A JIT IMPLEMENTATION USING HYBRID KANBAN-CONWIP MODELING APPROACH

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

IN INDUSTRIAL SYSTEMS ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF REGINA

By

Amod S. Kelkar

Regina, Saskatchewan

June, 1999

0-612-45329-4

Abstract

Because of the efforts of today's industries to continuously reduce every possible production cost, various new approaches have been developed as possible solutions. Just-In-Time, or JIT as is popularly known in the industrial world, is indeed a proven technique to achieve substantial savings on inventory, purchase and production cost incurred by manufacturers all over the world. However, a careful understanding of the JIT philosophy is required to assess its suitability to a particular production setup. Considerable investment of time and study is necessary to select a proper JIT scheme. Also, it's possible that every sample setup may not reap profits from JIT implementation sufficient enough to make the investment justifiable.

The present study is based on a real life scenario in a leading Canadian agricultural equipment manufacturing plant. The system is studied in detail as regard to the plant operation with all the existing constraints. A careful study is made of all the available JIT methods from the applicability point of view for the plant in consideration. A new hybrid policy is developed specially for this plant using the two most proven JIT techniques. A mathematical model along with the supporting application algorithms is constructed for JIT implementation. The objective of the model is to minimize the production and inventory costs at the said plant. Calculations and analysis are carried out with the real plant data using Mixed Integer Linear Programming (MILP) approach. Results obtained are reviewed and discussed assuring the success of the said hybrid methodology. Also, some other possible solution approaches are discussed for a similar kind of cost minimization plan, eventually revealing that the hybrid model is the most appropriate approach for the problem in consideration.

Acknowledgements

I am greatly indebted to Dr. M. Chen for his invaluable guidance, encouragement and financial support during the different phases of the coursework, research progress and thesis preparation.

I would like to thank my external examiner, Dr. James Mason, Department of Administration, University of Regina, for his constructive suggestions. I also thank my committee members, Dr. Y-C Jin and Dr. Bruce Cooke, for their valuable comments and suggestions. I am truly indebted to Dr. S. Sankaran for his advice throughout the research process. I must thank the staff at Brandt Industries, Regina and especially Mr. Don Vass and Mr. Jason Litowitz for all the information support they provided making this study truly comprehensive. I also thank Mr. Robert Jones, Robotics lab, University of Regina for all his help during the computation phase of this research.

I wish to express my sincere respect and gratitude towards Mrs. Leena Bhole and Dr. S. D. Bhole for all the support and encouragement they have been bestowing upon me throughout my stay in Regina. I thank my parents and brother Abhijeet for always being there for me. I thank Nandini for all her caring and my wonderful College West office team of Judy, Beverly and Pat for their cooperation during my work and stay at the CW residence. Heartening support and love from friends like Annette Pritchard, Leila Doyle, Shilpesh Katragada, Shiv Nagrajan and Wen Zhang is highly appreciated.

The financial help received from NSERC (Natural Sciences and Engineering Research Council of Canada) through Dr. Chen's research grant and from the Faculty of Graduate Studies and Research, University of Regina as well as the Faculty of Engineering, University of Regina is gratefully acknowledged.

Nomenclature

Abbreviations: The following is a summary of keyword abbreviations in the Thesis text.

BOM	Bill of Material
COPT	Combinatorial Optimization
CONWIP	Constant Work in Process
FCFS	First Come First Served
JIT	Just in Time
LPS	Lean Production Systems
MAN	Material As Needed
MILP	Mixed Integer Linear Programming
MIPS	Minimum Inventory Production Systems
MIS	Management Information System
MPS	Main Production Schedule
MRP	Material Requirement Planning
MRP II	Manufacturing Resource Planning
MVA	Mean Value Analysis
NLP	Non-Linear Programming
OPT	Optimized Production Technology
PSA	Paint Schedule Algorithm
TOC	Theory of Constraints
WIP	Work in Process

Symbols: The symbols used for the mathematical models are mentioned in chapter 4 along with the model formulations.

Contents

List of Tables

List of Figures

Chapter One

Introduction

To survive in the immense competition profitably, every mature industry has to strive hard for cost minimization. The saving potential should be explored in each possible area like production, purchasing, servicing and so on. The choice of the most effective technique for a given facility can be made through a wide variety of available methods or some specially developed policies like the one presented in this research.

Production systems can broadly be classified into two types based on the control techniques adopted: Push and Pull systems. The traditional Material Requirement Planning (MRP) and the recent Manufacturing Resource Planning (MRP II) represent the Push type of production control. On the other hand, JIT is an effective and proven Pull type of production control system.

The difference between Pull type manufacturing systems and Push type manufacturing systems is the difference between producing to order and producing to schedule as pointed out by researchers. In a pull management, upstream activities are geared to match the final assembly needs. When all component parts and materials are pulled through production in an exact correspondence to end-item demands, the theoretical ideal of 'stockless production' is achieved.

The Push type production control will run as per the predetermined Master Production Schedule (MPS). The lead times for all the products and for all their operations are known and are used as a basis for the MPS. Work orders are issued based on this schedule for all the production time-frame in consideration and then flow of production is rigorously followed up to ensure the timely completion.

A brief but clear explanation of all these terms is necessary before proceeding to the literature survey of research made on them.

1.1 Push Type Production System

As explained earlier, this system works on MPS and a continuous updating of the central computer database is carried out for each activity completed. As a result, quick and easy tracking of job progress can be done from any user terminal in the plant.

In this system, plant work in process (WIP) inventory is used as a means of absorbing uncertainties in processes and the changes in the demand. In practice, however, as one may find out, this system often creates one or both of the these problems (Singh, 1996):

1. It may lead to starvation and excessive stocks simultaneously at the different stages because of the imbalances of stocks between various stages.
2. It may lead to conditions where, manufacturer employs excessive capacities of equipment and/or manpower.

Even with these problems, it's considered robust and conservative when compared with Pull type of production in some aspects like provision for buffer stock, availability of commercial user-friendly software's, savings on investments of designing Pull type setups etc. Now follows the popular Push type production methods.

1.1.1 Manufacturing Requirement Planning (MRP)

The basic principles behind MRP have been used for many years. But, because of the high capacity computers required for calculations in MRP, these principles could not be exploited fully. MRP combines inventory with production planning. This coupling, if done manually, would take a long time because of the complex scheduling procedures and data updating being so dynamic in nature. But, once computer usage started, MRP became the ideal example for Management Information Systems (MIS). Thus MRP is sometimes called as a functional balance between sales orders, MPS, bills of materials (BOM), lead times on one side and purchase orders, WIP and inventory on the other side. Then, this balance will help in making manufacturing decisions like whether 'to make or buy?' or 'to ship or store?' or 'whether overall capacity shortage or excess capacity?' (Riggs, 1987).

The mechanics of MRP start with breaking the time frame into time buckets and exploding BOM into various levels. Then aggregate capacity calculations are balanced vis-à-vis occupied resource times by all components' demand quantities (total demand – stock in hand) in all the time buckets. The scheduled receipts from the suppliers are also considered to fit into the overall time horizon. The final schedule is obtained satisfying all of these constraints, and it might show some excess capacity in some time buckets or might ask for overtime and/or sub-contracting.

1.1.2 Manufacturing Resource Planning (MRP II)

The expansion of MRP to include more control apart from only planning was a next step for the manufacturing engineers. MPS is coupled with other dependent requirements like machine hours, labor hours and capital. Shop floor progress and vendor information are also considered to finalize the output quantity and time. All departments can have access to MRP-II database and few changes in manufacturing practices are enhanced in it. Commercial MRP-II softwares are composed of a manufacturing part (bills of materials, part routings etc.) and a control part for the production process. It's rightly called a closed loop MRP system.

1.2 Pull Type Production Systems

This is a simplified control technique, which is designed to respond quickly to the demand changes. It needs minimized record keeping and simple methods. The downstream machines pull the production from the upstream ones based on the demands created at their output buffers. Some kind of signal is sent (in the form of kanban cards, containers, tags etc.) to the upstream machines to indicate the demand for a particular component. So, the product flow and information flow, are in opposite directions to each other, the prior ones being in the forward direction.

The increased responsiveness of this system to the changes in product demand makes it a better choice in certain cases over the Push type of production. In Pull type of production control, MPS is used only as a broad outline of the requirements for resources at the different work centers. The major difference with Push type, as regards the usage of MPS lies in the fact that MPS is used for this broad outlining and not for the individual workstation' production rate. As described in the previous paragraph, the built schedules move in the backward direction with the help of some form of signals. The JIT philosophy is the original basis of this system.

1.2.1 Just-In-Time Production System

There is a very popular 'water-level in a lake' analogy for describing the purpose and functioning of JIT systems (Martinich, 1997). The same is used here to clarify the point. The whole production setup is considered as a lake in which smooth production flow is like navigating a small boat. The intention is obviously to travel without any accidents to the boat. Shop inventory is thought of as water in the lake covering the lake's bottom made of dangerously sharp rocks (which are shortfalls or weaknesses in the given production setup). Now, it is logical to think that lowering the water level will expose the rocks (problems) making it easier to rectify them and thus in turn, save the boat from dashing against them.

Apparently, in their attempt to save the boat, managers of the industries have been using high water levels (big batch sizes creating safety stocks) in order to cover the rocks, (problems like unexpected machine failures, absenteeism, severe demand fluctuations etc.) instead of using the above mentioned logical approach.

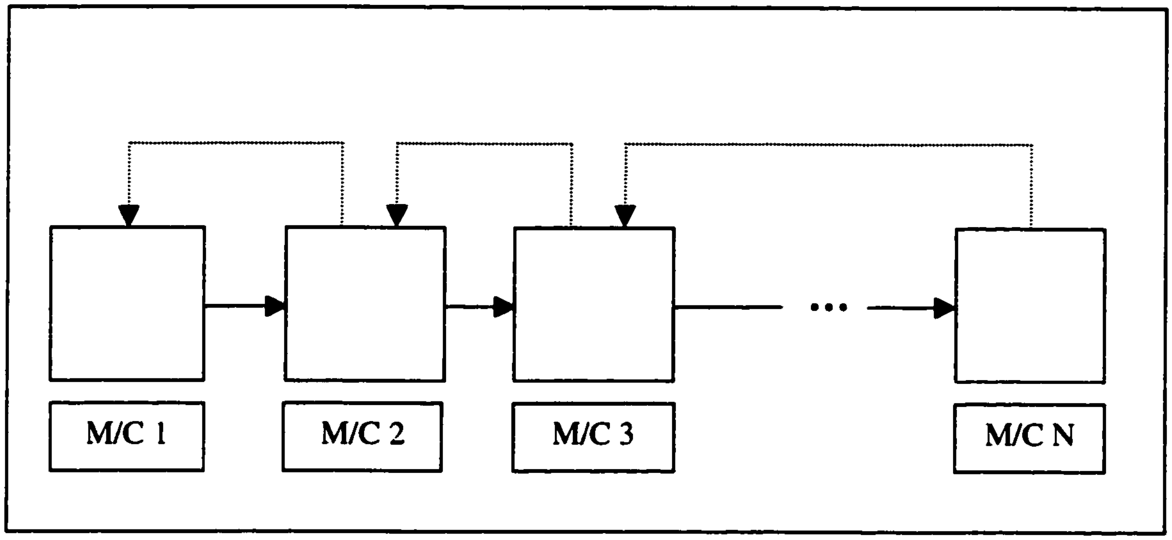
The JIT philosophy aims at reducing the production batch sizes to expose the production problems. Producing goods 'just in time', eliminates the need for buffer stocks. "Making (or buying in case of JIT purchases) right quantity at the right time at the right place" is another way of defining the JIT systems. When compared with the MRP-II system, which emphasizes on planning to anticipate problems, JIT stresses at execution to identify problems. To achieve low-cost, high quality and on-time production, this system removes stock accumulations between various production stages. The working is based on defect-free production concept. Any job rejection is immediately compensated by the prompt entry of the new component of that type. Fluctuations in daily quota are minimized by principles of 'production smoothing'. The results have frequently been spectacular. Some of them are explained briefly in Chapter Two, under the heading 'some success stories'. There are varieties of methods used to implement JIT, which can be summarized as followed in the coming pages.

Kanban System

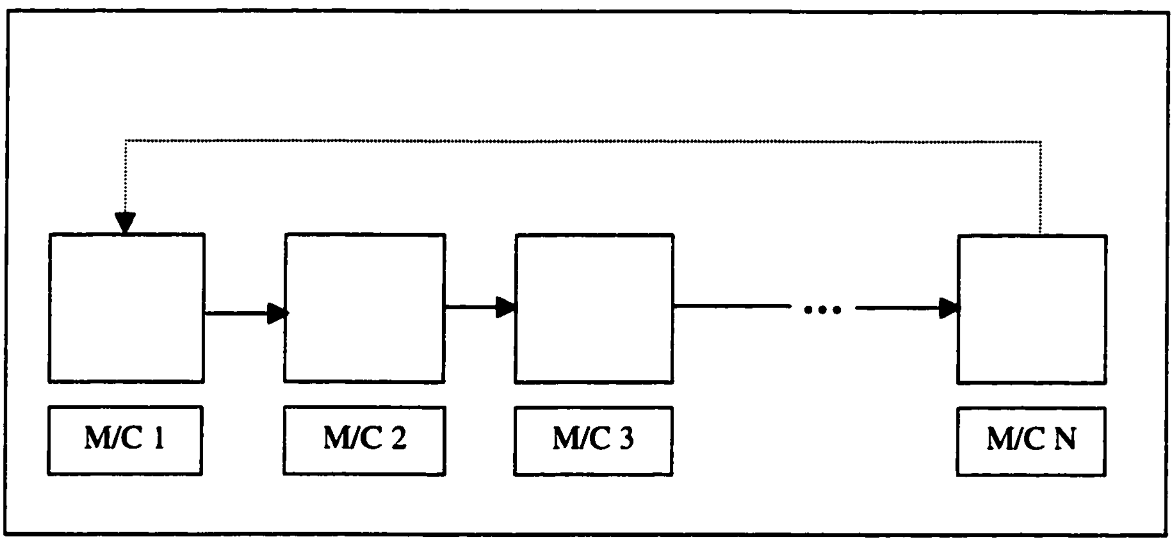
It's the original JIT model, which was developed by Toyota motor company in the 1950's. The Japanese word 'kanban' means visible record. There are mainly two types of kanbans used in Toyota system, namely, Movement kanban and Production kanban.

It is basically a signal (in the form of cards, tags or lights) operated Pull production philosophy. Production on every machine in a kanban system is triggered by a demand signal from its downstream machine. Thus, the originator of signaling is the final station (which is generally assembly) and the information flows backwards to the first machine, as shown in Figure 1.1. A card travels with the container and typically is marked with kanban identification number, job number, job description and place of issue with quantity of jobs per standard container. Thus, cards are substituting for the computers in MRP system for material flow tracking. The number of containers in shop, circulating at any time controls the total amount of WIP. A user from downstream machine is not authorized to move any container from an upstream machine unless it has got a conveyance card attached. Similarly, he is not allowed to produce any quantities until he has a raw material container with production card attached to it.

In addition to the above two main types of kanbans, an *Express kanban* is used when shortages of parts occur. It's immediately taken off the system after the priority production is done. The *Emergency kanban* is used as a temporary measure to make up for the defective units and other uncertainties such as resource failures or demand fluctuations. The *Through kanban* is used as a common card between two adjacent work-centers where there is no need to exchange the cards because of their physical vicinity.



Kanban type of Pull Production Control



CONWIP type of Pull Production Control

Figure 1.1 Kanban and CONWIP production systems

CONWIP System

All jobs on the constant work in process (CONWIP) line follow the same sequence of operations (Spearman et al., 1989). There is a waiting group of jobs at the entry buffer of the first machine in the line. Scheduling of the line for the most optimum costs is executed. There are a fixed number of containers moving in the line (CON WIP). The jobs enter the line as per the optimum schedule. When the preset WIP level is reached, no new job is allowed to enter the line unless some job leaves the line. Thus, job scheduling is done for the shop conditions rather than the times to enter the line. Obviously, forecasting throughput and cycle time is very important for the success of a CONWIP line. The first machine in CONWIP line triggers the production. The components flow thereafter throughout the CONWIP line on the first come first served discipline (FCFS). Hence compared with kanban, card movements at all stages are not needed as every machine is always authorized to process the incoming component on its line. In CONWIP, the finished goods buffer is always full and there are no internal inventory buffers as in kanban. This is because any part entering the system will go through its process cycle on FCFS basis. CONWIP has outperformed kanban in situations where shops are controlled by job orders. Also, in case of a machine failure, CONWIP performs better because of its working logic, in handling inventory pileup. If this happens, the jobs on the downstream side of the failed machine will progress until past the finishing process and upstream WIP will be stored at the machine's upstream buffer. Thus upstream production will automatically stop if the breakdown continues for a long time, thus avoiding inventory pileup. It is a kanban philosophy applied to a group of machines considered as a single kanban cell.

Some Other Pull Production Systems

There are some other JIT techniques developed by researchers. They are explained in short in the following text.

Drumbuffer System It is a scheduling method developed by Goldratt (1988), in which the flow of production is balanced in such a way that bottleneck stages are fully utilized and other stages are forced to produce in synchronization with them. The mechanism forces them to produce at the pace of the bottlenecks aiming at Synchronous Production.

Basestock System The basestock control limits the amount of inventory between each production stage and the demand process (Kimball, 1988). Each machine tries to maintain a certain amount of material in its output buffer, subtracting backlogged finished goods demand, if any. This amount is called basestock level of the machine. To operate, the end process transfers demand information to all the machines for which there are separate cards. Hence, if there is only one machine, it is same as kanban.

Periodic Pull System In this system, manual information processing time of a kanban method is replaced by online computerized processing. This results in reduced lead times and inventory and faster system response (Kim, 1985).

Long Pull System The triggering mechanism for this method works in the same way as that for a Pull system. However, the control of this method encompasses more than one workstation. In this system, one unit is allowed to enter the system at the same time one unit is pulled at the end of the pull line. The individual buffer capacities are not restricted, but the total number of units in the span of a long pull is limited. Once the job enters at the first machine in the span (after signaled by the last machine), it is pushed through the middle stages. Lambrecht and Segaert (1990) developed this scheme.

1.3 Organization of Thesis

Chapter 1 covered the introduction to various Push and Pull type of production control techniques regarding their individual function styles and differences with others. Chapter 2 is devoted to the literature survey on MRP and various JIT approaches developed so far. A review of research in the field is made with the help of citations from the relevant papers and models. The attempt is to cover the latest published materials in these areas. Chapter 3 describes the actual production environment under consideration. The plant structure is explained with all the constraints faced and the exact problem definition is shaped. Also, the basic product structure (with two different levels of the bill of material) is provided. We also present job data in a tabular form, for various shop-flows and attributes like colors, setups needed etc. The research work is developed for all the problems discussed with all the data presented, in this chapter. Chapter 4 is a presentation of the actual mathematical model development. It explains in detail, the MILP models and the supporting application algorithms developed with all the applicable constraints. Also, the solution procedures are briefly explained with respect to the individual modules. Chapter 5 deals with the calculations for and analysis of the obtained results using the developed hybrid model. The model validation is illustrated with the help of two separate cases. Analysis of the obtained results is performed. Chapter 6 presents summary and discussion for the results as well as the other possible research directions.

1.4 Main Objectives of this thesis research

An attempt is made to address an actual industrial problem from Western Canada. The focus is laid on the requirements of the manufacturer for production synchronization amongst the shops to save production and inventory costs. A mathematical programming model is developed for solving the cost saving objective function.

This model is very realistic and practical. As it is developed for a specific industrial problem, it is unique in its application. Also, a hybrid approach is used combining two effective JIT policies. Our literature search identified only one similar application in the past (Bonvik et al., 1996). That particular model associated itself with a simulation study for a 10 machine production line with assumed data. The objective was to study the behavior of the individual and hybrid approaches as regards service improvements and inventory.

This model can be noted as one of the very few efforts for applying such hybrid policy to a real life problem. It also studies the effect of capacity constraints and demand patterns on the overall costs of this plant. This research is an important application of JIT policies to serve cost minimization purposes in a real world manufacturing system.

Chapter Two

Literature Survey

2.1 Material Requirement Planning

Until recently, MRP has been the most popular method of production planning using Push type of controls. It provides a good solution to the complex scheduling problems of multiple jobs and machines. Also, because of its efficient computerized functions, it takes less computation time. It assumes known flow times and hence does not need estimated values. Most MRP systems check to ensure that system capacities are not exceeded (Herer and Masin, 1997). Complexity of operations because of its heavy computer dependence is a major disadvantage of MRP. Also, apparently, it does not perform well against inventory criterion. Theoretically, it aims at zero inventories. Since the calculations assume exact known processing times, exact production quantities are released to the shop. Then ideally, the jobs should follow the sequence and finish by the same date as predicted by MRP output. On the contrary, JIT schemes like kanban or CONWIP ask for intended inventory buffer for their operation. Duncan (1988) presented a good comparative study of MRP and JIT systems. He mentioned that MRP II inventory and production control techniques, while correct in theory, have in many cases automated generation of the wrong requirements. MRP, being one of such powerful generation systems, is completely reliant on accurate bills of materials, product routings, lead times etc. In order to minimize the errors in computations, researchers and practitioners have developed many complicated checks and algorithms which in turn made the system more complicated. MRP needs frozen production plans being almost hypothetical in today's dynamic market environments.

2.2 Just In Time Systems

JIT philosophy has proved to be a reliable tool which aims at: (i) preventing transmission of amplified fluctuations in demand or production volume of the process, (ii) minimizing fluctuation of in-process inventory and (iii) raising the level of production control through decentralization (Monden, 1983). But, it is also true that JIT is not equally beneficial to every type of industry and it takes considerable time and study to determine the correct JIT scheme for a given industry. Chang and Lee (1995) showed that American JIT firms have not achieved better organizational performance in terms of dollars/employee or operating profit margins or return on investment. But, they agreed that better performance in terms of finished goods inventory turnover, overall operational quality and flexibility are obtained.

Various approaches have been suggested to tackle the problem of JIT implementation. Spearman and Zazanis (1992) presented a good study of Pull v/s Push systems' controllability and robustness of operations. They concluded that because it is easier to control WIP and measure throughput, Pull type systems are more robust. In addition to these two, various other methods like Basestock, Drumbuffer, and some hybrid models are also proposed by researchers. A good reference can be found in a study by Bonvik (1996) and Spearman et al.(1989). Some of these proposed approaches are: A hybrid scheme by Groenevelt and Karmarkar (1988) similar to kanban but also responding to seasonally changing product mix. In this case, cards are directly sent to the bottleneck machine and then those parts are pushed till finished. Another one is by

Deleersnyder (1988) in which information is passed beyond the immediate stage, so that it triggers material releases upstream. The two popular JIT approaches have been kanban and CONWIP systems. This chapter addresses the publications mainly for these two methods.

JIT is a philosophy. It can briefly be described as a method to ensure that a job flows throughout its process cycle without being stopped or accumulated as WIP at any stage. In many cases, 'zero safety stock' is not justifiable, as safety stock is a trade-off between setup cost (plus down-time or idle cost) and inventory holding cost and acts as a guard against numerous uncertainties in production. These safety stocks also act as a cover-up for the inaccuracies and poor production control techniques. As pointed by Bitran and Chang (1987), managers very often accept the setup times without trying to reduce them. Also, they do not try to improve the lead-time or forecast inaccuracies. Instead, they use more and more safety stocks (higher batch sizes) to suppress the uncertainty factor.

Clendenen and Rinks (1996) presented a 'restoration model' to account for 4 types of inventories: in process, in-buffer, in-transit and in-wait inventory. They presented a cost minimization MILP model with respect to these different inventories. The model suggested that the increasing variation in demand should be offset by placing increased amount of safety stocks more or less uniformly over the whole network. This conclusion supports the one derived by Salameh et al. (1984), and Shmitt (1984).

Philipoom et al. (1996) employed JACKS (JIT algorithm for containers, kanbans and sequence). The basic assumption is to start with any work-center (generally the final workstation) to process the products. However, it may not be realistic to find such work-center in fabrication shop.

Landers et al. (1998) presented an in-depth study of various versions of JIT by different users. The basic philosophy remains the same in line with the pull philosophy. They gave an example of Harley Davidson attributing Harley's very existence to JIT application, which they name as MAN (Material As Needed) program. Omark Industries devised their own scheme called ZIPS (Zero Inventory Production Systems). The authors also cited other examples like MIPS (Minimum Inventory Production Systems), Stockless Production, CFM (Continuous Flow Manufacturing).

With JIT philosophy being so popular, new MRP-II software has been developed by Goldratt and named as Optimized Production Technology (OPT)(Riggs, 1987). It has characteristics of both the MRP-II and JIT. In this plan, any controlling parameter for maximum output is considered as a potential bottleneck. Total orders are broken into small batches and setup times are minimized. Queues are ensured for the bottlenecks and thus continuous work is maintained. The OPT program calculates nearly optimal schedule after considering all priorities and capacities.

Duncan (1988) presented a good study of JIT systems as applied to American manufacturers. He emphasized 8 primary components required to implement JIT successfully. Some of them are organization of JIT program, quality, process-oriented flow, simplified and synchronous production, employee involvement, among others. He also pointed out that 70 to 80 % of the total manufacturing lead-time is queue time. Setup costs can be reduced by effective group technology techniques. He also expressed the need for setting proper production control policies as an essential factor of JIT implementation, which include proper scheduling and kanban setting methods.

Bonvik and Couch (1996) presented a good comparative study of various policies of Lean Production Systems (LPS) v/s kanban and CONWIP. Bonvik (1996) concluded that both of them operate for the best results in different operational environments.

2.2.1 Kanban System

The credit of successful invention goes to the technical team at Toyota Motor Company, Japan. As mentioned by Ohno (original developer of the kanban system), it was born through their various efforts to catch up with the automotive industries of western advanced nations after World War-II. Total number of kanbans puts upper bound on the WIP. Sequencing for kanban system is an equally important issue. In the Toyota production system, kanban system is supported by smooth production flow, reduction of setup times, standardization of the jobs and improved activities along with autonomation (Monden, 1983). Bitran and Chang (1987) developed a mathematical model for kanban system in a deterministic multi-stage assembly production setting. They also investigated the solution procedures for this model so that it can be implemented practically. They provided three choices for the container sizes for line managers to choose from. Based on this work, Bard and Golany (1991) developed a MILP model for determining the optimum number of kanbans. They considered a single card kanban production system as a directed network in which nodes are workstations and arcs are representing material and information flows. The model is then solved as non-linear programming (NLP) by efficient solution (software) packages like GAMS/MINOS. They also suggested that kanban is best suited when demands are steady and lead times are short. They expressed the need for a better kanban model to solve the problems of inventory balance and setup minimization. Kanban outperforms the traditional heavy computer based MRP system when demand is stable and predictable with low setup times. Clearly, the shops with a large variety of parts will ask for higher number of setups and have higher cost. But, for repetitive manufacturing shops, the variety is less and setup times are small. So, Kanban is best suggested in such kind of manufacturing environments. Kanban system is not suitable for expensive and/or low volume jobs because the shop will carry the inventory all the time for a small one-time order. As pointed out in Riggs (1987), kanban moves with a batch of parts specifying what is to be done with the parts. This movement helps to highlight any change of flows in the system.

2.2.2 CONWIP System

Constant work in process or CONWIP, as named by founder Spearman (Spearman et al., 1990), has been a very popular methodology. It is a generalization of kanban system. Spearman et al. (1989) developed a hierarchical control model around the CONWIP concept. They mentioned the suitability of CONWIP for changing product mix, job order production and significant setups against kanban. They highlighted extreme predictability of CONWIP as its major advantage. As mentioned by Hall (1981), kanban is intrinsically a system for repetitive manufacturing. Also, for changing product mix situations, kanban's flexibility is limited (unless used in some form of hybrid system). On the other hand, CONWIP can be easily adjusted to such situations. Herer and Masin (1997) developed a mathematical model addressing the question of establishing efficient backlog list order. As mentioned by them, the procedure for achieving this order with infinite number of containers is a flow-shop sequencing problem with permutation schedules. They used the MVA (mean value analysis) approach for calculating shop WIP and throughput repeatedly in their model with a tandem queuing system. They also concluded that CONWIP is more predictable and simpler than MRP with almost no need of supervisors. Bonvik, Couch and Gershwin (1996) made comparisons amongst kanban, CONWIP, minimal blocking, basestock and hybrid models. They carried simulation study for single product short flow line to estimate the performance attributes such as 'total inventory carried' and service levels. They also concluded that CONWIP outperforms kanban with respect to average inventory levels for the same throughput and service level requirements. When system operates at its maximum capacity the hybrid policy further improves the performance. Gstettner and Kuhn (1996) presented a good comparison study between kanban and CONWIP systems. They analyzed the two methods with respect to production rate and WIP. Bonvik (1996) proposed a sensible approach for hybrid modeling with kanban and CONWIP. Some simulation experiments were carried out to check the behavior of the model. A shortcoming of CONWIP method was published. If the system is very heavily utilized or there is a bottleneck in the CONWIP line, the buffers towards the upstream end of CONWIP line will have quite high level of WIP. And kanban control is designed for putting upper limit on such kind of buffers. A hybrid control policy was developed using secondary kanban cells to supplement CONWIP control.

2.2.3 Some Success Stories

- In N. America, the automobile industry was the first to try the JIT philosophy in an effort to cut down on production costs. Now, Aerospace industry is mimicking them. Newest success story is **Boeing** adapting kanban system for their production facilities for 737-700 jumbojets and C-17 Fighter planes. Significant achievements like doubling of production without raising the employee count and minimizing production and WIP times to a great extent have been achieved (National Post, 1999).

- **B.F.Goodrich** has boosted their operating profits from 9.5% in 1995 to 15.6% in 1998. They credited it partially to the lean manufacturing techniques obtained through JIT system (National Post, 1999).
- **Hewlett Packard**'s Cupertino, California plant reduced their circuit board assembly time from 15 days to 11.3 hours and reduced WIP from $670,000 to $20,000 (Spearman et al., 1989).
- **General Electric** studied and implemented JIT in their dishwasher manufacturing facility. They achieved considerable results reducing 9 mile long traditional conveyor to a 2.9 mile computer controlled non-synchronous conveyor. They also reduced setup times by about 75%, human material handling from 28% to 1% and inventory investments from $9.7 million to $3.2 million (Duncan, 1988).
- **Harley Davidson** reduced their setup times by 75% and assembly throughput time from 3 days to ½ day. Direct, indirect and warranty costs per unit are reduced by 46%. Employee productivity was raised by 37% along with the reduction in operational defects by 53% (Landers et al.1998).
- **Omark industries** have been able to reduce their setup times for some operations from 6 ½ hours to 1 min. 40 sec. Space requirement is reduced by 40% and plant-wide inventories are reduced by 50% (Landers et. al.1998).
- The Japan management association (1989) reports a very relevant example of the benefits obtained by **Toyota** from their own Toyota Production System. They quoted,

"In 1976 and 1977, shortly after the first oil shock, when Toyota motors registered profits of ¥ 182.2 billion ($ 597.4 million) and ¥ 210 billion ($ 716.7 million) respectively, the company was criticized for making too much money."

2.3 Summary

In principle, MRP is based on zero WIP production, whereas, JIT systems need some amount of WIP for their operation. Apparently, JIT systems normally have less total WIP inventory than MRP. The reason for this lies in the fact that actual real life production is faced with many problems like job rejections, machine failures, raw material shortages, absenteeism etc. So, to cover up for these problems in an MRP system, the manufactured quantity is always made in little excess in the beginning. Then, because of these excess components, jobs start getting queued. This in turn causes longer lead times. As a result, the queue length further increases. And then it is like a 'vicious cycle' causing huge inventory pileups eventually (Herer and Masin, 1997). Also, because of the tight schedules based on the exact processing times in MRP, heavy supervisory control is a must. Anything going wrong with any of the resources has to be fixed immediately to avoid the adverse effect on overall job progress. Miltenburg (1997) presented a five step procedure to gradually shift from MRP to TOC (theory of constraints) or JIT using the common things (like workstations, resources and production planning hierarchy) between these two methods. He then illustrated the peculiar benefits achieved from all these three methods with the help of a Markov chain model. But, overall, MRP is out-performed by the other two methods. He concluded that MRP is flexible and can easily be transformed into JIT or TOC without being dismantled abruptly. He also suggested that traditional methods like MRP should only be used when competitors are using it. Otherwise JIT system will be very effective against all criterion like output, cycle time, inventory and shortages. The disadvantage is the cost incurred for the JIT implementation.

Chapter Three

Problem Description

This research is conducted based on the production system in a major agricultural machinery manufacturing plant in Western Canada. The plant layout is presented schematically in Figure 3.2. The facility is engaged in manufacturing a main product called 'Heavy Harrow' (HH), comprised of seven sub-assemblies, which in turn are comprised of several parts. This research associates itself with the development of JIT system for in-house manufactured components only. The purchased components are not considered in developing the model.

The actual parent-child relationships from 'bill of material' are used as a basis of this research work. To make it realistic, the second level parts in the bill of material are also considered. Refer to Figure 3.1 for all these parts. Other data as regard to the attributes of all the jobs like color and setup types, overall quantities and required operations are presented as a table in Table 3.1.

Each sub-assembly is a group of several components. Some of them are purchased components and their receipt schedules are controlled by MRP system at present. The in-house parts from any group share the same group color. The member of a particular group will either have that group's color or be colorless. The facility does not assemble the finished product (HH) because of the difficulty in transporting it as a single unit with its big size. After the group assemblies are done, they are transported to the customer's site to build a complete Heavy Harrow there.

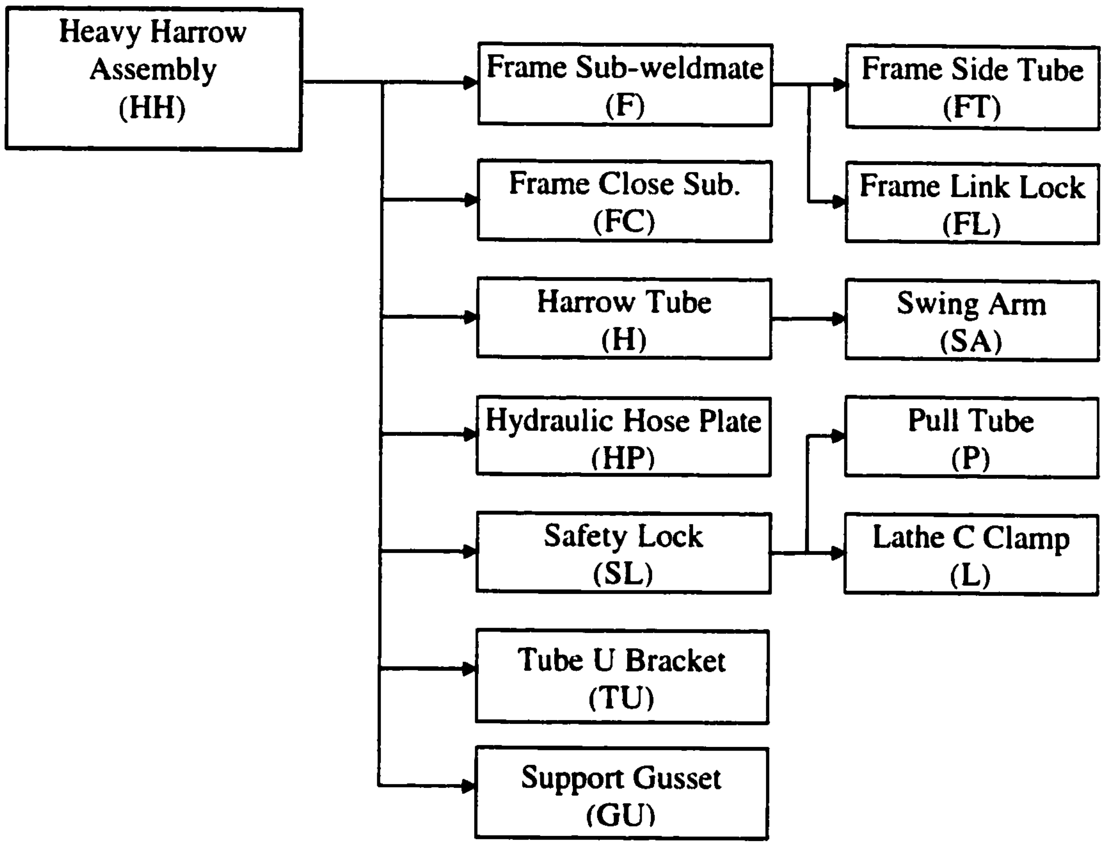


Figure 3.1 Part structure of Heavy Harrow

Table 3.1 Data and Production Flow for manufactured components

Sr. No.	Job	Qty per HH	Color	Setup Type	Setup Times	Process Times	Product's Flow In Various Shops
1	F	1	Blue	B	Data File ✤	Data File ✤	P→ A→ FS
2	FT	1	Black	B	--- do ---	--- do ---	RM→ M→ F→ SB→ P
3	FC	1	Black	A	--- do ---	--- do ---	P→ A→ FS
4	FL	1	Blue	B	--- do ---	--- do ---	RM→ M→ F→ SB→ P
5	SA	1	Black	A	--- do ---	--- do ---	RM→ M→ F→ SB→ P
6	H	2	Blue	A	--- do ---	--- do ---	P→ A→ FS
7	HP	1	Blue	B	--- do ---	--- do ---	P→ A→ FS
8	SL	1	Blue	A	--- do ---	--- do ---	P→ A→ FS
9	P	1	Blue	A	--- do ---	--- do ---	RM→ M→ F→ SB→ P
10	L	2	Blue	B	--- do ---	--- do ---	RM→ M→ F→ SB→ P
11	TU	1	No Color	A	--- do ---	--- do ---	R→ M→ A→FS
12	GU	1	No Color	B	--- do ---	--- do ---	R→ M→ A→FS

Legend:

✤ : Please refer to the Appendices
M: Machine Shop
SB: Shot Blasting Shop
A: Assembly Shop
RM: Raw Material Store
F: Fabrication Shop
P: Paint Line
FS: Finished Goods Storage

This research work was done to design a system to reduce production and inventory costs for the Heavy Harrow line. There is also a need to have a better control over the resource utilization rates. Increasing throughput and synchronization between the shops was another important requirement.

The plant under consideration employs batch production and is being changed towards a JIT shop. Production is initiated by a work order from the Planning Department. They release the order quantities with the help of a central MRP planning module. But, because of the non-availability of exact lead times and high variability in process times (leading to unreliable inputs), they can not fully rely on the MRP outputs. Excess quantities are used in dealing with these uncertainties. As a result, a large number of work orders remains open causing high WIP costs.

A sudden change in production philosophy from Batch Production to JIT is difficult for the workers. Especially the reduced batch sizes (and thus more setups), give a feeling of low output as they now spend their total time on a bigger variety of parts. This yields less daily production quantities per product. Even though it holds good for overall production schedules, the reduced production quantities create a feeling of insecurity in the workers.

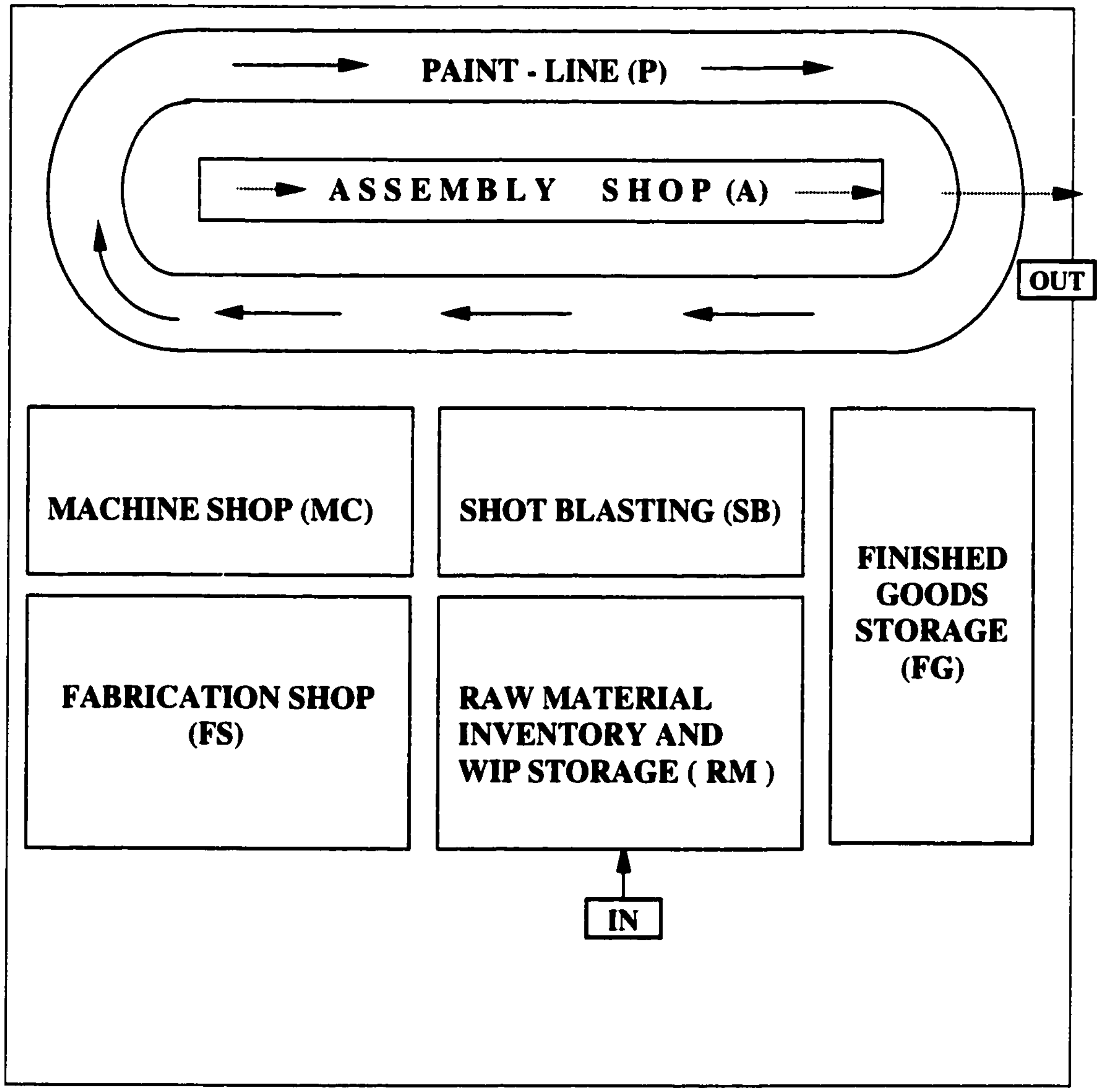


Figure 3.2 'Heavy Harrow' production facility at the model plant

The operation of the plant is described now. The production flow for the main in-house manufactured components is described along with the constraints in detail:

1. Most of the bar and sheet metal components are manufactured in-house. They are generally machined. Bar components mainly need operations like turning, milling, drilling etc. The sheets may either be drilled or bent on press brakes or cut on shearing press. These components are made in excess quantities by batch production to justify for the high setup times. Some components flow directly to Assembly shop from Machine shop. The others follow a serial production flow.

2. Some components are meant to be fabricated sub-assemblies. Whereas, the others are temporarily (tack) welded in color groups for their shared hookup on the paint-line load-bars. All such components needing welding are fabricated in the Fabrication shop. No check is currently present on the exact production quantities. Extra WIP starts piling up at this stage. The components, which are welded, are always painted. So they proceed to the next shop where shot blasting is performed.

3. All the components to be painted must be shot blasted first, to remove welding spatters, grease, scale, rust etc. The throughput capacity of the shot-blasting machine is low as compared to the Paint Shop. Also, because of the process time differences, the time required for shot-blasting operation is generally three times more than the painting time for any component. To meet the Paint Shop demand, shot-blasting operation has to be run continuously. This sometimes asks for extra capacity from manually operated blast-shop if needed.

4. The components (almost about 90%) need painting. The two colors used are blue and black with the quantity distribution of approximately 60 and 30 % respectively. Paint

Shop comprises of a continuous chain conveyor operated paint line. Because of the high investment and running costs for this line, the most important goal is to keep its utilization rate to the maximum. To achieve that, a uniform flow of products from Fabrication and Machine shops (in continuous batches of the same color) is needed. At a time, the paint line can be set for only one color. Switching from one color to another takes considerable time and hence costs. So, preference is given to make the change (if needed) at the end of the day. In that case, it is necessary to supply a minimum of one-day's (same color) workload to paint line. Large quantities of each color jobs are produced to justify a continuous operation with the same color setup. It causes a huge pileup of inventory at the shot blasting machine entry because of capacity constraint on that machine. A main concern is to solve this problem. Based on the paint-line loadbar capacity and velocity of conveyor, at most 161 loadbars can be hooked on to the conveyor per day (with 2 shifts). Currently it can be used to a maximum of 66% utilization. The main reason is the lack of proper synchronization between the expensive Paint Shop and its predecessor earlier shops.

5. The painted jobs enter the Assembly Shop at the left entrance point and then the assembly follows in a sequential way to get a finished product at the right hand side end of the line. Obviously, the heavy jobs get assembled near the entrance point and the light ones on the right side to avoid heavy material handling from paint-line to assembly point. Assembly Shop performs seven main subassemblies for the HH. So, the crux behind the synchronized paint line and assembly shop is the perfect timing of the component deliveries from the prior to the later one.

6. Most of the purchased components like motors, gearboxes and mild steel sheets come from Europe and Asia because of lower purchasing costs. To justify for the transportation and buying costs, they are purchased by larger quantities.

7. MRP system has been used for production planning. The total demand for components is fed into a central computer. It calculates the production quantities. Job orders are then passed on to the first shop (Machine Shop or Fabrication Shop) to withdraw raw material. Detailed schedule is obtained from the central master computer. Every job's stage-wise progress is input into the system to update the job progress data.

Apart from the above shop-wise constraints, another major concern is the quality of production. JIT is not only meant for saving on inventory or setup costs but also for increased quality and human respect through continuous fault finding and rectification or 'autonomation' (Monden, 1983). JIT also warrants for small batch sizes as compared to the batch production policy. Reducing setup times to the minimal is necessary to justify for the small batches.

When an assembly operation takes place, all of the parts must be available, and sometimes there are many parts. Each of them needs to be produced by either manufacturing or a purchasing function. Thus, timely flow of components from previous shops to Paint Shop and then from there to the Assembly Shop are the main targets to be achieved. Also, it needs to be done without the shield of excess safety stocks. This is a main challenge in implementing JIT.

The mathematical programming model presented in the next chapter is designed for three main purposes. It aims at:

1. reducing the production and inventory costs in the plant,

2. synchronizing the material flow between expensive Paint and Assembly Shops, and their suppliers like Fabrication and Shot Blasting Shops,
3. increasing the resource utilization of Paint Shop by ensuring a continuous flow of job batches with same color from its earlier shops.

A hybrid approach using kanban and CONWIP methods is selected. The rationale follows in Chapter 4. A mixed integer linear programming (MILP) model is formulated and solved along with application algorithms developed for color group formations. All constraints described in this chapter are considered to make the model as effective as possible.

Chapter 4

Model Development

Pertaining to this particular case (considering all the inputs from Chapter 3), a mathematical programming model is developed using hybrid kanban-CONWIP approach. Figure 4.1 shows a schematic diagram for the proposed hybrid model. The information and production flows are in opposite directions to each other as shown. The model is explained in the following text. The reasons for choosing such approach and its suitability to the plant in consideration can be summarized as follows.

4.1 Suitability of hybrid approach to the plant in consideration

1. The model could be developed as a CONWIP line to decide the optimum schedule for all the shops. There are some major drawbacks in such approach. CONWIP line can be developed for only the items sharing the same sequence of operations. In the current setup, this condition is clearly violated because of a variety of flows between the different shops (Table 3.1). Hence, developing a CONWIP line becomes a complex problem.
2. This shop could be designed as a kanban shop. In kanban modeling, there is no provision for scheduling. In order to ensure continuous and correct type of material flow to Paint Shop, a robust scheduling method was required while kanban alone could not be a solution for this plant, in that case.
3. Setup costs involved in the first two work shops (M/C and Fabrication) and the last shop (Assembly) are significant. Hence, kanban quantity calculation for minimum costs is suggested in these shops.

The proposed hybrid model consists of three different modules (Kanban-2, CONWIP and Kanban-1). Also, it contains supporting algorithms like MRP programs and Paint-Setup Algorithm, developed for these modules. Attention is diverted towards Figure 4.1 for a presentation of a working logic of these modules. The model uses two JIT schemes (kanban and CONWIP) to solve the given problem.

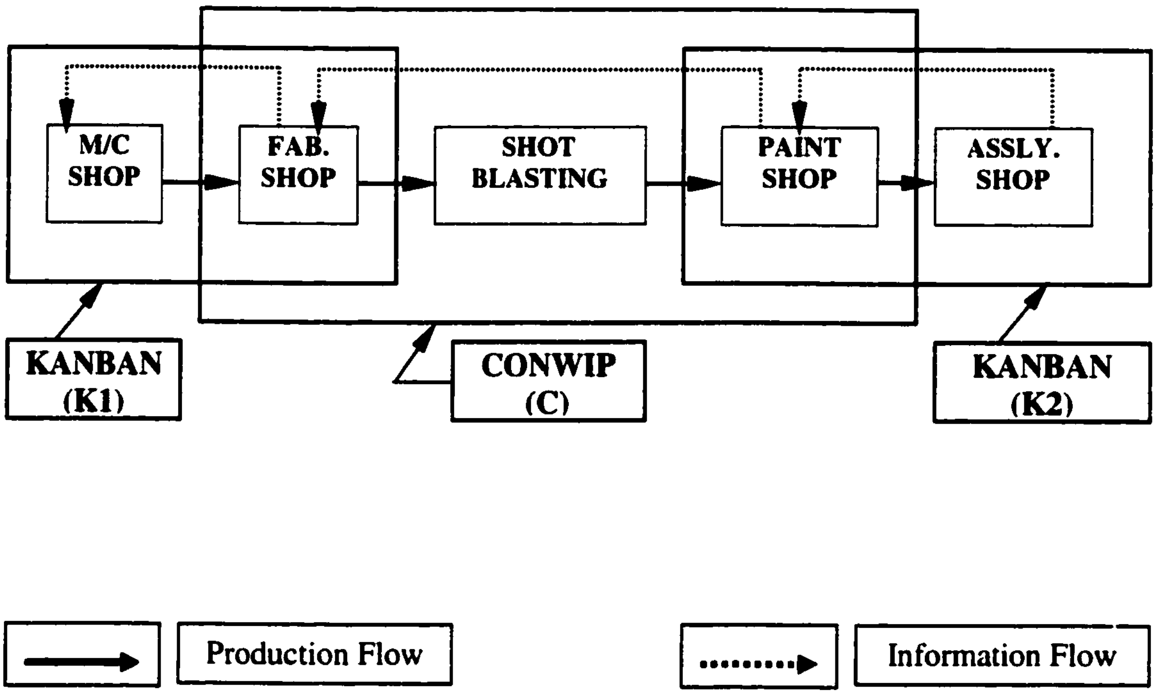


Figure 4.1 Hybrid Kanban-CONWIP production model

4.2 Method of working

The production is initiated based on the total demand from the Sales Department being conveyed to the Planning Department. (The hybrid model will be installed in that office). The input to the last module is this demand. The output from each module serves as the input for its earlier module thus conveying information in a backward direction and parts flow forward. The working logic is presented in Figure 4.2.

1. Planning Department receives the demands from the Sales Department in advance. Then they arc entered into the third module. (Module K2) MRP program in that module works to give total demand quantities. Demand for finished Heavy Harrows is combined with the spares' orders. Part structure (refer to Figure 3.1) is considered to calculate the total demand quantities.

2. The total demand quantities are fed to the kanban program, which calculates production quantities for all those demanded items.

3. The part list and their kanban quantities are then sorted out into color groups and also rearranged to have minimum number of setups. Also, at the same time, the capacity constraint for the Paint-Shop is also considered. The output from the K2 module is the item lists divided in different color groups and with kanban quantities for the total demand from the Sales Department.

4. CONWIP module works individually for every color group. A backlog list is developed for the group and then a scheduling program is run to minimize the total inventory and setup costs. The optimum schedule obtained is used for the costly operations in Paint Shop and its predecessors.

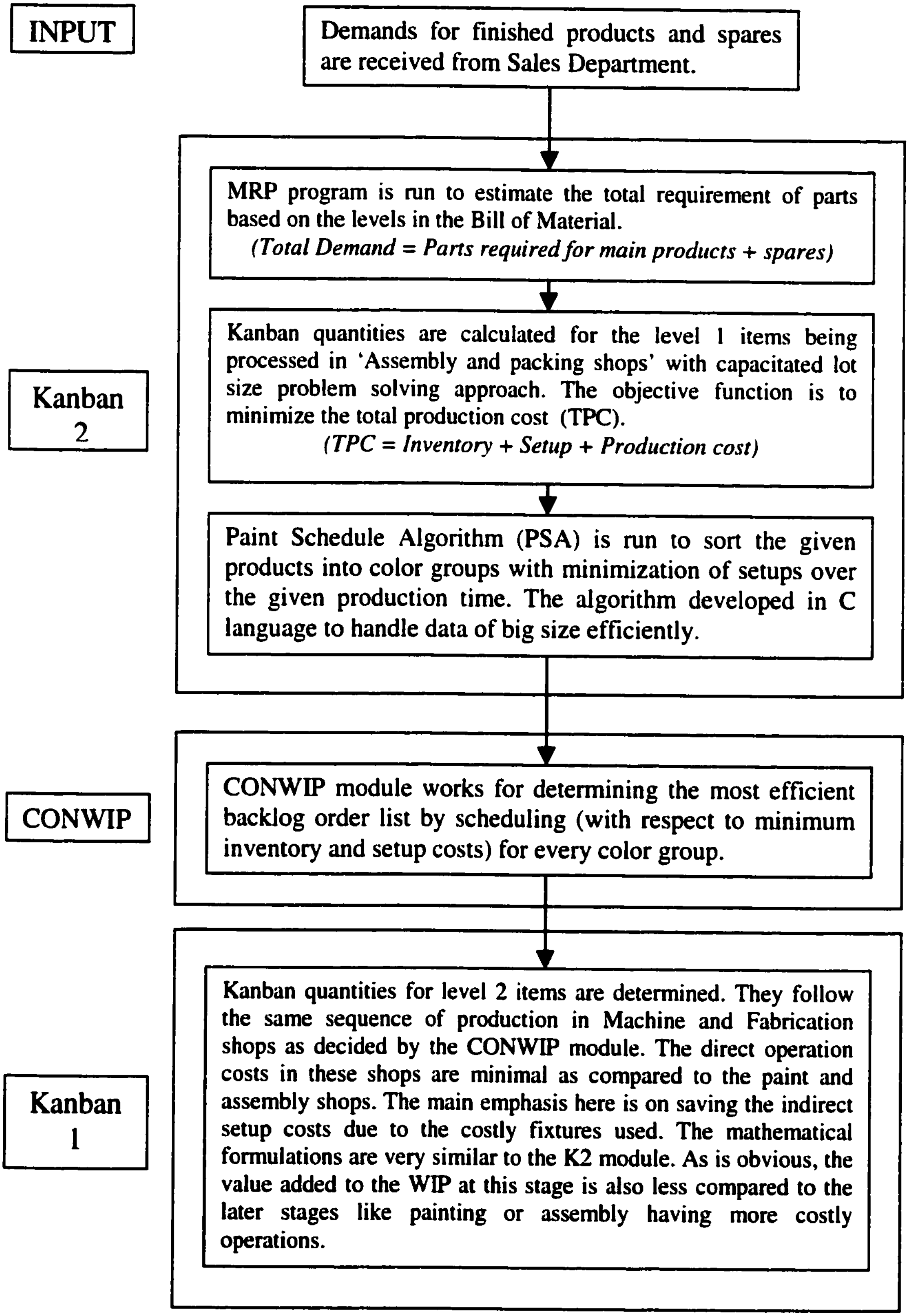


Figure 4.2 Function diagram of the hybrid Kanban-CONWIP model

5. The output of the CONWIP module is the list of items in the form of color groups with kanban quantities and CONWIP backlog list. At this stage, the parts considered are level 2 items from the BOM and their scheduling is done. In order to input the demand quantities into the K1 program, the MRP quantities estimated by K2MRP (MRP program in K2 Module) for the parent parts are considered.

The outputs from K1 module are:

1. kanban quantities for level 2 items for M/C Shop and Fabrication Shop,
2. production schedule for optimum backorder list so that Shot Blasting and Paint Shops have a steady and uniform flow of parts grouped in the color batches.

4.3 Mathematical Presentation of Developed Modules

Formulations for the modules in Figure 4.1 are given below. They are presented here in the same reverse working sequence from the pull control philosophy.

4.3.1 Module 3 – K2

MRP quantity calculation

Using the parent-child relationships and data shown in Figures 3.1 and 3.2, total demand quantities are estimated by MRP program coded in LINGO software (Hyper LINGO/PC, 1996). These quantities form the basis for all the further calculations. This program has an additional feature to assign lead time to the parts being assembled. But, it is used here for MRP purpose only. There is a constraint of the Assembly Shop capacity, which balances assembly activity properly to ensure that every subassembly is produced at the time, exactly or before the demand, and also the total demand is satisfied for a given scenario.

Kanban quantity calculation

A capacitated lot sizing modeling approach is used. The MILP model is programmed using LINGO software (Hyper LINGO/PC, 1996). This section provides the basic formulation. Individual module programs and data are provided in Appendix A.

The notations used for this model are as follows:

I Number of part types $(i = 1,2,...,N)$

M Set of machines $(m = 1,2,...,M)$

X_{it} Number of kanban quantities of part i to be produced in time period t

t Number of time periods $(t = 1,2,...., T)$

D_{it} Demand of part i at time T (numbers)

h_i Holding cost of in-process inventory of part i ($ per part per time period)

S_i Setup cost for the part i ($ per part per setup)

P_i Production cost per unit of part i ($ per part)

C_t Available capacity of shop for time period t (hours)

P_t Processing time for job i in shop (hours)

A_i Setup time for product i (hours)

I_{it} Numbers of item i in WIP inventory at the end of period t

$$Y_{it} = \begin{cases} 1, \text{if any quantity of } i \text{ is getting produced in time } t, \\ 0, \text{otherwise} \end{cases}$$

Objective Function is:

Minimize (Total Production Cost) = $\sum_{i=1}^{I}\sum_{t=1}^{T} h_i I_{it} + S_i Y_{it} + P_i X_{it} Y_{it}$ (1)

$s.t.$

$\sum_{t=1}^{T}\sum_{i=1}^{I} X_{it} = \sum_{t=1}^{T}\sum_{i=1}^{I} D_{it}$ $\quad \forall i,t$ (Demand must be met in time horizon) (2)

$I_{i,t-1} + X_{it} - I_{it} = D_{it}$ $\quad \forall i, t = 1,2,...T$ (Inventory balance equation) (3)

$C_t \geq A_i Y_{it} + P_t Y_{it}$ $\quad \forall i,t$ (Production capacity balance equation) (4)

$X_{it} \geq 0$

$I_{it} \geq 0$

We use a capacitated lot size modeling approach for the kanban modeling. The model is developed with a simplified approach. Inclusion of batch based setup costs brings non-linearity in the model making it very complex to solve. Further, production is modeled with a one part per container assumption. Various costs considered are finished goods (WIP)holding costs, setup costs and assembly (production) costs for the K2 (K1) module. Holding costs include cost of capital inventory, storage, insurance, taxes, spoilage and obsolescence costs. Production costs include direct costs (labor, tools, other consumable items) and indirect costs (other factory expenses and technical know how costs). Overheads like managers' salaries are not considered, as the level of production activity has no direct influence on these costs. Thus they can not be apportioned to the product costs. Also, when production is in batches, setup cost is incurred. Every product that needs a setup (or a group of parts having same setup) has to be identified and treated separately. Initially, an attempt to solve this problem by a 'set covering' approach

(Winston, 1991) was made. That approach was subsequently replaced by the current capacitated lot size modeling due to the excessive complexity associated with the set covering problem. The choice of capacitated or non-capacitated (as might be applicable) lot size modeling approach was also suggested for solving such problems (Hax and Candea , 1984) as an alternative method.

Module K2 and module K1 do not differ in their working logic. The only differences are the parts handled and input data used. Hence, this section is also used as a presentation for module K1 to avoid the repetition.

Paint Setup Algorithm (PSA)

The PSA is coded in C language and its working logic is described in Figures 4.3 and 4.4. It consists of 2 loops which sort out the given production part list first by colors and then by setups (amongst the created color batches), to minimize the total setups required. It is user friendly and its input and limiting parameters can be adjusted easily. The colors handled are 1 (blue), 2 (black) and 3 (no color) with setup types 1 (setup A) and 2 (setup B).

The program statement for PSA and sample input and output screens are presented in Appendix B.

Note:
The model divides shop times into 3 color groups based on the color percentages. Hence; blue, black and no-color are given 60, 30 and 10% of shop times respectively.

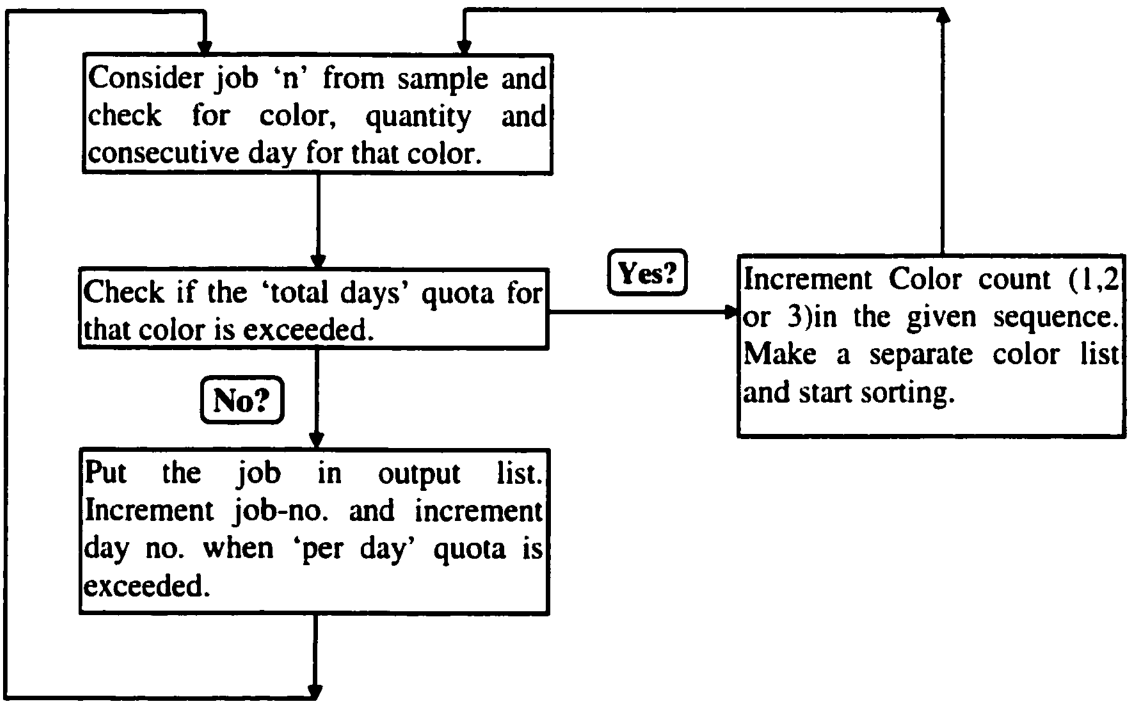


Figure 4.3 Paint Setup Algorithm - Loop 1: sorting paint-wise

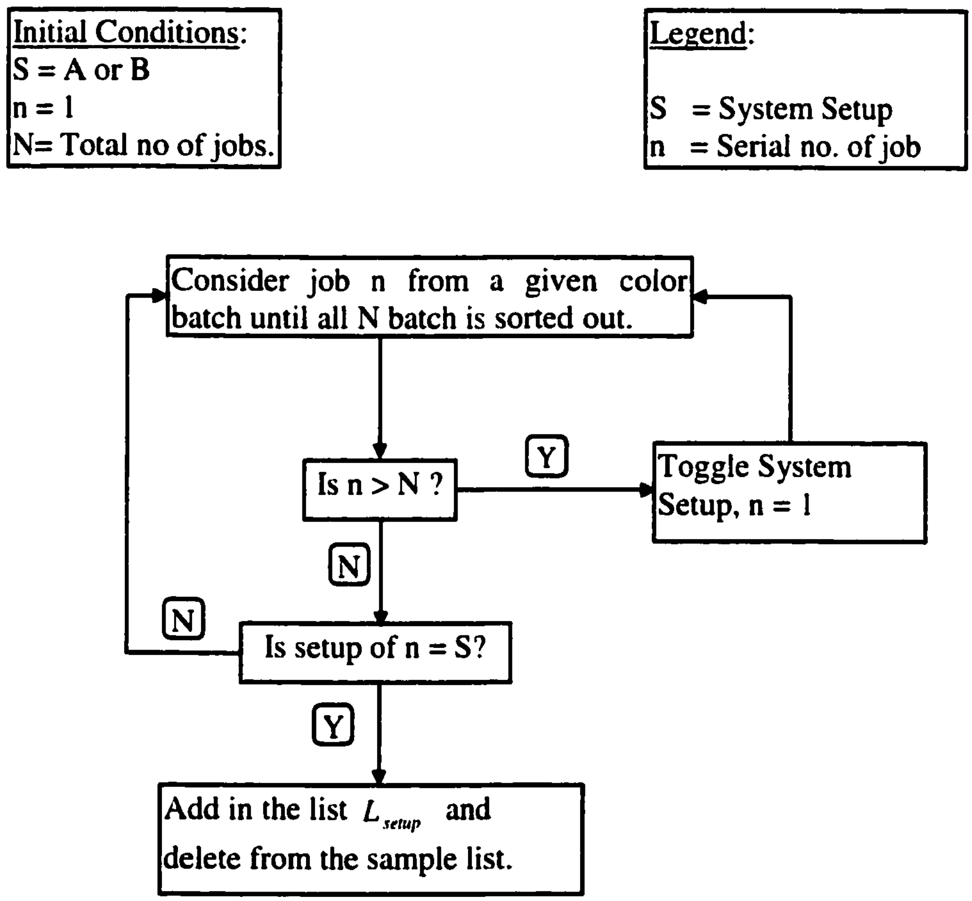


Figure 4.4 Paint Setup Algorithm - Loop 2: sorting setup-wise

4.3.2 Module 2 – C

Shop scheduling for the backlog list of the components for each color group is done using a MILP model based on CONWIP approach. It determines the optimal job sequence to minimize the production, inventory and shortage costs. The CONWIP module assumes a closed tandem queuing network environment with one job per container assumption. The parameters, variables and expressions are briefly stated as follows. Herer, Masin (1997) used Mean Value Analysis (MVA) approach for solving a simpler problem. Their model is modified and programmed using more linearized constraints and reducing the non-linear constraints. Notation is given below.

M Number of machines in the shop

I Number of part types (i = 1,2,...I)

N Number of kanbans from earlier module (n = 1,2,...N)

T Number of time periods (t = 1,2,...T)

D_{it} Demand of part i at time t

K Number of backlog list entries

h_i^w Holding cost of in-process inventory of part i ($ per container per time period)

h_i^f Holding cost of the finished goods inventory of part i ($ per container per period)

b_i Backorder cost of part i ($ per period per container)

p_{im} Processing time for part i on machine m

τ_{ijm} Sequence dependent setup time when machine m is subjected to part change from i to j

$$x_{ik} = \begin{cases} 1, \textit{if the } k^{th} \textit{ entry on the back} \log \textit{list represents part type } i \\ 0, \textit{otherwise} \end{cases}$$

$$z_{kt} = \begin{cases} 1, \textit{if the estimated release time for the part represented by } k^{th} \textit{ entry falls} \\ \quad \textit{in period } t \\ 0, \textit{otherwise.} \end{cases}$$

$$f_{kt} = \begin{cases} 1, \textit{if the estimated completion time for part represented by the } k^{th} \textit{ entry, falls} \\ \quad \textit{in period } t \\ 0, \textit{otherwise.} \end{cases}$$

$$v_{ikt} = \begin{cases} 1, \textit{if the } k^{th} \textit{ entry on back} \log \textit{list represents part type } i \textit{ and its estimated release} \\ \quad \textit{time falls in period } t \\ 0, \textit{otherwise.} \end{cases}$$

$$u_{ikt} = \begin{cases} 1, \textit{if the } k^{th} \textit{ entry on back} \log \textit{list represents part type } i \textit{ and its estimated} \\ \quad \textit{completion time falls in period } t \\ 0, \textit{otherwise.} \end{cases}$$

$$y_{ijk} = \begin{cases} 1, \textit{if the } k^{th} \textit{ entry on back} \log \textit{list represents part type } j \textit{ and it needs a setup change} \\ \quad \textit{from its predecessor component } i \\ 0, \textit{otherwise.} \end{cases}$$

R_k = Estimated release time of the part representing k^{th} entry on backlog list into the system (Hours after start of the production)

F_k = Estimated completion time of the part representing k^{th} entry on backlog list into the system (Hours after start of the production)

S_{it}^{-} = Lost Sales or backorders of part i at the end of period t (containers)

S_{it}^{+} = Finished goods inventory of part i at the end of period t (containers)

W_{it} = WIP inventory at the end of the period t (containers)

t_{mn} = Average time spent on machine m by part n (hours)

p_m = Mean effective processing time for machine m including all the processing times on that machine

Objective Function is of the MILP model :

$$\text{Min. (Total Production Cost)} = \sum_{i=1}^{I}\sum_{t=1}^{T} h_i^f S_{it}^+ + \sum_{t=1}^{T}\sum_{i=1}^{I} b_i S_{it}^- + \sum_{i=1}^{I}\sum_{t=1}^{T} h_i^w W_{it} \quad (1)$$

$s.t.$

$$S_{i,t-1}^- + S_{i,t-1}^+ + \sum_{k=1}^{K} u_{ikt} - D_{it} = S_{it}^- + S_{it}^+ \quad \forall i,t \text{ (Finished inv. balance equation)} \quad (2)$$

$$W_{i,t-1} + \sum_{k=1}^{K} (v_{ikt} - u_{ikt}) - D_{it} = W_{it} \quad \text{(WIP inventory balance)} \quad (3)$$

$$\sum_{k=1}^{K} x_{ik} = \sum_{t=1}^{T} D_{it} \quad \forall i \quad \text{(Demand be met in time horizon)} \quad (4)$$

$$\sum_{i=1}^{I} x_{ik} = 1 \quad \forall k \quad \text{(Each entry contains only one part)} \quad (5)$$

$$x_{ik} - 1.5 \le u_{ikt} \quad \forall i,k,t \quad \text{(Binary variables control)} \quad (6)$$

$$x_{ik} \ge 1.5 u_{ikt} \quad \forall i,k,t \quad \text{(Binary variables control)} \quad (7)$$

$$x_{ik} - 1.5 \le v_{ikt} \quad \forall i,k,t \quad \text{(Binary variables control)} \quad (8)$$

$$x_{ik} \ge 1.5 v_{ikt} \quad \forall i,k,t \quad \text{(Binary variables control)} \quad (9)$$

$$\delta_1 = \frac{\sum_{k=1}^{K}\left(\sum_{i=1}^{I} x_{ik} p_{im}\right)}{K} \quad (\delta_1 \text{ :Mean effective processing time}) \quad (10)$$

$$\delta_2 = \frac{\sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{I} y_{ijm} \tau_{ijm}}{K} \quad (\delta_2 \text{: Mean effective setup time}) \quad (11)$$

$$p_m = \delta_1 + \delta_2 \quad \text{(Total mean effective time of machine } m \text{)} \quad (12)$$

$t_{m1} = p_m$ (Average time spent by container on machine m if only 1 job is processed on the machine) (13)

$$t_{m,n-1} = p_m \left[1 + \frac{t_{m,n-1}(n-1)}{\sum_{j=1}^{M} t_{j,n-1}} \right]$$ (Mean value analysis estimating model) (14)

$$R_k = (k-1)\frac{\sum_{m=1}^{M} t_{m,n-1}}{n-1}$$ (Entering time into the CONWIP line) (15)

$$F_k = R_k + \sum_{m=1}^{M} t_{m,n-1}$$ (Completion time from the system) (16)

$x_{ik}, u_{ikt}, v_{ikt}, y_{ijk} \in (0,1)$ $\forall i, j, k, t$

$S_{it}^{+}, S_{it}^{-}, W_{it}, p_m, t_{m,n-1}, R_k, F_k \geq 0$ $\forall i, m, n, k, t$

A brief explanation of the above model is given below:

The objective function (1) aims at minimizing the sum of the holding cost for finished goods, holding cost for the WIP goods and the backorder costs. It considers the total time horizon for calculation along with all the items. Backorder costs include cost of expediting (overtime or added sub-contraction), cost of loss of customer goodwill and loss of sales revenues resulting from the shortage situations. Costs like those of goodwill losses or order losses might be difficult to be estimated in exact figures as they are relative in nature. But, overtime or sub-contraction can be estimated precisely. Constraint (2) is an inventory balance equation for total finished goods stock (or shortage) in hand. It ensures that demand for the given time period is satisfied. Constraint (3) is an inventory balance equation for WIP goods. Constraint (4) ensures that all the demand must be satisfied during the given planning horizon. Constraint (5) ensures that only one part is represented by each entry in the backlog list. Constraints (6), (7), (8), (9) ensure the proper behavior of the binary variable. Constraint (10) calculates the Mean Effective Processing Time. Constraint (11) calculates the Mean Effective Setup Time. Both of these constraints are required because p_m (Mean Processing Time for job i on machine m) is used in constraint (14), which represents the MVA approach. Constraint (12) gives the total mean effective operation time of machine m. Constraint (13) provides the value of p_m, when only one job $(i = 1)$ is processed on machine m. Constraint (14) is the formulation for MVA approach. In this, the mean flow time and mean throughput rate are calculated with iterations by continuously applying Little's law which states that (Mean Inventory = Mean Throughput Rate × Mean Cycle Time) This is the constraint which creates non-linearity in this model. The computation time of the model depends mainly on this constraint. Constraint (15) estimates the release time of a part i into the system by using the mean throughput rate and mean processing time. Mean time between departures is the inverse of the throughput rate. Hence entry of the part at k^{th} position in backlog list into the system can always be found by multiplying $k-1$ with the mean time between the departures. Constraint (16) is an estimation of finish time by adding mean flow-time of job i to the release time of job i into the system. The final two constraints define the mentioned variables of binary nature and positive values respectively.

Chapter Five

Calculations and Analysis

The performance for proposed JIT approach is measured in terms of the overall production costs saved. The detailed list and expressions for all the program files are included in the Appendices.

5.1 Calculations

The LINGO program runs were conducted on an IBM-PC (Pentium) compatible computer in the Robotics Laboratory, University of Regina. Paint Setup Algorithm (PSA) runs (coded in C language) were carried out on a Unix Workstation, Computer Science Laboratory, University of Regina. The computation times required were minimal (less than 1 second). Calculations were carried for two different cases. Input data and result tables for each of them are given as these cases are presented.

Before we proceed to any further section, the significance of the intended reverse fashion of calculations needs to be mentioned. The calculations start with module K2 and then follow in the reverse order until module K1. Actual production flow is in direction from K1 to K2, but in order to maintain the pull philosophy, information flows backward. Output from the last module K2 forms information input base for its previous module C, which in turn is the basis for module K1. Hence, the same reverse working fashion of these modules is followed for their explanations in this chapter.

5.2 Experiment 1

This section is divided into two parts broadly viz. input and output. The input section (5.2.1) covers the information about various input data and basic capacity assumptions. Output section (5.2.2) is a presentation and detailed analysis of results obtained from the calculations.

5.2.1 Input Data

In this first experiment, data representing a typical demand pattern for this type of industry are used. Input data are shown from Tables 5.1 to 5.3. The arrangement of the output tables is in the same sequence as the working of their respective modules. Hence, they are presented in the reverse direction starting with module K2.

Data in Table 5.1 is input for the MRP program (File: K2MRP1.LDT) to calculate the total demand for all levels in the product BOM. Tables 5.2 and 5.3 are the processing time and cost data for the parts and subassemblies considered in the experiment. In Table 5.1, demand for both the main product and spares is shown. Amongst the spares, both level-1 and level-2 items are demanded, which is quite realistic. Such spares' demands arise out of breakage of a single part from a subassembly.

Table 5.1 Module K2: Demand from Sales Department, Experiment 1

Serial Number	Job Name	Demand quantities (Period P5)	Demand quantities (Period P6)
1	HH	2	1
2	F	0	2
3	FT	0	1
4	FL	2	0
5	SA	0	2

Table 5.2 Process attributes for 7 sub-assemblies, module K2, Experiment 1

Serial Number	Attribute (Minutes)	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7
1	Setup Time	20	25	10	14	17	30	10
2	Assembly Time	1	1	3	3	3	2	2
3	Setup Cost	6	5	4	5	10	6	2
4	Assembly Cost	2	2	2	3	1	1	1
5	Holding Cost	1	1	1	2	1	1	1

Table 5.3 Process attributes for 7 sub-assemblies, module K1, Experiment 1

Serial Number	Attribute (Minutes)	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7
1	Setup Time	36	39	41	35	20	23	34
2	Production Time	12	22	11	21	11	13	11
3	Setup Cost	12	20	30	21	10	10	20
4	Production Cost	8	7	8	9	6	5	4
5	Holding Cost	11	11	11	12	11	11	12

Data from Table 5.2 is used by K2 module (K2C1.LNG) for calculating kanban quantities for level-1 items in Assembly Shop. Capacity of Assembly Shop is 100 minutes per period. This is different than that of the Machine and Fabrication shop capacities because of the different shop utilization rates. The Assembly Shop carries some other product assemblies apart from Heavy Harrow and only a part of this shop is occupied with HH activities. Hence, the manpower and equipment allotted for HH subassemblies in terms of time units are less than a full shift load of 480 minutes.

Table 5.3 is used by module K1 for calculating kanban quantities of all level-2 items (and level-1 items : components TU and GU) in the BOM handled by this module. Calculations are based on their MRP demand (File:K1C1.LNG). Machine Shop and Fabrication Shop capacity is 480 minutes (per time period).

5.2.2 Results and Analysis of Experiment 1

Output of MRP program in module K2 is shown in Table 5.4. These data then in turn were used as input to the kanban model to calculate kanban quantities in Assembly Shop (module K2). Main quantitative results are summarized and presented in Tables 5.4 to 5.9. Detailed results of this experiment can be viewed in Appendix A.

Table 5.4 presents the output of the MRP program. The total demand is obtained for all the required components. The demand is observed in the last two periods in line with the input of Table 5.1. Total number of HH required is 3 meaning that 3 sets of all the 7 subassemblies will be required. In addition to that, spares demand is taken into consideration. Periods 1 to 4 with no demand are also shown in this table in order to compare the final production schedule obtained from the first module (K1) in the end.

Table 5.4 Module K2: MRP quantities obtained, Experiment 1

Serial Number	Job Name	MRP Quantities (period-wise)					
		P1	P2	P3	P4	P5	P6
1	HH	0	0	0	0	2	1
2	F	0	0	0	0	2	3
3	FT	0	0	0	0	2	4
4	FC	0	0	0	0	2	1
5	FL	0	0	0	0	4	3
6	SA	0	0	0	0	4	4
7	H	0	0	0	0	4	2
8	HP	0	0	0	0	2	1
9	SL	0	0	0	0	2	1
10	P	0	0	0	0	2	1
11	L	0	0	0	0	4	2
12	TU	0	0	0	0	2	1
13	GU	0	0	0	0	2	1

Table 5.5 Module K2:Kanban quantities obtained, Experiment 1

Serial Number	Job Name	Kanban Quantities (period-wise)					
		P1	P2	P3	P4	P5	P6
1	F	0	0	0	0	5	0
2	FC	0	0	0	3	0	0
3	H	0	0	0	0	6	0
4	HP	0	0	0	0	3	0
5	SL	0	0	0	3	0	0
6	TU	0	0	0	3	0	0
7	GU	0	0	0	0	3	0

Table 5.5 represents the kanban quantities calculated by K2 module for all the level-1 subassemblies considering their setup, assembly and inventory costs. Most of the components are produced just in time. There is a variation between the production periods because of the need for availability of children parts getting assembled before the parent part. Total assembly time required from this output for subassemblies FC, SL and TU in the 4th time period (90 minutes) is approximately equal to that for F, H, HP and GU subassemblies (91 minutes) in the 5th time period (File: AK2C1). In the available period capacity of Assembly Shop (100 minutes), production is distributed smoothly over 2 periods without creating shortage. It also minimizes total assembly costs. Because of the high setup costs (as compared to the assembly costs), breaking of batches is avoided by the model. Holding costs are less than any of these costs, which justifies carrying over some stock for low inventory items for 1 period. Periods marked with all 0 production quantities reveal the excess capacity available after accounting for all demands. In case of periods with partial 0 quantities, total workload for the existing production has to be considered before deciding available capacity.

Table 5.6 shows output of K2 module for the PSA run. The basic purpose of this algorithm is to sort the production quantities into colors and then minimize the setups required in Shot Blasting Shop (Appendix B). The blue part list is followed by black and again toggle is made. Then setup-wise sorting is done among these color batches as can be seen in Table 5.6. The importance of PSA is apparent when dealing with a large number of components with a variety of setups. This output is further used by CONWIP module to schedule the backorder list components in each of these color batches individually.

Table 5.6 Module K2: PSA results for color/setup-wise sorting, Experiment 1

Serial Number	Color	Job	Setup Required
1	Blue 1	P	A
2	Blue 2	FL	B
3	Blue 3	L	B
4	Black 1	SA	A
5	Black 2	FT	B

Table 5.7 CONWIP module scheduling for color group 1 (Blue), Experiment 1

Serial Number	Job	Start Time	Finish Time
1	FL	00.00	29.37
2	P	29.37	58.75
3	P	58.75	88.12
4	P	88.12	117.50
5	FL	117.50	146.87
6	P	146.87	176.25
7	L	176.25	205.62
8	L	205.62	235.00
9	P	235.00	264.37
10	P	264.37	293.75
11	P	293.75	323.12
12	L	323.12	352.50
13	FL	352.50	381.87
14	L	381.87	411.25
15	L	411.25	440.62
16	L	440.62	470.00

Table 5.7 gives optimum production schedule for blue colored level-2 components which are processed along the CONWIP line. All of these parts follow the same sequence of Fabrication – Shot Blasting – Paint Shop operations. Component FL has high production times (AMKEL11.LNG). Hence it is distributed evenly over the total time period. The schedule is optimum for given production and holding costs.

Table 5.8 is the output for the black color batch CONWIP scheduling. In this module for two black colored jobs, significance of CONWIP module is more clear. Because of a higher backorder cost and comparatively lesser finished goods holding cost, component SA is finished first and then component FT is scheduled. This saves the manufacturer from the shortage cost that would be incurred if the parts were not produced in this sequence.

Finally, the kanban quantities obtained for all level-2 components (and level-1 components like TU and GU) are displayed in Table 5.9. The basis for this model is the schedule given by the earlier shop and MRP quantities from K2 module. As can be seen, the difference in production periods for K2 and K1 outputs is apparent because of the predecessor-successor job relationships. Hence, this table represents the final output for the shop supervisor to start the production with quantities and periods as given in the table.

Table 5.8 CONWIP module scheduling for color group 2 (Black), Experiment 1

Serial Number	Job	Start Time	Finish Time
1	SA	00.00	23.42
2	FT	23.42	46.85
3	SA	46.85	70.28
4	SA	70.28	93.71
5	SA	93.71	117.14
6	SA	117.14	140.57
7	SA	140.57	164.00
8	SA	164.00	187.42
9	FT	187.42	210.85
10	SA	210.85	234.28
11	FT	234.28	257.71
12	FT	257.71	281.14
13	FT	281.14	304.57
14	FT	304.57	328.00

Table 5.9 Module K1: Kanban quantities obtained, Experiment 1

Serial Number	Job Name	Kanban Quantities (Period-wise)					
		P1	P2	P3	P4	P5	P6
1	P	0	0	0	3	0	0
2	FL	0	0	0	3	0	0
3	L	0	0	4	3	0	0
4	FT	0	0	3	0	0	0
5	SA	0	0	4	4	0	0
6	TU	0	0	2	4	0	0
7	GU	0	0	2	4	0	0

5.3 Experiment 2

Input data used in this experiment represent a typical year-end demand pattern of the industry considered. In the final quarter of the year there usually is a high demand for all products including spares.

This section is divided into two parts broadly viz. input and output as done in Section 5.2. Section 5.3.1 covers the information about various input data and basic capacity assumptions. Section 5.3.2 presents detailed analysis of results obtained from the calculations.

5.3.1 Input Data

Input data are shown from Tables 5.10 to 5.12. The arrangement of the output tables is in the same sequence as the working of their respective modules. They are presented in the reverse direction starting with module K2.

Data in Table 5.10 is input into the MRP program (File: K2MRP2.LDT) to calculate the total demand for all levels in the product BOM. Tables 5.11 and 5.12 are the processing time and cost data for the parts and subassemblies considered in the experiment. In Table 5.10, accumulated demand in final period for the main product as well as spares is observed. Amongst the spares, both level-1 and level-2 items are demanded, which is quite realistic.

Table 5.10 Module K2: Demand from Sales Department, Experiment 2

Serial Number	Job Name	Demand Quantities (Period P6)
1	HH	2
2	F	1
3	FT	2
4	FL	2
5	SA	1
6	L	1

Table 5.11 Process attributes for 7 sub-assemblies, module K2, Experiment 2

Serial Number	Attribute (Minutes)	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7
1	Setup Time	20	25	10	14	17	30	10
2	Assembly Time	1	1	3	3	3	2	2
3	Setup Cost	6	5	4	5	10	6	2
4	Assembly Cost	2	2	2	3	1	1	1
5	Holding Cost	1	1	1	2	1	1	1

Table 5.12 Process attributes for 7 sub-assemblies, module K1, Experiment 2

Serial Number	Attribute (Minutes)	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7
1	Setup Time	36	39	41	35	20	23	34
2	Production Time	12	22	11	21	11	13	11
3	Setup Cost	12	20	30	21	10	10	20
4	Production Cost	8	7	8	9	6	5	4
5	Holding Cost	11	11	11	12	11	11	12

Data from table 5.11 is used by K2 module (K2C2.LNG) for calculating kanban quantities for level-1 items in Assembly Shop. Capacity of Assembly Shop is 100 minutes per period. This is different than that of the Machine and Fabrication shop capacities because of the same reasons explained in Section 5.2.

Table 5.12 is used by module K1 for calculating kanban quantities of all level-2 items (and level-1 items : components TU and GU) in the BOM handled by this module. Calculations are based on their MRP demand (File:K1C2.LNG). Machine Shop and Fabrication Shop capacity is 480 minutes (1 time period).

5.3.2 Results and Analysis of Experiment 2

Output of MRP program in module K2 is shown in Table 5.13. These data then in turn were used as input to the kanban module to calculate kanban quantities in Assembly Shop (module K2). Main quantitative results are summarized and presented in Tables 5.13 to 5.18.

Table 5.13 presents the output of the MRP program. The total demand is obtained for all the required components. The demand is observed in the last period in line with the input of Table 5.10. Periods 1 to 5 have no demand as shown in this table. This is to compare the final production schedule obtained from K1 module.

Table 5.13 Module K2: MRP quantities obtained, Experiment 2

Serial Number	Job Name	MRP Quantities (period-wise)					
		P1	P2	P3	P4	P5	P6
1	HH	0	0	0	0	0	2
2	F	0	0	0	0	0	3
3	FT	0	0	0	0	0	5
4	FC	0	0	0	0	0	2
5	FL	0	0	0	0	0	5
6	SA	0	0	0	0	0	5
7	H	0	0	0	0	0	4
8	HP	0	0	0	0	0	2
9	SL	0	0	0	0	0	2
10	P	0	0	0	0	0	2
11	L	0	0	0	0	0	5
12	TU	0	0	0	0	0	2
13	GU	0	0	0	0	0	2

Table 5.14 Module K2:Kanban quantities obtained, Experiment 2

Serial Number	Job Name	Kanban Quantities (period-wise)					
		P1	P2	P3	P4	P5	P6
1	F	0	0	0	0	0	3
2	FC	0	0	0	0	2	0
3	H	0	0	0	0	0	4
4	HP	0	0	0	0	0	2
5	SL	0	0	0	0	2	0
6	TU	0	0	0	0	2	0
7	GU	0	0	0	0	0	2

Table 5.14 gives the kanban quantities estimated for level-1 components by module K2. Because of the higher total assembly costs in their initial time periods (CCOF from output file AK2C2), components F, H, HP and GU are assembled in the last possible assembly period ensuring that delay is just avoided. Assembly costs for components FC, SL, TU are lower in the initial periods. Hence, they are processed earlier. Thus, the total shop capacity is utilized uniformly by the above two groups of assembly in this case.

Table 5.15 shows output of K2 module for the PSA run. The basic purpose of this algorithm is to sort the production quantities into colors and then minimize the setups required in Shot Blasting Shop (Appendix B). The blue part list is followed by black and again toggle is made. This is used as input to the algorithm. Then setup-wise sorting is done among these color batches as can be seen in Table 5.15. The importance of PSA is obvious when dealing with a large number of components with variety of setups. This output is further used by CONWIP module to schedule the backorder list components in each of these color batches individually.

Table 5.15 Module K2: PSA results for color/setup-wise sorting, Experiment 2

Serial Number	Color	Job	Setup Required
1	Blue 1	P	A
2	Blue 2	FL	B
3	Blue 3	L	B
4	Black 1	SA	A
5	Black 2	FT	B

Table 5.16 CONWIP module scheduling for color group 1 (Blue), Experiment 2

Serial Number	Job	Start Time	Finish Time
1	FL	00.00	27.25
2	FL	27.25	54.50
3	FL	54.50	81.75
4	L	81.75	109.00
5	FL	109.00	136.25
6	L	136.25	163.50
7	FL	163.50	190.75
8	L	190.75	218.00
9	L	218.00	245.25
10	P	245.25	272.50
11	P	272.50	299.75
12	L	299.75	327.00

Table 5.16 gives optimum production schedule for blue colored level-2 components which are processed along the CONWIP line. Because of the almost same operation timings, jobs are evenly distributed on the basis of their inventory costs. All of these parts follow a same sequence of Fabrication – Shot Blasting – Paint Shop operations. Component FL has high production times (AMKEL12.LNG). Hence it is distributed evenly over the total time period. The schedule is optimum for given production and holding costs. First component gets scheduled at time zero and then the rest follow as per the CONWIP philosophy.

Table 5.17 is the output for the CONWIP scheduling of black color batch. Similar backorder and finished goods holding costs compel uniform production of these two components scheduled.

Finally, Table 5.18 shows the kanban quantities obtained for all level-2 components (and level-1 components like TU and GU) by module K1. The basis for this model is the schedule given by the earlier shop and MRP quantities from K2 module. In this module, shop capacity is 480 minutes per time period. It is best utilized by FL, L, FT and SA (total production time = 480 minutes). Also, because of less quantity (and hence low overall production cost), items like P, TU and GU are produced before or after this group. Holding cost for all of them is almost similar. So, that factor is not significant here. Also, component TU is made in period 5. It can be estimated as to how much more capacity is required if TU needs to be manufactured in time period 4.

Table 5.17 CONWIP module scheduling for color group 2 (Black), Experiment 2

Serial Number	Job	Start Time	Finish Time
1	FT	00.00	11.00
2	SA	11.00	22.00
3	SA	22.00	33.00
4	FT	33.00	44.00
5	FT	44.00	55.00
6	SA	55.00	66.00
7	FT	66.00	77.00
8	SA	77.00	88.00
9	FT	88.00	99.00
10	SA	99.00	110.00

Table 5.18 Module K1: Kanban quantities obtained, Experiment 2

Serial Number	Job Name	Kanban Quantities (Period-wise)					
		P1	P2	P3	P4	P5	P6
1	P	0	0	2	0	0	0
2	FL	0	0	0	5	0	0
3	L	0	0	0	5	0	0
4	FT	0	0	0	5	0	0
5	SA	0	0	0	5	0	0
6	TU	0	0	0	0	2	0
7	GU	0	0	2	0	0	0

5.4 Summary of Results

In the agricultural machinery industry, one would expect that there will be significant variations in production lead times because of uncertainties involved. Referring to the results in Sections 5.2.2 and 5.3.2 (Tables 5.5 to 5.9 and Tables 5.14 to 5.18), the production is observed to be smoothened over the given time periods, because of the minimization of setup costs involved. Many runs were made for different capacities (capacities can be observed in any program file in Appendix A). Because of the objective function making tradeoff between (setup + production) cost and inventory carrying cost, every time, a leveled production was achieved. After reducing a capacity below a certain limit, the LINGO stopped execution because of unfeasibility formed. Just before this condition, production got distributed unevenly in different time periods depending on whether setup costs or holding costs are higher. Thus, a secondary application of this model is found for capacity estimation purposes, i.e., to solve capacity related problems like requirement of total machine hours for a given demand within a given time period.

5.5 Implementation

As discussed earlier, the output information from one module is used as input by its previous module (Pull Production Control). The transfer of data is done manually at this stage. Once the model is implemented on a larger scale at the said facility, special software can be developed for total computerized integration. When fully automated, it will ask for the demand and due dates (for finished assembly and spares) as its input and give all the results (presented in result tables here) directly as its output.

Chapter Six

Conclusion

6.1 Summary

Through this research, a mathematical programming model is being presented for JIT implementation. A hybrid kanban-CONWIP approach is used for the proposed JIT setup. This study was done for a Canadian medium scale agricultural machinery manufacturer aiming at solving their production output related problems. We have attempted to minimize their production costs through this study by using the hybrid model.

Two cases were studied for the validation purpose. In both of the experiments, tendency of the model was to distribute the production evenly over the planning horizon and within resource capacity. It minimized the production, WIP and other inventory costs. It also attempted to solve the most cost-significant material intake flow problem of the Paint Shop in the factory. Kanban quantity calculations and CONWIP scheduling are achieved through MILP models. We have given general solution procedure also with reference to these two experiments.

This research when coupled with implementation refinements and efficient computerization, offers good potential for achieving planning benefits of MRP and capacitated lot sizing along with the simplicity of JIT operation. It also can be used as a tool for measuring the capacity of plant as against a particular product demand. The accuracy of this model depends on the accuracy of input data used. Even though only two cases were presented , a large number of demand and capacity variations were tested to verify the intended working logic of the model. In all cases, the behavior of the model was satisfactory as mentioned above.

6.2 Discussion

The JIT implementation is certainly not a simple task. It took Toyota over 20 years to have a fully developed JIT scheme. JIT scheduling and lean manufacturing require substantial changes not only to the mechanics of the systems and the work of its employees, but also to the corporate philosophy and the work culture. Cutting inventory stocks blindly will lead to delivery delays or reduced machine utilization rates. Martinich (1997) pointed out that although very high inventory reduction may take some time, it's not uncommon to observe some immediate benefits like 20-40% of inventory reduction and 5-10% of productivity growth per annum within the first 2-3 years. To conclude, he suggested three guidelines for a successful JIT implementation such as – (i) patient, flexible and adaptive behavior on part of the implementers, (ii) customized implementations and (iii) an excess capacity maintenance in the manufacturing facility. The view 'simpler is better' should be followed while designing JIT setup. 'Least possible resources set in the least complicated fashion' should be the motto. Karmarkar and Kekre (1987) established the basic four requirements for kanban (JIT) implementation. They are: Short lead times, Quick changeovers, Level-demand and Low setup costs. Applicability of JIT vis-à-vis MRP has to be studied carefully for every case individually. For example, consider a production system producing a wide variety of customer-specified products that are produced only twice a year. Hence, there is a clear possibility that, there is no WIP for that product at all or there is some being carried forward for not less than six months in the shop causing high inventory costs. Also, process type of industries (chemical plants, paper mills, food processing industries) pose a greater challenge in terms of setup time reduction and process flow improvement because of their connected operations.

In the coming years, hybrid concepts will be a powerful tool to exercise better control over the shop. More and more MRP-II software manufacturers, who are incorporating JIT features in it, support this prediction (Rao, 1989).

6.3 Future Work

Any type of research is an ongoing process in which a continuous improvement with respect to time is expected. The evaluation of such work should be done considering the usage of techniques and resources available at the time. With that perspective, the present study has certain limitations too. The choice of MILP models for minimization problems is common and appropriate because of the principles of Operations Research being so suitable for them.

Other possible options were also considered such as combinatorial optimization (COPT) techniques from computer science for solving the same problem using max-flow or mini-flow pipe network analogies. One might think that the scheduling CONWIP model could be constructed in the linear programming with COPT approach. But, COPT may give different schedules depending on the weights given to various constraints. Also, assigning weights is not at all a trivial task in itself. Such techniques can be used as generalized solution for a wide span of problems than being for a specialized ones. The features of COPT like heavy computer programming (requiring for very high capacity machines) made it somewhat unsuitable as applied to this particular research.

The other possible option would be to develop a CONWIP line for the entire plant. The main requirement for the CONWIP line is that the parts on that line must follow the same sequence of operations. Because of the high setup costs involved in the first two and last shops, kanban quantity calculations for minimum costs were suggested.

There is an emerging philosophy called 'Quick Response Manufacturing (QRM)' which helps to overcome some of the shortcomings of JIT (Suri, 1998). At the heart of it lie the queuing network theory and the aim of reducing lead times. It is an expansion of the time-based competition (TBC). It applies to all the areas of a company. New software's are getting developed on this concept, which are able to create a picture of a shop floor and address important trade-offs such as lot-sizes v/s setups and account for scrap and rework components. The basic function of the model is estimate for "what if" conditions with schedules developed for the optimum usage of costly resources. Hence, as development advances, a hybrid model between CONWIP, MRP-II and QRM might be useful, as QRM is flexible for combinations.

References

1. Bard, J., Golani, B., 1991, Determining the number of kanbans in a multi-product, multistage production system. *International Journal of Production Research*, **29**, 881-895.
2. Bitran, G., Chang, L., 1987, A mathematical programming approach to a deterministic kanban system. *Management Science*, **33** (4), 427-441.
3. Bonvik, A., Couch, C., Gershwin, S., 1997, A comparison of production-line control mechanisms. *International Journal of Production Research*, **35**, 789-804.
4. Bonvik, A., 1996, *How to control a lean manufacturing system*. (Research for manufacturing - Group 5, MIT, MA, USA)
5. Chang, D., Lee S. M., 1995, Impact of JIT on organizational performance of US firms. *International Journal of Production Research*, **33**, 3053-3068.
6. Clendenen, G. W., Rinks, D. B., 1996, The effect of labor costs, holding costs and uncertainty in demand on pull inventory control policies. *International Journal of Production Research*, **34**, 1725-1738.
7. Deleersnyder, J. L., 1988, Stochastic Models for Evaluating the effects of Information Flow Structures on the Hybrid Push-Pull System Performance. Paper presented at *The Fifth Euro Summer Institute on Production Planning and Control*.
8. Duncan, W., 1988, *Just-In-Time in American Manufacturing*. (Society of Manufacturing Engineers, MI, USA).
9. Goldratt, E. M., 1988, Computerized shop floor scheduling. *International Journal of Production Research*, **26**, 443-455.

10.Gstettner, G., Kuhn, H., 1996, Analysis of production control systems: kanban and CONWIP. *International Journal of Production Research*, **34**, 3253-3273.

11.Groenevelt, H., Karmarkar, U. S., 1988, A dynamic kanban system case study. *Production and Inventory Management Journal*, **29**, 46-51.

12.Hax, Candea, 1984, *Production and inventory management.* (Prentice-hall Inc., NJ, USA).

13.Herer, Y. T., Masin M., 1997, Mathematical programming formulation of CONWIP based production lines and relationships to MRP. *International Journal of Production Research*, **35**, 1067-1076.

14.Japan Management Association, 1989, *Kanban: Just-In-Time at Toyota - Management begins at the workplace.* (Productivity Press, MA, USA).

15.Karmarkar, U. S., and Kekre, S., 1987, Batching policy in kanban systems. *Working paper QM 8706, Graduate School of Business, University of Rochester, Rochester,* New York, USA.

16.Kim, T., 1985, Just-in-time manufacturing system: A periodic pull system. *International Journal of Production Research*, **23**, 553-562.

17.Kimball, G., 1988, General principles of inventory control. *Journal of Manufacturing and Operations Management*, **1**, 119-130.

18.Landers, T., Brown, W., Fant, E., Malstrom, E. and Schmitt, N., 1994, *Electronics manufacturing processes*. (Prentice-Hall Inc., USA)

19.*LINGO User's Guide*, 1995, (LINDO systems Inc., IL, USA).

20.Martinich, J. S., 1997, *Production and Operations Management – An Applied Modern Approach.* (John Wiley & Sons, Inc., NY, USA)

21.Miltenburg, J., 1997, Comparing JIT, MRP and TOC, and embedding TOC into MRP. *International Journal of Production Research*, **35**, 1147-1169.

22.Monden, Y., 1983, *Toyota Production System.* (Industrial Engineering and Management Press, IIE, GA, USA).

23.Philipoom et al., 1996, Simultaneously determining the number of kanbans, container sizes and the final assembly sequence of products in a just-in-time shop. *International Journal of Production Research*, **34**, 51-69.

24.Rao, A., 1989, A Survey of MRP-II Software Suppliers' Trends in Support of Just-In-Time. *Production and Inventory Management Journal*, **30**, 14-17.

25.Riggs, James, 1987, *Production Systems: Planning, Analysis, and Control.* (John Wiley and Sons, NY, USA).

26.Review Article, 1999, Aerospace industry mimics Toyota. *National Post (A Daily newspaper in Canada)*, Vol.**114** (March 10, 1999).

27.Salameh et al., 1984, Safety stock analysis in a multi-level inventory systems. *IIE Transactions*, **16**, 348-354.

28.Shmitt, 1984, *Journal of Operations Management*, **4**, 331-345.

29.Singh, Nanua, 1996, *Systems Approach To Computer-Integrated Design And Manufacturing.* (John Wiley and Sons, NY, USA)

30.Spearman, M. L., Hopp, W. J., Woodruff, D. L., 1989, A hierarchical control architecture for constant work-in-process (CONWIP) production systems. *Journal of Manufacturing and Operations Management*, **2**, 147-171.

31.Spearman, M. L., Hopp, W. J., Woodruff, D. L., 1990, CONWIP: A pull alternative to kanban. *International Journal of Production Research*, **28**, 879-894

32.Spearman, Zazanis, 1992, Push and pull production systems. *Operations Research*, **40**, 521-531.

33. Suri, Rajan, 1998, *Quick Response Manufacturing – A company-wide Approach to Reducing Lead Times*. (Productivity Press, MA, USA.)

34.Winston, W.L., 1991, *Operations Research: Applications and Algorithms*. (PWS-Kent Publishing, MA, USA)

APPENDICES

Appendix A: LINGO Programs for kanban and CONWIP modules

The present study uses Hyper LINGO/PC (Release 3.0) software package for programming and analyzing JIT setup performance for the production facility under consideration. Programming using LINGO involves Data file, Program file and the Output file.

1. The Data file is used to input the known attribute values required by the Program file for executing the program.
2. The Program file contains the actual mathematical formulations for objective function and constraints in LINGO syntax.
3. The results of every LINGO model can be saved in the form of an Output file for easy and better analysis and presentation.

This adjoining appendix presents the Data, Program and the Output files developed and used in this research. The section starts with a summary of all LINGO files.

In this appendix, a full set of Data, Program and Solution files for Experiment 1 is provided. There is no difference between Experiment 1 and Experiment 2 from the working logic point of view. They differ in input data/parameter values. Hence, same set can be referred to, for the programming logic of Experiment 2. For individual summaries, data and result values for both the experiments are recorded in Chapter 5.

CASE 1:

FILE	DESCRIPTION
K2MRP1.LDT	Module K2, MRP Data, Case 1
K2MRP1.LNG	Module K2, MRP Calculation Program, Case 1
AK2MRP1	Module K2, MRP Results, Case 1
K2C1.LNG	Module K2, Kanban Quantity Calculation Program, Case 1
AK2C1	Module K2, Kanban Results, Case 1
CDATA11.LDT	Module C, CONWIP Data, Group 1, Case 1
AMKEL11.LNG	Module C, CONWIP Calculation Program, Case 1
ACC11	Module C, CONWIP Schedule Results, Group 1, Case 1
CDATA21.LDT	Module C, CONWIP Data, Group 2, Case 1
AMKEL21.LNG	Module C, CONWIP Calculation Program, Case 1
ACC21	Module C, CONWIP Schedule Results, Group 2, Case 1
K1C1.LNG	Module K1, Kanban Quantity Calculation Program, Case1
AK1C1	Module K1, Kanban Results, Case 1

(Copies of all above Case 1 files attached in Appendix A)

CASE 2

FILE	DESCRIPTION
K2MRP2.LDT	Module K2, MRP Data, Case 2
K2MRP2.LNG	Module K2, MRP Calculation Program, Case 2
AK2MRP2	Module K2, MRP Results, Case 2
K2C2.LNG	Module K2, Kanban Quantity Calculation Program, Case 2
AK2C2	Module K2, Kanban Results, Case 2
CDATA12.LDT	Module C, CONWIP Data, Group 1, Case 2
AMKEL12.LNG	Module C, CONWIP Calculation Program, Case 2
ACC12	Module C, CONWIP Schedule Results, Group 1, Case 2
CDATA22.LDT	Module C, CONWIP Data, Group 2, Case 2
AMKEL22.LNG	Module C, CONWIP Calculation Program, Case 2
ACC22	Module C, CONWIP Schedule Results, Group 2, Case 2
K1C2.LNG	Module K1, Kanban Quantity Calculation Program, Case2
AK1C2	Module K1, Kanban Results, Case 2

K2MRP1.LDT

```
! Parts list;
  HH, ! Heavy Harrow;
  F,  ! Frame Subwldmet;
  FT, ! Frame Side Tube;
  FL, ! Frame Link Lock;
  FC, ! Frame Close Sub;
  H,  ! Harrow Tube;
  SA, ! Swing Arm;
  HP, ! Hyd. Hose Plate;
  SL, ! Safety Lock;
  P,  ! Pull Tube;
  L,  ! Lathe C Clamp;
  TU, ! Tube U Bracket;
  GU~ ! Support Gusset;

!  The set of periods;
  1..6 ~

! The parent-child use relationships (i.e. Heavy Harrow, Swing Arm, P...);
 HH F, HH FC, HH H, HH HP, HH SL, HH TU, HH GU, F FT, F FL, H SA, SL P,
 SL L ~

! The number of child parts required in each parent-child relationship;
 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2 ~

! The lead times for each part;
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~

! The external demands or master schedule;
! Time period;
!              1  2  3  4  5  6;
  ! HH;        0, 0, 0, 0, 2, 1,
  ! F ;        0, 0, 0, 0, 0, 2,
  ! FT;        0, 0, 0, 0, 0, 1,
  ! FL;        0, 0, 0, 0, 2, 0,
  ! FC;        0, 0, 0, 0, 0, 0,
  ! H ;        0, 0, 0, 0, 0, 0,
  ! SA;         0, 0, 0, 0, 0, 2,
  ! HP;        0, 0, 0, 0, 0, 0,
  ! SL;        0, 0, 0, 0, 0, 0,
  ! P;         0, 0, 0, 0, 0, 0,
  ! L;         0, 0, 0, 0, 0, 0,
  ! TU;        0, 0, 0, 0, 0, 0,
  ! GU;        0, 0, 0, 0, 0, 0
```

K2MRP1.LNG

```
MODEL:
! Data for specific K2MRP problem comes from K2MRP1.LDT;

SETS:
! The set of parts;
 PART/ @FILE( K2MRP1.LDT)/ : LT;
 ! LT(i) = Lead time to produce part i;

! The set of time periods;
 TIME / @FILE( K2MRP1.LDT)/;

! A relationship called USES between pairs of parts;
 USES( PART, PART) /@FILE( K2MRP1.LDT) / : NEEDS;
 ! Parent part i needs (i,j) units of
   child part j;

! For each part and time period we are interested in;
 PXT( PART, TIME): ED, TD;
 ! ED(i, j) = External demand for part i at time j;
 ! TD(i, j) = Total demand for part i at time j;
ENDSETS
! Set NP = no. of time periods in the problem;
 NP = @SIZE( TIME);

! For each part I and period J, the total demand =
  external demand + demand generated by parents
  one lead time in the future;
 @FOR( PXT( I, J) | J + LT( I) #LE# NP :
 TD(I, J) = ED(I, J + LT(I)) + @SUM( USES( PR, I):
  TD( PR, J + LT( I)) * NEEDS( PR, I));
);

DATA:
! Get no. parts needed in each parent child relation;
 NEEDS = @FILE( K2MRP1.LDT);

! Get the lead times from the file;
 LT  = @FILE( K2MRP1.LDT);

! Get the external demands over time for each part;
 ED  = @FILE( K2MRP1.LDT);

ENDDATA
 END
```

AK2MRP1

The output of MRP calculations after considering all HH and spares demands:

TD(HH, 5)	2.000000
TD(HH, 6)	1.000000
TD(F, 5)	2.000000
TD(F, 6)	3.000000
TD(FT, 5)	2.000000
TD(FT, 6)	4.000000
TD(FL, 5)	4.000000
TD(FL, 6)	3.000000
TD(FC, 5)	2.000000
TD(FC, 6)	1.000000
TD(H, 5)	4.000000
TD(H, 6)	2.000000
TD(SA, 5)	4.000000
TD(SA, 6)	4.000000
TD(HP, 5)	2.000000
TD(HP, 6)	1.000000
TD(SL, 5)	2.000000
TD(SL, 6)	1.000000
TD(P, 5)	2.000000
TD(P, 6)	1.000000
TD(L, 5)	4.000000
TD(L, 6)	2.000000
TD(TU, 5)	2.000000
TD(TU, 6)	1.000000
TD(GU, 5)	2.000000
TD(GU, 6)	1.000000

K2C1.LNG

```
MODEL:
! KANBAN QUANTITY CALCULATION FOR MODULE K2;
SETS:
 PROD/1..7/:          ! Each subassly has a :...;
 ST,                  ! Assembly Setup time;
 AT,                  ! Assembly time per unit;
 SC,                  ! Setup cost for individual assly's;
 AC,                  ! Assembly cost per unit;
 HC;                  ! Holding cost per unit per period;
 TIME/1..6/:;         ! Total time periods considered;
 PXT(PROD,TIME): ! Each product in each period has...;
 DEM,                 ! Demand;
 KANBAN2,             ! Optimum kanban qty.to assemble;
 Y;                   ! = 1 if anything is assembled;

ENDSETS

DATA:
 CAP = 100;                          ! Capacity per period;
 ST = 20 25 10 14 17 30 10;          ! Setup times;
 AT = 1 1 3 3 3 2 2;                 ! Assembly time per subassly.;
 SC = 6 5 4 5 10 6 2;                ! Setup costs;
 AC = 2 2 2 3 1 1 1;                 ! Cost per unit to assemble;
 HC = 1 1 1 2 1 1 1;                 ! Holding cost per unit;
 DEM = ! Demands;

     0 0 0 0 2 3                     ! Frame Sub Wldmt. (F);
     0 0 0 0 2 1                     ! Frame Close sub. (FC);
     0 0 0 0 4 2                     ! Harrow Tube (H);
     0 0 0 0 2 1                     ! Hyd. Hose Plate (HP);
     0 0 0 0 2 1                     ! Safety Lock (SL);
     0 0 0 0 2 1                     ! Tube U Bracket (TU);
     0 0 0 0 2 1;                    ! Support Gusset (GU);


ENDDATA
!-----------------------------------------------------------------;
SETS:
 PXTXT( PROD, TIME, TIME)| &3 #GE# &2:
  PCOF,
  CCOF,
  X;
ENDSETS
! Compute cost and assembly Kanban quantities for various production runs;
```

```
@FOR( PROD( I):
 @FOR( TIME( S):
   PCOF( I, S, S) = DEM( I, S);
   CCOF( I, S, S) = AC( I) * DEM( I, S);
   @FOR( TIME( T)| T #GT# S:
    PCOF( I, S, T) = PCOF( I, S, T - 1) +
      DEM( I, T);
    CCOF( I, S, T) = CCOF( I, S, T - 1) +
     ( AC( I) + HC( I) * ( T - S)) * DEM( I, T);
    );
  ););
! The objective;
 MIN = TCOST;
 TCOST = @SUM( PXTXT: CCOF * X) +
      @SUM( PXT( I, S): SC( I) * Y( I, S));

 @FOR( PROD( I):
!  In period 1, some production run must be started;
!  Note, watch out for periods without demand;
   @SUM( PXTXT( I, S, T) | S #EQ# 1:
     X( I, S, T)) = 1;
   @FOR( TIME( K)| K #GT# 1:
!  If we ended a run in period K - 1...;
     @SUM( PXTXT( I, S, T)| T #EQ# K - 1: X( I, S, K - 1))
!  then we must start a run in period k;
    = @SUM( PXTXT( I, K, T): X( I, K, T));
     );
!  Setup forcing;
  @FOR( TIME( S):
    Y( I, S) =  @SUM( PXTXT( I, S, T)
         : ( PCOF( I, S, T) #GT# 0) * X( I, S, T));
!  Calculate the Kanban qty. made each period;
    KANBAN2( I, S) = @SUM( PXTXT( I, S, T):
      PCOF( I, S, T) * X( I, S, T)); );
    );

!  The capacity constraints;
  @FOR( TIME( S):
   @SUM( PROD( I): ST( I) * Y( I, S)) +
    @SUM( PXTXT( I, S, T):
    AT( I) * PCOF( I, S, T) * X( I, S, T)) <= CAP;
    );
!  Make the Y's integer;
   @FOR( PXT: @GIN( Y););

  END
```

AK2C1

Rows= 134 Vars= 232 No. integer vars= 42 (all are linear)
Nonzeros= 1023 Constraint nonz= 729(423 are +- 1) Density=0.033
Smallest and largest elements in absolute value= 1.00000 100.000
No. < : 6 No. =: 127 No. > : 0, Obj=MIN, GUBs <= 59
Single cols= 42

Optimal solution found at step: 175
Objective value: 104.0000
Branch count: 3

Variable	Value	Reduced Cost
CAP	100.0000	0.0000000E+00
TCOST	104.0000	0.0000000E+00
KANBAN2(1, 5)	5.000000	0.0000000E+00
KANBAN2(2, 4)	3.000000	0.0000000E+00
KANBAN2(3, 5)	6.000000	0.0000000E+00
KANBAN2(4, 5)	3.000000	0.0000000E+00
KANBAN2(5, 4)	3.000000	0.0000000E+00
KANBAN2(6, 4)	3.000000	0.0000000E+00
KANBAN2(7, 5)	3.000000	0.0000000E+00
PCOF(1, 1, 5)	2.000000	0.0000000E+00
PCOF(1, 1, 6)	5.000000	0.0000000E+00
PCOF(1, 2, 5)	2.000000	0.0000000E+00
PCOF(1, 2, 6)	5.000000	0.0000000E+00
PCOF(1, 3, 5)	2.000000	0.0000000E+00
PCOF(1, 3, 6)	5.000000	0.0000000E+00
PCOF(1, 4, 5)	2.000000	0.0000000E+00
PCOF(1, 4, 6)	5.000000	0.0000000E+00
PCOF(1, 5, 5)	2.000000	0.0000000E+00
PCOF(1, 5, 6)	5.000000	0.0000000E+00
PCOF(1, 6, 6)	3.000000	0.0000000E+00
PCOF(2, 1, 5)	2.000000	0.0000000E+00
PCOF(2, 1, 6)	3.000000	0.0000000E+00
PCOF(2, 2, 5)	2.000000	0.0000000E+00
PCOF(2, 2, 6)	3.000000	0.0000000E+00
PCOF(2, 3, 5)	2.000000	0.0000000E+00
PCOF(2, 3, 6)	3.000000	0.0000000E+00
PCOF(2, 4, 5)	2.000000	0.0000000E+00
PCOF(2, 4, 6)	3.000000	0.0000000E+00
PCOF(2, 5, 5)	2.000000	0.0000000E+00

PCOF(2, 5, 6)	3.000000	0.0000000E+00
PCOF(2, 6, 6)	1.000000	0.0000000E+00
PCOF(3, 1, 5)	4.000000	0.0000000E+00
PCOF(3, 1, 6)	6.000000	0.0000000E+00
PCOF(3, 2, 5)	4.000000	0.0000000E+00
PCOF(3, 2, 6)	6.000000	0.0000000E+00
PCOF(3, 3, 5)	4.000000	0.0000000E+00
PCOF(3, 3, 6)	6.000000	0.0000000E+00
PCOF(3, 4, 5)	4.000000	0.0000000E+00
PCOF(3, 4, 6)	6.000000	0.0000000E+00
PCOF(3, 5, 5)	4.000000	0.0000000E+00
PCOF(3, 5, 6)	6.000000	0.0000000E+00
PCOF(3, 6, 6)	2.000000	0.0000000E+00
PCOF(4, 1, 5)	2.000000	0.0000000E+00
PCOF(4, 1, 6)	3.000000	0.0000000E+00
PCOF(4, 2, 5)	2.000000	0.0000000E+00
PCOF(4, 2, 6)	3.000000	0.0000000E+00
PCOF(4, 3, 5)	2.000000	0.0000000E+00
PCOF(4, 3, 6)	3.000000	0.0000000E+00
PCOF(4, 4, 5)	2.000000	0.0000000E+00
PCOF(4, 4, 6)	3.000000	0.0000000E+00
PCOF(4, 5, 5)	2.000000	0.0000000E+00
PCOF(4, 5, 6)	3.000000	0.0000000E+00
PCOF(4, 6, 6)	1.000000	0.0000000E+00
PCOF(5, 1, 5)	2.000000	0.0000000E+00
PCOF(5, 1, 6)	3.000000	0.0000000E+00
PCOF(5, 2, 5)	2.000000	0.0000000E+00
PCOF(5, 2, 6)	3.000000	0.0000000E+00
PCOF(5, 3, 5)	2.000000	0.0000000E+00
PCOF(5, 3, 6)	3.000000	0.0000000E+00
PCOF(5, 4, 5)	2.000000	0.0000000E+00
PCOF(5, 4, 6)	3.000000	0.0000000E+00
PCOF(5, 5, 5)	2.000000	0.0000000E+00
PCOF(5, 5, 6)	3.000000	0.0000000E+00
PCOF(5, 6, 6)	1.000000	0.0000000E+00
PCOF(6, 1, 5)	2.000000	0.0000000E+00
PCOF(6, 1, 6)	3.000000	0.0000000E+00
PCOF(6, 2, 5)	2.000000	0.0000000E+00
PCOF(6, 2, 6)	3.000000	0.0000000E+00
PCOF(6, 3, 5)	2.000000	0.0000000E+00
PCOF(6, 3, 6)	3.000000	0.0000000E+00
PCOF(6, 4, 5)	2.000000	0.0000000E+00
PCOF(6, 4, 6)	3.000000	0.0000000E+00
PCOF(6, 5, 5)	2.000000	0.0000000E+00
PCOF(6, 5, 6)	3.000000	0.0000000E+00
PCOF(6, 6, 6)	1.000000	0.0000000E+00

PCOF(7, 1, 5)	2.000000	0.0000000E+00
PCOF(7, 1, 6)	3.000000	0.0000000E+00
PCOF(7, 2, 5)	2.000000	0.0000000E+00
PCOF(7, 2, 6)	3.000000	0.0000000E+00
PCOF(7, 3, 5)	2.000000	0.0000000E+00
PCOF(7, 3, 6)	3.000000	0.0000000E+00
PCOF(7, 4, 5)	2.000000	0.0000000E+00
PCOF(7, 4, 6)	3.000000	0.0000000E+00
PCOF(7, 5, 5)	2.000000	0.0000000E+00
PCOF(7, 5, 6)	3.000000	0.0000000E+00
PCOF(7, 6, 6)	1.000000	0.0000000E+00
CCOF(1, 1, 5)	12.00000	0.0000000E+00
CCOF(1, 1, 6)	33.00000	0.0000000E+00
CCOF(1, 2, 5)	10.00000	0.0000000E+00
CCOF(1, 2, 6)	28.00000	0.0000000E+00
CCOF(1, 3, 5)	8.000000	0.0000000E+00
CCOF(1, 3, 6)	23.00000	0.0000000E+00
CCOF(1, 4, 5)	6.000000	0.0000000E+00
CCOF(1, 4, 6)	18.00000	0.0000000E+00
CCOF(1, 5, 5)	4.000000	0.0000000E+00
CCOF(1, 5, 6)	13.00000	0.0000000E+00
CCOF(1, 6, 6)	6.000000	0.0000000E+00
CCOF(2, 1, 5)	12.00000	0.0000000E+00
CCOF(2, 1, 6)	19.00000	0.0000000E+00
CCOF(2, 2, 5)	10.00000	0.0000000E+00
CCOF(2, 2, 6)	16.00000	0.0000000E+00
CCOF(2, 3, 5)	8.000000	0.0000000E+00
CCOF(2, 3, 6)	13.00000	0.0000000E+00
CCOF(2, 4, 5)	6.000000	0.0000000E+00
CCOF(2, 4, 6)	10.00000	0.0000000E+00
CCOF(2, 5, 5)	4.000000	0.0000000E+00
CCOF(2, 5, 6)	7.000000	0.0000000E+00
CCOF(2, 6, 6)	2.000000	0.0000000E+00
CCOF(3, 1, 5)	24.00000	0.0000000E+00
CCOF(3, 1, 6)	38.00000	0.0000000E+00
CCOF(3, 2, 5)	20.00000	0.0000000E+00
CCOF(3, 2, 6)	32.00000	0.0000000E+00
CCOF(3, 3, 5)	16.00000	0.0000000E+00
CCOF(3, 3, 6)	26.00000	0.0000000E+00
CCOF(3, 4, 5)	12.00000	0.0000000E+00
CCOF(3, 4, 6)	20.00000	0.0000000E+00
CCOF(3, 5, 5)	8.000000	0.0000000E+00
CCOF(3, 5, 6)	14.00000	0.0000000E+00
CCOF(3, 6, 6)	4.000000	0.0000000E+00
CCOF(4, 1, 5)	22.00000	0.0000000E+00
CCOF(4, 1, 6)	35.00000	0.0000000E+00

CCOF(4, 2, 5)	18.00000	0.0000000E+00
CCOF(4, 2, 6)	29.00000	0.0000000E+00
CCOF(4, 3, 5)	14.00000	0.0000000E+00
CCOF(4, 3, 6)	23.00000	0.0000000E+00
CCOF(4, 4, 5)	10.00000	0.0000000E+00
CCOF(4, 4, 6)	17.00000	0.0000000E+00
CCOF(4, 5, 5)	6.000000	0.0000000E+00
CCOF(4, 5, 6)	11.00000	0.0000000E+00
CCOF(4, 6, 6)	3.000000	0.0000000E+00
CCOF(5, 1, 5)	10.00000	0.0000000E+00
CCOF(5, 1, 6)	16.00000	0.0000000E+00
CCOF(5, 2, 5)	8.000000	0.0000000E+00
CCOF(5, 2, 6)	13.00000	0.0000000E+00
CCOF(5, 3, 5)	6.000000	0.0000000E+00
CCOF(5, 3, 6)	10.00000	0.0000000E+00
CCOF(5, 4, 5)	4.000000	0.0000000E+00
CCOF(5, 4, 6)	7.000000	0.0000000E+00
CCOF(5, 5, 5)	2.000000	0.0000000E+00
CCOF(5, 5, 6)	4.000000	0.0000000E+00
CCOF(5, 6, 6)	1.000000	0.0000000E+00
CCOF(6, 1, 5)	10.00000	0.0000000E+00
CCOF(6, 1, 6)	16.00000	0.0000000E+00
CCOF(6, 2, 5)	8.000000	0.0000000E+00
CCOF(6, 2, 6)	13.00000	0.0000000E+00
CCOF(6, 3, 5)	6.000000	0.0000000E+00
CCOF(6, 3, 6)	10.00000	0.0000000E+00
CCOF(6, 4, 5)	4.000000	0.0000000E+00
CCOF(6, 4, 6)	7.000000	0.0000000E+00
CCOF(6, 5, 5)	2.000000	0.0000000E+00
CCOF(6, 5, 6)	4.000000	0.0000000E+00
CCOF(6, 6, 6)	1.000000	0.0000000E+00
CCOF(7, 1, 5)	10.00000	0.0000000E+00
CCOF(7, 1, 6)	16.00000	0.0000000E+00
CCOF(7, 2, 5)	8.000000	0.0000000E+00
CCOF(7, 2, 6)	13.00000	0.0000000E+00
CCOF(7, 3, 5)	6.000000	0.0000000E+00
CCOF(7, 3, 6)	10.00000	0.0000000E+00
CCOF(7, 4, 5)	4.000000	0.0000000E+00
CCOF(7, 4, 6)	7.000000	0.0000000E+00
CCOF(7, 5, 5)	2.000000	0.0000000E+00
CCOF(7, 5, 6)	4.000000	0.0000000E+00
CCOF(7, 6, 6)	1.000000	0.0000000E+00

CDATA11.LDT

```
!TIME PERIODS;
  1..2~

  !PART TYPES;
  1..3~

  !MACHINES (SHOT BLASTING AND PAINTING);
  1..2~

  !NUMBER OF COMPONENTS IN BACKLOG LIST;
  1..16~

  !NUMBER OF CONTAINERS;
  1..3~

  !PART_SEQ;
  1..3~

  !PART-FOLL_PART-MACHINE;
  1,2,1 1,2,2 1,3,1 1,3,2
  2,1,1 2,1,2 2,3,1 2,3,2
  3,1,1 3,1,2 3,2,1 3,2,2~
```

AMKEL11.LNG

```
MODEL:
SETS:
        TIME_P /@FILE(CDATA11.LDT) /;
        PART /@FILE(CDATA11.LDT) /;
        MACHINE /@FILE(CDATA11.LDT) /;
        N_BACKLOG /@FILE(CDATA11.LDT) /;
        CONTAINER /@FILE(CDATA11.LDT) /;
        FOLL_PART /@FILE(CDATA11.LDT) /;
        RQSETUP(PART,FOLL_PART,MACHINE) /@FILE(CDATA11.LDT) /;


      PART_MACH(PART,MACHINE): P;
      SWSETUP_T(RQSETUP): TT;
      PARTS(PART):NN,WHC,FHC,BC,SPLUS0,SM0,W0;
      PART_TIME(PART,TIME_P): D,SPLUS,SMINUS,W;
      BACK_MACH(N_BACKLOG,MACHINE):DELT1,DELT2;
      MACH_CONT(MACHINE,CONTAINER): AT;
      MACH(MACHINE): MP;
      AVERAGE_THR(CONTAINER):BETA;
      PART_BACK(PART,N_BACKLOG): X;
      PART_BACK1(FOLL_PART,N_BACKLOG):X1;
      RELEASE_T(PART, N_BACKLOG): RK;
      FINISH_T(PART, N_BACKLOG): FK;
      ZKT(RELEASE_T,TIME_P): Z;
      FKT(FINISH_T,TIME_P): F;
      VIKT(ZKT): V;
      UIKT(FKT): U;
      YIJK(RQSETUP): Y;


ENDSETS
DATA:
     !THE TOTAL NUMBER OF CONTAINERS;
     NN = 1,1,1;

        !HOLDING COST OF WIP INVENTORY;
        WHC = 1,2,1;

        !HOLDING COST OF FINISHED GOODS;
        FHC = 2,2,2;

        !BACKORDER COST OF FINISHED GOODS;
        BC = 1,1,2;
```

```
!INITIAL FGI AND BACKORDERS;
SPLUS0 = 0,0,0;

SM0 = 0,0,0;

W0= 0,0,0;

!PROCESSING TIME;
P = 3,2
    4,3
    3,2;

!SEQUENCE DEPENDENT SET-UP TIME FOR SWITCHING PART I TO J;
TT = 2,4  1,2
     2,4  1,2
     1,2  1,2;


!PART DEMAND FOR TIME PERIODS 1 TO 2;

D = 4,3
    2,1
    4,2 ;


ENDDATA

MIN =@SUM(TIME_P(T): @SUM(PART(I): FHC(I)*SPLUS(I,T)))
+@SUM(TIME_P(T): @SUM(PART(I): BC(I)*SMINUS(I,T)))
+@SUM(TIME_P(T): @SUM(PART(I): WHC(I)*W(I,T)));


@FOR (PART_TIME(I,T)|T #EQ# 1:
 SPLUS0(I)-SM0(I)
+@SUM(FKT(I,K,T):U(I,K,T))
-D(I,T)=SPLUS(I,1)-SMINUS(I,1)
);

@FOR (PART_TIME(I,T)|T #NE# 1:
SPLUS(I,T-1)-SMINUS(I,T-1)+@SUM(FKT(I,K,T):U(I,K,T))
-D(I,T)=SPLUS(I,T)-SMINUS(I,T)
);

@FOR (PART_TIME(I,T)|T #EQ#1:
@FOR (PART_BACK(I,T):
W0(I)+@SUM(ZKT(I,K,T): V(I,K,T))
```

```
-@SUM(FKT(I,K,T):U(I,K,T))=W(I,1)
));

@FOR (PART_TIME(I,T)|T #NE#1:
@FOR (PART_BACK(I,T):
W(I,T-1)+@SUM(ZKT(I,K,T): V(I,K,T))
-@SUM(FKT(I,K,T):U(I,K,T))=W(I,T)
));

@FOR(PART(I):
@SUM(PART_BACK(I,K): X(I,K))=@SUM(TIME_P(T): D(I,T))
);

@FOR(N_BACKLOG(K):
@SUM(PART_BACK(I,K):X(I,K))=1
);

@FOR(PART_BACK(I,K):
@FOR(ZKT(I,K,T):
X(I,K)-1.5<=V(I,K,T);
X(I,K)>=1.5*V(I,K,T)));

@FOR(PART_BACK(I,K):
@FOR(FKT(I,K,T):
X(I,K)-1.5<=U(I,K,T);
X(I,K)>=1.5*U(I,K,T)));


@FOR(PART_BACK1(J,K)|K #NE# 1:
@FOR(YIJK(I,J,M):
X(I,K-1)+X1(J,K)-1.5<=Y(I,J,M);
X(I,K-1)+X1(J,K)>=1.5*Y(I,J,M)));

@FOR(MACHINE(M):
@FOR(N_BACKLOG(K):
DELT1(K,M)=@SUM(PART_BACK(I,K):X(I,K)*P(I,M));
));


@FOR(MACHINE(M):
@FOR(N_BACKLOG(K):
DELT2(K,M)=@SUM(PART_BACK(J,K):
@SUM(YIJK(I,J,M):Y(I,J,M)*TT(I,J,M)))
));
```

```
@FOR(MACHINE(M):
MP(M)=@SUM(N_BACKLOG(K):DELT1(K,M)+DELT2(K,M))
/(@SUM(PART(I):@SUM(TIME_P(T):D(I,T))))
);

@FOR(MACHINE(M):
AT(M,1)=MP(M)
);

@FOR(CONTAINER(N):
BETA(N)=@SUM(MACHINE(M):AT(M,N))
);

@FOR(CONTAINER(N)|N #NE#1:
@FOR(MACHINE(M):
AT(M,N)*BETA(N-1)=MP(M)*(BETA(N-1)+AT(M,N-1)*(N-1));
));

@FOR(RELEASE_T(I,K):
@FOR(N_BACKLOG(K):
@FOR(CONTAINER(N)|N #EQ# NN(I):
RK(I,K)=(K-1)*@SUM(MACHINE(M): AT(M,N))/NN(I)
)));
@FOR(FINISH_T(I,K):
@FOR(N_BACKLOG(K):
@FOR(CONTAINER(N)|N #EQ# NN(I):
FK(I,K)=RK(I,K)+@SUM(MACHINE(M):AT(M,N))
)));

@FOR(PART_BACK(I,K):
@BIN(X(I,K)));
@FOR(PART_BACK(I,K):
@FOR(VIKT(I,K,T):
@BIN(V(I,K,T))));
@FOR(PART_BACK(I,K):
@FOR(UIKT(I,K,T):
@BIN(U(I,K,T))));
@FOR(PART_BACK1(J,K):
@FOR(YIJK(I,J,M):
@BIN(Y(I,J,M))

));

END
```

ACC1

Rows= 947 | Vars= 681 | No. integer vars= 252
Nonlinear rows= 4 | Nonlinear vars= 10 | Nonlinear constraints= 4
Nonzeros= 3457 | Constraint nonz= 3036 | Density=0.005

Rows= 947	Vars= 681	No. integer vars= 252
Nonlinear rows= 4	Nonlinear vars= 10	Nonlinear constraints= 4
Nonzeros= 3457	Constraint nonz= 3036	Density=0.005

Optimal solution found at step: 300
Objective value: 36.00000
Branch count: 22

Variable	Value	Reduced Cost
P(1, 1)	3.000000	0.0000000E+00
P(1, 2)	2.000000	0.0000000E+00
P(2, 1)	4.000000	0.0000000E+00
P(2, 2)	3.000000	0.0000000E+00
P(3, 1)	3.000000	0.0000000E+00
P(3, 2)	2.000000	0.0000000E+00
AT(1, 1)	11.18750	0.0000000E+00
AT(1, 2)	15.44827	0.0000000E+00
AT(1, 3)	18.88642	0.0000000E+00
AT(2, 1)	18.18750	0.0000000E+00
AT(2, 2)	29.44827	0.0000000E+00
AT(2, 3)	42.04637	0.0000000E+00
MP(1)	11.18750	0.0000000E+00
MP(2)	18.18750	0.0000000E+00
BETA(1)	29.37500	0.0000000E+00
BETA(2)	44.89654	0.0000000E+00
BETA(3)	60.93280	0.0000000E+00
X(1, 2)	1.000000	0.0000000E+00
X(1, 3)	1.000000	0.0000000E+00
X(1, 4)	1.000000	0.0000000E+00
X(1, 6)	1.000000	0.0000000E+00
X(1, 9)	1.000000	0.0000000E+00
X(1, 10)	1.000000	0.0000000E+00
X(1, 11)	1.000000	0.0000000E+00
X(2, 1)	1.000000	0.0000000E+00
X(2, 5)	1.000000	0.0000000E+00
X(2, 13)	1.000000	0.0000000E+00
X(3, 7)	1.000000	0.0000000E+00
X(3, 8)	1.000000	0.0000000E+00
X(3, 12)	1.000000	0.0000000E+00
X(3, 14)	1.000000	0.0000000E+00
X(3, 15)	1.000000	0.0000000E+00
X(3, 16)	1.000000	0.0000000E+00

CDATA21.LDT

```
!TIME PERIODS;
        1..2~

        !PART TYPES;
        1..2~

        !MACHINES (SHOT BLASTING AND PAINTING);
        1..2~

        !NUMBER OF COMPONENTS IN BACKLOG LIST;
        1..14~

        !NUMBER OF CONTAINERS;
        1..2~

    !PART_SEQ;
    1..2~

        !PART-FOLL_PART-MACHINE;
        1,2,1 1,2,2
        2,1,1 2,1,2~
```

AMKEL21.LNG

```
MODEL:
SETS:

TIME_P /@FILE(CDATA21.LDT) /;
PART /@FILE(CDATA21.LDT) /;
MACHINE /@FILE(CDATA21.LDT) /;
N_BACKLOG /@FILE(CDATA21.LDT) /;
CONTAINER /@FILE(CDATA21.LDT) /;
FOLL_PART /@FILE(CDATA21.LDT) /;
RQSETUP(PART,FOLL_PART,MACHINE) /@FILE(CDATA21.LDT) /;


    PART_MACH(PART,MACHINE): P;
    SWSETUP_T(RQSETUP): TT;
    PARTS(PART):NN,WHC,FHC,BC,SPLUS0,SM0,W0;
    PART_TIME(PART,TIME_P): D,SPLUS,SMINUS,W;
    BACK_MACH(N_BACKLOG,MACHINE):DELT1,DELT2;
    MACH_CONT(MACHINE,CONTAINER): AT;
    MACH(MACHINE): MP;
    AVERAGE_THR(CONTAINER):BETA;
    PART_BACK(PART,N_BACKLOG): X;
    PART_BACK1(FOLL_PART,N_BACKLOG):X1;
    RELEASE_T(PART, N_BACKLOG): RK;
    FINISH_T(PART, N_BACKLOG): FK;
    ZKT(RELEASE_T,TIME_P): Z;
    FKT(FINISH_T,TIME_P): F;
    VIKT(ZKT): V;
    UIKT(FKT): U;
    YIJK(RQSETUP): Y;


ENDSETS
DATA:
    !THE TOTAL NUMBER OF CONTAINERS;
    NN = 1,1;

      !HOLDING COST OF WIP INVENTORY;
      WHC = 1,2;

    !HOLDING COST OF FINISHED GOODS;
      FHC = 3,2;

    !BACKORDER COST OF FINISHED GOODS;
      BC = 2,3;
```

```
!INITIAL FGI AND BACKORDERS;
 SPLUS0 = 0,0;

 SM0 =   0,0;

 W0=    0,0;


!PROCESSING TIME;
 P = 5,8
     3,2;

!SEQUENCE DEPENDENT SET-UP TIME FOR SWITCHING PART I TO J;
TT = 3,4
      2,6;


   !PART DEMAND FOR TIME PERIODS 1 TO 2;

   D = 2,4
       4,4;


ENDDATA

   MIN =@SUM(TIME_P(T): @SUM(PART(I): FHC(I)*SPLUS(I,T)))
   +@SUM(TIME_P(T): @SUM(PART(I): BC(I)*SMINUS(I,T)))
   +@SUM(TIME_P(T): @SUM(PART(I): WHC(I)*W(I,T)));


@FOR (PART_TIME(I,T)|T #EQ# 1:
 SPLUS0(I)-SM0(I)
+@SUM(FKT(I,K,T):U(I,K,T))
-D(I,T)=SPLUS(I,1)-SMINUS(I,1)
 );

   @FOR (PART_TIME(I,T)|T #NE# 1:
   SPLUS(I,T-1)-SMINUS(I,T-1)+@SUM(FKT(I,K,T):U(I,K,T))
-D(I,T)=SPLUS(I,T)-SMINUS(I,T)
);

   @FOR (PART_TIME(I,T)|T #EQ#1:
@FOR (PART_BACK(I,T):
W0(I)+@SUM(ZKT(I,K,T): V(I,K,T))
-@SUM(FKT(I,K,T):U(I,K,T))=W(I,1)
```

```
));

@FOR (PART_TIME(I,T)|T #NE#1:
@FOR (PART_BACK(I,T):
W(I,T-1)+@SUM(ZKT(I,K,T): V(I,K,T))
-@SUM(FKT(I,K,T):U(I,K,T))=W(I,T)
));


@FOR(PART(I):
@SUM(PART_BACK(I,K): X(I,K))=@SUM(TIME_P(T): D(I,T))
);

@FOR(N_BACKLOG(K):
@SUM(PART_BACK(I,K):X(I,K))=1
);

@FOR(PART_BACK(I,K):
@FOR(ZKT(I,K,T):
X(I,K)-1.5<=V(I,K,T);
X(I,K)>=1.5*V(I,K,T)));

@FOR(PART_BACK(I,K):
@FOR(FKT(I,K,T):
X(I,K)-1.5<=U(I,K,T);
X(I,K)>=1.5*U(I,K,T)));


@FOR(PART_BACK1(J,K)|K #NE# 1:
@FOR(YIJK(I,J,M):
X(I,K-1)+X1(J,K)-1.5<=Y(I,J,M);
X(I,K-1)+X1(J,K)>=1.5*Y(I,J,M)));

@FOR(MACHINE(M):
@FOR(N_BACKLOG(K):
DELT1(K,M)=@SUM(PART_BACK(I,K):X(I,K)*P(I,M));
));



@FOR(MACHINE(M):
@FOR(N_BACKLOG(K):
DELT2(K,M)=@SUM(PART_BACK(J,K):
@SUM(YIJK(I,J,M):Y(I,J,M)*TT(I,J,M)))
));
```

```
@FOR(MACHINE(M):
MP(M)=@SUM(N_BACKLOG(K):DELT1(K,M)+DELT2(K,M))
/(@SUM(PART(I):@SUM(TIME_P(T):D(I,T)))));

@FOR(MACHINE(M):
AT(M,1)=MP(M));

@FOR(CONTAINER(N):
BETA(N)=@SUM(MACHINE(M):AT(M,N)));

@FOR(CONTAINER(N)|N #NE#1:
@FOR(MACHINE(M):
AT(M,N)*BETA(N-1)=MP(M)*(BETA(N-1)+AT(M,N-1)*(N-1));
));

@FOR(RELEASE_T(I,K):
@FOR(N_BACKLOG(K):
@FOR(CONTAINER(N)|N #EQ# NN(I):
RK(I,K)=(K-1)*@SUM(MACHINE(M): AT(M,N))/NN(I)
)));

@FOR(FINISH_T(I,K):
@FOR(N_BACKLOG(K):
@FOR(CONTAINER(N)|N #EQ# NN(I):
FK(I,K)=RK(I,K)+@SUM(MACHINE(M):AT(M,N))
)));

@FOR(PART_BACK(I,K):
@BIN(X(I,K)));

@FOR(PART_BACK(I,K):
@FOR(VIKT(I,K,T):
@BIN(V(I,K,T))));

@FOR(PART_BACK(I,K):
@FOR(UIKT(I,K,T):
@BIN(U(I,K,T))));

@FOR(PART_BACK1(J,K):
@FOR(YIJK(I,J,M):
@BIN(Y(I,J,M))
));


END
```

ACC21

Rows= 473	Vars= 416	No. integer vars= 144
Nonlinear rows= 2	Nonlinear vars= 7	Nonlinear constraints= 2
Nonzeros= 1638	Constraint nonz= 1438	Density=0.008

Optimal solution found at step:	35
Objective value:	52.00000
Branch count:	3

Variable	Value	Reduced Cost
P(1, 1)	5.000000	0.0000000E+00
P(1, 2)	8.000000	0.0000000E+00
P(2, 1)	3.000000	0.0000000E+00
P(2, 2)	2.000000	0.0000000E+00
X(1, 1)	0.0000000E+00	0.0000000E+00
X(1, 2)	1.000000	0.0000000E+00
X(1, 9)	1.000000	0.0000000E+00
X(1, 11)	1.000000	0.0000000E+00
X(1, 12)	1.000000	0.0000000E+00
X(1, 13)	1.000000	0.0000000E+00
X(1, 14)	1.000000	0.0000000E+00
X(2, 1)	1.000000	0.0000000E+00
X(2, 3)	1.000000	0.0000000E+00
X(2, 4)	1.000000	0.0000000E+00
X(2, 5)	1.000000	0.0000000E+00
X(2, 6)	1.000000	0.0000000E+00
X(2, 7)	1.000000	0.0000000E+00
X(2, 8)	1.000000	0.0000000E+00
X(2, 10)	1.000000	0.0000000E+00

K1C1.LNG

```
MODEL:
! KANBAN QUANTITY CALCULATIONS FOR MODULE K1(MACHINE SHOP
AND FABRICATION);

SETS:
 PROD/1..7/:          ! Each part has a :...;
 ST,                  ! Production setup time;
 PT,                  ! Actual Production time per unit;
 SC,                  ! Setup cost for any batch;
 PC,                  ! Production cost per part;
 HC;                  ! Holding cost per part per period;
 TIME/1..6/:;         ! Total time periods considered;
 PXT(PROD,TIME):      ! Each product in each period has...;
 DEM,                 ! Demand;
 KANBAN1,             ! Amount to produce;
 Y;                   ! = 1 if anything is produced;

ENDSETS

DATA:
 CAP =  480;                          ! Capacity per period;
 ST  =  36 39 41 35 20  23  34;       ! Setup times per batch;
 PT  =  12 22 11 21 11  13  11;       ! Production time per part;
 SC  =  12 20 30 21 10  10  20;       ! Setup cost;
 PC  =  8  7  8  9  6   5   4;        ! Production cost;
 HC  =  11 11 11 12 11  11  12;       ! Holding cost per unit;
 DEM =       0  0  0  3  0  0         ! Demands;
             0  0  0  3  0  0
             0  0  4  3  0  0
             0  0  2  1  0  0
             0  0  4  2  0  0
             0  0  2  4  0  0
             0  0  4  4  0  0;
ENDDATA

!-----------------------------------;
SETS:
 PXTXT( PROD, TIME, TIME)| &3 #GE# &2:
  PQOF,
  CCOF,
  X;
ENDSETS
! Compute cost and assembly Kanban quantities for various production runs;
@FOR( PROD( I):
```

```
  @FOR( TIME( S):
    PQOF( I, S, S) = DEM( I, S);
    CCOF( I, S, S) = PC( I) * DEM( I, S);
    @FOR( TIME( T)| T #GT# S:
     PQOF( I, S, T) = PQOF( I, S, T - 1) +
      DEM( I, T);
     CCOF( I, S, T) = CCOF( I, S, T - 1) +
      ( PC( I) + HC( I) * ( T - S)) * DEM( I, T);
    );
  ););
! The objective;
 MIN = TCOST;
 TCOST = @SUM( PXTXT: CCOF * X) +
     @SUM( PXT( I, S): SC( I) * Y( I, S));

 @FOR( PROD( I):
! In period 1, some production run must be started;
!  Note, watch out for periods without demand;
   @SUM( PXTXT( I, S, T) | S #EQ# 1:
     X( I, S, T)) = 1;
   @FOR( TIME( K)| K #GT# 1:
!  If we ended a run in period K - 1...;
    @SUM( PXTXT( I, S, T)| T #EQ# K - 1: X( I, S, K - 1))
!  then we must start a run in period k;
    = @SUM( PXTXT( I, K, T): X( I, K, T));
    );
! Setup forcing;
   @FOR( TIME( S):
    Y( I, S) =  @SUM( PXTXT( I, S, T)
         : ( PQOF( I, S, T) #GT# 0) * X( I, S, T));
! Calc amount made each period;
    KANBAN1( I, S) = @SUM( PXTXT( I, S, T):
      PQOF( I, S, T) * X( I, S, T)); );
   );
! The capacity constraints;
 @FOR( TIME( S):
  @SUM( PROD( I): ST( I) * Y( I, S)) +
  @SUM( PXTXT( I, S, T):
   PT( I) * PQOF( I, S, T) * X( I, S, T)) <= CAP;
   );
! Make the Y's integer;
 @FOR( PXT: @GIN( Y););
END
```

AK1C1

Rows=134 Vars= 232 No.integer vars= 42 (all are linear)

Nonzeros= 1023 Constraint nonz= 817(439 are +- 1) Density=0.033

Smallest and largest elements in absolute value= 1.00000 480.000
No. <:6 No.=:127 No.>:0, Obj=MIN, GUBs <= 64
Single cols= 42

Optimal solution found at step: 51
Objective value: 431.0000
Branch count: 0

Variable	Value	Reduced Cost
CAP	480.0000	0.0000000E+00
TCOST	431.0000	0.0000000E+00
KANBAN1(1, 4)	3.000000	0.0000000E+00
KANBAN1(2, 4)	3.000000	0.0000000E+00
KANBAN1(3, 3)	4.000000	0.0000000E+00
KANBAN1(3, 4)	3.000000	0.0000000E+00
KANBAN1(4, 3)	3.000000	0.0000000E+00
KANBAN1(5, 3)	4.000000	0.0000000E+00
KANBAN1(5, 4)	2.000000	0.0000000E+00
KANBAN1(6, 3)	2.000000	0.0000000E+00
KANBAN1(6, 4)	4.000000	0.0000000E+00
KANBAN1(7, 3)	4.000000	0.0000000E+00
KANBAN1(7, 4)	4.000000	0.0000000E+00
PQOF(1, 1, 4)	3.000000	0.0000000E+00
PQOF(1, 1, 5)	3.000000	0.0000000E+00
PQOF(1, 1, 6)	3.000000	0.0000000E+00
PQOF(1, 2, 4)	3.000000	0.0000000E+00
PQOF(1, 2, 5)	3.000000	0.0000000E+00
PQOF(1, 2, 6)	3.000000	0.0000000E+00
PQOF(1, 3, 4)	3.000000	0.0000000E+00
PQOF(1, 3, 5)	3.000000	0.0000000E+00
PQOF(1, 3, 6)	3.000000	0.0000000E+00
PQOF(1, 4, 4)	3.000000	0.0000000E+00
PQOF(1, 4, 5)	3.000000	0.0000000E+00
PQOF(1, 4, 6)	3.000000	0.0000000E+00
PQOF(2, 1, 4)	3.000000	0.0000000E+00
PQOF(2, 1, 5)	3.000000	0.0000000E+00
PQOF(2, 1, 6)	3.000000	0.0000000E+00
PQOF(2, 2, 4)	3.000000	0.0000000E+00

PQOF(2, 2, 5)	3.000000	0.0000000E+00
PQOF(2, 2, 6)	3.000000	0.0000000E+00
PQOF(2, 3, 4)	3.000000	0.0000000E+00
PQOF(2, 3, 5)	3.000000	0.0000000E+00
PQOF(2, 3, 6)	3.000000	0.0000000E+00
PQOF(2, 4, 4)	3.000000	0.0000000E+00
PQOF(2, 4, 5)	3.000000	0.0000000E+00
PQOF(2, 4, 6)	3.000000	0.0000000E+00
PQOF(3, 1, 3)	4.000000	0.0000000E+00
PQOF(3, 1, 4)	7.000000	0.0000000E+00
PQOF(3, 1, 5)	7.000000	0.0000000E+00
PQOF(3, 1, 6)	7.000000	0.0000000E+00
PQOF(3, 2, 3)	4.000000	0.0000000E+00
PQOF(3, 2, 4)	7.000000	0.0000000E+00
PQOF(3, 2, 5)	7.000000	0.0000000E+00
PQOF(3, 2, 6)	7.000000	0.0000000E+00
PQOF(3, 3, 3)	4.000000	0.0000000E+00
PQOF(3, 3, 4)	7.000000	0.0000000E+00
PQOF(3, 3, 5)	7.000000	0.0000000E+00
PQOF(3, 3, 6)	7.000000	0.0000000E+00
PQOF(3, 4, 4)	3.000000	0.0000000E+00
PQOF(3, 4, 5)	3.000000	0.0000000E+00
PQOF(3, 4, 6)	3.000000	0.0000000E+00
PQOF(4, 1, 3)	2.000000	0.0000000E+00
PQOF(4, 1, 4)	3.000000	0.0000000E+00
PQOF(4, 1, 5)	3.000000	0.0000000E+00
PQOF(4, 1, 6)	3.000000	0.0000000E+00
PQOF(4, 2, 3)	2.000000	0.0000000E+00
PQOF(4, 2, 4)	3.000000	0.0000000E+00
PQOF(4, 2, 5)	3.000000	0.0000000E+00
PQOF(4, 2, 6)	3.000000	0.0000000E+00
PQOF(4, 3, 3)	2.000000	0.0000000E+00
PQOF(4, 3, 4)	3.000000	0.0000000E+00
PQOF(4, 3, 5)	3.000000	0.0000000E+00
PQOF(4, 3, 6)	3.000000	0.0000000E+00
PQOF(4, 4, 4)	1.000000	0.0000000E+00
PQOF(4, 4, 5)	1.000000	0.0000000E+00
PQOF(4, 4, 6)	1.000000	0.0000000E+00
PQOF(5, 1, 3)	4.000000	0.0000000E+00
PQOF(5, 1, 4)	6.000000	0.0000000E+00
PQOF(5, 1, 5)	6.000000	0.0000000E+00
PQOF(5, 1, 6)	6.000000	0.0000000E+00
PQOF(5, 2, 3)	4.000000	0.0000000E+00
PQOF(5, 2, 4)	6.000000	0.0000000E+00
PQOF(5, 2, 5)	6.000000	0.0000000E+00
PQOF(5, 2, 6)	6.000000	0.0000000E+00

PQOF(5, 3, 3)	4.000000	0.0000000E+00
PQOF(5, 3, 4)	6.000000	0.0000000E+00
PQOF(5, 3, 5)	6.000000	0.0000000E+00
PQOF(5, 3, 6)	6.000000	0.0000000E+00
PQOF(5, 4, 4)	2.000000	0.0000000E+00
PQOF(5, 4, 5)	2.000000	0.0000000E+00
PQOF(5, 4, 6)	2.000000	0.0000000E+00
PQOF(6, 1, 3)	2.000000	0.0000000E+00
PQOF(6, 1, 4)	6.000000	0.0000000E+00
PQOF(6, 1, 5)	6.000000	0.0000000E+00
PQOF(6, 1, 6)	6.000000	0.0000000E+00
PQOF(6, 2, 3)	2.000000	0.0000000E+00
PQOF(6, 2, 4)	6.000000	0.0000000E+00
PQOF(6, 2, 5)	6.000000	0.0000000E+00
PQOF(6, 2, 6)	6.000000	0.0000000E+00
PQOF(6, 3, 3)	2.000000	0.0000000E+00
PQOF(6, 3, 4)	6.000000	0.0000000E+00
PQOF(6, 3, 5)	6.000000	0.0000000E+00
PQOF(6, 3, 6)	6.000000	0.0000000E+00
PQOF(6, 4, 4)	4.000000	0.0000000E+00
PQOF(6, 4, 5)	4.000000	0.0000000E+00
PQOF(6, 4, 6)	4.000000	0.0000000E+00
PQOF(7, 1, 3)	4.000000	0.0000000E+00
PQOF(7, 1, 4)	8.000000	0.0000000E+00
PQOF(7, 1, 5)	8.000000	0.0000000E+00
PQOF(7, 1, 6)	8.000000	0.0000000E+00
PQOF(7, 2, 3)	4.000000	0.0000000E+00
PQOF(7, 2, 4)	8.000000	0.0000000E+00
PQOF(7, 2, 5)	8.000000	0.0000000E+00
PQOF(7, 2, 6)	8.000000	0.0000000E+00
PQOF(7, 3, 3)	4.000000	0.0000000E+00
PQOF(7, 3, 4)	8.000000	0.0000000E+00
PQOF(7, 3, 5)	8.000000	0.0000000E+00
PQOF(7, 3, 6)	8.000000	0.0000000E+00
PQOF(7, 4, 4)	4.000000	0.0000000E+00
PQOF(7, 4, 5)	4.000000	0.0000000E+00
PQOF(7, 4, 6)	4.000000	0.0000000E+00
CCOF(1, 1, 4)	123.0000	0.0000000E+00
CCOF(1, 1, 5)	123.0000	0.0000000E+00
CCOF(1, 1, 6)	123.0000	0.0000000E+00
CCOF(1, 2, 4)	90.00000	0.0000000E+00
CCOF(1, 2, 5)	90.00000	0.0000000E+00
CCOF(1, 2, 6)	90.00000	0.0000000E+00
CCOF(1, 3, 4)	57.00000	0.0000000E+00
CCOF(1, 3, 5)	57.00000	0.0000000E+00
CCOF(1, 3, 6)	57.00000	0.0000000E+00

CCOF(1, 4, 4)	24.00000	0.0000000E+00
CCOF(1, 4, 5)	24.00000	0.0000000E+00
CCOF(1, 4, 6)	24.00000	0.0000000E+00
CCOF(2, 1, 4)	120.0000	0.0000000E+00
CCOF(2, 1, 5)	120.0000	0.0000000E+00
CCOF(2, 1, 6)	120.0000	0.0000000E+00
CCOF(2, 2, 4)	87.00000	0.0000000E+00
CCOF(2, 2, 5)	87.00000	0.0000000E+00
CCOF(2, 2, 6)	87.00000	0.0000000E+00
CCOF(2, 3, 4)	54.00000	0.0000000E+00
CCOF(2, 3, 5)	54.00000	0.0000000E+00
CCOF(2, 3, 6)	54.00000	0.0000000E+00
CCOF(2, 4, 4)	21.00000	0.0000000E+00
CCOF(2, 4, 5)	21.00000	0.0000000E+00
CCOF(2, 4, 6)	21.00000	0.0000000E+00
CCOF(3, 1, 3)	120.0000	0.0000000E+00
CCOF(3, 1, 4)	243.0000	0.0000000E+00
CCOF(3, 1, 5)	243.0000	0.0000000E+00
CCOF(3, 1, 6)	243.0000	0.0000000E+00
CCOF(3, 2, 3)	76.00000	0.0000000E+00
CCOF(3, 2, 4)	166.0000	0.0000000E+00
CCOF(3, 2, 5)	166.0000	0.0000000E+00
CCOF(3, 2, 6)	166.0000	0.0000000E+00
CCOF(3, 3, 3)	32.00000	0.0000000E+00
CCOF(3, 3, 4)	89.00000	0.0000000E+00
CCOF(3, 3, 5)	89.00000	0.0000000E+00
CCOF(3, 3, 6)	89.00000	0.0000000E+00
CCOF(3, 4, 4)	24.00000	0.0000000E+00
CCOF(3, 4, 5)	24.00000	0.0000000E+00
CCOF(3, 4, 6)	24.00000	0.0000000E+00
CCOF(4, 1, 3)	66.00000	0.0000000E+00
CCOF(4, 1, 4)	111.0000	0.0000000E+00
CCOF(4, 1, 5)	111.0000	0.0000000E+00
CCOF(4, 1, 6)	111.0000	0.0000000E+00
CCOF(4, 2, 3)	42.00000	0.0000000E+00
CCOF(4, 2, 4)	75.00000	0.0000000E+00
CCOF(4, 2, 5)	75.00000	0.0000000E+00
CCOF(4, 2, 6)	75.00000	0.0000000E+00
CCOF(4, 3, 3)	18.00000	0.0000000E+00
CCOF(4, 3, 4)	39.00000	0.0000000E+00
CCOF(4, 3, 5)	39.00000	0.0000000E+00
CCOF(4, 3, 6)	39.00000	0.0000000E+00
CCOF(4, 4, 4)	9.000000	0.0000000E+00
CCOF(4, 4, 5)	9.000000	0.0000000E+00
CCOF(4, 4, 6)	9.000000	0.0000000E+00
CCOF(5, 1, 3)	112.0000	0.0000000E+00

CCOF(5, 1, 4)	190.0000	0.0000000E+00
CCOF(5, 1, 5)	190.0000	0.0000000E+00
CCOF(5, 1, 6)	190.0000	0.0000000E+00
CCOF(5, 2, 3)	68.00000	0.0000000E+00
CCOF(5, 2, 4)	124.0000	0.0000000E+00
CCOF(5, 2, 5)	124.0000	0.0000000E+00
CCOF(5, 2, 6)	124.0000	0.0000000E+00
CCOF(5, 3, 3)	24.00000	0.0000000E+00
CCOF(5, 3, 4)	58.00000	0.0000000E+00
CCOF(5, 3, 5)	58.00000	0.0000000E+00
CCOF(5, 3, 6)	58.00000	0.0000000E+00
CCOF(5, 4, 4)	12.00000	0.0000000E+00
CCOF(5, 4, 5)	12.00000	0.0000000E+00
CCOF(5, 4, 6)	12.00000	0.0000000E+00
CCOF(6, 1, 3)	54.00000	0.0000000E+00
CCOF(6, 1, 4)	206.0000	0.0000000E+00
CCOF(6, 1, 5)	206.0000	0.0000000E+00
CCOF(6, 1, 6)	206.0000	0.0000000E+00
CCOF(6, 2, 3)	32.00000	0.0000000E+00
CCOF(6, 2, 4)	140.0000	0.0000000E+00
CCOF(6, 2, 5)	140.0000	0.0000000E+00
CCOF(6, 2, 6)	140.0000	0.0000000E+00
CCOF(6, 3, 3)	10.00000	0.0000000E+00
CCOF(6, 3, 4)	74.00000	0.0000000E+00
CCOF(6, 3, 5)	74.00000	0.0000000E+00
CCOF(6, 3, 6)	74.00000	0.0000000E+00
CCOF(6, 4, 4)	20.00000	0.0000000E+00
CCOF(6, 4, 5)	20.00000	0.0000000E+00
CCOF(6, 4, 6)	20.00000	0.0000000E+00
CCOF(7, 1, 3)	112.0000	0.0000000E+00
CCOF(7, 1, 4)	272.0000	0.0000000E+00
CCOF(7, 1, 5)	272.0000	0.0000000E+00
CCOF(7, 1, 6)	272.0000	0.0000000E+00
CCOF(7, 2, 3)	64.00000	0.0000000E+00
CCOF(7, 2, 4)	176.0000	0.0000000E+00
CCOF(7, 2, 5)	176.0000	0.0000000E+00
CCOF(7, 2, 6)	176.0000	0.0000000E+00
CCOF(7, 3, 3)	16.00000	0.0000000E+00
CCOF(7, 3, 4)	80.00000	0.0000000E+00
CCOF(7, 3, 5)	80.00000	0.0000000E+00
CCOF(7, 3, 6)	80.00000	0.0000000E+00
CCOF(7, 4, 4)	16.00000	0.0000000E+00
CCOF(7, 4, 5)	16.00000	0.0000000E+00
CCOF(7, 4, 6)	16.00000	0.0000000E+00

Appendix B: C Program for Paint Setup Algorithm

B.1 Program statement for PSA

```
#include<iostream.h>
#include<fstream.h>
#define MAXJOB 10000
int main()
{
int color[3]={1,2,3};
int max_clr_jobs[3];
char clr[10];
int setA_fin_clr[100];
int setB_fin_clr[100];
int setA_fin_job[100];
int setB_fin_job[100];
int setA_fin_set[100];
int setB_fin_set[100];
int job_num[1000],job_clr[1000],job_set[1000];
int j=0,k=0,i=0,dayjobs=0,jobs=0,l=0,SC=0,Act_jobs=0,jobcount=0;
ifstream infile;
ofstream outfile;

cout << "****Enter the no of jobs****" << endl;
cin >> Act_jobs;
for(i=0;i<Act_jobs;i++)
{
cout << "***Enter the job#*******" << endl;
cin >> job_num[i];
cout << "****Enter the job color:1,2,3****" << endl;
cin >> job_clr[i];
cout << "*****Enter the job setup:1,2******" << endl;
cin >> job_set[i];
}
cout<<"****Enter the number of Blue parts/week****"<<endl;
cin>>max_clr_jobs[0];
cout<<"****Enter the number of Black parts/week****"<<endl;
cin>>max_clr_jobs[1];
cout<<"****Enter the number of NoColor parts/week****"<<endl;
cin>>max_clr_jobs[2];
cout<<endl;
cout<<endl;
cout<<"INPUT FOR THE PSA"<<endl;
cout<<endl;
cout<<endl;
cout<<"Job#"<<" " <<"clr"<<" "<<"job setup"<<endl;
for(i=0;i<Act_jobs;i++)
{
if(job_num[i]>9)
 cout<<"0"<<job_num[i]<<" "<<" "<<" "<<" "<<job_clr[i]<<" "<<" "<<" "<<" "
<<job_set[i]<<endl;
else
 cout<<"00"<<job_num[i]<<" "<<" "<<" "<<" "<<job_clr[i]<<" "<<" "<<" "<<" "
<<job_set[i]<<endl;
```

```
cout<<endl;
cout<<endl;

cout<<"RESULTS OF THE PSA"<<endl;
cout<<endl;
cout<<endl;
cout<<"Job#"<<"  "<<"  "<<"clr"<<"  "<<"  "<<"  "<<"  "<<"job setup"<<endl;

while((jobcount<MAXJOB)&&(jobcount<Act_jobs))
{
 SC=color[j];
  dayjobs=0,jobs=0,k=0,l=0;
  while((dayjobs<max_clr_jobs[j])&&(jobs<MAXJOB)&&(jobs<Act_jobs))
   {


      if(job_clr[jobs]==SC)
        {
          if(job_set[jobs]==1)
            {
              setA_fin_clr[k]=job_clr[jobs];
              setA_fin_job[k]=job_num[jobs];
              setA_fin_set[k]=job_set[jobs];
              job_clr[jobs]=200;
              k++;
            }
           else
            {
              setB_fin_clr[l]=job_clr[jobs];
              setB_fin_job[l]=job_num[jobs];
               setB_fin_set[l]=job_set[jobs];
              job_clr[jobs]=200;
              l++;
            }
           dayjobs++;
           jobcount++;
      }
       jobs++;
    }

  for(i=0;i<k;i++)
  {
    if(setA_fin_clr[i]==1)
     strcpy(clr,"Blue");
    if(setA_fin_clr[i]==2)
     strcpy(clr,"Blck");
    if(setA_fin_clr[i]==3)
     strcpy(clr,"Nclr");
    if(setA_fin_job[i] > 9)
cout<<"0"<<setA_fin_job[i]<<"  "<<"  "<<"  "<<"  "<<clr<<"  "<<"  "<<"  "<<"  "<<"
<<"  "<<setA_fin_set[i]<<endl;
    else
 cout<<"00"<<setA_fin_job[i]<<"  "<<"  "<<"  "<<"  "<<clr<<"  "<<"  "<<"  "<<"  "<
<<"  "<<setA_fin_set[i]<<endl;
```

```
    }
   for(i=0;i<l;i++)
    {
      if(setB_fin_clr[i]==1)
      strcpy(clr,"Blue");
     if(setB_fin_clr[i]==2)
      strcpy(clr,"Blck");
     if(setB_fin_clr[i]==3)
      strcpy(clr,"Nclr");
 .if(setB_fin_job[i] > 9)
 cout<<"0"<<setB_fin_job[i]<<" "<<" "<<" "<<" "<<clr<<" "<<" "<<" "<<" "<
<<" "<<setB_fin_set[i]<<endl;
     else
 cout<<"00"<<setB_fin_job[i]<<" "<<" "<<" "<<" "<<clr<<" "<<" "<<" "<<" "
<<" "<<setB_fin_set[i]<<endl;
    }
jobs=0;
 j++;
 if(j>=3)
  j=0;
 }

return 1;
}
```

B.2 Sample Input Screen for Experiment 1:

```
**********************************************************************
**** Enter the number of jobs****
22
****Enter the job #*****
1
****Enter the job color:1,2,3*****
1
****Enter the job setup:1,2****
1
****Enter the job #*****
2
****Enter the job color:1,2,3*****
1
****Enter the job setup:1,2****
1
****Enter the job #*****
3
****Enter the job color:1,2,3*****
1
****Enter the job setup:1,2****
2
****Enter the job #*****
4
****Enter the job color:1,2,3*****
1
****Enter the job setup:1,2****
2
****Enter the job #*****
5
****Enter the job color:1,2,3*****
1
****Enter the job setup:1,2****
2

... (Data input is completed for total part list in the same way).
**********************************************************************
```

B.3 Sample Output Screen for Experiment 1:

****

INPUT FOR THE PSA

Job #	clr	job setup
001	1	1
002	1	1
003	1	2
004	1	2
005	1	2
006	1	2
007	1	2
008	2	2
009	2	2
010	2	2
011	2	2
012	2	2
013	1	2
014	1	2
015	1	2
016	1	2
017	1	2
018	2	1
019	2	1
020	2	1
022	2	1

RESULTS OF THE PSA

Job #	clr	job setup
001	Blue	1
002	Blue	1
003	Blue	2
004	Blue	2
005	Blue	2
006	Blue	2
007	Blue	2
013	Blue	2
014	Blue	2
015	Blue	2
016	Blue	2
017	Blue	2
018	Blck	1
019	Blck	1

020	Blck	1
021	Blck	1
022	Blck	1
008	Blck	2
009	Blck	2
010	Blck	2
011	Blck	2
012	Blck	2

****