_2 = 3$, $k_3 = 5$, and $k_4 = 15$ (actually, their article gives $c = 13$, but in a letter to us they confirm that the correct value is 15). The 95% confidence interval for WIP in this system is 13.93 ± 0.03; for service it is 99.907 ± 0.007.

For the same production line as Bonvik *et al.* (1997) studied, we obtain significantly different pull systems through our customization approach:

(i) *New structure*. Our procedure (EA, RSM fine-tuning) results in a type of pull system – see Figure 23 – that does not belong to any predefined type of pull system: it does not match the typology of pull systems presented in Chapter 2. Indeed a noticeable property of this solution is that it does not have a Conwip loop. Therefore, it is completely different from Bonvik's Hybrid.

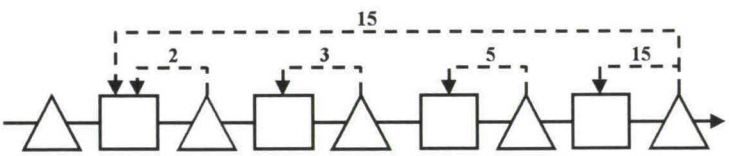
(ii) *Improved performance*. The following performance measures are averaged over 30 replications. For comparing our solution to the best Hybrid found by Bonvik *et al.* (1997), we use the paired t-test with 95% confidence (see Law and Kelton, 1991, pp. 587). Our system yields an overall WIP of 13.37 units against 13.93 units for Bonvik's best Hybrid. The paired t-test shows that this is a significant difference. Apart from statistical significance, this difference represents a decrease by more than 0.5 units,

which is an important gain considering that Hybrid already outperforms systems such as Kanban, Conwip, and Base stock. Given the service level constraint of 99.9%, we have $S = 99.88\%$ for customized, against $S = 99.91\%$ for Hybrid. Hypothesis testing, however, does not show a significant difference between the two systems, at a confidence level of 95%. Hence, our system yields a significantly lower WIP level with the same service level.

(iii) *Lower complexity*. Our customized pull system is also less complex as it has only three parameters (three control loops), whereas Hybrid has five parameters for this specific example of production line.

In summary, our solution is a type of pull system that has not been considered in the literature before; it yields significantly better performance – with a lower complexity – than the best system so far, namely, Bonvik's Hybrid. Therefore, this illustration indicates that our customization approach can yield results of practical interest in terms of performance and complexity.

Best Hybrid in Bonvik *et al.* (1997)



Our best customized solution

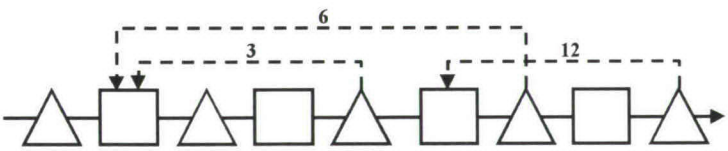


Figure 23. Customized generic model for the example in Bonvik *et al.* (1997)

3.7 Conclusion

We propose a novel approach to the design of pull control systems for single-product flow production lines. Instead of limiting these systems to traditional Kanban, Conwip, and Base stock systems, we design *customized* systems. For this customization we use a *generic* control model that connects each stage of a given production line with each preceding stage. *Optimization* of the generic model shows which potential control loops should actually be implemented. A single optimization is enough for selecting and configuring a pull system among all possible instances of pull structures. This is major improvement

compared with the selection technique that is usually proposed in the literature (one optimization per type of pull system under consideration). Furthermore, we exploit structural properties of the generic optimization model in order to limit the size of the search space. Optimization combines an evolutionary algorithm for searching with discrete-event simulation for performance evaluation of the generic model configurations.

Our approach may result not only in the three traditional pull systems, but also in combinations of these three systems. Moreover, this approach extends the pull concept to control systems that have never been investigated in the literature.

We illustrate our approach with the example of a production line taken from the literature on pull systems (Bonvik *et al.*, 1997). The outcome of our customization procedure is a pull system that has never been considered before, that shows significantly better performance than the best pull system so far, and that has a lower complexity in terms of number of parameters. Thus, our approach may be of practical interest.

The objective of the next chapter is to gain more insight into customization. Therefore we shall investigate customization for a wide variety of production lines: we define a sample of production lines that are compatible with models studied in the literature, and we customize the pull system for each of these lines.

Chapter 4
Customization for a Variety of Production Lines

Abstract

To gain more insight into customization and its benefits, we apply our methodology to a variety of production lines. We review the pull literature to determine this variety. Through that review we identify ten factors - such as line length, demand variability, and machine breakdowns - and classify these factors into three categories: (i) process, (ii) demand, and (iii) performance factors. To select typical values for each factor, we further examine the literature. Then we use a Plackett-Burman experimental design to generate a sample of twelve production line configurations. For each production line we apply the customization methodology proposed in Chapter 3. The results provide many interesting conclusions. First, none of the pull structures presented in our typology in section 2.1 is best for all production lines. Second, the resulting control systems are quite simple; Conwip, which is the simplest pull system, yields a performance that is often close to our customized system. Sometimes, however, customization can reduce the inventory value by up to 17% compared with Conwip, and still satisfy the service target. Third, the customized systems reveal three important structural patterns for their control loops. Hence, an optimization model that combines only these three patterns might be preferred to the complex generic optimization model, as it reduces the computational time requirements.

4.1 A sample of twelve production lines

In the previous chapter we presented a methodology for designing a customized pull control system for a given production line. To evaluate that methodology we now apply it to a sample of production lines. In order to select that sample, we analyze the characteristics of production lines that have already been studied in the pull literature (mainly on Kanban). Our analysis yields ten factors, together with their typical values. These factors may be classified as characterizing the (i) process, (ii) demand, and (iii) performance. Next, we review the factor values investigated in the literature, and we choose two values or *levels* per factor. (Table 12 gives an overview of our ten factors and their levels; we shall return to that table.)

4.1.1 Process factors

4.1.1.1 Line length

Chu and Shih (1992) point out that most of the Kanban models in the literature are relatively small: the usual line length is four or five stages; the largest length is nine (an outlier is 50, studied in Krajewski *et al.*, 1987). Conwip models, however, are simpler so they might easily be studied for larger systems. Table 7 gives some references for line length. For our sample we choose a low level of four, and a high level of eight stages.

Table 7. Line length in the pull literature

Reference	Control system	Line length
Krajewski *et al.* (1987)	Kanban	50
Sarker and Fitzsimmons (1989)	Kanban	9
Spearman *et al.* (1990)	Conwip	10
Meral and Erkip (1991)	Kanban	3, 4, 5, 6
Savsar and Al-Jawini (1995)	Kanban	3, 5, 7
Bonvik *et al.* (1997)	Kanban, Conwip, Hybrid, Base stock	4

4.1.1.2 Line imbalance

A line is said to be non-balanced (bottlenecked) when machines along the line do not have the same production rate. Sarker and Harris (1988) and Gupta and Gupta (1989) claim that balanced pull systems outperform non-balanced ones. The literature, however, does not always agree: for instance, Villeda *et al.* (1988) show that some imbalance patterns can improve output rates. Various patterns are defined and analyzed (Hillier and Boling, 1966); for example, bowl (the machines at the two ends of the line have the highest mean

processing times), funnel (mean processing times are getting shorter from stage to stage), and reversed funnel (mean processing times are getting longer from stage to stage). In theory the imbalance factor should also consider processing time variances and machine breakdown rates as possible causes of bottlenecks. We, however, focus on the mean processing times, because we will consider process variability and machine reliability as separate factors. (For assembly systems, Powell and Pyke (1998) analyze imbalance in both processing time means and variances).

An interesting measure of imbalance is defined by Meral and Erkip (1991): the Degree of Imbalance (*DI*) of a line is

$DI = \max\{TWC/N - \min(PT_i); \max(PT_i) - TWC/N\}N/TWC$

where *PTi* is the mean Processing Time at workstation *i* in an *N*-station line, and *TWC/N* is the mean processing time at a workstation on the balanced *N*-station line; *TWC* stands for Total Working Capacity.

For our sample we define two factors: *DI* and imbalance pattern. Table 8 reviews some of the values for *DI* in the Kanban literature. As *DI* levels we choose 0 (perfect balance) and 0.5. For the latter *DI* level we consider two imbalance patterns: funnel and reversed funnel (we see the bowl pattern as a combination of funnel and reversed funnel, so we do not study bowl).

Table 8. Degree of imbalance in the Kanban literature

Reference	DI
Villeda *et al.* (1988)	0.0 to 1.4 (step 0.2)
	0.0 to 0.7 (step 0.1)
Meral and Erkip (1991)	0.0, 0.1, 0.2, 0.45
Yavuz and Satir (1995)	0.0, 0.1, 0.3, 0.5

4.1.1.3 Processing time variability

It is well known that performance is sensitive to processing time variability. As a measure of variability we use the Coefficient of Variation (CV), which is the standard deviation divided by the mean. Table 9 reviews CV values used in the study of Kanban systems; most values are between 0.0 and 1.0. For our sample, we use a low level of 0.1 and a high level of 0.5.

Table 9. Coefficient of Variation of processing times in the Kanban literature

Reference	Coefficients of variation, CV
Sarker and Fitzsimmons (1989)	0.0, 0.1, 0.2, 0.3, 0.4, 0.6, 0.8, 1.0
Meral and Erkip (1991)	1.0, 1.5, 2.0
Swinehart and Blackstone (1991)	0.2, 0.5, 1.0
Savsar and Al-Jawini (1995)	0.2, 0.6, 1.2, 1.8
Yavuz and Satir (1995)	0, 0.1, 0.5, 0.9

4.1.1.4 Machine reliability

To model machine breakdowns we need two random variables: Time Between Failures (TBF) and Time To Repair (TTR). Many different distributions have been used in the literature, especially exponential (most popular), uniform, and normal. Table 10 reviews TBF and TTR distributions in the Kanban literature. We define two levels for machine reliability: either all machines are considered as perfectly reliable, or TBF and TTR are exponentially distributed.

Table 10. Distributions for machine breakdowns in the Kanban literature

Reference	TBF Distribution	TTR Distribution
Krajewski *et al.* (1987)	Normal	Normal
So and Pinault (1988)	Exponential	Exponential
Sarker and Fitzsimmons (1989)	Normal	Exponential
Wang and Wang (1990)	Exponential	Exponential
Yavuz and Satir (1995)	Uniform	Uniform
Bonvik *et al.* (1997)	Exponential	Exponential

4.1.2 Demand factors

4.1.2.1 Demand rate

In a production system the demand rate may change frequently. So it is a key issue to know whether the choice of a control system depends on the demand rate: once a control system is implemented, it might not be changed easily. Demand rate has to be defined relatively to line capacity. Thus, we take as one of our factors the ratio of demand rate and line capacity. We select these ratios equal to 0.8 and 0.9 respectively.

4.1.2.2 Demand variability

For demand variability we use the same principle as for processing time variability: we take its CV. Table 11 gives a sample of values in studies on pull control systems. We choose the

levels 0.0 and 0.5 (0.0 may correspond to dependent demand; for example, in Bonvik *et al.* (1997) the line feeds an assembly system that processes parts at a constant rate).

Table 11. Demand variability in the Kanban literature

Reference	Demand CV
Berkley (1996)	0.1, 0.045
Yavuz and Satir (1995)	0.052, 0.075, 0.115, 0.133
Savsar and Al-Jawini (1995)	0, 0.1, 0.316, 0.447, 0.707, 1

4.1.2.3 Customer attitude

We define customer attitude as willingness to wait for finished products. We consider two extreme cases: *lost sales* versus *backorders*. In the first case, customers do not accept any waiting, and do not order if they cannot be satisfied from stock. In the second case, the company backlogs orders that cannot be filled from stock. Most publications on pull production consider backorders only. Bonvik *et al.* (1997), however, study lost sales.

4.1.3 Performance factors

In this section we further discuss the performance criteria described in section 2.2.2: we see them as factors that may influence the outcome of our customization. Thus, we define several levels for (1) the service level target, and (2) the inventory value.

4.1.3.1 Service level target

The *service level* (fill rate) is the proportion of demand immediately supplied from stock. Obviously, the higher the level of finished good products, the higher the service level. Thus, the choice of a target value for this level affects overall WIP. This target should be set by managers; it varies with the type of production system. Targets close to 100% may be used for systems with lost sales, whereas lower targets may correspond with systems with backorders. So, it is necessary to consider service as a factor (not only as a performance measure). Setting a target for the service level has not often been done in the literature; yet, Bonvik *et al.* (1997) do use a target, namely 99.9%. We will also use service levels close to 100%: a low level of 95% and a high level of 99%.

4.1.3.2 Inventory value and added value

The ideal of pull production control is zero inventories. Thus, the inventory level is a major performance indicator. In some cases, however, managers may need to account for the value of inventories. Indeed, whereas keeping a high finished good inventory may be good for the service level, the added value may make this policy prohibitively expensive. Goldrat and Fox (1986) emphasize that inventory is money invested; minimizing this investment may improve competitiveness. We use the total value of inventories along the line as a

performance measure. We assume that each stage adds the same value. We define the total-value factor as the ratio of the finished good value and the raw material value. Our two levels are 1.0 (value not of interest to managers; total inventory value equals total inventory level) and 2.0. We expect these two cases to yield different inventory allocations along the production line, and different customized pull systems.

4.1.4 Summary of factors and levels

Table 12 gives an overview of our ten factors, now labeled from A through J, and their levels. Given these ten factors and their two levels, we can easily design an experiment

Table 12. Design factors and levels

Factor	Levels		Letter
Line length	4	8	A
Line imbalance	0	0.5	B
Imbalance pattern	Funnel	Reversed funnel	C
Processing time CV	0.1	0.5	D
Machine reliability	Perfect	Breakdowns	E
Demand CV	0	0.5	F
Demand rate/capacity	0.9	0.8	G
Service level target (%)	99	95	H
Inventory value ratio	1	2	I
Customers' attitude	Lost sales	Backorders	J

Table 13. Plackett-Burman design for production lines configurations

Legend: + and - denote the two values per factor

	Factors									
Line #	A	B	C	D	E	F	G	H	I	J
1	+	+	-	+	+	+	-	-	-	+
2	+	-	+	+	+	-	-	-	+	-
3	-	+	+	+	-	-	-	+	-	+
4	+	+	+	-	-	-	+	-	+	+
5	+	+	-	-	-	+	-	+	+	-
6	+	-	-	-	+	-	+	+	-	+
7	-	-	-	+	-	+	+	-	+	+
8	-	-	+	-	+	+	-	+	+	+
9	-	+	-	+	+	-	+	+	+	-
10	+	-	+	+	-	+	+	+	-	-
11	-	+	+	-	+	+	+	-	-	-
12	-	-	-	-	-	-	-	-	-	-

through the statistical theory on Design Of Experiment (DOE); see Kleijnen (1998) for an introduction to DOE in simulation. DOE combines the various factor levels, which define a (systematic, non-random) sample of line configurations. To minimize the number of configurations that must be simulated and optimized, we use the Plackett-Burman design for ten factors combined in twelve configurations (see Appendix 2); see Table 13. This table must be read as follows: line configuration 1 has factor A (line length; see Table 12) at its + level (the line has 4 stages; see Table 12 again), B (line imbalance) is + (value 0), ..., I (inventory ratio) is – (value 2), and J (customer attitude) is + (lost sales).

4.2 Results of customizing the generic pull system

Because it would be difficult to use analytical techniques to study the line configurations in Table 13, we choose simulation to evaluate the performance of these configurations. For this simulation we use the SIMAN simulation language (Pegden *et al.*, 1991). We estimate the performance measures through a single long run per pull system configuration; each run has 240,000 time units, after elimination of a start-up period of 9,600 time units (these figures are the same as in Bonvik *et al.*, 1997). The next step is to find the optimal configuration of the generic pull control system for each line.

We build twelve generic models that correspond with the twelve production lines in Table 13. Each of these models must be optimized. These models have $N(N + 1)/2$ control loops; in Table 12, N is 4 or 8 (factor A), so there are 10 or 36 loops. Per loop we try to optimize the number of cards, $k_{i;j}$; for these numbers we consider 21 integer values including infinity. So the search space consists of 21^{10} or 21^{36} possible solutions. We further simplify our customized control systems as follows: we measure the WIP level within each segment of the line controlled through a loop; if the maximal WIP level for that segment remains below the corresponding number of cards, then this control loop does not need to be implemented.

As a yardstick for the results of our methodology we select Conwip, because Conwip is the simplest policy and it has been proven to be very efficient. Since this pull system has only one parameter, we find its optimal configuration (lowest number of cards that still guarantees a service above target) very easily by running the simulation models for increasing numbers of Conwip cards, starting from a low number: when the service level performance exceeds the target for the first time in our search, we have the optimal number of Conwip cards.

A detailed description of the results of our methodology is given by Gaury *et al.* (1998); here we summarize these results, as follows. We expect that the 'best' customized control systems may be (i) one of the traditional pull systems (Kanban, Conwip, Base stock), (ii) a segmented, (iii) a joint, or (iv) a new type of control system. Indeed, the twelve best control systems – for the line configurations in Table 13 – do turn out to belong to these four types.

We note that since we use a heuristic optimization technique, we cannot be sure of having found optimal solutions.

(i) For line configuration #1 the result (see Figure 24) is a *traditional Kanban* system; see section 2.1.1.1. The card (optimal) numbers are one, for each stage. So this is a highly synchronized system: as soon as a machine stops, the upstream stages are not authorized to produce anymore and the downstream stages starve. An advantage is extremely low inventory. Such a system, however, is highly sensitive to variations (demand variability, process variability, breakdowns, etc). In fact, it is well known that a single Kanban card per control loop suffices if the production environment is ideal. Indeed, we saw in section 2.2.3 that Japanese managers use the following empirical formula to determine the number of Kanban:

$$y_i \geq D_i L_i(1 + \alpha_i)/a,$$

where y_i is the number of Kanbans at stage i, D_i is the average demand per time unit at stage i, L_i is the average production lead-time at i, α_i is a variable for safety stock at i, and a is the container capacity (a single Kanban is attached to each container). If demand is met with probability one, then $D_i L_i = 1$. If there is no variation, then safety stocks are not needed, so $\alpha_i = 0$. Moreover, if transport and switchover costs are unimportant, then there is no need for containers so $a = 1$. Thus, in an ideal production environment, the minimum number of Kanbans at each stage is one: $y_i \geq 1$. In line configuration #1, demand is indeed constant, processing time variability is low, and customer pressure is low compared to line capacity (service level of 95% and no backlog); see Table 12 and line 1 of Table 13. Thus, it seems reasonable to have a single Kanban per loop.

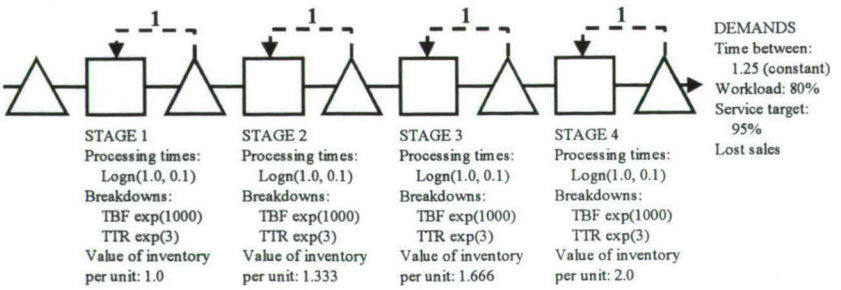


Figure 24. Customized system for line configuration #1

(ii) Though none of the twelve lines yields a *segmented* system (see section 2.1.2), the customized system obtained for configuration #2 is almost a segmented Conwip system; see Figure 25.

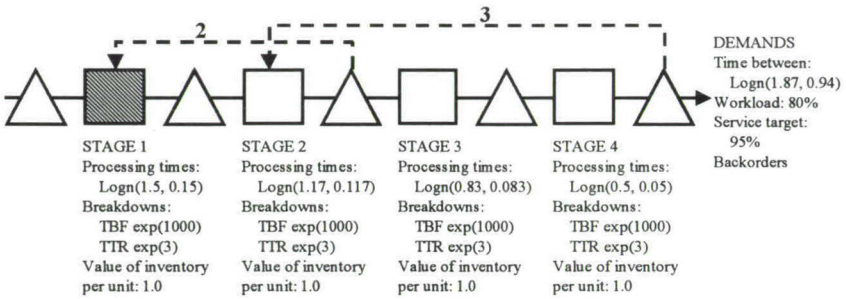


Figure 25. Customized system for line configuration #2

(iii) Some line configurations give *joint* control systems, which combine several policies on the same portion of the line. In configurations with eight stages, these systems can be rather complex; they yield little benefit compared with the simpler Conwip system.

(iv) For some other configurations *new control types* result that connect each stage of the line to the first stage. A typical example is configuration #5; see Figure 26. This system permits release of raw materials (stage 1) only if each stage of the line allows production – one loop (from stage 3) does not need to be implemented. Once a part is released, it does not require any additional authorization to progress. Another example of a new control type is found for line configuration #10; see Figure 27. For the first two stages this configuration uses the new policy derived for line configuration #5; for the last three stages it uses Integral Control.

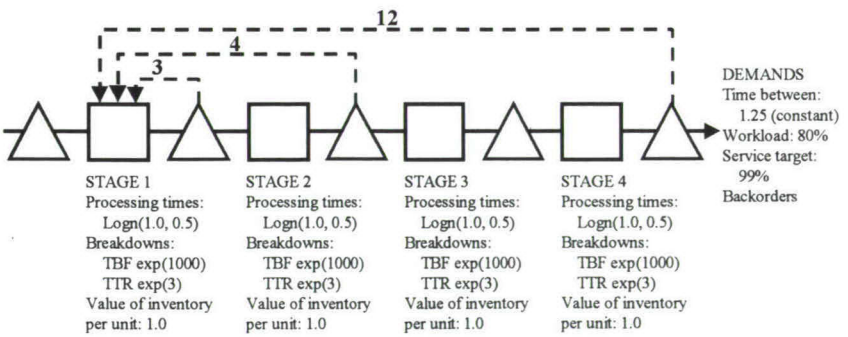


Figure 26. Customized system for line configuration #5

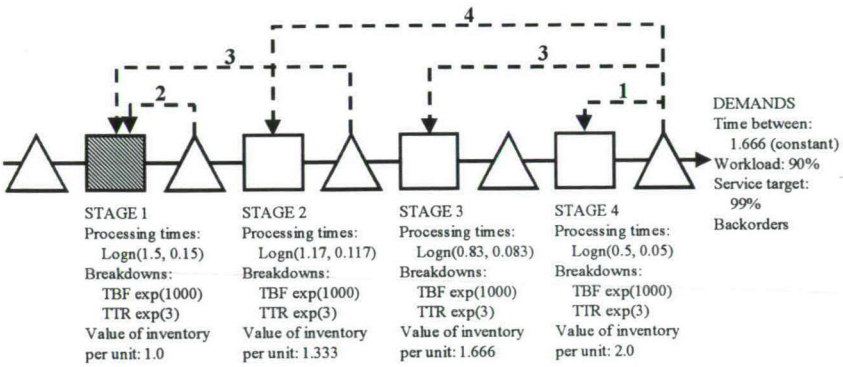


Figure 27. Customized system for line configuration #10

In summary, optimization of the generic model leads to *simpler* control systems than we expected: the generic model has $N(N+1)/2$ loops, whereas the optimized control systems rarely have more than $N+1$ loops! For many configurations, however, we prefer *Conwip* because the customized system does not reduce the inventory value substantially. Most of these configurations have eight stages. More generally, we observe that the relative benefit of the generic system often depends on how close to target Conwip's service level is. Indeed, the customized systems realize some of the largest WIP reductions when Conwip

Table 14. Performance of Generic versus Conwip for each of the twelve configurations in Table 13

Line #	Inventory value			Service level		
	Generic	Conwip	Gain (%)	Generic (above target)	Conwip (above target)	Loss (%)
1	5.99	6.38	6.11	98.91 (3.91)	100.00 (5.00)	1.09
2	4.10	4.73	13.32	96.31 (1.31)	99.21 (4.12)	2.92
3	17.42	17.96	3.01	99.01 (0.01)	99.17 (0.17)	0.16
4	9.36	9.95	5.93	95.13 (0.13)	95.49 (0.49)	0.38
5	11.34	11.70	3.08	99.05 (0.05)	99.31 (0.31)	0.26
6	15.18	16.35	7.16	99.00 (0.00)	99.17 (0.17)	0.17
7	5.30	6.00	11.67	97.48 (2.48)	99.07 (4.07)	1.60
8	7.91	7.99	1.00	99.03 (0.03)	99.12 (0.12)	0.09
9	14.90	15.14	1.59	99.04 (0.04)	99.12 (0.12)	0.08
10	8.52	10.33	17.52	99.00 (0.00)	99.36 (0.36)	0.36
11	41.21	42.36	2.71	95.11 (0.11)	96.00 (1.00)	0.93
12	16.68	17.28	3.47	95.06 (0.06)	96.01 (1.01)	0.99

overshoots the service target: see lines #2 and #7 in Table 14. An exception is configuration #10: inventory value in our customized solution is 17.52% lower, compared to the best Conwip system, for a service level decrease of 0.36% only. In general, customization may yield significant WIP reductions along the line, and still satisfy the service target.

4.3 General results: simplification through meshing

The results of customizing the generic pull system for a variety of production lines provide information that can be exploited for generalization. In this section we derive commonalties in the structure of the customized pull systems. Three main structural patterns seem to emerge; they correspond to different ways of reducing inventory value.

(i) First, the release of raw materials (at stage 1) can be limited so that the overall inventory level is reduced. In most of the customized pull systems, the release of raw materials requires authorizations from several stages, not just from the last stage as in Conwip or Base stock. Thus, if one machine in the line fails, the release of raw materials may be blocked much earlier than in Conwip system: the control system reacts faster.

(ii) Second, when value is added along the line, the total inventory value can be reduced through a more efficient allocation of WIP along the line. Indeed, when inventory value is considered, the fact that Conwip pushes parts to the last stage is a disadvantage in terms of costs; our customized systems result in lower costs. To achieve this cost reduction our systems often have several control loops that link the last stage to upstream stages. The extreme case of this structural pattern is Integral control (see Base stock in section 2.1.1.3); our customized systems do not link the last stage to all preceding stages but only to a few of them. Configuration #10 (see again Figure 27) illustrates the gains achieved through this second structural pattern. Table 15 clearly shows that the WIP allocation along the line differs completely between these two solutions: Conwip concentrates WIP at the end of line, whereas our customized system allocates more WIP at the first stage and in the middle of the line. The result is a reduction of total inventory value by 17.5% in our solution.

Table 15. Performance of the customized and best Conwip systems for line configuration #10

		Service	WIP at stage			
	Value	*Target is 99%*	1	2	3	4
Best Generic	8.52	99.00%	1.94	0.99	1.96	0.99
Conwip 6	10.33	99.36%	0.91	0.71	0.50	3.82

(iii) The third structural pattern is less obvious than the two patterns: it appears only when one of the machines is a bottleneck. In those cases, control loops link the input inventory of the bottleneck to the previous stages so that inventory in front of the bottleneck does not grow unnecessarily. Goldratt has emphasized the importance of

focusing on bottleneck machines in his famous book *The goal* (Goldratt and Fox, 1986); he further explained his theory in the Theory Of Constraints (TOC). In our investigation, however, bottleneck machines do not seem to play such a predominant role in the positioning of control loops. The reasons may be that only three line configurations in our sample of twelve have a bottleneck machine within the line (not at one extremity) and that the structural patterns presented in (i) and (ii) control the whole line, including the bottleneck resources.

Many customized systems combine these three structural patterns. A typical example is configuration #7, shown in Figure 28. In this example, pattern (i) loops are displayed in solid lines, pattern (ii) in long-dashed lines, and pattern (iii) in round-dotted lines.

Line configuration #7 is not an exception: all four-stage lines and many eight-stage lines are a superimposition of the three structural patterns only; the few remaining eight-stage lines add some control loops that do not match the three patterns. This is an important conclusion that leads us to focus on these three structural patterns. So instead of using the generic optimization model described in section 3.3.2, we now limit our investigation to a simpler optimization model that combines the three structural patterns only. We call this concept *meshing*; the terminology is inspired by the field of structural engineering, which studies the resistance of mechanical structures ranging from bolts to bridges and buildings. Such studies are based on computer models that decompose structures into small elements; the technique is called finite elements decomposition. A key principle is to define elements of smaller size, that is, make a fine meshing in those places where constraints are expected to be important. We use the same approach for our customized pull systems: we put potential control loops at the places where constraints are expected to be important. The gain in terms of reduced complexity is important: our original customization model had

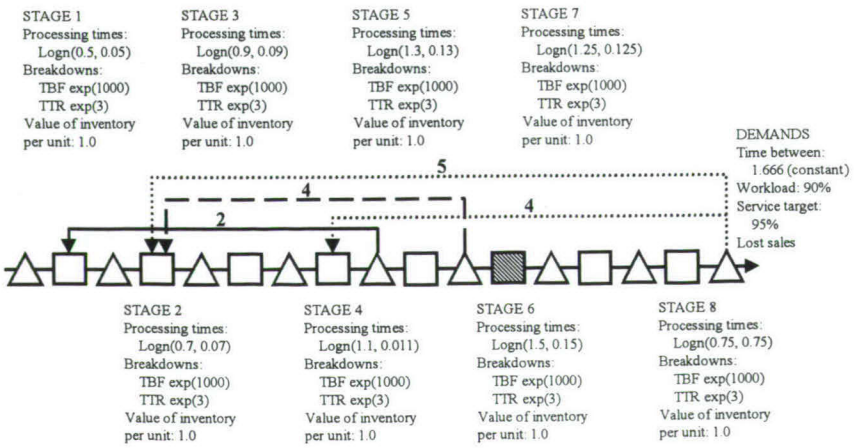


Figure 28. Customized system for line configuration #7: combination of three control patterns

$N(N+1)/2$ potential control loops, whereas our new meshed model has only $2N - 1 + P$ control loops, with $0 \leq P < N - 1$ depending on the position of the bottleneck in the production line (order N^2 versus N). The gain becomes particularly important as the production line length N grows. Table 16 compares the complexity of our original customization model with the complexity of the meshed model when the bottleneck is positioned at stage 1 or N ($P = 0$, minimal complexity), and at stage $N - 1$ ($P = N - 2$, maximal complexity).

Table 16. Complexity of customization versus meshed optimization model

# stages	# possible loops		# possible structures	
	Custom	Meshed*	Custom	Meshed*
4	10	7 / 9	1024	128 / 512
10	55	19 / 27	3.6×10^{16}	5.2×10^{5} / 1.3×10^{8}
20	210	39 / 57	1.6×10^{63}	5.5×10^{11} / 1.4×10^{17}

* min./max. complexity

4.4 Effects of production line characteristics

Another important result of our sample customization is that WIP varies drastically with the characteristics of the production lines. For instance, the number of cards (determining WIP) in the best Conwip system vary between 4 and 12 for four-stage lines, and between 6 and 27 for eight-stage lines. Hence overall WIP in an eight-stage line can increase by a factor up to 4.5, depending on the characteristics of the line! This result agrees with a general conclusion of Krajewski *et al.* (1987): improving the production environment is a potential source of bigger gains than improving the control system only. Indeed, the biggest gain obtained by changing the control system from Conwip to a customized system is a 17.5% reduction of total inventory value. However, many factors in the production environment are not within control of a manager, whereas the production control system is.

Our sample enables us to study the effects of the ten factors defined in section 4.1. For this purpose we use *analysis of variance* (ANOVA), which consists of fitting a regression model to the input/output data of the simulation. As input we use the standardized values in Table 12 where the symbol - now means -1 and + means +1; this standardization allows a fair comparison of the factor effects independently of their measurement scales (see Kleijnen, 1998 for details). As output we select the optimized number of Conwip cards. This ANOVA shows that a first-order approximation (main effects only) is not very adequate: the adjusted coefficient of determination (Adj.R^2) is only 0.517 (an exact fit would yield a value of 1.0): see Table 17). Hence the following discussion of these estimated main effects is only approximate (estimating higher-order effects would require many more configurations to be simulated and optimized). Five of the ten factors seem to have an important effect: line length (factor A), line imbalance (B), processing time's CV

(D), ratio of demand rate and capacity (G), and customer attitude (J). These results suggest where to focus the efforts to improve the performance of production lines controlled by a customized pull system. Surprisingly, the demand CV (factor F) and the service target (H) seem to be unimportant. These conclusions, however, also depend on the ranges over which we change the factors in our experiment. We emphasize that our goal is not to derive general recommendations based on extensive simulation experiments; instead, we wish to show that environmental factors may have a major effect on performance. A difficulty is that a manager rarely has direct control over these environmental factors. Thus these effects may be difficult to avoid or limit.

Table 17. Main effects of ten factors on the optimal number of Conwip cards
$R^2 = 0.956$, Adj.$R^2 = 0.517$

Factor	A	B	C	D	E	F	G	H	I	J
Regression Coefficient	-2.58	3.08	0.92	-2.42	1.08	0.08	2.08	-0.08	-0.75	-2.25

4.5 Conclusion

To gain more insight into customization and its benefits, we applied our methodology to a variety of production lines, namely twelve configurations. These combinations were selected through an experimental design (Plackett-Burman) with ten factors, such as line length, demand variability, and machine breakdowns. The results provide the following important conclusions.

There is not a single dominant type of pull control system. Indeed, depending on the characteristics of the production line, the best system may be traditional, segmented, joint, or novel. This conclusion further validates the need for customization.

Despite the complexity of our generic optimization model, the resulting control systems are quite simple. Conwip, however, remains the simplest system; its performance is often close to our customized system. Therefore we prefer Conwip for most lines with eight stages. Nevertheless, customization can reduce the inventory value by up to 17% compared with Conwip, while our system still satisfies the service target.

Our results also reveal three important *structural patterns* for control loops. One pattern links each stage to the first stage; another pattern links the last stage to each preceding stage (Integral Control); the last pattern links the input inventory of the bottleneck machine to each preceding stage. These patterns characterize most of our customized solutions. The first pattern and its combination with the other patterns have not been mentioned in the literature. Since our generic optimization model is rather complex – especially for long

production lines – we might prefer an optimization model that combines the three patterns only.

We also studied the performance effects of the production line characteristics. Our conclusion is that bigger gains may be achieved by modifying the production environment instead of changing the production control system only. Krajewski *et al.* (1987) refer to this approach as 'shaping the environment'. A manager, however, may not have control over all ten factors studied in this chapter. Moreover a production environment is rarely stable over time. So 'shaping the environment' may not be the best solution. The objective of the next three chapters is to review and develop ways of dealing with the effects of the production environment on performance. We shall see that a famous technique in quality control developed by Taguchi, consists in designing products or systems that are less sensitive to the effects of the environment, instead of continuously 'shaping the environment'.

Chapter 5
Uncertain and Dynamic Environments

Abstract

In this chapter, we identify three sources of uncertainty in simulation: (i) stochastic uncertainty, which is due to the use of (pseudo)random numbers in our discrete-event simulation, (ii) subjective uncertainty, which results from our need to estimate the probability distributions based on either sampled data or expert opinions, and (iii) dynamic uncertainty, which results from variations in the real production environment. Through simple examples we illustrate the possible effects of these three sources of uncertainty. We conclude that simulation studies for system design should assess the effects of these uncertainties, and try to integrate them in the design process. In the literature, various techniques have been developed to tackle different aspects of uncertainty assessment and integration. These techniques, however, have been used in different fields and have never been integrated. We present two main techniques, namely, uncertainty/risk analysis and robust design (Taguchi), and we show how they can be combined to support system design through simulation.

5.1 Introduction

A simulation model can be seen as a black box that processes inputs to produce outputs. There are two categories of inputs: the environmental parameters, which are used to model the production environment, and the decision variables, which characterize the factors that are controllable by the designer or manager of the production system. Simulation outputs can be any kind of performance measures. The purpose of design through simulation is to decide which values to give to the decision variables so that either an optimal or a satisfactory level of estimated performance is achieved. The main difficulty of system design through simulation is that the outputs are not only functions of the decision variables: they also depend on the simulation model itself. Indeed, such a model is one specific representation of an existing or future system; the assumptions used for building this model are critical. Many uncertainties arise when building and using a simulation model, because it is difficult to model reality though mathematics and statistics, and that reality is dynamic. In this dissertation, we shall focus on three sources of uncertainty that are due to the input data of simulation only. Uncertainty in the model structure is rarely considered in the literature (Helton, 1997).

In such circumstances, decision making is not simple. We illustrate this through the following metaphor of continuous improvement found in the literature (also see section 1.2). The company is a boat floating on a sea of inventory, and rocks are problems that may arise when the inventory is too low. When this 'sea of inventory' is placid (no uncertainties), it is easy to decide what should be the inventory level. When, however, this sea is turbulent, the decision is much more difficult and involves a higher risk: uncertain parameters such as the amplitude and the frequency of the waves have to be accounted for; compare Figure 1 to Figure 29. In this chapter, we identify various sources of uncertainty involved in system design through simulation, and we investigate ways of incorporating them in the design process.

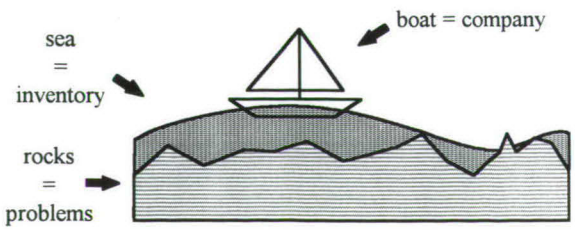


Figure 29. Managing inventory levels under uncertainty

5.2 System design using stochastic simulation

5.2.1 Stochastic uncertainty

As mentioned in section 2.2.1, we define the production environment as the set of factors that are not completely within control of the designer or manager of the production system. This environment includes processing times, demand rate, time between failures, etc. These factors change over time in a fashion that may be interpreted as randomness by an analyst. Yet, there may be an explanation for such changes; for instance, it is possible to explain why a component failed by looking at its microscopic structure. However, if the objective of the simulation analysis does not require such a level of precision, then the microscopic phenomena are modeled using probabilistic theories. Ziegler (1976, pp. 42) discusses thoroughly this modeling issue. In simulation, the environmental factors (say) X_i are modeled through random variables: a probability distribution f_i is associated with one or several X_i's and values are randomly selected from this (multi-variate)distribution. Thus, the output of a simulation is also a random variable. This randomness is inherent to the real system, and it cannot be reduced. It is often called *stochastic uncertainty*, but other designations can be found in the literature, such as aleatory uncertainty, irreducible uncertainty, and variability (Helton, 1997). This type of uncertainty differentiates a stochastic model from a deterministic one.

5.2.2 Subjective uncertainty

A difficult stage of simulation modeling is the definition of the probability distributions f_i. Several decisions have to be made concerning the shape or type of the distribution (such as exponential, normal, lognormal) and its parameters $\alpha_1, \ldots, \alpha_{f_i}$ (which fix the mean, median, variance). Many simulation studies assume that the production environment is known and modeled with certainty. This would mean that the input distributions are known with certainty. Yet, these parameters can only be estimated through real data (also called *objective data*) from an existing system or forecasts and expert knowledge (*subjective data*) for a system in its design phase. Because of the limited availability of data, analysts cannot know these parameters with certainty. This type of uncertainty is called *subjective uncertainty*; other designations in the literature include epistemic, analyst, and reducible uncertainty (Helton, 1997).

Both stochastic and subjective uncertainties are involved in any simulation-based analysis. Most simulation studies do account for stochastic uncertainty (using pseudo-random numbers); they rarely consider subjective uncertainty. Even books that are dedicated to simulation teaching do not mention this issue. For instance, Law and Kelton (1991) focus on model validation only: "if output is sensitive to some aspect, then that aspect must be modeled carefully". A recent paper that considers both types of uncertainty

is Helton (1997), which refers to Hacking (1975) as one of the first paper to make the distinction between the two types of uncertainties.

5.2.3 Dynamic uncertainty

Another characteristic of many studies is that the production environment is considered to be stable over time; this is a *static* formulation. Yet, a particular production environment does vary over time. For instance, the demand for a product evolves over time because of the product life cycle or seasonal trends. Also the production system may be improved: a bottleneck resource may be removed by adding capacity. Thus the characteristics $\alpha_{i,j}$ of the probability distributions f_i should be functions of time: $\alpha_{i,j} = g(t)$. It is often difficult to distinguish between randomness as defined in section 5.2.1 and a change in the characteristics of the production environment. Indeed, if a machine starts failing more often, is it just an effect of randomness (bad luck) or is it because the machine is getting old (a trend)? Furthermore, dynamic behaviors cause uncertainties, which we call *dynamic uncertainties* and can be interpreted partly as subjective uncertainties. Indeed, dynamics imply that the future environment is not known with certainty: the analyst can only forecast the evolution of the environment over time (in the field of quality control, Statistical Process Control is such a technique that tries to identify trends in product quality). Thus, we shall see that there is a strong relationship among the various sources of uncertainty; the approaches that deal with subjective uncertainty can also be used for dynamic uncertainty. However, we will continue to distinguish between subjective and dynamic uncertainties because the literature treated these two problems independently.

When the objective of a simulation study is to design a system, it is very risky to assume that the environment is known and modeled with certainty (we shall illustrate the effect of uncertainties on the simulation outputs). Thus we claim that in practice it is important to consider the uncertainty in our design issues for pull systems. In this dissertation we consider all three types of uncertainty. For the subjective uncertainty, however, we assume the type of distribution to be known. In Figure 30 we recapitulate the various sources of

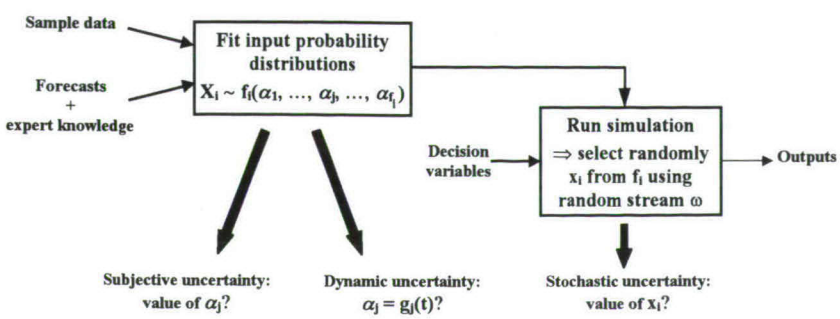


Figure 30. Three sources of uncertainty in design through simulation

uncertainty involved in any design study through simulation. Next, we look at the effects of uncertainties and dynamic environments.

5.2.4 Effects of uncertainties

In this section, we show the importance of the uncertainties discussed above. The outputs of a simulation (performance measures) can be defined as a function of the decision variables (card numbers, for instance), the shape and characteristics α_j of the probability distributions f_i, and the set of random numbers ω. For a given set of inputs of the simulation model, we want to study the effect of uncertainties on the outputs. Important issues are:

- What if the estimates for the α_j's are wrong?
- What if the production environment is evolving over time?

Let us consider a simple example, namely a bank office with a single office counter. Customers arrive at time intervals that seem random from the viewpoint of the analyst; they are served on a first arrived / first served basis. The service time also seems random: some customer's demands require a short service time only, whereas other more complex demands may require a much longer time. The analysts are interested in estimating the daily average number of customers waiting for service, given a design option (such as a single office counter). Thus, they build a model: using sample data they estimate that the time between customer arrivals X_1 follows an exponential probability distribution with mean $\mu_{1,0} = 5$ minutes and the service time at the office counter X_2 follows an exponential probability distribution too, but with mean $\mu_{2,0} = 4$ minutes. We denote this as $X_1 \sim f_1(\mu_{1,0}) = \text{Exp}(5)$ and $X_2 \sim f_2(\mu_{2,0}) = \text{Exp}(4)$. For this simple model, queuing theory provides a formula for the determination of the steady-state mean number of customers waiting for service (say) Q (see Gross and Harris, 1998, p.63):

$$Q = \rho^2/(1 - \rho), \quad . \tag{6}$$

with traffic intensity $\rho = \mu_2/\mu_1$.

When building the model the analysts estimate the mean values $\mu_{1,0}$ and $\mu_{2,0}$ of X_1 and X_2 from sample data. Now, as we saw in the previous section, the 'actual' means can be different from the analysts' expectation. Figure 31 shows the effect of μ_1 and μ_2 on Q; we vary the values of μ_1 and μ_2 within the respective intervals $[\mu_{1,0} - 2.5\%; \mu_{1,0} + 2.5\%]$ and $[\mu_{2,0} - 2.5\%; \mu_{2,0} + 2.5\%]$, and we compute the corresponding Q values from (6). The resulting Q values range from 2.4 to 4.5 units; for $\mu_{1,0}$ and $\mu_{2,0}$ the value (say) q_0 of Q is 3.2. This means that varying μ_1 and μ_2 by 2.5% only around their expected values yields values of Q in the interval $[q_0 - 25\%; q_0 + 40.6\%]$! Thus, in this example, small subjective uncertainties yield very large uncertainties for the output.

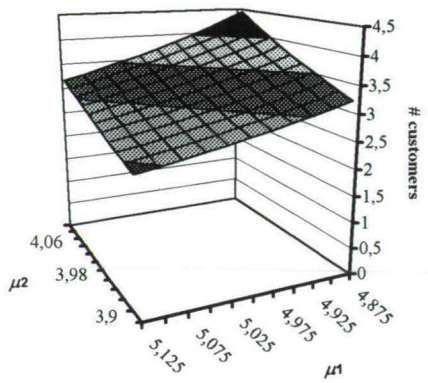


Figure 31. Sensitivity of Q to μ_1 and μ_2

This queuing analysis does not involve stochastic uncertainty: no random number are used. Considering stochastic uncertainty makes the analysis more difficult. Indeed, for fixed values of the inputs μ_1, and μ_2, the output of the simulation is different depending on which set of random numbers ω is used. Figure 32 shows the evolution over time of the number of customers waiting for service during a day; when we use two particular, different random streams ω_1 and ω_2. Notice that the behavior of the system changes dramatically. This, however, is also true in the real system: some days show a high activity, whereas others are not busy at all. Thus, in real life as well as in simulated life, our analysis

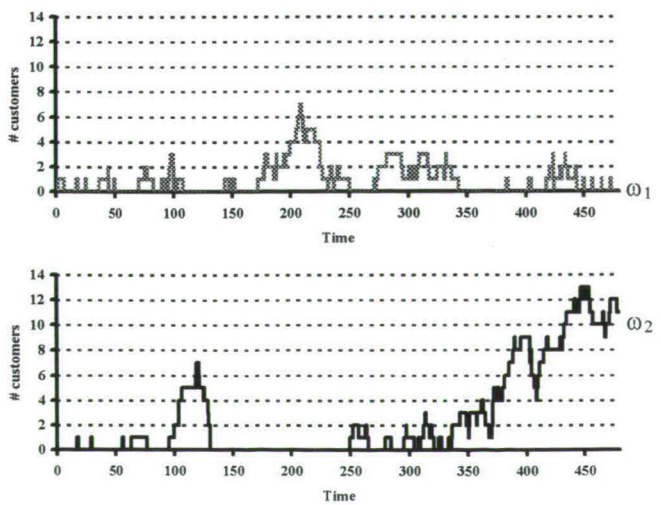


Figure 32. Time series of the number of customers in queue for two different sets of random numbers, ω_1 (upper part) and ω_2 (lower part)

should not rely on a few observations. In Figure 33 we plot the simulated average daily number of customer waiting for service as a function of μ_1, and μ_2 for two random streams. The curves have similar orientations, but their characteristics are completely different.

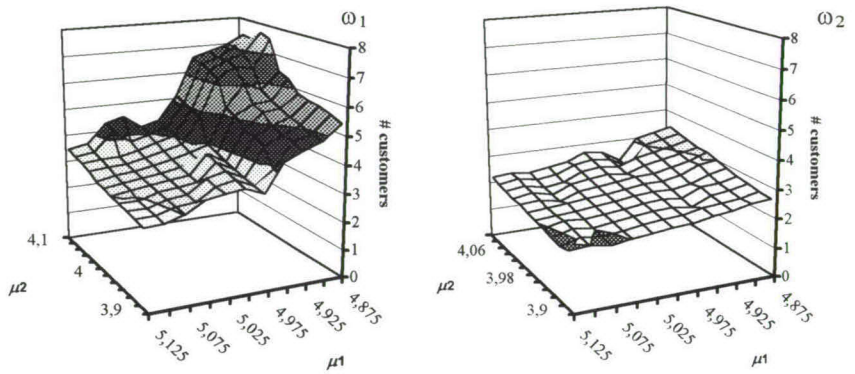


Figure 33. Sensitivity of daily number of customers to μ_1 and μ_2 for two different sets of random numbers, ω_1 (left) and ω_2 (right)

A higher degree of complexity arises if we consider μ_1 and μ_2 evolving over time. We again study a simple example for which the time between customer arrivals (μ_1) has seasonal variations; we keep μ_2 constant over time, namely equal to 4. We model the seasonal variations of μ_1 as a sinusoidal function - see Figure 34 - and we use this function in the simulation model of the bank office. Figure 35 shows the time series of Q (daily number of customers waiting for service). The effect of the dynamic variations of μ_1 is an increase in the variability of Q. Obviously, the amplitude of this variation depends on the amplitude of the sinusoidal function for μ_1. However, such variations in Q may reach levels

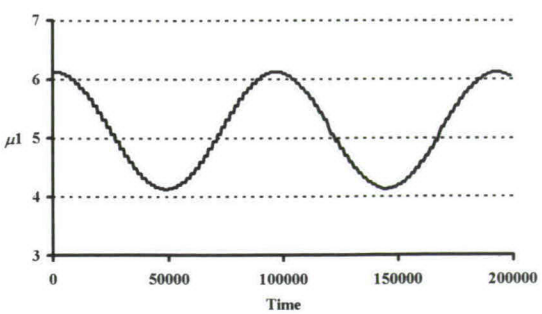


Figure 34. Seasonal variations in the mean time between customer arrivals (μ_1)

that are not acceptable for the bank manager: to avoid this kind of unacceptable performance, corrective actions, such as adding an office counter, may be needed.

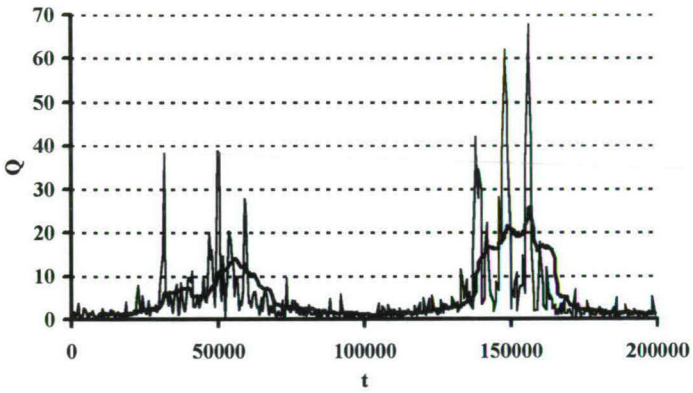


Figure 35. Seasonal variations (in μ_1) and resulting daily number of customers waiting for service: individual values and moving average (bold curve)

Thus, uncertainties should not be neglected when designing a system through simulation. Unfortunately, designing a system under uncertainty is a complex problem. The literature often focuses on a single aspect of environmental uncertainty. Therefore a collection of techniques for dealing with specific aspects of the design problem have been developed in fields such as simulation, decision making, engineering, and control theory. To our knowledge, in production control these techniques have never been integrated within a common framework. Yet we see potential synergies in combining them. Indeed we shall see that they share common concepts, but implement them in different ways.

Traditionally stochastic uncertainties have been tackled using statistical techniques for building confidence intervals. We give a short overview of these techniques in section 5.3. Subjective uncertainties can be handled through Uncertainty/Risk Analysis (URA), which includes risk assessment and risk management; in section 5.4 we describe URA techniques and give a short overview of their applications in the literature. Dynamic uncertainties are the main concern of Taguchi's approach to robust design; in section 5.5 we review the literature on Taguchi's robust design. We consider that URA and Taguchi's approach should be used during the design phase; dynamic control may be used for dealing with subjective and dynamic uncertainties during the operational phase. In section 5.6 we review applications of dynamic control in Kanban systems. Our conclusion is that dynamic control would be too complex to implement in Customized pull systems. In Table 18 we

recapitulate the various sources of uncertainty involved in design through simulation and the techniques proposed in the literature for dealing with these sources.

Table 18. Sources of uncertainty and solutions proposed in the literature

	Stochastic uncertainty	Subjective uncertainty	Dynamic uncertainty
Cause	Random number generators in simulation	Estimates of simulation parameters	Dynamic behavior of real production environment
Solutions in literature	Confidence intervals	Uncertainty/Risk analysis	Taguchi's robust design
		Dynamic control	

5.3 Confidence intervals

Stochastic uncertainty is modeled through the set(s) of random numbers ω used in the simulation. The approach for dealing with stochastic uncertainty consists in building confidence intervals for the simulation output: replicate – that is, run simulations with different set(s) of random numbers $\omega_1, \ldots, \omega_n$ – and look at the resulting variability in the output $X_1, \ldots, X_n$. In the following we present two techniques for building confidence intervals, namely standard statistical techniques and bootstrapping. When simulation is used to compare design alternatives or perform optimization, a complementary approach for dealing with stochastic uncertainty is to make sure that all simulations use the same common random numbers and initial conditions.

5.3.1 Standard techniques

Let $X_1, \ldots, X_n$ be a sample of independent identically distributed (IID) random variables with a common probability distribution fixed by its parameters. The purpose of confidence intervals is to quantify the accuracy of the (sample) average (say) $\overline{X}(n)$ as an estimate for the (population) mean or expected value (say) μ by measuring the *standard error* given a certain confidence level $1 - \alpha$; we denote the standard error by $s_{err, 1-\alpha}$. Standard statistics provides formulas for the standard error: the $100(1 - \alpha)$ percent confidence interval for μ is given by $\overline{X}(n) \pm s_{err}$. This formula means that only 100α sample means out of 100, for samples of size n, will be outside the range of this confidence interval. Standard statistical techniques for building confidence intervals in simulation are thoroughly discussed in Law and Kelton (1991).

General case: IID random variables X_i

If n is sufficiently large, then the random variable Z_n defined as $Z_n = [\overline{X}(n) - \mu]/\sqrt{s^2/n}$ follows a normal distribution with mean 0.0 and variance 1.0 (central limit theorem); s^2 is the sample variance defined as $\sum_i (X_i - \overline{X}(n))^2/(n - 1)$. Thus, for n sufficiently large, an estimate of the standard error for μ given a $(1 - \alpha)$ confidence level is $z_{1-\alpha/2}\sqrt{s^2/n}$, where $z_{1-\alpha/2}$ is the $\alpha/2$ point of the standard normal distribution. This formula, however, is only an *approximation*: depending on the shape of the probability distribution for X_i, the value of n should be more or less "large". For instance, the less symmetric the shape of the probability distribution, the larger n should be.

Special case: IID, normally distributed random variables X_i

If the random variables X_i are IID and follow a normal distribution with mean μ and variance σ^2, then the random variable t_n defined as $t_n = [\overline{X}(n) - \mu]/\sqrt{s^2/n}$ follows a Student t probability distribution with $n - 1$ degrees of freedom. For n larger than 2, an *exact* estimate of the standard error for μ given a $(1 - \alpha)$ confidence level is $t_{n-1,\,1-\alpha/2}\sqrt{s^2/n}$, where $t_{n-1,\,1-\alpha/2}$ is obtained from a table of t-values and is the upper $\alpha/2$ point of the Student-t distribution with $n - 1$ degrees of freedom.

5.3.2 Bootstrapping

The standard techniques discussed in previous section give formulas for confidence intervals for the mean μ. In practice, however, we may be interested in estimating performance criteria (say Y) such as a quantile – the population quantile (say) ξ_p of order p satisfies $P(X \leq \xi_p) = p$. For such criteria, standard statistics do not provide any simple formula for the standard error – such formulas do not even exist in many cases. A resampling technique called bootstrapping provides an easy and cheap way (in terms of simulation) to compute standard errors for any performance criterion. The only assumption is that $X_1, \ldots, X_n$ are IID random variables. The principle of bootstrapping is as follows:

- Draw randomly with replacement a number (say b) of 'bootstrap samples' of size n from the original data set $\{x_1, \ldots, x_n\}$. (Technically, this implies that the values $x_1, \ldots, x_n$ receive a multinomially distributed weight w_i with values 0, 1, ..., n such that $\Sigma w_i = n$).
- Compute the performance criterion Y_j for each 'bootstrap sample' ($j = 1, \ldots, b$).
- Estimate the variability of the original Y by the observed variability in the b bootstrap Y_j.

Much theoretical work undertaken by statisticians has shown the validity of the bootstrap technique for building confidence intervals. Experiments demonstrate that in the case of confidence intervals for the mean (μ), bootstrapping provides an estimate of the standard error that is close to the value obtained from the standard formulas. For other criteria,

bootstrapping yields an accuracy estimate that has excellent theoretical properties; see Efron and Tibishrani (1993) for details about bootstrapping.

5.4 Uncertainty and Risk Analysis (URA)

5.4.1 Uncertainty and risk

In the decision theory literature the definitions of risk and uncertainty are rather unsettled: there does not seem to be consensus among researchers. In Knight (1971), for instance, risk refers to probability functions with unknown parameters and functional forms that can be estimated from historical data (objective probability), whereas in uncertainty, one has to rely on subjective probability. More recently Norman and Shimer (1994) use similar definitions: "A risk decision is defined as a stochastic optimization problem where the parameters and the functional forms required to determine the optimal decision are known. And an uncertain decision is defined as a stochastic optimization problem where at least one parameter or functional form must be estimated". Bayesians, however, do not make distinctions between objective and subjective probabilities: they consider all probabilities to be subjective (see Cyert and De Groot, 1987, p.13).

According to Morgan and Henrion (1990, p. 1), risk involves an 'exposure to a chance of injury or loss'. Such an exposure indeed occurs in our case. Designing a system for a specific environment does not guarantee good performance for other environments: there is a risk associated with the design chosen; another design may lead to a lower risk. Now, as discussed in the previous sections, simulated environments involve different sources of uncertainty. Therefore we may associate a risk estimate with each set of decision variable values (design). In this dissertation, we see risk as a consequence of uncertainties, whatever their source. We define the risk associated with a design as the probability of poor performance. This definition is close to the one used in Bayesian decision theory: the risk of a decision is the expected loss (see Cyert and De Groot, 1987, p.10).

URA is a general framework that consists of several components. Balson *et al.* (1992) see the following two main components:

(i) *Risk assessment* is the qualitative or quantitative evaluation of risks.

(ii) *Risk management* is the process of determining whether an identified risk is acceptable and what action (if any) should be taken to mitigate or control a risk that has been identified or assessed. It includes the following steps: determine acceptable risk, define management alternatives, evaluate alternatives, select and implement alternatives.

A possible third component of URA is *risk communication*. It raises such issues as: what levels of risk are present, what is the importance of the risks, how are risks to be managed or controlled?

Bodily (1992) gives a typical example of risk analysis in finance: "(...) in evaluating a capital project, a company carries out a risk analysis to determine its financial risk in making the investment. The approach might incorporate a cash flow model, and the risk analysis might involve a Monte Carlo simulation of the uncertainty in net present value or other financial performance measures. Risk management in such a context would relate to reducing risk of the project, if it is not acceptable, either by diversifying, risk sharing, or contingency planing to protect against unwanted scenarios."

5.4.2 Procedure

Quantifying risk through Monte Carlo[1] requires determining the output probability distribution. For this purpose risk assessment proceeds as follows (see Figure 36): sample each unknown parameter from a statistical distribution function, combine the sampled parameter values into scenarios, and conduct a simulation experiment for each scenario. The outcome of this procedure is an estimated probability distribution of the performance measures. Among the many sampling techniques, crude Monte Carlo sampling is probably best known. Basically, the principle is to select values at random from the distribution per input. Other techniques try to yield better samples than Monte Carlo sampling. Latin Hypercube sampling or LHS (Iman and Shortencarier 1984), for instance, is stratified sampling that divides the range of each input parameter into non-overlapping intervals of equal probability; from each interval, one value is selected at random according to the probability distribution in that interval. A refined technique is Median Latin Hypercube sampling, which selects systematically the middle value of the intervals; thus, the sample of each input parameter depends only on the sample size. "Hammersley points" are designed using a procedure based on "low discrepancy" pseudo-random numbers; for details, we refer to Hammersley (1960) and Kalagnanam and Diwekar (1997). We use LHS only.

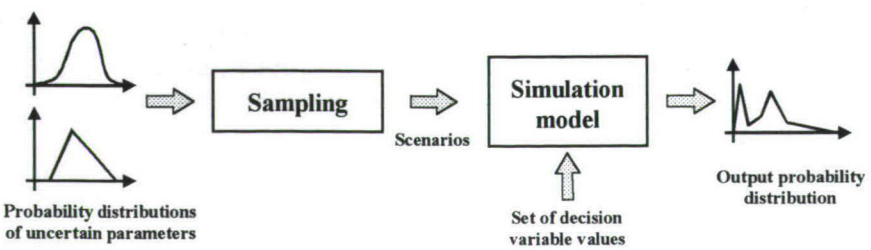


Figure 36. Risk assessment through simulation

[1] According to Kleijnen and Van Groenendaal (1992, pp. 12): "We speak of a *Monte Carlo* method whenever the solution makes use of random-numbers, which are uniformly and independently distributed over the interval from zero to one. We speak of *simulation* whenever the model has a time dimension (the model is called dynamic) and it is solved numerically."

URA is popular in finance and investment problems, and compulsory for the design of potentially dangerous systems, such as nuclear and waste isolation plants (Helton *et al.* 1997), and certain industrial activities, such as chemical industry (Palle 1994); also see Brehmer *et al.* (1994). However, to our knowledge, it has never been used to design systems based on stochastic discrete-event models (such as production-control systems): actually, most of the models used in the URA literature are deterministic.

Risk management does not seek for an optimal solution but for a solution that yields an acceptable risk. Bayesians, however, do search for the decision that minimizes a single criterion: the expected loss. This expected loss is the negative of the utility function, which is the expression of preferences of the decision-maker with respect to perceived risk and expected return. This loss is used to compare output probability distributions, and to choose among them. Preferences, however, are individual perceptions: a preference may not be unanimous. First- and second-order stochastic dominance tests identify probability distributions that are unanimously preferred by all decision-makers with monotone utility functions and monotone, strictly concave utility functions respectively; see Wolfstetter (1996). We shall see applications of dominance tests in the next chapter.

5.5 Taguchi's robust designs

5.5.1 Parameter design problem: concepts

Taguchi's *robust design* consists in searching for a product design that guarantees low variations in the performance level when the environment changes. Traditional design aims at a product that is optimal for a single specific environment (noise configuration). Thus Taguchi emphasizes that the effect of environmental variations depends on the decision variable values. Instead of spending time and money trying to control the sources of environmental variability (fire fighting), robust design focuses on finding a set of decision variable values that yields good average performance and low sensitivity to environmental variations. The product design that achieves this objective is said to be "robust". This approach, also known as parameter design, has been stated and popularized by Taguchi (Taguchi and Phadke 1984; Taguchi 1986).

5.5.2 Procedure

Besides the concept of robust design, Taguchi also proposed technical solutions for designing robust products and systems. These techniques focus on (i) estimating the sensitivity to environmental variations for a given set of decision variable values, and (ii) selecting a good set of values. Mayer and Benjamin (1992) proposes the following six-step procedure for robust design that integrates Taguchi's techniques:

(i) Identify factors and specify targets

Distinguish between (a) design factors, say D_i with $1 \le i \le k$, which are the parameters with values (presumably) within the control of the designer, and (b) noise factors, say N_i with $1 \le i \le l$, which are not within the control of the designer. Define performance measure(s) and possible target values. Taguchi proposes robustness measures called *signal to noise* (S/N) *ratios* that aggregate information on the average performance and its variability (location and dispersion); see step iii.

(ii) Formulate the design of experiment (DOE): crossed arrays

Design factors are varied according to an orthogonal array (Taguchi 1959), called *inner array*. For each combination in this array, noise factors are systematically varied according to another orthogonal array called *outer array*. Thus, if there are m and n factor combinations in the design and noise arrays respectively (with $m \ge k$ and $n \ge l$), then m x n combinations have to be examined: see Table 19. In the following, this DOE for robustness study is called a crossed array.

Table 19. Crossed Arrays for Robust Design

				Outer array: 1	j	n		
				-	...	-	N_l	
				...	...	...	...	
		D_1	D_k	-	...	+	N_1	
Inner array	1	-	-		...		S/N_1	S/N ratios
	i	...	...	...	y_{ij}	...	S/N_i	
	m	+	-		...		S/N_m	

(iii) Execute the runs and compute the performance statistics

Execute the m x n runs. Then, for each combination of design factors compute the S/N ratios, which measure the effect of systematic noise variations on the performance of the product. The following three types of S/N ratios are standard.

- The smaller Y, the better: $S/N_i = -10 \log(1/n \sum y_{ij}^2)$.
- The larger Y, the better: $S/N_i = -10 \log(1/n \sum 1/y_{ij}^2)$.
- And, the closer Y to target, the better: $S/N_i = 10 \log(\bar{y}^2 / s^2)$.

(iv) Find parameter settings that maximize S/N

Perform an analysis of variance (ANOVA) with the S/N ratios as response. Identify design factors with a significant effect on S/N. Then, set these factors at levels that maximize S/N.

(v) Tune performance to target

Perform a second ANOVA using the performance measure(s) averaged over the n noise combinations, as response. Identify design factors with significant effects on performance measure(s), among the factors that have non-significant effects on S/N (identified in step iv). Adjust these factors to improve performance.

(vi) Perform confirmation runs

Does the model perform as predicted? If not, assumptions are not valid (for instance, ignoring factor interactions may be wrong). Go back to ii.

5.5.3 Critique and alternative tactical choices

Taguchi's contribution to robust design is undeniable. However, his choices for robust design implementation are not unanimously accepted. For instance, Nair (1992) reports on a thorough panel discussion that criticized the use of S/N ratios and crossed arrays. Yet, there seems to be consensus about the fundaments of robust design: conducting experiments in order to study the effects of controllable factors on both the location and the dispersion of the response. Thus, Pignatiello and Ramberg (1987) propose to distinguish between the strategic aspect (namely, Taguchi's philosophy of robustness) and the tactical issues (for instance, S/N ratios and DOE).

Many tactical alternatives can be found in the literature. Table 20 shows that researchers sometimes prefer using loss functions or studying the location and dispersion of the performance separately (instead of S/N ratios). Moreover, crossed designs (such as shown in Table 19) may be replaced by combined designs, that is, a single array that does not treat noise factors separately from design factors.

Taguchi originally proposed his technique for product design. Later, a few researchers have also applied robust design to simulated systems. For instance, Wild and Pignatiello (1991), Dooley and Mahmoodi (1992), Benjamin *et al.* (1995), and Sanchez *et al.* (1996) propose simulation-based methodologies for the design of robust jobshop manufacturing systems. Simulation allows the use of larger samples than crossed and combined arrays. We now discuss two recent examples of alternative experimental techniques.

Moeeni *et al.* (1997) proposes an original approach – based on simulation and *Frequency Domain Experiments* (FDE) – to design robust Kanban systems. FDE consists in

generating levels x_{ij} for each noise factor x_i and for each noise configuration $j = 1 \ldots m$, according to a sinusoidal function:

$$x_{ij} = \tfrac{1}{2}(u_i + l_i) + \tfrac{1}{2}(u_i - l_i)\cos(2\pi.\omega_i.j), \tag{7}$$

for $i = 1,\ldots,n$ and where u_i and l_i are the upper and lower bound of factor x_i respectively, ω_i is the oscillation frequency of x_i, that is $\omega_i = T_i/m$ where T_i is called the driving integer for x_i. Jacobson *et al.* (1991) proposes an algorithm for determining the driving integers so that main effects, quadratic effects, and two-factor interactions are not confounded. Moeeni *et al.* (1997) uses FDE to measure the robustness of a Kanban system design.

FDE has originally been designed for sensitivity analysis (Schruben and Cogliano 1981): the effect of each input factor is assumed to be measured by the contribution of its characteristic frequency to the output. This contribution is determined through discrete

Table 20. Literature on tactical issues for robustness studies

Reference	# Design/ Noise factors	Robustness measures	Design of Experiments
Sanchez *et al.* (1996)	5 / 2	quadratic loss function	Comparison: - combined: 2^{7-2} + two center points - crossed: (2^{5-1} + center points) x 2^2
Mayer and Benjamin (1992)	4 / 2	close-to-target S/N	Crossed: 2^{4-1} x 2^2
Lim *et al.* (1996)	4 / 6	smaller-the-better S/N for flowtime; larger-the-better for throughput	Crossed: L_{27} x L_8 L_{27} : 3^{13-10} L_8 : 2^7
Dooley and Mamoodi (1992)	2 / 4	signal-to-noise ratios of the performance mean and dispersion.	2^2 x 2^{4-1}
Sanchez *et al.* (1993)	4 / 1	$\overline{Y}(x)$, log(S(x))	2^{4-1} x 2^1, replicated four times
Moeeni *et al.* (1997)	7 / 34	quadratic loss function	2^{7-1} & noise factors oscillations, replicated four times (frequency domain)

Fourier analysis. Research prior to Schruben and Cogliano (1981) also used sinusoidal functions to examine sensitivity to inputs; the approach is known as the Fourier Amplitude Sensitivity Test (Cukier *et al.* 1973), and is used for uncertainty analyses (Morgan and Henrion 1990, p209). In the latter approach factor values change in a similar way as in FDE:

$$x_{ij} = \mathrm{E}[x_i] + v_i \sin(\omega_i . s_j), \tag{8}$$

where v_i is the half-range of the variations (x_{ij} varies within $[\mathrm{E}[x_i] - v_i; \mathrm{E}[x_i] + v_i]$), $\{\omega_i\}$ is a set of frequencies so that factors are not correlated, and s_j is a parameter to discretize the sinusoidal function and has equally spaced values. So (7) and (8) are equivalent indeed.

5.5.4 Robust optimization

As Mayer and Benjamin (1992) mentions, Taguchi's procedure focuses on performance improvement and does not search for an optimum, namely, the most robust design. Indeed, in practice the search is often limited to the local area defined by the inner array (for design factors). Thus, they suggest that optimization techniques such as RSM (see section 3.6.3) could be coupled with Taguchi's procedure. Indeed, RSM moves from one small area to another based on estimated steepest path. Wild and Pignatiello (1991) also mention this possibility of using RSM. An application can be found in Benjamin *et al.* (1995), who use RSM for the optimization of two criteria, namely, the performance characteristic and the sensitivity of the performance characteristic to environmental variations. We shall use the concept of robust optimization in section 5.8.

5.6 Dynamic control

5.6.1 Two issues: when to act and what to do?

Robust design may not be sufficient to keep the effects of environmental variability within an acceptable region. Actually, additional actions may be needed during the *operational* phase. The purpose of dynamic control is to modify the value of decision variables on-line, in order to maintain an acceptable level of performance. In pull systems control actions consist in adjusting the card numbers. Two main issues arise when trying to implement a dynamic control procedure, namely, when to act and what to do? Possible control actions may be considered (i) on a real-time basis (i.e., whenever an event happens), (ii) at fixed time intervals, or (iii) only when the system is out of control. Real-time control actions require some prior knowledge about what values should be assigned to the decision variables as a function of the system state: action has to be taken immediately. Such prior knowledge, however, may not be easy to obtain and can only be partial since the space of possible system states is rather large. Taking control actions at fixed intervals requires techniques for finding the best control action to be taken. Acting upon an out-of-control

state requires additional techniques for monitoring the system and detecting such states. Next, we review four applications of dynamic control in Kanban systems, and we detail the techniques used to deal with the issues of when to act and what to do.

5.6.2 Dynamic control of pull systems

After an extensive literature search we found four publications dealing with dynamic control of Kanban systems; we detail these four publications in this section. Also, an approach inspired from SPC (Statistical Process Control) for the dynamic control of Conwip systems is discussed in Hopp and Roof (1998).

• Takahashi and Nakamura (1997)

Prerequisite. Graph of best card numbers at each stage as a function of the finished product demand rate. Their technique for building this graph consists in optimizing the card numbers with the goal of achieving a required level of mean waiting time with minimum inventory. They repeat this optimization for several values of the mean demand interarrival time. The outcome of the procedure is a response surface with the optimal card numbers as functions of mean demand interarrival time.

Monitoring demand. They use exponential smoothing to filter the time series data of product demand. Exponential smoothing gives higher weight to recent data through an exponentially weighted moving average (EWMA):

$H_i = \alpha x_i + (1 - \alpha) H_{i-1}$ where H_i is the i-th EWMA, x_i is the i-th real data, and α is the exponential smoothing constant ($0 \leq \alpha \leq 1$).

Detecting out-of-control cases. Out-of-control cases are characterized by an upper and lower control limit – denoted by *UCL* and *LCL* respectively – for the mean interarrival time μ:

$UCL = \mu + \delta\sqrt{,\alpha/(2 - \alpha)}\sigma_x,$

$LCL = \mu - \delta\sqrt{,\alpha/(2 - \alpha)}\sigma_x,$

δ is usually assumed to be 3; for α, Takahashi and Nakamura consider values of 0.1, 0.2, 0.3, 0.4, and 0.5; σ_x denotes the variance of the interarrival time.

Controlling the card numbers. The card numbers must be changed when an unstable change in product demand is detected (see 'monitoring demand'), i.e., if EWMA moves below *LCL* or above *UCL*. The new card numbers are selected using the graph (see 'prerequisite') so that the required level of mean waiting time of product demand is assumed to be achieved with minimal mean WIP.

• Chang and Yih (1998)

Prerequisite. Fuzzy rule-based system that gives the desired number of kanbans for a given set of system attribute values (demand, number of lots waiting for kanbans, average

utilization, processing time, number of kanbans currently in the system). The fuzzy system is trained on a set of examples.

Control procedure. At each job arrival, the fuzzy system is used to re-estimate the desired number of kanbans, given the current system state. Thus, the number of kanbans is controlled on a real-time basis.

• Liberatore *et al.* (1996)

Monitoring system throughput. System throughput is monitored and averaged over the m most recent observations.

Searching for best control action. They consider adding or removing one kanban. The throughput of the two corresponding systems is estimated using perturbation analysis (Ho and Cao, 1991).

Controlling the card numbers. For each possible control action, the estimated throughput of the corresponding system is averaged over the m most recent observations. The control action that yields the best performance is implemented. This control procedure is performed each time a given number of parts is delivered. Thus control intervals have a variable duration.

• Rees *et al.* (1987)

Rees *et al.*'s approach is based on the following well-known equation for determining the number of kanbans (see, e.g., Monden, 1993):

$$n = [DL(1 + \alpha)] \tag{9}$$

where D is the average demand expressed in containers, L is the average lead-time for the product, α is a safety factor for protection against stochastic variations and anomalies (such as machine breakdowns), and $[x]$ is the smaller integer greater than or equal to x. Their idea is to compute the card numbers using 'real-time' updated information about the lead-times, and periodically updated forecasts for the demand level.

Measuring leadtime characteristics. Rees *et al.* consider two measuring periods needed to (1) estimate the autocorrelation function of leadtimes, and (2) estimate the leadtime density function based on independent observations.

Forecasting demand. To estimate the demand for the next period, they use standard forecasting procedures.

Determining the number of kanbans. The density function of the card number is derived from (9). A card number is chosen so that the sum of holding costs and shortage costs is assumed to be minimized. Rees *et al.* emphasize that if shortage costs overwhelm holding costs, then L can be replaced by L_{max}, the maximal lead-time, or by L_q, the qth percentile of the lead-time density function with a q% service target.

Taking action. Set the number of kanbans to the value determined in the previous step. Before repeating the whole control procedure, ensure that the system had sufficient time to settle down.

The concept of dynamic control is seducing. The review above, however, shows that its implementation is rather difficult even for Kanban, which is among the simplest pull system. Thus, dynamic control of more complex systems such as Hybrid or some of the results obtained in section 0 seems hardly feasible. We focus our research on robust design and URA.

5.7 Robust design and URA

Obviously, Taguchi's approach to robust design and URA deal with different problems: variability in the production environment versus uncertainty in the model inputs. Yet, the main differences are at the implementation level, not at the conceptual level. Next, we review these implementation differences, and we try to determine how robust design and URA may be combined.

5.7.1 Physical versus Simulation Experiments

Mayer and Benjamin (1992) points out the main differences between product design and system design. In product design, robustness is achieved through prototypes and physical experiments; the designed products are intended for production in large quantities. In system design, a robustness study has to be performed on a model using simulation experiments. Moreover, only a single system is to be implemented. Thus, product design can use only small experiments, for feasibility reasons – it may not be easy to reproduce a specific environment – and cost reasons – prototypes usually are expensive. System design, however, is limited only by time constraints (any type of environment can be simulated, and the major experimental cost is computer and analysts' time).

Originally, robust design addressed product design problems: it is a method designed for physical experimentation. Now, physical experimentation is rarely possible for systems. URA, however, is designed for simulated experiments. Thus, the second approach seems better suited for system parameter design.

5.7.2 Sampling

Most robustness studies consider at most three levels for each environmental parameter. Next, mathematical techniques are used to choose the combinations of parameter values to be experimented. Risk analysis, on the other hand, uses a large sample size in which each parameter has many different values. For instance, Latin Hypercube Sampling (LHS) requires a sample size of 100 at least. Moreover, Taguchi's designs select extreme parameter value combinations, whereas risk analysis samples values for each parameter

over the whole domain. The parameter probability distribution functions are specified by the analysts, possibly with the support of experts.

5.7.3 Dispersion versus Risk

Robust design is based on the estimation of the performance location and its dispersion over environmental variations. Any deviation from the mean is penalized (quadratic loss function): dispersion does not distinguish between good and bad performance. Decision makers, however, may be interested only in avoiding "bad" performance, that is, performance below a prespecified target. Furthermore, the use of signal-to-noise ratios does not give much flexibility to the decision makers: the information about location and dispersion of the performance is aggregated through a single criterion that is not particularly easy to interpret (Nair, 1992). Quantifying a probability of poor performance seems more appropriate for decision making.

5.7.4 Combining robust design and URA

As mentioned in section 5.2, dynamic uncertainties are closely related to subjective uncertainty. Therefore it is not surprising that recently studies solve the robust design problem through URA techniques: the optimization concern of robust design is preserved, but URA techniques are used.

Kalagnanam and Diwelar (1997) suggest an optimization procedure for the design of robust systems based on simulation and Monte Carlo methods, which they apply to the design of a chemical tank reactor. They state the design problem as a stochastic optimization problem in the sense that they have uncertain input variables; they do not consider stochastic uncertainty – they use a deterministic mathematical model for performance evaluation – nor dynamic uncertainty. An interesting aspect of this paper is that the robustness of the system is not just studied for a few extreme environments, but for a large sample of environments. The focus of their research is on a new sampling technique and its comparison with four other techniques, namely, Monte Carlo, Latin Hypercube, Median Latin Hypercube, and Hammersley points.

Next we propose a more complete procedure for designing systems under uncertainty. Our objective is to integrate the three types of uncertainty presented in section 5.2, namely, stochastic, subjective, and dynamic uncertainties.

5.8 Procedure for designing systems under uncertainty

We use the same framework as Kalagnanam and Diwelar (1997): an optimization procedure calls URA techniques that estimate performance under uncertainties. The main difference is that we use discrete-event simulation for performance evaluation, which adds the issue of stochastic uncertainty to the problem treated in Kalagnanam and Diwelar (1997). We deal with this added uncertainty through several means (see Figure 37). First,

we consider the (pseudo)random number generator seed as an environmental factor: we change it for each environmental scenario. The difference with the other factors (processing time, time between failure, etc.) is that there is no order for random number generators: LHS cannot be used to define a sample. Thus we let the simulation software select different seeds for each replication (simulated scenario). We also use the same initial conditions for all scenarios. Second, we use common random numbers for system evaluation through LHS, which means that all simulated systems are submitted to the same stochastic uncertainty. Third, we may use bootstrapping to build confidence intervals on probabilistic performance measures. Indeed, traditional statistical techniques for building confidence intervals do not apply to performance measures such as the probability of poor performance.

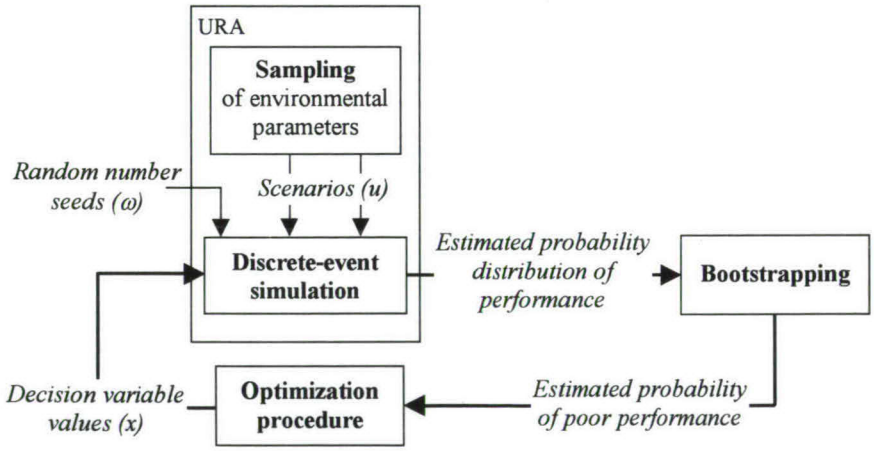


Figure 37. Our procedure for designing systems under uncertainty

We state the optimization problem as follows:

$$\begin{aligned} &\text{Min } F(x, u, \omega), \\ &\text{s.t. } G_1(x, u, \omega) = 0, \\ &\quad G_2(x, u, \omega) \geq 0, \end{aligned} \tag{10}$$

where x is the set of decision variables, u is the set of uncertain parameters, and ω is the random seed. F is the main criterion, G_1 and G_2 are constraints. As demonstrated throughout this chapter, the difficulty of this optimization problem is to estimate F, G_1, and G_2 for a given x, accounting for the effects of u and ω. The solution we propose is to estimate the probability distribution of F, G_1, and G_2 for a given x, and characterize these distributions through appropriately chosen metrics (say) f, g_1, and g_2 respectively. These metrics are then functions of x only.

5.9 Conclusion

The purpose of simulation in our context is to estimate performance measures and to find the optimal values of the decisions parameters that optimize those measures. The main difficulty, however, is that simulation outputs depend not only on decision parameters but also on the way uncertainty is handled. We identified three sources of uncertainty in simulation, namely, stochastic uncertainty (due to random numbers), subjective uncertainty (due to input data estimation), and dynamic uncertainty (due to variations in the real production environment). We proposed a novel approach that integrates URA and robust design (Taguchi), to tackle uncertainty and obtain performance measures that depend only on decision parameters.

In the next chapter we apply our procedure to pull production control systems. The idea is to design pull systems under uncertainty. Thus, we shall detail the procedure described above.

Chapter 6
Robust Customization of Pull Systems

Abstract

The objective of this chapter is to design pull systems, given the three sources of uncertainty identified in Chapter 5. For this purpose we state the robust design problem and specify robustness criteria in terms of service and WIP. We detail the procedure for estimating these robustness criteria, and we compare their values for four pull systems. We also investigate the robustness effects of managerial decisions. These decisions include the card numbers, the type of probability distributions used in LHS, and various parameters that specify the managers' attitude towards risk and characterize their preference. Using these results, we apply the robust design procedure to the production system studied in Bonvik et al. (1997) controlled through Conwip. Customization under uncertainty, however, would require a computational cost that is currently not affordable.

6.1 Introduction

Under the three sources of uncertainty identified in Chapter 5, the optimization problem (1) (page 14) stated in the same terms as (10) (page 80) becomes:

$$\begin{aligned} &\min_{x} WIP(x, u, \omega), \\ &\text{s.t. } S(x, u, \omega) \geq \tau, \end{aligned} \tag{11}$$

where x is the set of card numbers in the generic pull system, u is the set of uncertain parameters, namely, the average processing times, the variability of process times, the average times between failures, the average times to repair, and the demand rate, ω is the random seed for stochastic simulation and τ is the service target. For a four-stage production line we have ten card numbers (see section 3.6) and 17 uncertain parameters.

6.2 Robustness criteria and notation

We use the procedure defined in section 5.8 to solve this optimization problem. So we generate a sample of $n = 100$ scenarios for u and ω. Each scenario is simulated over a period of one month – 22 days, two shifts (900 minutes) per day (thus, 44 shifts per month, that is, 19800 minutes), plus a warming-up period of three days (2700 minutes); thus the total simulated time is 19800 + 2700 = 22500 minutes. From this simulation we estimate the average WIP level and the probability of the service level per shift dropping below target. We speak of a *disaster* when a shift has a service level below target. Our choice of disaster probability as a performance criterion is motivated by the fact that service has to remain at a high level in a *short-term* horizon. Indeed, managers are under pressure when one or more shifts during a month yield poor customer service, that is, when the disaster probability is unacceptably high. For WIP, however, the concern is not short-term performance. Indeed, a temporary increase in WIP is acceptable, as long as WIP's *long-term* performance remains satisfactory.

We use the following notation. Upper case letters denote random variables, lower case letters denote realizations of random variables and deterministic variables, and Greek letters represent parameters to be estimated.

$\mu = E(WIP)$: expected average WIP per month;

Y: service level per shift ($0 \leq y \leq 1$; percentage of demand per shift satisfied from stock);

$\pi = P(Y < c_y)$: probability of Y dropping below a prespecified manager's target (say) c_y.

The performance measures μ and π are estimated through

$$\hat{\mu} = \int_{2700}^{22500} WIP_t \, dt / 19800,$$

$$\hat{\pi} = \sum_{i=1}^{44} I(y_i < c_y)/44,$$

where WIP_t is the WIP level at simulated (continuous) time t, y_i is the service level realized in shift i, and $I(a)$ is an indicator function, that is, $I(a)$ is equal to one if statement a is true, zero otherwise (see Figure 38).

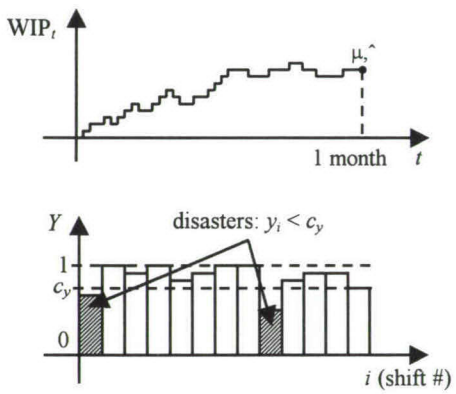


Figure 38. Performance criteria for robust optimization

Since we repeat this simulation for different environmental scenarios (say) S, our symbols need a subscript:

$\mu_S = E(WIP|\ S = s)$;

$Y_S = (Y|\ S = s)$;

$\pi_S = P(Y < c_y|\ S = s) = P(Y_S < c_y)$.

The performance measures μ_S and π_S are realizations of random variables (say) M and Π respectively, and repeating the simulation over all possible scenarios yields their joint probability distribution. Our objective is to perform robust optimization; thus we search for the pull system with the best performance distribution. Stochastic dominance theory permits comparison of probability distributions (see section 6.3.2 for an illustration). Its integration within an optimization algorithm, however, is not simple. Moreover, stochastic dominance helps only when unanimous decision can be made (see Appendix 3). Therefore, we decide to characterize the estimated joint performance distribution through the following two robustness measures:

$\eta = E_S(M_S) = E_S[E(WIP|\ S = s)]$: average monthly WIP averaged over all scenarios;

$\rho = P(\Pi_S \geq c_\pi)$: probability of Π_S exceeding another managerial threshold (say) c_π, under various scenarios. This probability expresses the risk of a high disaster probability over the scenarios. Management should decide which risk ρ they are willing to accept in terms of service, at which price η in terms of WIP level. We denote this acceptable risk by c_ρ.

We use URA to estimate the two robustness measures for a given input distribution of scenarios. The resulting estimators are:

$\bar{\hat{\mu}} = \sum_{s=1}^{100} \hat{\mu}_S/100$, the average monthly WIP averaged over the 100 sampled scenarios,

$\hat{\rho} = \sum_{s=1}^{100} I(\hat{\pi}_S \geq c_\pi)/100$, the fraction of $\hat{\pi}_S$ that exceeds c_π among the 100 realizations.

Our robust optimization problem is meant to guarantee an acceptable risk level ρ in terms of service, while minimizing the price η in terms of WIP level. This problem can be stated as follows:

$$\begin{aligned} &\text{Min } \bar{\hat{\mu}}(x), \\ &\text{s.t. } \hat{\rho}(x) < c_\rho, \end{aligned}$$

which can be further developed as follows:

$$\begin{aligned} &\underset{x}{\text{Min}} \sum_{s=1}^{100} \int_{2700}^{22500} WIP_{t,s}(x)\,dt/19800/100, \\ &\text{s.t. } \hat{\rho}(x) = \sum_{s=1}^{100} I(\sum_{i=1}^{44} I(y_{i,s}(x) < c_y)/44 \geq c_\pi)/100 < c_\rho. \end{aligned} \tag{12}$$

This optimization problem is complex, and involves substantial computing for a given x. Therefore, we expect computing times to be rather long. Designing a pull system, however, is not an every day activity, and the financial risks may be high. Thus, long computing times may still be acceptable. We shall discuss ways of reducing this cost, at the end this chapter.

Problem statement (12) relies on the definition of several managerial parameters, which makes it more understandable than traditional approaches to robustness: c_y is the targeted service level per shift, c_π is maximum number of disasters tolerated per month, and c_ρ characterizes the risk that managers are willing to take. In section 6.3.3 we shall study the effects of these parameters on the performance in terms of service and raise the issue of how to select their values.

Next, we illustrate our approach by comparing the performance under uncertainty of four pull systems, namely, Kanban, Conwip, Hybrid, and our customized system; for all four systems we take the configurations found for the example in Bonvik *et al.* (1997).

6.3 Illustration: robustness of four pull systems

6.3.1 URA of pull systems

The production line in Bonvik *et al.* (1997) was described in detail in section 3.6.1. The uncertain parameters are the various processing time averages and variances, the average times between failures and times to repair, and the demand rate; altogether 17 parameters. Our *base scenario* is the set of values used for these parameters in Bonvik *et al.* (1997). We choose to study a range of ±5% around this base scenario. LHS is used to generate a sample of $n = 100$ environmental scenarios from these ranges. In our academic examples we have

no information on the likelihood of the various scenarios. Therefore we assume that all scenarios are equally likely; that is, we use a uniform prior distribution per parameter, and assume independent parameters. The card numbers found in Bonvik *et al.* (1997) for Kanban, Conwip, and Hybrid (see Table 21), and for our customized system (see section 3.6.3), hold for the base scenario (all uncertain parameters at their mid values). We do not expect them to be optimal when uncertainty is considered. The purpose of this section is to compare the performance of these four pull control systems under uncertainty.

Table 21. Recapitulation of optimal card numbers found by Bonvik *et al.*; shaded cells and remaining card numbers have infinite values

	$k_{4;1}$	$k_{1;1}$	$k_{2;2}$	$k_{3;3}$	$k_{4;4}$
Conwip	15				
Kanban		2	2	4	10
Hybrid	15	2	3	5	15

For each scenario sampled by LHS, we run a simulation corresponding to one month of production (22 days, two shifts per day). All simulation runs use the same initial conditions, but different random seeds and environmental parameters within URA. In Bonvik *et al.* (1997), the objective was to achieve a 99.9% service level target, while minimizing the WIP level; both performance measures were estimated over a long simulation run. We use the average WIP level estimated over a month, μ_S, and the monthly proportion of shifts with a service level below target, π_S. We choose the same target as Bonvik *et al.* (1997), so $c_y = 99.9\%$. URA yields the joint distribution of the two performance measures. The distribution of the disaster probability π_S has more or less a 'bath tub' shape: relatively high probabilities either of low or high realizations; Figure 39 shows a bar chart for our

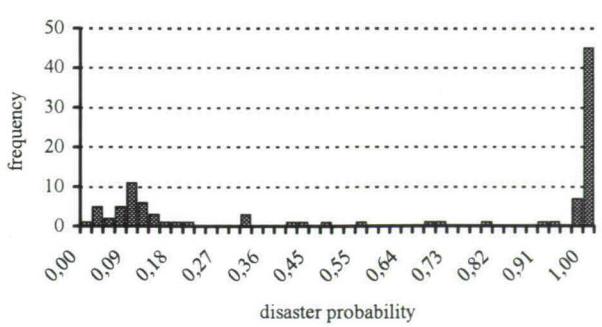


Figure 39. Distribution of the disaster probability π_S for our customized system

customized system.

Cumulative distributions, however, give a clearer picture, and facilitate comparison of systems. Figure 40 displays the cumulative distributions of μ_S for the four pull systems. Figure 41 shows the cumulative distributions of π_S for these same four pull systems.

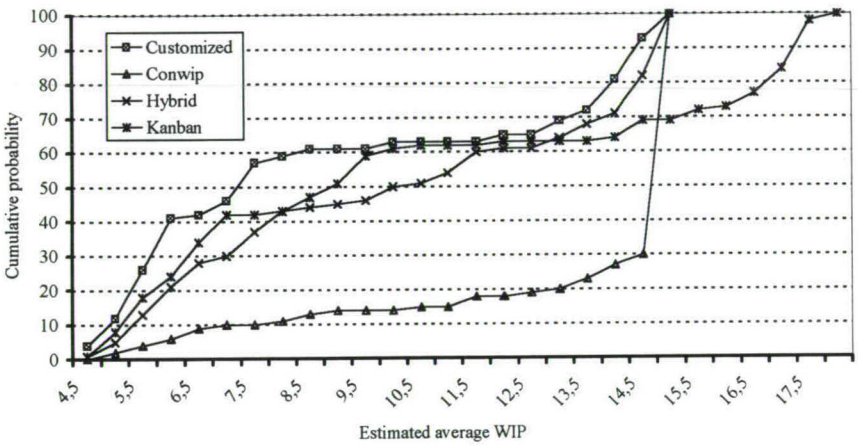


Figure 40. Cumulative distribution of μ_S for the four pull systems

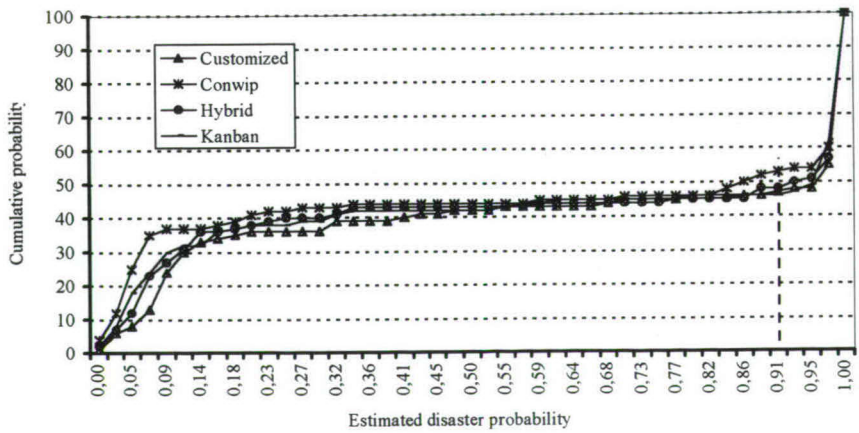


Figure 41. Cumulative distributions of π_S for the four pull systems

6.3.2 Comparison of pull systems in terms of robustness

As we mentioned in section 6.2, there are two ways of comparing performance under various scenarios. The first way is to use stochastic dominance theory to compare the estimated density functions of the estimated disaster probability $\hat{\Pi}$ and average WIP $\hat{M}$. In

Appendix 3 we give a short summary of stochastic dominance theory. We compute dominance tests for the disaster probability. The outcome is the following: Conwip first-order stochastically dominates Hybrid and our Customized system, and second-order stochastically dominates Kanban; Kanban and Hybrid both second-order stochastically dominate our customized system, but it is not possible to choose unanimously between them. For the estimated average WIP, we find that our customized system first-order stochastically dominates all the other systems; Hybrid and Kanban both first-order stochastically dominate Conwip; it is not possible to choose unanimously between Hybrid and Kanban. Thus, Conwip would be unanimously preferred for its disaster probabilities, Π, but at the cost of the worst estimated average WIP, M. Also our Customized system would be unanimously preferred in terms of estimated average WIP, M, but at the cost of the worst disaster probabilities, Π. Therefore, Kanban and Hybrid may be the best compromises, but managers would have to express their preference concerning the risk/cost compromise to make a decision.

The second way of comparing performance under various scenarios is to use the robustness criteria ($\bar{\hat{\mu}}$, $\hat{\rho}$) proposed in section 6.2. In the computation of $\hat{\rho}$ we select $c_y = 0.999$ (same service as in Bonvik *et al.*) and $c_\pi = 0.9$ – so we look at the high disaster probabilities. From the density functions shown in Figure 40 and Figure 41, we derive the values in Table 22. Note that we measure $\hat{\rho}$ in Figure 41 by drawing a vertical line at $\hat{\pi} = c_\pi$ and reading the corresponding cumulative probability. This cumulative probability, however, is $1 - \hat{\rho}$, since $\rho = P(\pi_S \geq c_\pi)$ and the cumulative probability read on the chart is $P(\pi_S < c_\pi)$. The values in Table 22 seem to confirm the results of the comparisons through stochastic dominance theory: Conwip has the best performance in terms of service ($\hat{\rho}$), but the worst in terms of WIP ($\bar{\hat{\mu}}$), whereas our Customized system has the best performance in terms of WIP, but one of the worst in terms of service.

Table 22. Estimated robustness measures for the four pull systems

	Kanban	Conwip	Hybrid	Customized
$\bar{\hat{\mu}}$	10.4	13.3	10.0	8.9
$\hat{\rho}$	0.54	0.47	0.52	0.53

In order to support decision making, bootstrapping (see section 5.3.2) can be used to build confidence intervals around the values in Table 22. We resample the 100 scenario outcomes $b = 200$ times from the estimated density functions of $\hat{\Pi}$ and $\hat{M}$. For each of these new samples we recompute the robustness criteria ($\bar{\hat{\mu}}$, $\hat{\rho}$), which yields the bootstrapped joint density function of these criteria. We perform the bootstrapping procedure for each of the four pull systems. The outcome of this procedure for Conwip is shown in Figure 42. We

note that bootstrapping is much faster than simulating, because bootstrapping involves little computing compared with discrete-event simulation.

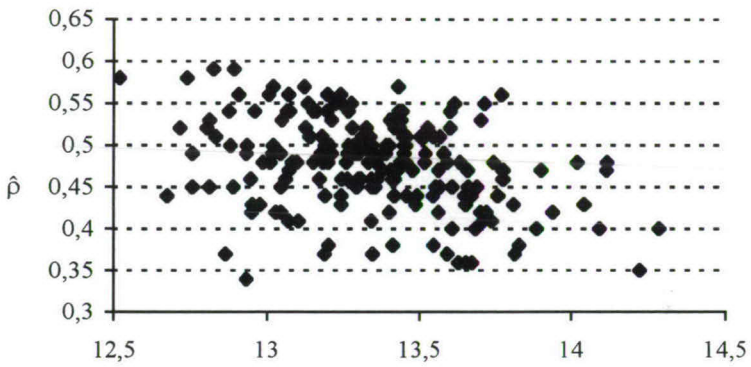


Figure 42. Bootstrapped joint density function of the estimated robustness criteria ($\bar{\hat{\mu}}$, $\hat{\rho}$) for Conwip

Using these four estimated bivariate density functions, we can build confidence ellipsoids for the two estimated robustness criteria for each of the four pull systems. We assume that the bootstrapped variables are *bivariate normal*. To test this assumption we apply Johnson and Wichern (1992, pp. 158-164), as follows. Denote the sample multivariate observations by X_j with $j = 1, \ldots, b$ (in our case x_j equals ($\bar{\hat{\mu}}$, $\hat{\rho}$)). Define the squared generalized distance D_j^2:

$$D_j^2 = (\boldsymbol{X}_j - \bar{\boldsymbol{X}})' \boldsymbol{S}^{-1} (\boldsymbol{X}_j - \bar{\boldsymbol{X}}) \text{ with } j = 1, 2, \ldots, b \qquad (13)$$

with bold letters for matrices and vectors,

$$\bar{\boldsymbol{X}} = \sum_{j=1}^{b} \boldsymbol{X}_j,$$

$$\boldsymbol{S} = \sum_{j=1}^{b} (\boldsymbol{X}_j - \bar{\boldsymbol{X}})(\boldsymbol{X}_j - \bar{\boldsymbol{X}})'/(b - 1).$$

Then v-variate normality (here $v = 2$) is not rejected if (i) roughly half of the d_j^2 are less than $\chi_v^2(0.50)$, which denotes the 50% quantile of the chi-square distribution χ_v^2, and (ii) a plot of the b ordered distances versus the b quantiles $\chi_v^2([j - 0.5]/b)$ gives a straight line. This test with $v = 2$ gives Figure 43, which suggests that the normality assumption may indeed be used for our Customized system. We assume that the assumption also holds for the other three systems.

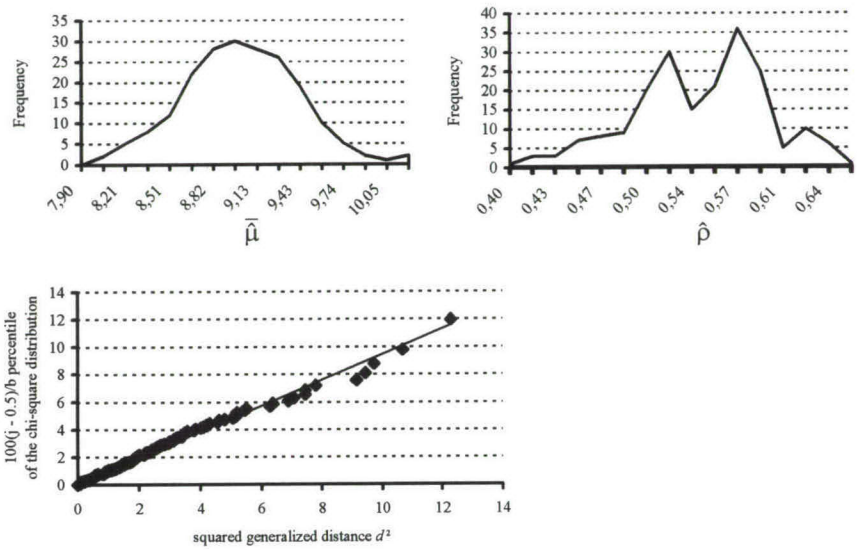


Figure 43. Testing normality of the bootstrapped $\bar{\hat{\mu}}$ and $\hat{\rho}$ for the our Customized system

Johnson and Wichern (1992, pp. 189) gives a 1 - α confidence region for v variates with means (say) ξ (in our case $\xi = (\mu, \rho)$):

$$b(\boldsymbol{X}_j - \xi)'\boldsymbol{S}^{-1}(\boldsymbol{X}_j - \xi) \leq f^{\alpha}_{2,\, b-2}\, 2(b-1)/(b-2)$$

where $f^{\alpha}_{2,\, b-2}$ denotes the upper α point (or 1 - α quantile) of the F statistic with degrees of freedom 2 and b - 2. We might apply this formula to each of the four pull systems with a type-I error rate of α. However, our selection of a pull system depends on all four confidence intervals simultaneously. Therefore we use Bonferroni's inequality: we replace α by α /4, which keeps the overall type-I error rate below α; see Kleijnen (1987). Taking α = 0.10 yields Figure 44. This figure suggests that our Customized system yields the lowest WIP cost $\bar{\hat{\mu}}$, but at the highest service risk $\hat{\rho}$, whereas Conwip yields the lowest service risk at the price of the highest WIP cost. Between these two extremes, Hybrid and Kanban are possible tradeoffs between service risk and WIP cost. Hybrid, however, dominates Kanban since both $\bar{\hat{\mu}}$ and $\hat{\rho}$ are lower. These conclusions are consistent with those made using stochastic dominance theory. Even though no unanimous decision can be made, managers can use Figure 44 for decision support.

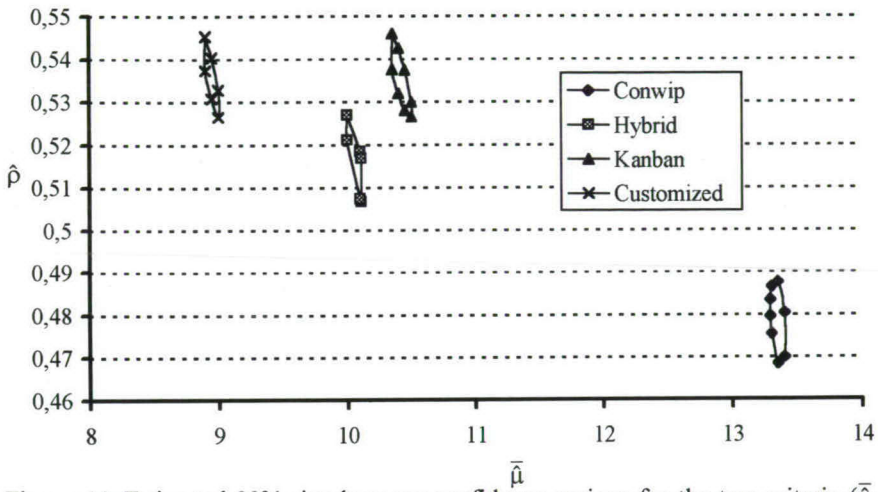


Figure 44. Estimated 90% simultaneous confidence regions for the two criteria ($\hat{\bar{\mu}}$, $\hat{\rho}$) for the four pull systems

6.3.3 Managerial decisions

To solve the robust design problem stated in (12), managers need to decide on values for several thresholds: which service target c_y should they use, and which risk do they judge to be acceptable (c_π and c_ρ)? Other decisions have to be made, such as which input probability distributions should be used to define the environmental scenarios? The purpose of this section is to investigate the possible effects of these managerial decisions on the robustness measures.

6.3.3.1 Effects of the card numbers

We have already seen (when comparing the four pull control systems in section 6.3.2) that different pull designs do yield different responses under uncertainty. We can further study the effects of the card numbers on the robustness measures by considering (say) Conwip with different amounts of cards. The effect of increasing the card number on the average WIP averaged over the scenarios $\hat{\bar{\mu}}$ is quite straightforward. Indeed, Figure 45 illustrates that $\hat{\bar{\mu}}$ increases linearly with the number of cards. The service risk $\hat{\rho}$, however, is not a linear function of the card number. Besides, it seems to be more sensitive to environmental disturbances than $\hat{\bar{\mu}}$. Indeed, $\hat{\rho}$ should be a decreasing function of the card number, but we observe that its value for 60 cards is higher than for 50 cards. Yet, the density functions of π_S in Figure 46 do show that a Conwip system with a given number of cards dominates any other Conwip system with lower card numbers. The noise sensitivity in Figure 45 may be due to the definition of $\hat{\rho}$ itself. Indeed, the indicator functions $I(a)$ in $\hat{\rho}$ (see formula (12))

have the value 0 or 1 depending on statement a that involves the thresholds c_y and c_π. Thus, small variations of $y_{i,s}$ in (12) can have dramatic effects on $\hat{\rho}$ – one might say that $\hat{\rho}$ is not a robust performance measure. These effects may be reduced by considering longer simulation runs, that is, by increasing the number of simulated shifts. This solution, however, has a high computational cost since simulation is repeated for 100 scenarios. We also tried to characterize service through another robustness measure. We used $\bar{\hat{\mu}}$, the average monthly service averaged over the scenarios. An interesting result is that the comparison results obtained in section 6.3.2 still hold. The behavior of $\bar{\hat{\mu}}$ as a function of the number of cards is smoother than that of $\hat{\rho}$ in Conwip. Conceptually, however, an average ($\bar{\hat{\mu}}$) is less interesting as a robustness measure than the probability of poor performance ($\hat{\rho}$).

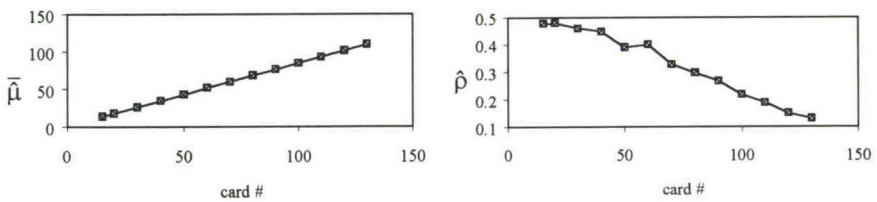


Figure 45. Robustness measures $\bar{\hat{\mu}}$ and $\hat{\rho}$ as functions of the number of cards, in Conwip

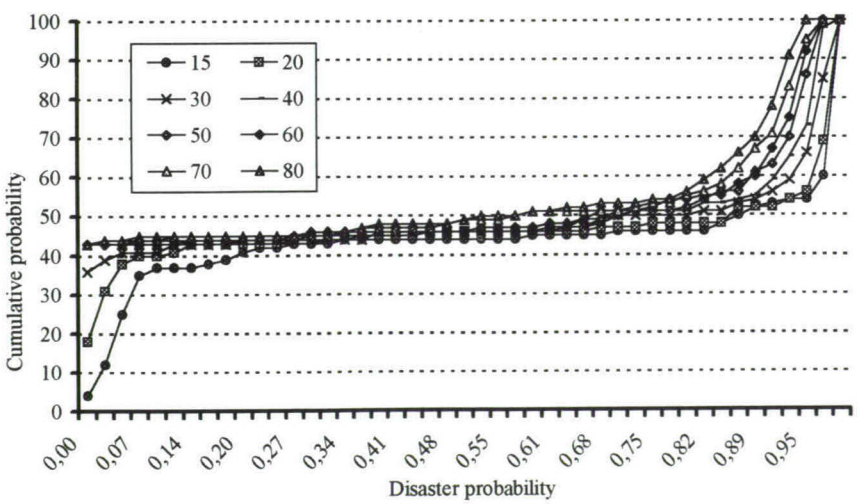


Figure 46. Effect of the number of cards c on the disaster probability in Conwip with $c_y = 0.999$, estimated from $n = 100$ scenarios

6.3.3.2 Choosing a value for the maximum number of disasters c_π

Choosing a value for c_π means drawing a vertical line in the plot of the distribution function of the disaster probability, and reading the corresponding cumulative probability. $\hat{\rho}$ is the complement to 1.0 of this cumulative probability. Figure 46 shows that the choice of a value for c_π is critical: for disaster probabilities between 0.1 and 0.8 the cumulative density functions have almost the same values, whatever the number of Conwip cards. Thus, for c_π values between 0.1 and 0.8 the robustness measure $\hat{\rho}$ is not a powerful measure for comparing pull systems. The extreme case occurs for $c_\pi = 0.3$; Conwip systems with 15 to 80 cards yield almost the same $\hat{\rho}$ value; see Table 23. Yet, these systems are not equivalent in terms of service performance – we saw in the previous section that a Conwip system with a given number of cards dominates any other Conwip systems with lower card numbers.

Table 23. $\hat{\rho}$ for Conwip with various numbers of cards, given $c_\pi = 0.3$

# cards	15	20	30	40	50	60	70	80
$\hat{\rho}$	0.57	0.55	0.56	0.56	0.54	0.54	0.54	0.55

Selecting either low or high values of c_π yields more relevant results. For instance, for $c_\pi = 0$ – no disaster; see the left-hand side in Figure 46 – the cumulative probability is an increasing function of the Conwip card number and tends to a limit of 0.43 corresponding to $\hat{\rho} = 0.57$ (= 1 - 0.43). This result makes sense: increasing the number of cards does improve service performance, but system capacity is eventually reached.

6.3.3.3 Effects of the service target c_y

Obviously, the choice of a service level target influences only the service related performance measures. Figure 47 shows the distribution functions of the disaster probability for Conwip with 15 cards, for target values of 95%, 97%, and 99.9% (see solid curves in Figure 47). The lower the target, the higher the probability of no disaster: $P(\pi_S = 0 | c_y = 0.95) > P(\pi_S = 0 | c_y = 0.999)$ (see left-hand side).

A less obvious observation is that changing the service target does not seem to affect the ranking of different types of pull systems in terms of $\hat{\rho}$. Figure 47 also shows the distribution functions of the disaster probability for our Customized system (dashed curves): Conwip dominates Customized for any given service target.

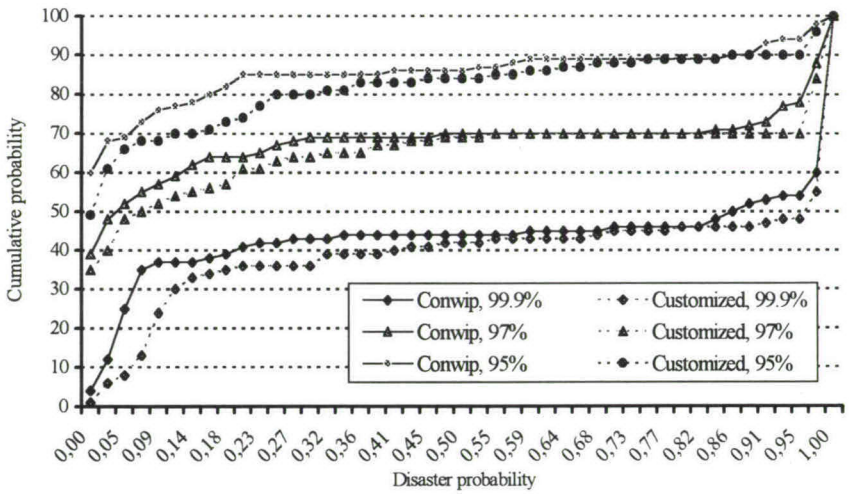


Figure 47. Effect of service target c_y on estimated disaster probability, for Conwip (with 15 cards) and the customized system

6.3.3.4 Effect of LHS input distributions

Another issue is the selection of input distributions for URA. As we explained in section 5.4.2 (also see Figure 36), URA samples each unknown parameter from a statistical distribution function. Experts should define these functions and specify their ranges and

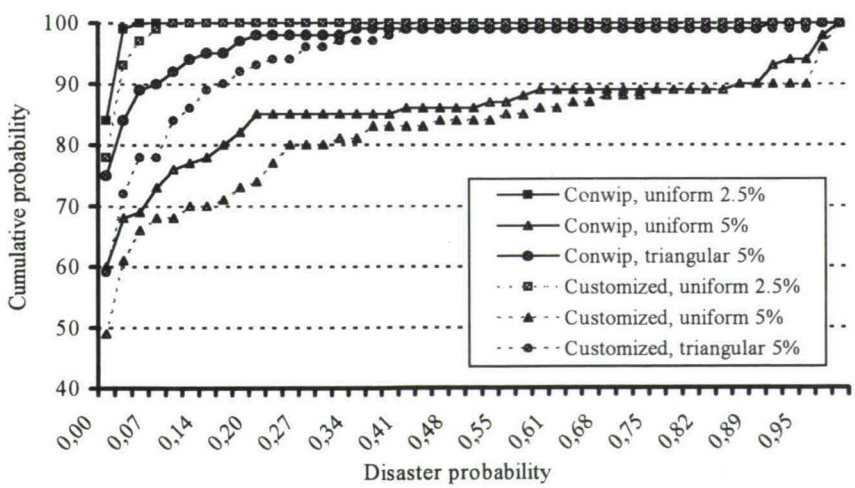


Figure 48. Effect of LHS input range and distribution shape on estimated disaster probability for Conwip (15 cards) and Customized system given the service target $c_y = 0.95$

shapes. A question then arises: to which environments should the system be robust? To answer this question, we perform a few experiments with various LHS input ranges and distribution shapes for two pull systems, namely, Conwip with 15 cards and our Customized system. We study LHS input ranges of 5% and 2.5% around the base scenario, and uniform and triangular distribution shapes. Figure 48 shows that the choice of LHS input distributions has a critical effect on the service performance. Obviously, for a given distribution shape, the larger the range of the input values, the bigger the probability of disasters: the disaster probability for uniform input distributions with 2.5% ranges first-order dominates the disaster probability for uniform input distributions with 5% ranges. Also, for a given input range, uniform distributions yield bigger disaster probabilities than triangular distributions: triangular distributions have lower probabilities of extreme input values, so less extreme scenarios are generated by LHS. For the three LHS input distributions that we consider, Conwip dominates our Customized system (solid curves versus dashed curves). This suggests that the choice of LHS input distributions does not have an effect on the ranking of different types of pull systems in terms of service.

6.3.3.5 Value of the risk level c_ρ

The choice of a value for c_ρ in (12) is also critical. Indeed, its purpose is to characterize the managers' preference concerning the $\bar{\mu}/\hat{\rho}$ compromise. To illustrate this issue, we use the robustness measures of four pull systems displayed before in Table 22 (obtained for $c_\pi = 0.9$). If managers select $c_\rho = 0.6$, then they prefer the system with lowest inventory (because all four systems have $\hat{\rho} < 0.6$), which is our Customized system. If, however, they choose $c_\rho = 0.525$, then Conwip and Hybrid are the only systems that satisfy the constraint on $\hat{\rho}$. Hybrid would be preferred over Conwip, since Hybrid's $\bar{\mu}$ value is lower. An important issue is how to select a value for c_ρ. The study of c_π in section 6.3.3.2 provides good support for this selection. Indeed the analysis of Figure 46 showed that the value of $\hat{\rho}$ in a Conwip system tends to a limit when the number of cards becomes large. For instance, we saw that $\hat{\rho}$ tends to a value of 0.57 for $c_\pi = 0$. Thus managers may choose to select $c_\rho = 0.57$. More generally, the procedure consists in building a figure similar to Figure 46 (that is, $\hat{\rho}$ in a Conwip system as a function of the number of cards) and study the value of $\hat{\rho}$ for a given value of c_π.

6.4 Example of robust optimization

The objective of this section is to illustrate our robust optimization procedure for an example production system. We consider the production system that we used at several other occasions in this dissertation, namely, the system in Bonvik *et al.* (1997). First, we consider Conwip systems only. Second, we consider robust customization, that is, customization under uncertainty.

For each set of decision parameters (namely, the card numbers), we simulate 100 scenarios over a period of 44 shifts each (plus a warming-up of 4 days). Thus, we use exactly the same experimental conditions as in sections 6.2 and 6.3. Furthermore, we choose to use the following values for the decision parameters: $c_\pi = 0$, $c_y = 0.999$, $c_\rho = 0.5985$ (= 0.57 + 5%; we choose a slightly easier target than the asymptotic value found in section 6.3.3.2). We selected those values according to the results obtained in section 6.3. The statement of the robust optimization problem (12) becomes:

$$\begin{aligned} &\min_x \sum_{s=1}^{100} \int_{2700}^{22500} WIP_{t,s}(x)\,dt/19800/100, \\ &\text{s.t. } \hat{\rho}(x) = \sum_{s=1}^{100} I(\sum_{i=1}^{44} I(y_{i,s}(x) < 0.999)/44 \geq 0)/100 < 0.5985 \end{aligned} \tag{14}$$

In the case of Conwip, the set of decision parameters reduces to a single number of cards. Thus, the robust optimization problem (14) can be solved easily through exhaustive search. We estimate $\hat{\rho}$ and $\bar{\hat{\mu}}$ through bootstrapping according to the procedure detailed in section 6.3.2 (we use the center of the ellipsoid). This estimation is rather expensive in terms of computing time. Indeed, the estimation of the robustness measures ($\hat{\rho}$, $\bar{\hat{\mu}}$) through simulation, LHS, and bootstrapping requires about 40 minutes on a Pentium 90 MHz, and 7 minutes on a Pentium II 350 MHz. For the robust optimization of Conwip, this cost is acceptable since the search space is very limited. We find that the Conwip system that satisfies the constraint on $\hat{\rho}$ with minimal $\bar{\hat{\mu}}$ has 36 cards. We note that for the same production system and under the base scenario, the best Conwip system found by Bonvik *et al.* (1997) had 15 cards only. This large difference shows that considering uncertainties does have a major influence on the outcome. Of course, we might have found a lower number of cards if we had taken other values for the managerial decisions (see section 6.3.3).

Customization is much more difficult when aiming at robustness. Indeed, the search space is much larger and requires the use of a heuristic optimization technique. We estimate that using the evolutionary algorithm of section 3.5, a solution to problem (14) would be obtained after approximately one week of computation on a Pentium II 350 MHz. Such a cost is unfortunately not affordable in our research. Actually, this cost may seem extremely high, in general. It is important, however, to emphasize that most of the research presented in this thesis would not have been possible a few years ago. Computer power has increased dramatically during the last few years, and we are convinced that robust customization will quickly become affordable. Besides, the reduction of computational costs can be accelerated through the use of parallel evolutionary algorithms (Paris and Pierreval, 1997).

6.5 Conclusion

In this chapter, we examined the robustness issue when customizing pull systems. The challenge was to adapt the procedure developed in Chapter 5. First we stated the robust customization problem, and we defined rigorous notation. We chose two robustness measures, namely, (i) the average monthly WIP averaged over 100 LHS scenarios and (ii) the proportion of disaster probabilities that exceeds a threshold c_π in these scenarios.

Second, we studied these two measures for four "optimized" pull systems, already studied in Bonvik *et al.* (1997) and section 3.6.3. Two comparison procedures, namely, stochastic dominance and confidence ellipsoids built through bootstrapping, yielded consistent conclusions: our customized system yields the lowest WIP risk, at the cost of the highest service risk. Hybrid and Kanban yield a tradeoff between the two robustness measures. Conwip has the lowest service risk, at the cost of the highest WIP risk. A type of pull system can be selected only if managers specify their attitude towards risk and characterize their preferences.

Third, to support managers we investigated the effects of the various parameters within their control, namely the numbers of cards (x), the various thresholds in statement (12) (c_y: manager's target for the service level per shift, c_π: target for the proportion of disastrous shifts per month, and c_ρ: acceptable risk in terms of service), and the LHS input distributions. Our conclusion is that all these parameters have major effects on the robustness measures. Furthermore, the value of c_π should be chosen such that $\hat{\rho}$ is sensitive to variations in the numbers of cards. The value of c_ρ should not exceed the asymptotic value of $\hat{\rho}$ when the card number tends to infinity.

Fourth, we applied the robust optimization procedure to the production system studied in Bonvik *et al.* (1997) controlled through Conwip. The results show that considering uncertainties yields results completely different from those obtained for a production environment known with certainty (base scenario). Thus, designing a pull system for a single scenario may be extremely risky. Our procedure provides one solution for minimizing this risk, provided managers can specify their attitude towards risk and characterize their preferences.

Chapter 7
Conclusions and Further Research

Throughout this dissertation we considered two design issues: which type of pull system to choose, and how to set the various parameters of the chosen system. We contributed to these issues in several ways.

- We proposed a new classification of pull systems studied in the literature, namely three classes: *traditional*, *segmented*, and *joint* systems.
- We raised the problem of selecting a pull system among known pull systems and systems not considered in the literature so far. In order to solve this complex problem, we proposed a generic model and a selection procedure based on a single optimization. We called this procedure *customization*.
- We illustrated the benefits of our customization approach through an example production system taken from the literature: we find a customized pull system that performs significantly better than the best system so far, which was Kanban/Conwip Hybrid.
- We applied our methodology to a variety of production lines. Twelve lines that we selected using statistical techniques, yielded various new types of pull systems of quite low complexity and high levels of performance. Three structural patterns of pull control were highlighted through these experiments.
- We raised the issue of uncertainty in performance estimation through simulation: the production environment is not known with certainty. We identified three sources of uncertainty and proposed a novel procedure for designing systems under such uncertainties. The objective of this procedure is to minimize the risk of poor performance. The procedure combines tools such as uncertainty and risk analysis, robust design, and bootstrapping.
- Next, we investigated to what extend risk considerations impact the choice of a pull system. We showed that managers might prefer systems with characteristics completely different than those chosen for known production environments (base scenario). A limitation of our risk-based approach, however, is its high computational cost.

During our research we thought about many exciting *perspectives*. Because of time constraints, however, we had to restrict our research scope. Future research may include the following topics.

- Throughout the dissertation, we considered single-product flow lines. A necessary step before application of our research in industry is to extend customization to multiple product systems, as well as assembly and disassembly systems. For this purpose, the literature on Kanban and Conwip is a valuable source of inspiration.
- Some logistical aspects should also be included. We considered the supply of raw materials as perfect. This assumption is critical, so the effect of imperfect supply should be investigated. More generally, we could study logistical chains as a whole, including raw material ordering policies for one or several suppliers, and delivery policies (multiple customers, delivery splitting).
- Information plays a key role in pull systems in general, and in our customized systems in particular. However, some patterns of information flow (superposition of control loops), may be difficult to implement. It is particularly difficult when information is materialized through cards (as in Kanban). The use of electronic signals is one possible solution that insures pull system integrity (cards can be lost or forgotten) and instantaneous transmission of information. Such implementation issues should be investigated in detail.
- Because of computational limitations we could not exploit the full potential of robust customization. One perspective is to try and identify robust patterns of pull control. For this purpose we might use the same technique as for customization in known production environments, that is, perform customization for a variety of production systems. Since the main challenge is to reduce computational costs, we might also use parallel computing or find a more efficient risk assessment technique (replace LHS by another technique).

We conclude this dissertation by considering possible implications of our research in *education*. We see two main points that disserve a specific attention.

First, in many course books dealing with production control and inventory management, pull control is limited to Kanban systems only. Yet, other types of pull systems yield much higher levels of performance, for equivalent levels of complexity. We showed in this dissertation that it is extremely difficult to say a priori which type of pull system is best for a given production environment. The three patterns identified in Chapter 4 give a good basis for the design of pull systems. Students in the field of production control and inventory management should have a broader view of pull control than Kanban only.

Second, simulation is often used as a magic tool that always yields results. Yet, a common saying among computer users is 'garbage in, garbage out', which means that the quality of input data has the utmost influence on the outcome of simulation. In practice, however, it is rare to know input data with certainty. The three sources of uncertainty identified in the dissertation can be identified in any simulation study. Yet, simulation often supports critical decisions such as system design that involve major financial investments. Thus, underestimating uncertainty might be extremely risky. Simulation practitioners should be aware of such risks, and should be able to quantify them. A key role of education

in simulation is to develop this awareness and provide the right tools for dealing with risks and uncertainties. A possible solution could be to include systematically sensitivity and uncertainty analyses in simulation teaching.

Appendices

Appendix 1. Demonstration of Property 2

The idea of Property 2 is to search for the maximal number of parts authorized for production by the control loops on the line portion of interest, that is, control loops that start and finish in that portion. This maximal number depends on the number of cards circulating in the control loops in that portion. Thus we are looking for the control loops in the line portion that, taken together, constraint the flow of parts the most.

We denote by MAX(i, $i+n$) the maximal number of parts allowed in stages i to $i+n$ by the control loops starting and finishing between stages i and $i+n$. We will demonstrate by generalized recurrence the following proposition:

(P_0) $\text{MAX}(i, i) = k_{i;i}$,

(P_n) $\text{MAX}(i, i+n) = \text{Min}(k_{i+n;i}, \text{Min}_{i \le l < i+n} [\text{MAX}(i, l) + k_{i+n;l+1}]), \forall n \neq 0$

Demonstration

- (P_0). Only one control loop starts and finishes at stage i, namely, $CL_{i;i}$. Thus the maximal number of parts allowed in stage i by the control loops starting and finishing at stages i is equal to the number of cards circulating in $CL_{i;i}$. Therefore, $\forall i$, $\text{MAX}(i, i) = k_{i;i}$.
- (P_1). We are considering a two-stage line portion as shown in Figure 49.

The maximal number of parts allowed in stages i and $i + 1$ by the control loops starting and finishing between stages i and $i + 1$ is either fixed locally by $CL_{i;i}$ and $CL_{i+1;i+1}$ together or by $CL_{i+1;i}$ only. The control mechanism that has the lowest number of cards yields the strongest constraint.

Thus, $\text{MAX}(i, i + 1) = \text{Min}\ (k_{i+1;i}, k_{i;i} + k_{i+1;i+1})$.

Since (P_0): $\forall i$, $\text{MAX}(i, i) = k_{i;i}$, we have $k_{i;i} + k_{i+1;i+1} = \text{MAX}(i, i) + k_{i+1;i+1}$.

So $\text{MAX}(i, i + 1) = \text{Min}\ (k_{i+1;i}, \text{MAX}(i, i) + k_{i+1;i+1})$

$= \text{Min}\ (k_{i+1;i}, \text{Min}_{l=i} [\text{MAX}(i, l) + k_{i+1;l+1}])$

$= \text{Min}\ (k_{i+1;i}, \text{Min}_{i \le l < i+1} [\text{MAX}(i, l) + k_{i+1;l+1}])$

Therefore (P_1) is true.

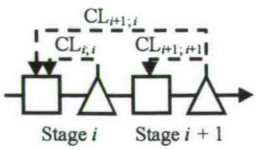


Figure 49. Two-stage line portion and control loops starting and finishing between stages i and $i + 1$

- (P_n). We suppose that (P_m) is true for $m < n$, and we want to prove that (P_n) is true. By assumption (P_{n-1}) is true. When adding stage $i + n$ to the portion of the line including stage i to stage $i + n - 1$, we add the control loops starting at stage $i + n$ to all preceding stages; see

the bold arrows in Figure 50. Since parts at stage $i + n$ have cards from at least one of the n control loops starting at stage $i + n$, we know that one of these control loops contributes to MAX(i, i+n). If $CL_{i+n;i+n}$ is this control loop, then we know that the maximal number of parts allowed in stage $i + n$ by the control loops starting and finishing between stages i and $i + n$ is $k_{i+n;i+n}$; the contribution to MAX(i, i+n) by the remaining stages, that is, stages i to $i + n - 1$, is the maximal number of parts allowed in stages i to $i + n$ - 1 by the control loops starting and finishing between stages i and $i + n - 1$: MAX(i, i+n-1). By extension, if $CL_{i+n;m}$ is the control loop that contributes to MAX(i, i+n), then the contribution to MAX(i, i+n) by the remaining stages (that is, stages i to $i + n - m$) is MAX(i, i+n-m-1). In total, there are n possibilities and MAX(i, i+n) is the possibility that has the smallest number of cards:

$$\begin{aligned}\text{MAX}(i, i+n) = \text{Min}\ \{&\text{MAX}(i, i+n-1) + k_{i+n;i+n};\ \ldots;\\ &\text{MAX}(i, i+n-m-1) + k_{i+n;i+n-m};\\ &\text{MAX}(i, i) + k_{i+n;i+1};\\ &k_{i+n;i}\}\end{aligned}$$

We define $l = i + n - m - 1$; since m varies between zero and $n - 1$ in the preceding equality, l varies between i and $i + n - 1$. Then, $\text{MAX}(i, i+n) = \text{Min}_{i \le l < i+n}\ \{\text{MAX}(i, l) + k_{i+n;l+1};\ k_{i+n;i}\}$. Since $k_{i+n;i}$ is independent of l, we obtain the same formula as (P_n). Therefore, (P_n) is true.

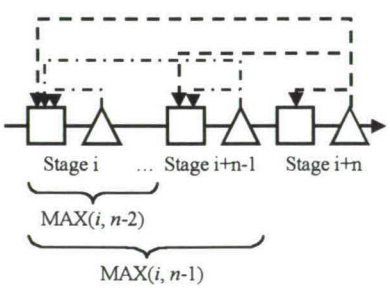


Figure 50. Effect of adding one stage to the line portion

Thus we demonstrated through generalized recurrence that (P_n) is true for all n. Let us consider the case $n > 0$. Then, $\text{MAX}(i, i+n) = \text{Min}(k_{i+n;i}, \text{Min}_{i \le l < i+n}\ [\text{MAX}(i, l) + k_{i+n;l+1}])$. Our objective is to determine a value for the number of cards $k_{i+n;i}$ in the control loop $CL_{i+n;i}$. If $k_{i+n;i}$ is strictly larger than $\text{Min}_{i \le l < i+n}\ [\text{MAX}(i, l) + k_{i+n;l+1}]$, then the control loop $CL_{i+n;i}$ does not constraint the flow of parts and it does not need to be implemented. However, if $k_{i+n;i}$ is smaller than $\text{Min}_{i \le l < i+n}\ [\text{MAX}(i, l) + k_{i+n;l+1}]$, then the control loop does constraint the flow of parts.

During the third part of the demonstration, that is (P_n), we saw that the added stage $i + n$ can contribute only once to MAX(i, i+n). Since MAX() is a recursive function, this

statement holds also for each stage in the line: each stage contributes only once to MAX(). This means that two control loops considered for the computation of MAX() can not control the same stage; they cannot overlap. Therefore, $\mathrm{Min}_{i \le l < i+n}\,[\mathrm{MAX}(i, l) + k_{i+n;l+1}]$ is equal to the number of cards that can be found in the most constraining non-overlapping sequence of loops controlling all stages in the line.

Appendix 2. Central composite and Plackett-Burman designs

Design of Experiments (DOE) is the science of selecting factor combinations, among a large number of possibilities, to be studied experimentally. The motivation for this selection is that experimentation is often time consuming; therefore, the number of experiments to be performed should be reduced as much as possible. This reduction is only possible by assuming particular types of the relationships among the factors (inputs of the experiment) and the measured results (outputs of the experiment): these relationships may be expressed as mathematical equations (for example, regression models). Depending on the assumptions, the required number of experiments can be limited more or less. Statistical techniques are required to check the validity of the assumptions, and to estimate the mathematical models.

This appendix gives a few examples of two types of experimental designs, namely, Plackett-Burman and central composite. More details can be found in Kleijnen (1987), pp. 312-314 and pp. 329-336 respectively. DOE and its applications are discussed thoroughly in Kleijnen (1998).

• Plackett-Burman designs

Plackett-Burman designs minimize the number of experiments for studying first-order (main) effects. Such designs are built through generators (Plackett and Burman, 1946) such as shown in the following (N is the maximum number of factors that can be studied with the design).

$N = 12$ + + - + + + - - - + -

$N = 20$ + + - - + + + + - + - + - - - - + + -

$N = 24$ + + + + + - + - + + - - + + - - + - + - - - -

The generator corresponds to the first column of the design. The next columns are obtained through cyclical permutation of the generator. One line of minuses is added to the design. These example generators show that the number of factors is a multiple of four; otherwise, "dummy" factors can be introduced.

• Central composite designs

Central composite designs combine a factorial design, a star design (one factor at a time design), and a central point. An example for three factors (say) X, Y, and Z is shown in the following table, where $a \neq 1$, $a \neq 0$.

Run #	X	Y	Z
1	+1	+1	+1
2	-1	+1	+1
3	+1	-1	+1

2^3 factorial	4	-1	-1	+1
Design	5	+1	+1	-1
	6	-1	+1	-1
	7	+1	-1	-1
	8	-1	-1	-1
	9	+a	0	0
	10	-a	0	0
Star	11	0	+a	0
Design	12	0	-a	0
	13	0	0	+a
	14	0	0	-a
Central point	15	0	0	0

Appendix 3. Stochastic dominance theory

The theory of *stochastic dominance* (Wolfstetter 1996) can be used to rank the four PPCSs, as follows. Let X and Y be two random variables with x and y corresponding realizations.

• X *first-order stochastically dominates* Y ($X \geq_{FSD} Y$) if $Pr\{X>z\} \geq Pr\{Y>z\}$ for all z. X is *unanimously preferred* to Y by all agents (managers for instance) with monotone increasing utility functions if and only if $X \geq_{FDS} Y$. A *utility function* is a mathematical expression that assigns a value to all possible choices. In investment theory the utility function is the expression of preferences with respect to perceived risk and expected return. The higher the values of the utility function, the better.

Example of first-order stochastic dominance

For the example shown in Figure 51 we clearly have $G(z) \geq F(z)$ for all z (where F and G are the cumulative probability functions of X and Y respectively), which is equivalent to $Pr\{Y \leq z\} \geq Pr\{X \leq z\}$, for all z and to $Pr\{X>z\} \geq Pr\{Y>z\}$, for all z.

Thus, X is unanimously preferred to Y by all agents with monotonic increasing utility functions. We can interpret this result as follows: since the utility function is monotone increasing, we are looking for high probabilities of realization for high values of the random variables. Thus, the steeper the cumulative function for high values of the random variables, the better. Ideally, we would like to have $Pr\{X = b\} = 1$.

We note that the preference ranking is inverted if the utility function is monotone decreasing. Thus, in the example of Figure 51 one would prefer Y to X, because low realization values are more interesting.

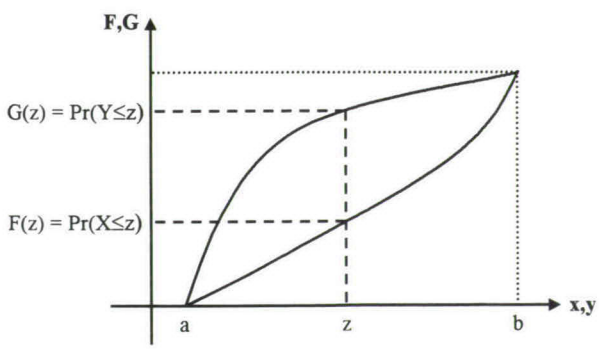


Figure 51. X first-order stochastically dominates Y (monotone increasing utility function)

• X *second-order stochastically dominates* Y ($X \geq_{SSD} Y$) if

$$\int_a^k \Pr\{X > x\}dx \geq \int_a^k \Pr\{Y > y\}dy, \text{ for all } k \qquad (15)$$

X is *unanimously preferred* to Y by all agents with monotone increasing and strictly concave utility functions if and only if $X \geq_{SSD} Y$. Y is called "stochastically more risky" than X. The strictly concave condition on the utility function expresses the risk aversion of the agent.

The inequation (15) can be reformulated as follows: $\int_a^k \Pr\{Y \leq y\}dy \geq \int_a^k \Pr\{X \leq x\}dx$, for all k. Thus, $X \geq_{SSD} Y$ if and only if the area below the cumulative probability function of Y is larger than for X, on any [a; k] interval.

Example of second-order stochastic dominance

For the example shown in Figure 52 we clearly have $S_X(k) \geq S_Y(k)$ for all k.

Thus, Y second-order stochastically dominates X; hence Y is unanimously preferred to X by all agents with monotone increasing and strictly concave utility functions.

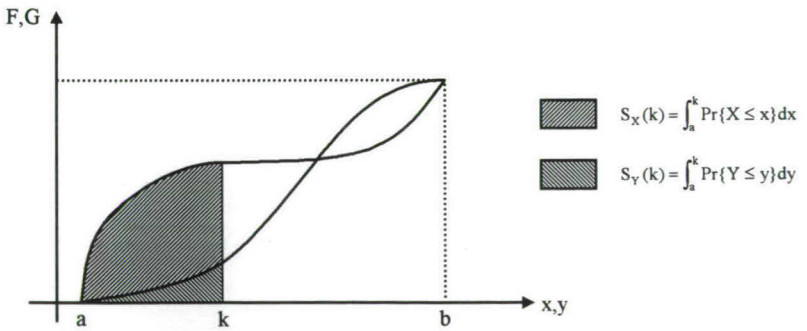


Figure 52. Y second-order stochastically dominates X (monotone increasing and strictly convex utility function)

Bibliography

Amin, M., and T. Altiok. 1997. Control policies for multi-product multi-stage manufacturing systems: an experimental approach. *International Journal of Production Research*, 35(1): 201-223.

Aytug, H., C.A. Hogan, and G. Bezmez. 1996. Determining the number of kanbans: a simulation metamodelling approach. *Simulation* 67: 23-32.

Bäck, T. 1996. *Evolutionary algorithms in theory and practice*. Oxford University Press, New York.

Bäck, T., and H.P. Schwefel. 1993, An overview of evolutionary algorithms for parameter optimisation. *Evolutionary computation* 1(1): pp 1-23.

Baker, J.E. 1985. Adaptive selection methods for genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. J.J. Grefenstette, ed., Erlbaum.

Balson, W.E., Welsh, J.L., and D.S. Wilson. 1992. Using decision analysis and risk analysis to manage utility environmental risk. *Interfaces* 22(6): 126-139.

Benjamin, P.C., M. Erraguntla, and R.J. Mayer. 1995, Using simulation for robust system design. *Simulation* 65(2), 116-128.

Benton, W.C., and H. Shin. 1998. Manufacturing planning and control: the evolution of MRP and JIT integration. *European Journal of Operational Research* 110: 411-440.

Berkley, B.J. 1992. A review of the Kanban production control research literature. *Production and Operations Management* 1: 393-411.

Berkley, B.J. 1996. A simulation study of container size in two-card Kanban systems. *International Journal of Production Research* 34(12): 3417-3446.

Bertrand, J.W.M. 1983. The use of workload information to control job lateness in controlled and uncontrolled release production systems. *Journal of Operations Management* 3(2): 79-92.

Bonvik, A.M., C.E. Couch and S.B. Gershwin. 1997. A comparison of production-line control mechanisms. *International Journal of Production Research*. 35(3): 789-804.

Brehmer, B., E.A. Eriksson, and P. Wulff. 1994. Risk management (feature issue). *European Journal of Operational Research* 75: 477-566.

Buzacott, J.A. 1989. Queueing models of Kanban and MRP controlled manufacturing systems. *Engineering Cost and Production Economics* 17: 3-20.

Buzacott, J.A., S. Price, and J.G. Shanthikumar. 1991. Service level in multi-stage MRP and Base Stock controlled production systems. *Proceedings of the Conference on New Directions for Operations Research in Manufacturing*.

Buzacott, J.A., and G.J. Shanthikumar. 1993. *Stochastic models of manufacturing systems*. Prentice Hall, Englewood Cliffs, New Jersey.

Chang T.-M., and Y. Yih. 1994. Determining the number of kanbans and lotsizes in a generic kanban system: a simulated annealing approach. *International journal of production research* 32(8): 1991-2004.

Chang, T.-M., and Y. Yih. 1998. A fuzzy-based approach for dynamic control of kanbans in a generic kanban system. *International Journal of Production Research* 36(8): 2247-2257.

Chu, C.-H., and W.-L. Shih. 1992. Simulation studies in JIT production. *International Journal of Production Research* 30: 2573-2586.

Cochran, J.K., and S.-S. Kim. 1998. Optimum junction point location and inventory levels in serial hybrid push/pull production systems. *International Journal of Production Research* 36(4): 1141-1155.

Cukier, R.I, C.M. Fortuin, K.E. Schuler, A.G. Petschek, and J.H. Schaibly. 1973. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients: I Theory. *Journal of Chemical Physics* 59: 3873-3878.

Cyert, R.M., and M.H. DeGroot. 1987. *Bayesian analysis and uncertainty in economic theory*. Chapman and Hall, London.

Dallery, Y., and G. Liberopoulos. 1995, A new Kanban-type pull control mechanism for multi-stage manufacturing systems. *Proceedings of the 3rd European Control Conference*, Rome, September 5-8, 4(2): 3543-3548.

Davis, W.J., and S.J. Stubitz. 1987. Configuring a kanban system using discrete optimization of multiple stochastic responses. *International Journal of Production Research* 25(5): 71-740.

De Jong, K.A. 1975. *An analysis of the behavior of a class of genetic adaptive systems*. Ph.D. Thesis. University of Michigan, Ann Arbor.

De La Maza, M., and B. Tidor. 1993. An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection. *Proceedings of the Fifth International Conference on Genetic Algorithms*. S. Forrest, ed., Morgan Kaufmann.

Di Mascolo, M., Y. Frein, and Y. Dallery. 1996. An analytical method for performance evaluation of Kanban controlled production systems. *Operations Research* 44(1): 50-64.

Dooley, K.J., and F. Mahmoodi. 1992. Identification of robust scheduling heuristics: application of Taguchi methods in simulation studies. *Computers and Industrial Engineering* 22(4): 359-368.

Duri C. 1997. Etude comparative des gestions à flux tiré. *Thèse de doctorat de l'Institut National Polytechnique de Grenoble*, Laboratoire d'Automatique de Grenoble, Janvier 1997.

Efron, B., and R.J. Tibshirani. 1993. *An introduction to the bootstrap*. Chapman & Hall, New York.

Ettl, M., and M. Schwehm. 1995. Determining the optimal network partition and kanban allocation in JIT production lines. In *Evolutionary algorithms in management applications*. Biethahn, J., and V. Nissen, eds., Springer-Verlag, pp. 139-152.

Fogel, L.J., A.J. Owens, and M.J. Walsh. 1966. *Artificial intelligence through simulated evolution*. Wiley, New York.

Forrest, S. 1985. Scaling fitnesses in the genetic algorithm. In documentation for Prisoners Dilemna and NORMS programs that use the genetic algorithm. Unpublished manuscript.

Frein, Y., M. Di Mascolo, and Y. Dallery. 1995. On the design of generalized Kanban control systems. *International Journal of Operations in Production Management* 15(9): 158-184.

Gaury, E.G.A., H. Pierreval, and J.P.C. Kleijnen. 1997a. Modélisation et simulation dans l'étude de systèmes de production gérés en Juste-A-Temps. *Proceedings of the First French-speaking AFCET/FRANCOSIM/SCS conference on modeling and simulation* 113-123. MOSIM'97, 5-6 June, Rouen, France.

Gaury, E.G.A., H. Pierreval, and J.P.C. Kleijnen. 1998. New species of hybrid pull systems. CentER Discussion Paper No. 9831, Tilburg University, Netherlands.

Gaury, E.G.A., J.P.C. Kleijnen, and H. Pierreval. 1997b. Configuring a pull production-control strategy through a generic model. CentER Discussion Paper No. 97101. Tilburg University, Netherlands.

Goldberg, D.E. 1989. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley Publishing Company.

Goldberg, D.E. 1990. A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. *Complex Systems* 4: 445-460.

Goldberg, D.E., and K. Deb. 1991. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*. G. Rawlins, ed., Morgan Kaufmann.

Goldrat, E.M., and R.E. Fox. 1986. *The race*. New York North River Press.

Golhar, D.Y., and C.L. Stamm. 1991. The just-in-time philosophy: A literature review. *International Journal of Production Research* 29(4): 657-676.

Gross, D., and C.M. Harris. 1998. *Fundamentals of queueing theory*. Wiley, New York.

Gstettner, S., and H. Kuhn. 1996. Analysis of production control systems Kanban and Conwip. *International Journal of Production Research* 34(11): 3253-3274.

Gupta, P.Y,. and C.M. Gupta. 1989. A system dynamic model for multi-stage multi-line dual-card JIT-Kanban system. *International Journal of Production Research* 27: 309-352.

Hacking, I. 1975. *The emergence of probability*. Cambridge University Press.

Hammersley, J.M. 1960. Monte Carlo methods for solving multivariate problems. *Annals of the New York Academy of Science* 86, 844-874.

Helton, J.C. 1997. Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. *Journal Statistical Computation and Simulation* 57: 3-76.

Hillier, F.S., and R.W. Boling. 1966. The effect of some design factors on the efficiency of production lines with variable operation times. *Industrial Engineering* 17: 651-658.

Ho, Y.-C., and X.-R. Cao. 1991. *Perturbation Analysis of Discrete Event Dynamic Systems*. Kluwer Academic Press.

Holland, J.H. 1975. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, Michigan.

Hopp, W.J., and M.L. Roof. 1998. Setting WIP levels with statistical throughput control (STC) in Conwip production lines. *International Journal of Production Research* 36(4): 876-882.

Huang, C.-C., and A. Kusiak. 1996. Overview of Kanban systems. *International Journal of Computer Integrated Manufacturing* 9: 169-189.

Huang, C.-C., and A. Kusiak. 1998. Manufacturing control with a push-pull approach. *International Journal of Production Research* 36(1): 251-275.

Huang, P.Y., L.P. Rees, and B.W. Taylor 1983. A simulation analysis of the Japanese Just-In-Time technique (with Kanbans) for a multiline, multistage production system. *Decision Sciences* 14: 326-344.

Hurion, R.D. 1997. An example of simulation optimization using a neural network metamodel: finding the optimum number of kanbans in a manufacturing system. *Journal of the Operations Research Society* 48: 1105-1112.

Iman, R.L., and M.J. Shortencarier. 1984. A FORTRAN 77 program and user's guide for the generation of Latin Hypercube and random samples for use with computer models. NUREG/CR-3624, SAND83-2365, Sandia National Laboratories, Albuquerque, New Mexico.

Jacobson, S.H., A.H. Buss, and L.W. Schruben. 1991. Driving frequency selection for frequency domain simulation experiments. *Operations Research* 39(6): 917-924.

Janssen, F.B.S. 1998. *Inventory Management Systems: control and information issues*. CentER Dissertation, Tilburg University, The Netherlands.

Johnson, R.A., and D.W. Wichern. 1992. *Applied multivariate statistical analysis*. Prentice-Hall International, Englewood Cliffs, New Jersey.

Jothisankar, M.C., and H.P. Wang. 1993. Metamodelling a Just-In-Time Kanban system. *International Journal of Operations and Production Management* 13(8): 18-36.

Kalagnanam, J.R., and U.M. Diwekar. 1997. An efficient sampling technique for off-line quality control. *Technometrics* 39(3): 308-319.

Karaesmen, F., and Y. Dallery. 1998. A performance comparison of pull type control mechanisms for multi-stage manufacturing. *Proceedings of the tenth international working seminar in Production Economics*, Igls, Austria.

Kimball, G. 1988. General principles of inventory control. *Journal of Manufacturing and Operations Research* 1(1): 119-130.

Kleijnen, J.P.C. 1987. *Statistical tools for simulation practitioners*. Marcel Dekker, New York.

Kleijnen, J.P.C. 1998. Experimental design for sensitivity analysis, optimization, and validation of simulation models. In *Handbook of simulation*. Jerry Banks, ed. Wiley, New York.

Kleijnen, J.P.C., and W. Groenendaal. 1992. *Simulation: A statistical perspective*. John Wiley & Sons, New York.

Kleijnen, J.P.C., and J. Helton. 1998. Statistical analysis of scatter plots to identify important factors in large-scale simulations. *Reliability Engineering and Systems Safety* 65(2): 147-185.

Knight, F.H. 1971. *Risk uncertainty and profit*. University of Chicago Press, Chicago, IL.

Krajewski, L.J., B.E. King, L.P. Ritzman, and D.S. Wong. 1987. Kanban, MRP, and shaping the manufacturing environment. *Management Science* 33(1): 39-57.

Lambrecht, M., and A. Segaert. 1990. Buffer stock allocation in serial and assembly type production lines. *International Journal of Operations & Production Management* 10(2): 47-61.

Law, A.M., and W.D. Kelton. 1991. *Simulation modeling and analysis*. McGraw-Hill, New York.

Lee, Y.-J., and P. Zipkin. 1992. Tandem queues with planned inventories. *Operations Research* 40(5): 936-947.

Liberatore, G., S. Nicosia, and P. Valigi. 1996. Dynamic scheduling and kanban allocation in manufacturing systems. *Preprints of the IFAC '96 World Congress* B: 79-84, San Francisco, CA, July 1-5, 1996.

Liberopoulos, G., and Y. Dallery. 1997. A unified framework for pull control mechanisms in multi stage manufacturing systems. *Technical report*. Laboratoire d'Informatique de Paris 6 (LIP6-CNRS), Université Pierre et Marie Curie, Paris, France.

Lim, J.-M., K.-S. Kim, B.-J. Yum, and H. Hwang. 1996. Determination of an optimal configuration of operating policies for direct-input-output manufacturing systems using the Taguchi method. *Computers and Industrial Engineering* 31(3/4): 555-560.

Mayer, R.J., and P.C. Benjamin. 1992. Using the Taguchi paradigm for manufacturing system design using simulation experiments. *Computers and Industrial Engineering* 22(2): 195-209.

Meral, S., and N. Erkip. 1991. Simulation analysis of a JIT production line. *International Journal of Production Economics* 24: 147-156.

Michalewicz, Z. 1992. Genetic algorithms + data structures = evolution programs. Springer-Verlag.

Michalewicz, Z., D. Dasgupta, R. Le Riche, and M. Schoenauer. 1996. Evolutionary algorithms for constrained engineering problems. *Computers and Industrial Engineering* 30(4) : 851-870.

Mitchell, M. 1996. *An introduction to genetic algorithms*. The MIT press.

Moeeni, F., S.M. Sanchez, and A.J. Vakharia. 1997. A robust design methodology for Kanban system design. *International Journal of Production Research* 35(10): 2821-2838.

Monden, Y. 1993. *Toyota Production System*, 2nd edition, Norcross: Institute of Industrial Engineers.

Morgan, M.G., and M. Henrion. 1990. *Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis*. Cambridge University Press.

Nair, V.N. 1992. Taguchi's parameter design: a panel discussion. *Technometrics* 34(2): 127-161.

Norman, A.L., and D.W. Shimer. 1994. Risk, uncertainty, and complexity. *Journal of Economic Dynamics and Control* 18(1): 231-249.

Olhager, J., and B. Ostlund. 1990. An integrated push-pull manufacturing strategy. *European Journal of Operations Research* 45: 135-142.

Ou, J., and J. Jiang. 1997. Yield comparison of push and pull control methods on production systems with unreliable machines. *International Journal of Production Economics* 5011-12.

Palle, H. 1994. Overview of problems of risk management of accidents with dangerous chemicals in Europe. *European Journal of Operational Research* 75(3): 488-498.

Paris, J.-L., and H. Pierreval. 1997. Configuration of a multiproduct Kanban system using a distributed evolutionary algorithm. *Proceedings of the IFAC/IFIP Conference on Management and Control of Production and Logistics (MCPL'97)*. Las Campinas, Brazil, 31 Aug. - 3 Sept.

Pegden, C. D., R.E. Shannon, and R.P. Sadowski. 1991. *Introduction to Simulation using SIMAN / C*. McGraw-Hill, New York.

Philipoom, P.R., and T.D. Fry. 1992. Capacity-based order review/release strategies to improve manufacturing performance. *International Journal of Production Research* 30(11): 2559-2572.

Pierreval, H., and L. Tautou. 1997. Using evolutionary algorithms and simulation for the optimization of manufacturing systems. *IIE Transactions* 29(3): 181-189.

Pignatiello, J.J.Jr., and J.S. Ramberg. 1987. Discussion of performance measures independent of adjustment. *Technometrics* 29: 274-277.

Plackett, R.L., and J.P. Burman, 1946. The design of optimum multifactorial experiments. *Biometrica* 33: 305-325.

Powell, S.G., and D.F. Pyke. 1998. Buffering unbalanced assembly systems. *IIE Transactions* 30(1), 55-65.

Price, W., M. Gravel, and A.L. Nsakanda. 1994. A review of optimization models of Kanban-based production systems. *European Journal of Operational Research* 75: 1-12.

Rechenberg, I. 1965. *Cybernetic solution path of an experimental problem*. Royal Aircraft Establishment. Libr. transl. 1122. Farnborough, Hants., UK.

Rees, L.P., P.R. Philipoom, B.W. Taylor, and P.Y. Huang. 1987. Dynamically adjusting the number of kanbans in a Just-In-Time production system using estimated values of leadtime. *IIE Transactions* 19(2): 199-207.

Roderick, L.M., D.T. Phillips, and G.L. Hogg. 1992. A comparison of order release strategies in production control systems. *International Journal of Production Research* 30: 611-626.

Roderick, L.M., J. Toland, and F. Rodriguez. 1994. A simulation study of Conwip versus MRP at Westinghouse. *Computers and Industrial Engineering* 26: 137-142.

Sanchez, S.M., J.S. Ramberg, J. Fiero, and J.J.Jr. Pignatiello. 1993. Quality by design. In *Concurrent engineering: automation, tools, and techniques*, ed. Andrew Kusiak, 271-277. John Wiley & Sons, Inc.

Sanchez, S.M., P.J. Sanchez, J.S. Ramberg, and F. Moeeni. 1996. Effective engineering design through simulation. *International Transactions in Operational Research* 3(2): 169-185.

Sarker, B.R., and J.A. Fitzsimmons. 1989. The performance of push and pull systems: a simulation and comparative study. *International Journal of Production Research* 27(10): 1715-1731.

Sarker, B.R., and R.D. Harris. 1988. The effect of imbalance in a Just-In-Time production system: a simulation study. *International Journal of Production Research* 26(1): 1-18.

Savsar, M., and A. Al-Jawini. 1995. Simulation analysis of just-in-time production systems. *International Journal of Production Economics* 42: 67-78.

Schonberger, R.J. 1982. *Japanese Manufacturing Technique*. The Free Press, New York.

Schroer, B.J., J.T. Black, and S.X. Zhang. 1985. Just-In-Time (JIT), with Kanban, manufacturing system simulation on a microcomputer. *Simulation* 45(2): 62-70.

Schruben, L.W., and V.J. Cogliano. 1981. Simulation sensitivity analysis: a frequency domain approach. In *Proceedings of the 1981 Winter Simulation Conference*, 455-459. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sing, N., and J.K. Brar. 1992. Modelling and analysis of Just-In-Time manufacturing systems: A review. *International Journal of Operations & Production Management* 12: 3-14.

So, K.C., and S.C. Pinault. 1988. Allocating buffer storages in a pull system. *International Journal of Production Research* 26(12): 1959-1980.

Spearman, M.L., and M.A. Zazanis. 1992. Push and pull production systems: issues and comparisons. *Operations Research* 40(3): 521-532.

Spearman, M.L., D.L. Woodruff, and W.J. Hopp. 1990. Conwip: a pull alternative to Kanban. *International Journal of Production Research* 28(5): 879-894.

Spears, W.M., K.A. De Jong, T. Bäck, D. Fogel, and H. Degaris. 1993. An overview of evolutionary computation. *Proceedings of the European Conference on Machine Learning*. Springer, New-York, 442-459.

Sugimori, Y., K. Kusunoki, F. Cho, and S. Uchikawa. 1977. Toyota production system and Kanban system materialization of Just-In-Time and Respect-For-Human system. *International Journal of Production Research* 15(6): 553-564.

Swinehart, K.D., and J.H . Blackstone. 1991. Simulating a JIT/Kanban production system using GEMS. *Simulation* 57(4): 262-269.

Syrjakow, M., and H. Szczerbicka. 1994. Optimization of simulation models with REMO. *Proceedings of the Conference on Modelling and Simulation*. Guash, X.X., and Y.Y. Huber, eds., 274-281.

Taguchi, G. 1986. *Introduction to quality engineering: designing quality into products and processes*, White Plains, New York: Kraus International Publications.

Taguchi, G., and M.S. Phadke. 1984. Quality engineering through design optimization. In *Conference record of the 1984 IEEE Globecom Conference*, Atlanta, Georgia, (3): 1106-1113. New York: Institute of Electrical and Electronics Engineers.

Takahashi, K., and N. Nakamura. 1997. Comparing reactive Kanban and reactive Conwip. Proceedings of the world congress on systems simulation (WCSS'97). Teo, Y.M., Wong W.C., Oren T.I., and Rimane R, eds.

Tayur, S.R. 1993. Structural properties and a heuristic for Kanban-controlled serial lines. *Management Science* 39(11): 1347-1368.

Veatch, M.H., and L.M. Wein. 1994. Optimal control of a two-station tandem production / inventory system. *Operations Research* 42: 337-350.

Villeda, R., R. Dudek, and M.L. Smith. 1988. Increasing the production rate of a just-in-time production system with variable operation times. *International Journal of Production Research* 26: 1749-1768.

Wang, H., and H.-P. Wang. 1990. Determining the number of Kanbans: a step toward non-stock-production. *International Journal of Production Research* 28(11): 2101-2115.

Wild, R.H. and J.J.Jr. Pignatiello. 1991. An experimental design strategy for designing robust systems using discrete-event simulation, *Simulation*, 57(6): 358-368.

Wolfstetter, E. 1996. Stochastic dominance: theory and applications, *series Sonderforschungsbereich 373*, Humboldt Universitaet Berlin. Available online from http://econwpa.wustl.edu/WoPEc/data/Papers/wophumbsf960040.html [accessed June 16, 1998].

Yavuz, I.H., and A. Satir. 1995. A Kanban-based simulation study of a mixed model just-in-time manufacturing line. *International Journal of Production Research* 33(4): 1027-1048.

Zeigler, B.P. 1976. *Theory of modelling and simulation*. John Wiley & Sons, New York.

Zipkin, P. 1989. A Kanban like production control system: analysis of simple models. *Research Working Paper No. 89-1.* Graduate School of Business, Columbia University, New York.

Samenvatting (summary in Dutch)

Dit proefschrift behandelt het ontwerp van *pull* systemen voor productielijnen met één product, waarbij we ons onderzoek beperken tot productie op voorraad. Verder behandelen we alleen de goederenstroom binnen een bedrijf. We nemen dus aan dat grondstoffen en onderdelen continu en onbeperkt aangevoerd worden.

Het proefschrift is als volgt ingedeeld. In hoofdstuk 2 geven we een overzicht van de pull systemen die in de literatuur behandeld worden, waarbij we een nieuwe classificatie voorstellen. Hierbij komen twee vragen naar voren, namelijk welk pull systeem men moet kiezen en hoe de verschillende parameters van het gekozen systeem ingesteld moeten worden. We formuleren ons ontwerpprobleem op verschillende manieren, afhankelijk van de veronderstellingen die we bij het modelleren van de productieomgeving maken. (De productieomgeving wordt nader gedefinieerd in hoofdstuk 3 tot en met 6). In hoofdstuk 3 en 4 nemen we de productie-omgeving als gegeven aan, in hoofdstuk 5 en 6 onderzoeken we productieomgevingen die niet met zekerheid bekend zijn en dynamisch gedrag kunnen vertonen. Aan het eind van hoofdstuk 2 tonen we de noodzaak van nieuwe ontwerpbenaderingen aan.

In hoofdstuk 3 tonen we aan dat de keuze van een specifiek pull systeem een complex probleem is dat in de literatuur niet bestudeerd is. Onze bijdrage is het ontwerp van een generiek model, waarin alle in hoofdstuk 2 beschreven modellen gerepresenteerd kunnen worden. Om het generieke systeem toe te passen op een gegeven productiesysteem en een gegeven productieomgeving, stellen we een methode voor die gebaseerd is op evolutionaire berekening en simulatie; we noemen deze methode *customization* ofwel maatwerk. Het resultaat geeft aan welk pull systeem moet worden geïmplementeerd. De voordelen van dit maatwerk worden getoond in een productiesysteem dat aan de literatuur is ontleend. Voor dit systeem zijn de optimale configuraties voor verschillende pull-systemen al eens bepaald. We hebben echter een pull-systeem ontdekt dat significant betere resultaten oplevert dan het beste tot nu toe beschreven systeem.

In hoofdstuk 4 vergroten we ons inzicht in customization en de voordelen daarvan door onze methode op verschillende soorten productielijnen toe te passen. We bepalen deze soorten op grond van de literatuur, en gebruiken proefopzetten om een steekproef van twaalf configuraties van productielijnen te genereren. Op elke configuratie passen we de in hoofdstuk 3 beschreven methode toe. Uit de resultaten is een aantal conclusies te trekken met betrekking tot het resultaat en de complexiteit van verschillende pull-structuren.

In hoofdstuk 5 wijzen we drie oorzaken van onzekerheid aan die bij het ontwerp van pull-systemen met behulp van simulatie kunnen ontstaan: (i) stochastische onzekerheid als gevolg van het gebruik van (pseudo)toevalsgetallen in onze discrete simulatie, (ii) subjectieve onzekerheid als gevolg van de noodzaak om stochastisch gedrag te modelleren met behulp van kansverdelingen die op steekproeven of meningen van experts gebaseerd zijn en (iii) dynamische onzekerheid als gevolg van veranderingen in de productieomgeving

in de loop van de tijd. Met een aantal eenvoudige voorbeelden illustreren we de mogelijke gevolgen van deze drie oorzaken van onzekerheid, en we leggen de nadruk op de noodzaak om het effect van deze onzekerheden te bepalen en in het ontwerpproces op te nemen. Hieraan leveren we een bijdrage met een nieuwe methode die gebaseerd is op onzekerheidsanalyse (*Uncertainty/Risk Analysis*: URA) en Taguchi's robuuste ontwerp.

In hoofdstuk 6 passen we onze methode toe op het ontwerp van pull-systemen onder onzekerheid. We geven twee criteria voor robuustheid, één die gebaseerd is op de service-graad en één die gebaseerd is op de hoeveelheid onderhanden werk (*Work in Progress* : WIP), en we geven een exacte definitie van het robuuste maatwerk probleem. Vervolgens bekijken we hoe de robuustheid van pull-systemen vergeleken moet worden. We bestuderen de relatieve prestaties van vier pull-systemen met twee vergelijkingsmethoden, namelijk stochastische dominantie en betrouwbaarheidsellipsoiden die met bootstrapping geconstrueerd zijn. We komen tot de conclusie dat een beheersingssysteem alleen kan worden gekozen als managers hun houding tegenover risico bekend maken en hun voorkeuren opgeven. Om managers houvast te geven onderzoeken we het effect van de parameters die zij kunnen beheersen (het aantal kanbans, de kansverdelingen die in URA gebruikt worden en de parameters met betrekking tot hun houding tegenover risico en hun voorkeuren). De volledige methode voor robuust maatwerk passen we toe op het productiesysteem dat door Bonvik *et al.* (1997) bestudeerd is.

Hoofdstuk 7 bevat een samenvatting van de belangrijkste conclusies van dit proefschrift en richtlijnen voor verder onderzoek.

Résumé (summary in French)

L'objet de cette thèse est la conception de systèmes gestion en flux tiré pour des lignes production produisant un seul type de pièces. Nous nous intéressons plus particulièrement à des systèmes produisant pour stock. La principale hypothèse que nous faisons est de considérer uniquement les flux de production internes aux lignes, c'est à dire que l'approvisionnement en matières premières et composants est continu et infini.

La thèse se décompose de la façon suivante. Dans le Chapitre 2, nous passons en revue les types de gestion en flux tirés développés dans la littérature et nous proposons un nouvelle classification. Deux problèmes se posent : quel type de gestion choisi, puis comment régler les divers paramètres de la gestion choisie ? Cette problèmatique de conception est au coeur de cette thèse. Nous distinguons plusieurs formulation de notre problèmatique selon les hypothèse faite quant à la modélisation de l'environnement de production (que nous définissons plus précisemment dans les Chapitres 3 à 6 : dans les Chapitres 3 et 4 nous considérons l'environnement de production comme étant donné, alors que dans les Chapitres 5 et 6 nous étudions des environnements qui ne sont pas connus avec certitude et peuvent avoir un comportement dynamique). Nous concluons le Chapitre 2 en montrant la nécessité de développer de nouvelles approches de conception.

Dans le Chapitre 3, nous montrons que sélectionner un système de gestion en flux tirés parmi toutes les possibilités, est un problème complexe qui n'a pas été étudié dans la littérature. Notre contribution à ce problème réside dans la conception d'un modèle générique, c'est à dire une représentation commune à l'ensemble des gestions en flux tirés identifiés au Chapitre 2. Nous proposons un procédure basée sur l'algorithmique évolutionniste et la simulation afin de configurer notre système générique pour une ligne de production et un environnement donnés ; nous appelons cette procédure *conception sur mesure*. Le résultat de cette procédure quel type de gestion en flux tirés doit être mise en place. Les avantages de notre conception sur mesure sont illstrés par un example tiré de la littérature, pour lequel les configurations optimales de plusieurs types de gestion ont déjà été déterminés : nous aboutissons à un système dont les performances sont significativement meilleures que le meilleur système de la littérature.

Dans le Chapitre 4, nous analysons plus en détail le principe de conception sur mesure et ses avantages en appliquant notre méthodologie à un échantillon de lignes de production. Nous construisons cet échantillon à partir d'une synthèse des lignes étudiées dans la littérature et nous utilisons la technique des plan d'expérience pour générer douze configurations de lignes de production. Pour chacune de ces lignes, nous appliquons la méthode de conception sur mesure présentée au Chapitre 3. Les résultats fournissent de précieuses conclusions quant à la structures des meilleurs systèmes de gestion en flux tirés, leur performance et leur complexité.

Dans le Chapitre 5, nous identifions trois sources d'incertitude pouvant apparaître lors de la conception par simulation de systèmes de gestion de production en flux tirés : (i)

incertitude stochastique, causée par l'utilisation de nombres (pseudo)aléatoires dans nos simulations à événements discrets, (ii) incertitude subjective, qui résulte de la nécessité de modéliser les comportements stochastiques par des distributions de probabilités basée sur des échantillon de données ou des opinions d'experts, et (iii) incertitude dynamique, due aux fluctuations de l'environnement de production au cours du temps. Au travers d'exemple simples, nous illustrons les effets éventuels de ces trois source d'incertitudes et nous soulignons la nécessité d'évaluer et d'intégrer ces effets dès la phase de conception. Notre contribution à ce problème est le développement d'une procédure basée sur les techniques d'analyse de risques/incertitudes et de conception robuste (Taguchi).

Dans le Chapitre 6, nous appliquons notre procédure à la conception de systèmes gérés en flux tirés sous incertitude. Deux critères de robustesse sont spécifiés – l'un basé sur la qualité de service et l'autre sur la quantité d'en-cours – et nous donnons une définition rigoureuse du problème de conception robuste sur mesure. Nous nous attachons alors à comparer la robustesse de quatre systèmes gérés en flux tirés. Pour cela, nous utilisons deux procédure de comparaison : dominance stochastique d'une part et ellipsoïdes de confiance construits par bootstrapping. Nous concluons que le choix d'un système géré en flux tirés selon des critères de robustesse ne peut se faire que si les décideurs spécifient leur attitude vis à vis des risques et caractérisent leurs préférence. Afin d'aider les décideurs dans leur choix, nous étudions l'effet des divers paramètres qu'ils contrôlent (les nombres de cartes, le type de distributions de probabilités utilisés dans l'analyse de risques/incertitudes, et plusieurs paramètres qui permettent de spécifier l'attitude des décideurs vis à vis des risques et de caractériser leurs préférence). Nous appliquons la procédure de conception robuste sur mesure complète à un exemple de système de production traité dans Bonvik *et al.* (1997).

Dans le Chapitre 7 nous résumons les principales conclusions de cette thèse et nous donnons plusieurs perspectives de recherche.

Center for Economic Research, Tilburg University, The Netherlands
Dissertation Series

No.	Author	Title	Published
1	P.J.J. Herings	Static and Dynamic Aspects of General Disequilibrium Theory; ISBN 90 5668 001 3	June 1995
2*	Erwin van der Krabben	Urban Dynamics: A Real Estate Perspective - An institutional analysis of the production of the built environment; ISBN 90 5170 390 2	August 1995
3	Arjan Lejour	Integrating or Desintegrating Welfare States? - a qualitative study to the consequences of economic integration on social insurance; ISBN 90 5668 003 X	Sept. 1995
4	Bas J.M. Werker	Statistical Methods in Financial Econometrics; ISBN 90 5668 002 1	Sept. 1995
5	Rudy Douven	Policy Coordination and Convergence in the EU; ISBN 90 5668 004 8	Sept. 1995
6	Arie J.T.M. Weeren	Coordination in Hierarchical Control; ISBN 90 5668 006 4	Sept. 1995
7	Herbert Hamers	Sequencing and Delivery Situations: a Game Theoretic Approach; ISBN 90 5668 005 6	Sept. 1995
8	Annemarie ter Veer	Strategic Decision Making in Politics; ISBN 90 5668 007 2	October 1995
9	Zaifu Yang	Simplicial Fixed Point Algorithms and Applications; ISBN 90 5668 008 0	January 1996
10	William Verkooijen	Neural Networks in Economic Modelling - An Empirical Study; ISBN 90 5668 010 2	February 1996
11	Henny Romijn	Acquisition of Technological Capability in Small Firms in Developing Countries; ISBN 90 5668 009 9	March 1996
12	W.B. van den Hout	The Power-Series Algorithm - A Numerical Approach to Markov Processes; ISBN 90 5668 011 0	March 1996
13	Paul W.J. de Bijl	Essays in Industrial Organization and Management Strategy; ISBN 90 5668 012 9	April 1996

No.	Author	Title	Published
14	Martijn van de Ven	Intergenerational Redistribution in Representative Democracies; ISBN 90 5668 013 7	May 1996
15	Eline van der Heijden	Altruism, Fairness and Public Pensions: An Investigation of Survey and Experimental Data; ISBN 90 5668 014 5	May 1996
16	H.M. Webers	Competition in Spatial Location Models; ISBN 90 5668 015 3	June 1996
17	Jan Bouckaert	Essays in Competition with Product Differentiation and Bargaining in Markets; ISBN 90 5668 016 1	June 1996
18	Zafar Iqbal	Three-Gap Analysis of Structural Adjustment in Pakistan; ISBN 90 5668 017 X	Sept. 1996
19	Jimmy Miller	A Treatise on Labour: A Matching-Model Analysis of Labour-Market Programmes; ISBN 90 5668 018 8	Sept. 1996
20	Edwin van Dam	Graphs with Few Eigenvalues - An interplay between combinatorics and algebra; ISBN 90 5668 019 6	October 1996
21	Henk Oosterhout	Takeover Barriers: the good, the bad, and the ugly; ISBN 90 5668 020 X	Nov. 1996
22	Jan Lemmen	Financial Integration in the European Union: Measurement and Determination; ISBN 90 5668 021 8	December 1996
23	Chris van Raalte	Market Formation and Market Selection; ISBN 90 5668 022 6	December 1996
24	Bas van Aarle	Essays on Monetary and Fiscal Policy Interaction: Applications to EMU and Eastern Europe; ISBN 90 5668 023 4	December 1996
25	Francis Y. Kumah	Common Stochastic Trends and Policy Shocks in the Open Economy: Empirical Essays in International Finance and Monetary Policy; ISBN 90 5668 024 2	May 1997
26	Erik Canton	Economic Growth and Business Cycles; ISBN 90 5668 025 0	Sept. 1997

No.	Author	Title	Published
27	Jeroen Hoppenbrouwers	Conceptual Modeling and the Lexicon; ISBN 90 5668 027 7	October 1997
28	Paul Smit	Numerical Analysis of Eigenvalue Algorithms Based on Subspace Iterations; ISBN 90 5668 026 9	October 1997
29	Uri Gneezy	Essays in Behavioral Economics; ISBN 90 5668 028 5	October 1997
30	Erwin Charlier	Limited Dependent Variable Models for Panel Data; ISBN 90 5668 029 3	Nov. 1997
31	Rob Euwals	Empirical Studies on Individual Labour Market Behaviour; ISBN 90 5668 030 7	December 1997
32	Anurag N. Banerjee	The Sensitivity of Estimates, Inferences, and Forecasts of Linear Models; ISBN 90 5668 031 5	December 1997
33	Frans A. de Roon	Essays on Testing for Spanning and on Modeling Futures Risk Premia; ISBN 90 5668 032 3	December 1997
34	Xiangzhu Han	Product Differentiation, Collusion and Standardization; ISBN 90 5668 033 1	January 1998
35	Marcel Das	On Income Expectations and Other Subjective Data: A Micro-Econometric Analysis; ISBN 90 5668 034 X	January 1998
36	Jeroen Suijs	Cooperative Decision Making in a Stochastic Environment; ISBN 90 5668 035 8	March 1998
37	Talitha Feenstra	Environmental Policy Instruments and International Rivalry: A Dynamic Analysis; ISBN 90 5668 036 6	May 1998
38	Jan Bouwens	The Use of Management Accounting Systems in Functionally Differentiated Organizations; ISBN 90 5668 037 4	June 1998
39	Stefan Hochguertel	Households' Portfolio Choices; ISBN 90 5668 038 2	June 1998
40	Henk van Houtum	The Development of Cross-Border Economic Relations; ISBN 90 5668 039 0	July 1998
41	Jorg Jansen	Service and Inventory Models subject to a Delay-Limit; ISBN 90 5668 040 4	Sept. 1998

No.	Author	Title	Published
42	F.B.S.L.P. Janssen	Inventory Management Systems: control and information issues; ISBN 90 5668 041 2	Sept.1998
43	Henri L.F. de Groot	Economic Growth, Sectoral Structure and Unemployment; ISBN 90 5668 042 0	October 1998
44	Jenke R. ter Horst	Longitudinal Analysis of Mutual Fund Performance; ISBN 90 5668 043 9	Nov. 1998
45	Marco Hoeberichts	The Design of Monetary Institutions; ISBN 90 5668 044 7	December 1998
46	Adriaan Kalwij	Household Consumption, Female Employment and Fertility Decisions: A microeconometric analysis; ISBN 90 5668 045 5	February 1999
47	Ursula Glunk	Realizing High Performance on Multiple Stakeholder Domains: A Resource-based Analysis of Professional Service Firms in the Netherlands and Germany; ISBN 90 5668 046 3	April 1999
48	Freek Vermeulen	Shifting Ground: Studies on the Intersection of Organizational Expansion, Internationalization, and Learning; ISBN 90 5668 047 1	February 1999
49	Haoran Pan	Competitive Pressures on Income Distribution in China; ISBN 90 5668 048 X	March 1999
50	Laurence van Lent	Incomplete Contracting Theory in Empirical Accounting Research; ISBN 90 5668 049 8	April 1999
51	Rob Aalbers	On the Implications of Thresholds for Economic Science and Environmental Policy; ISBN 90 5668 050 1	May 1999
52	Abe de Jong	An Empirical Analysis of Capital Structure Decisions in Dutch Firms; ISBN 90 5668 051 X	June 1999
53	Trea Aldershof	Female Labor Supply and Housing Decisions; ISBN 90 5668 052 8	June 1999
54	Jan Fidrmuc	The Political Economy of Reforms in Central and Eastern Europe; ISBN 90 5668 053 6	June 1999
55	Michael Kosfeld	Individual Decision-Making and Social Interaction; ISBN 90 5668 054 4	June 1999

No.	Author	Title	Published
56	Aldo de Moor	Empowering Communities - A method for the legitimate user-driven specification of network information systems; ISBN 90 5668 055 2	October 1999
57	Yohane A. Khamfula	Essays on Exchange Rate Policy in Developing Countries; ISBN 90 5668 056 0	Sept. 1999
58	Maurice Koster	Cost Sharing in Production Situations and Network Exploitation; ISBN 90 5668 057 9	December 1999
59	Miguel Rosellon Cifuentes	Essays on Financial Policy, Liquidation Values and Product Markets; ISBN 90 5668 058 7	December 1999
60	Sharon Schalk	Equilibrium Theory: A salient approach; ISBN 90 5668 059 5	December 1999
61	Mark Voorneveld	Potential Games and Interactive Decisions with Multiple Criteria; ISBN 90 5668 060 9	December 1999
62	Edward Droste	Adaptive Behavior in Economic and Social Environments; ISBN 90 5668 061 7	December 1999
63	Jos Jansen	Essays on Incentives in Regulation and Innovation; ISBN 90 5668 062 5	January 2000
64	Franc J.G.M. Klaassen	Exchange Rates and their Effects on International Trade; ISBN 90 5668 063 3	January 2000
65	Radislav Semenov	Cross-country Differences in Economic Governance: Culture as a major explanatory factor; ISBN 90 5668 065 X	February 2000
66	Alexandre Possajennikov	Learning and Evolution in Games and Oligopoly Models; ISBN 90 5668 064 1	March 2000
67	Eric Gaury	Designing Pull Production Control Systems: Customization and Robustness; ISBN 90 5668 066 8	March 2000

* Copies can be ordered from ThelaThesis,Prinseneiland 305, 1013 LP Amsterdam, The Netherlands, phone + 31 20 6255429; fax: +31 20 6203395; e-mail: office@thelathesis.nl

Eric Gaury graduated in mechanical engineering from the French Institute of Advanced Mechanical Engineering (IFMA) in Clermont-Ferrand, France, in 1997. He completed a master's degree in computer science at Blaise Pascal University (UBP), Clermont-Ferrand, in 1997. He carried out his Ph.D. research both in the operations research group at the Center for Economic Research (CentER), Tilburg University, and in the IFMA research group on manufacturing systems at the Laboratory of Computer Science for Modeling and Optimization of Systems (LIMOS). Since September 1999 he has a position as an engineer at Renault automobile company, where he is in charge of methodological issues in simulation.

In this dissertation we address the issues of selecting and configuring pull production control systems for single-product flowlines. We start with a review of pull systems in the literature, yielding a new classification. Then we propose a novel selection procedure based on a generic system that we test on a case also studied in the literature. We further study our procedure for a variety of twelve production lines. We find new types of pull systems that perform well. Next, we raise the issue of designing pull systems under uncertainty. We propose a novel procedure to minimize the risk of poor performance. Results show that risk considerations strongly influence the selection of a specific pull system.

Cette dissertation s'intéresse aux problèmes de sélection et configuration de systèmes de gestion en flux tirés pour des lignes de production mono-produit. Nous commençons par une revue des systèmes à flux tirés dans la littérature aboutissant à une nouvelle classification. Nous proposons alors une procédure de sélection originale basée sur un modèle générique que nous testons pour un cas étudié dans la littérature. Nous continuons l'analyse de notre procédure grâce à un échantillon de douze lignes de production. Nous aboutissons à de nouveaux types de systèmes à flux tirés qui permettent d'atteindre de hauts niveaux de performance. Ensuite, nous posons le problème de concevoir des systèmes à flux tirés sous incertitude. Nous proposons une nouvelle procédure avec pour objectif de minimiser le risque de mauvaise performance. Les résultats montrent que tenir compte de ce risque a une influence forte sur le choix d'un système à flux tiré spécifique.

ISBN: 90 5668 066 8