1-1-2003

A simulation model for comparison of an innovative hybrid system and the conventional cellular manufacturing

Erika Valery Lopez Roa
Ryerson University

A SIMULATION MODEL FOR COMPARISON OF AN INNOVATIVE HYBRID SYSTEM AND THE CONVENTIONAL CELLULAR MANUFACTURING

by

Erika Valery Lopez Roa
Bachelor of Industrial Engineering
University of Guadalajara. Guadalajara, Jalisco, Mexico
January 2001

A thesis

presented to Ryerson University

in partial fulfilment of the

requirements for the degree of

Master of Applied Science

in the Program of

Mechanical Engineering

Toronto, Ontario, Canada, 2003

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

Cellular manufacturing has tested positive in significantly reducing material handling and setup time as compared to a job shop, but it falls behind job shop in terms of flexibility. In this thesis a new system is proposed that takes advantage of the flexibility of a job shop while it keeps the setup time at a reduced level. This new system is referred to as *hybrid* system.

In this thesis the performance of the proposed hybrid system is compared to the conventional cellular manufacturing system. Both systems are evaluated within a cellular layout and utilize group scheduling rules DDSI (due date truncated shortest processing time) and MSSPT (minimum setup shortest processing time). A simulation model, with random due dates and quantities is developed and tested. Performance measures are mean flowtime, tardiness and earliness.

Overall results indicate that, in terms of mean flowtime and tardiness, the hybrid system outperforms the cellular system when the MSSPT rule was applied, while the cellular system outperforms the hybrid system when the DDSI rule is implemented. With regard to the earliness performance measure, the cellular system shows in most cases better performance than the hybrid system, regardless of the scheduling rule used. Finally, the results indicate that the hybrid system performs better than the cellular system with respect to the number of parts produced.

ACKNOWLEDGEMENTS

My most sincere gratitude to Dr. Saeed Zolfaghari, Professor at Ryerson University for providing me with his time and expertise in manufacturing systems. I greatly value his advice through the undertaking of this Master's program and as my thesis supervisor.

I would also like to thank Dr. Kawall, Dr. Fang and Leah Stanwick for their excellent student support.

I appreciate to a great extent the National Council of Science and Technology (CONACYT) for its support and trust. In the same way I am thankful to the University of Guadalajara in Mexico, for its providing in my overseas studies.

Finally, I would like to thank my parents for their encouragement to always pursue my academic goals and to Man for his love and caring while being in Canada.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Deterministic Equations

i	index of machines, $i = 1,\ldots,M$
j	index of parts, $j = 1,\ldots,N$
k	index of operations, $k = 1,\ldots,K$
$\bar{t}$	average time per part
t_{ijk}	total time of operation k of part j on machine i
N	number of part types
Q_j	batch size of part j
T_i	total processing time on machine i
L_i	average processing time per part on machine i
U_i	utilization of machine
X_{ijk}	$\begin{cases} 1 & \text{if machine } i \text{ is to process operation } k \text{ of part } j \\ 0 & \text{else} \end{cases}$

Simulation Model

i	index of machines, $i = 1,\ldots,M$
j	index of parts, $j = 1,\ldots,N$
h	index of family, $h = 1,\ldots,H$
k	index of operations, $k = 1,\ldots,K$
s_{ij}	total setup time for part j on machine i
s_{ij}^{f}	family setup time for part j on machine i
s_{ij}^{p}	part setup time for part j on machine i
β	ratio of the family setup time to the total setup
f_{ih}	family setup time of family h on machine i

Ω_h	set of parts in family h
$\overline{N}$	average number of part types in an order
P	probability of a part being selected for processing
N	number of part types
$\overline{Q}$	average order size of a part
$\bar{l}$	average lead time per order
γ	variable used in the due date assignment

Output Analysis

j	index of batch, j=1,…,J
r	index of replication, r=1,…,R
θ	point estimate or true performance measure
$\hat{\theta}_r$	point estimator of run r
F	true flowtime performance measure
T	true tardiness performance measure
E	true earliness performance measure
P	probability of a part type being selected for processing
t	time of performance measure observation
I_0	initial conditions at initialization phase
I	initial conditions at steady-state
T_0	time of first event in the steady-state or data-collection phase
T_E	time of last event E in the initalization phase
$\overline{Y}_{rj}$	mean value of replication r for batch j
$\overline{Y}_j$	ensemble average for batch j
$\overline{Y}..(j,d)$	ensemble average of batch j when deleting d observations
$\overline{Y}_r(n,d)$	sample mean of the undeleted n-d observations from the rth replication
$\overline{Y}..(n,d)$	mean point estimator of the undeleted n-d observation across all replications

S^2	sample variance of the mean point estimator
α	level of significance for the performance measure
ε	level of accuracy
R^{new}	new number of replications

CHAPTER 1: INTRODUCTION AND LITERATURE REVIEW

1.1 Research Background

In the last four decades challenging planning, operational and control manufacturing problems have been solved, and many others still remain as the scope of extensive research. New difficulties are continuously emerging as new manufacturing technologies evolve. Lower unit cost and higher quality are among the main objectives for managers. The unit cost, product quality, and process efficiency are influenced by lead time, setup, processing and material handling times. Flowtime and lateness are among the most commonly used performance measures of efficient processes.

Previous work in improving these performance measures and consequently maximizing the efficiency of the manufacturing systems, has led to the development of many scheduling heuristics and processing layouts. Group scheduling heuristics, flexible manufacturing systems and cellular manufacturing systems are among those innovative solutions. Such solutions focus on problems that are constrained to specific conditions (e.g., either in size or in the complexity of the system). In this situation, as the number of constraints increases the system reduces its flexibility. It is essential to attain this flexibility in the manufacturing system with minimum penalties in time, cost and performance.

Thus, this thesis investigates the feasibility of a cellular manufacturing system when non-family parts are assigned to it. When non-family parts are added to the cellular system, its flexibility is increased since the system is able to process not only family parts but also non-family parts. This increase in the system flexibility will provide a wider operational range and reduce the number of machines or tools required apart from the cells.

It is worth noting that most of scheduling research either assumes no setup time or considers it as a component of the processing time. While this assumption can reflect certain applications it also simplifies the analysis of the system. Although simplified assumptions may be found useful in some problem settings (e.g., setup times are very small and are considered negligible), they may not apply to manufacturing problems that require explicit treatment of the setup cost (e.g., when dealing with part families setups). This thesis considers scheduling rules where setup times are significant.

1.2 Overview of Flexible Manufacturing Systems

Major changes in production processes in the industry have created new opportunities and challenges in the scheduling area. These changes are inspired by the introduction of new process technologies, such as flexible manufacturing systems (FMSs) (Mosier and Mahmoodi 2002). Technological flexibility acquisition requires significant initial investment where decisions are characterised by being risky and strategic in nature. There were several attempts by researchers in the field to configure flexible systems and integrate them into production planning and control systems (Garg et al. 2001). Implementing just-in-time (JIT) policy requires manufacturing systems to be flexible (responsive) to changes in system environment. For a comprehensive literature review of manufacturing flexibility, see De Toni and Tonchia (1998).

The choice of the extent of flexibility is driven by the market and economic factors. Flexibility is more suitable for a multiple product manufacturing system with correlated demand. Simulation studies revealed that there can be no unique answer to the flexibility: inventory mix and alternative strategies have to be evaluated to determine the best scenario for a given situation (Garg et al. 2001).

Flexible manufacturing systems (FMSs) are capable of manufacturing a wide variety of products. Their flexibility arises from machines or robots that are capable of performing more than one operation (Potts and Whitehead 2001).

FMSs have received increasing attention in the last two decades. This is partly due to the fact that flexibility is required by manufacturing companies to stay in a highly competitive and changing business environment. Over the years, various types of FMSs have been designed and implemented worldwide. The existing implementations have demonstrated a number of benefits in terms of cost reductions and increase productivity, etc. However, these benefits are not easy to realize. Successful implementation of FMSs requires solutions of various decision problems faced during design and operation stages of these systems (Sabuncuoglu 1998).

A flexible manufacturing system (FMS) can be defined as a system consisting of workstations, automated material handling system(s), and a computer controlled network (Browne et al. 1984, Buzacott and Shanthikumar 1980). More specifically, an FMS may be viewed as an embellished job shop with some added constructs, such as automatic tool changers and tool magazines, sophisticated fixtures and pallets, central computer control, common material handling system(s), and automated loading, and unloading capabilities (Millar and Yang 1996). Flexible manufacturing systems are among the relatively new emerging technologies that have had a great impact in industry, and are becoming an attractive substitute for the conventional means of batch manufacturing (Stecke 1983). The aim of an FMS is to achieve the efficiency of automated mass production, while utilizing the flexibility of a manual job shop to simultaneously machine several part types.

FMSs are known to improve machine utilization and throughput rate, and reduce production lead time and work-in-process inventory by reducing time for tool changes and part movements. However, several difficult decision problems must be solved to obtain such benefits. Managing production for an FMS is more difficult than for production lines and job shops for the following reasons: (1) each machine is quite versatile and capable of performing many different operations, (2) the system can machine several part types simultaneously, and (3) each part type may have alternative routes through the system. These additional capabilities and planning options increase both the number of decision variables and the constraints associated with setting up an FMS (Stecke 1983).

1.3 Overview of Innovative Processing Layouts

Flowshop, job shop and cellular manufacturing processes have received much attention from researchers and practitioners since they are widely utilized in the industry and are the basis of more complex production systems. In a flow shop, products follow along direct linear routes where large batches of several standardized products are produced. Because products are produced in batches, the production system must be changed over when a different product is to be produced (Gaither 1996).

The typical job shop (or functional) layout tends to be on a functional basis, i.e., machines capable of carrying out similar operations grouped together. This normally means that, for any component to be produced it has to be moved from group to group over the shop floor; it can also result in component waiting beside machines for a particular operation to be carried out on them which, in turn, can result in a large amount of capital being tied up in work-in-process. Traditionally, the manufacturing of batch/discrete lots of component parts has taken place in a process layout where similar machines are grouped together in one area of the production plant. Thus, during the production process, batches move through various work centres according to specified machining sequences (Kattan 1997).

Cellular manufacturing offers a number of advantages over traditional job shop process configurations, including: reduced work-in-process (WIP) inventory and lower manufacturing lead times (Shafer and Charnes 1993). These advantages arise because processing parts with similar manufacturing requirements and setup characteristics reduces the average number of major setups (setups between parts in different families). A more detailed description of cellular manufacturing systems will be provided in section 1.4. Alternative manufacturing systems, however, had also been developed that provide good results although they are not as common.

1.3.1 Hybrid multi-cell flexible manufacturing system

A new classification scheme for FMS was presented by MacCarthy and Liu (1993). He identified three types of FMSs: single flexible machine (SFM), flexible manufacturing cell (FMC) and multi-cell flexible manufacturing system (MCFMS). An SFM is a computer controlled production unit which consists of a single NC or CNC machine tool served by a robot as a material handling device with a part storage buffer (Black 1983, 1988). An FMC is a type of FMS which consists of a group of SFMs with three or fewer machines sharing one common material handling device (MacCarthy and Liu 1993), whereas an MCFMS is a type of FMS which consists of a number of FMCs, and their operations are linked by an automated material handling system. The operation of the entire system is integrated and controlled by a computer network.

A MCFMS centre is a production system that has the advantage of the production rate of an assembly line (product line), yet the flexibility of a job shop. Application of GT is a key ingredient in forming appropriate machine cells for MCFMS. Arranging the cell in a U-shape form has many advantages; it provides flexibility for worker movement and provides the opportunity for machine coupling.

The workload of the machines and the intercell (movement of parts between cells)/intracell (movement of parts within the cell) movements are considered in the process of balancing the GT cell's workload and improving the MCFMS alternatives. However, the final selection of MCFMS depends on each special case of the application. In each application, the cost evaluation and comparison should be conducted for the total cost of the machines, space, labour and material handling equipments among all alternatives. Thus, alternative groupings will be generated which provide the manufacturers with more flexibility and opportunity to evaluate different options and choose an alternative one which is more flexible and the most cost effective.

The objective of the hybrid multi-cell flexible manufacturing system is to minimize machine duplication by increasing its utilization, minimize intercell moves, simplifying the scheduling problem and increasing the flexibility of the manufacturing system. A four step integrated approach of design and scheduling alternative hybrid multi-cell flexible manufacturing system (MCFMS) was developed for such a purpose (Kattan 1997).

1.3.2 Virtual cellular manufacturing systems

Considerable attention has been given to improving the efficiency of small batch manufacturing systems. Small batch manufacturing has traditionally been carried out in a job shop environment using a process layout. An alternative approach has been to utilize the cellular manufacturing principles of group technology (Kannan and Ghosh 1996). GT allows the advantage of flow production in batch manufacturing, through the sub-division of production facilities and procedures into discrete and independent cells. Organizing production around groups of items with similar production requirements (families) can yield improved lead time and due date performance (Greene and Sadowski 1984, Suresh and Meredith 1985).

The concept of physically dividing the factory into smaller units enhances the agility of large manufacturing systems. But in the case of small and medium industries, it may not always be possible to do this physical separation for a variety of reasons. Often small industries may not need GT at all, as the range of operations may not warrant subdivision. However, in the case of medium industries, application of GT may offer considerable benefits, but physical separation of the operations into cells can be constrained by practical, technical and organisational factors. In such situations, these industries may want to have the benefits that GT offers, but without actually breaking down the factory into cells.

It is towards this direction that the formation of virtual cells can offer considerable help. These virtual cells are in fact conceptual cells formed in the computer software and do not actually exist on the shop floor. A virtual cell is not identifiable as a fixed physical grouping of workstations but as data files and processes in a controller. Virtual Cellular Manufacturing (VCM) combines the setup efficiency typically obtained by cellular manufacturing systems with the routing flexibility of a job shop (Babu et al. 2000).

Using family-oriented (group) scheduling rules, machines are allocated temporarily to part families, based on prevailing production requirements and machine availability. Cells can expand and contract in size, and can increase capacity by obtaining access to multiple machines of the same type. On the other hand, the simulation comparison made between virtual cell shop and classical GT proved the superiority of virtual cell configuration for several criteria. Moily et al. (1992) state that, although implementing logical or virtual cells do not provide benefits of reduced materials flow, they do help to simplify the scheduling problem as well as provide the advantages of reduced setups.

1.4 Overview of Cellular Manufacturing Systems

The increasingly competitive global marketplace has forced manufacturers to look for ways to improve plant efficiency quickly. Cellular manufacturing (CM) is one option that is often investigated (Shafer and Charnes 1993). CM is the implementation of the group technology. Group technology (GT) is a philosophy that capitalizes on exploiting product similarities and was designed as a means of improving design and manufacturing productivity to cope with the rapidly expanding product diversity (Wirth et al. 1993). GT also exploits the 'sameness' or similarity of operation processes in design and manufacture (Salvendy 1982). GT takes advantage of the similarities in design and manufacturing attributes of different parts and groups them into families according to these similarities (Snead 1989). The similarities among the parts within group technology families lead to substantial setup time reductions. To further exploit the part similarities, machines should also be arranged into manufacturing cells so that each cell (i.e., a group of dissimilar machines) is dedicated to the production of particular families of parts (Ranson 1972).

Cellular manufacturing leads to smaller lot sizes, which in turn results in reduced work-in-process inventory and shorter manufacturing lead times (Huber and Hyer 1985, Mahmoodi and Martin 1997). Other benefits mentioned in the literature: improved human relations, improved operator expertise, reduced material handling (Greene and Sadowski 1984, Groover and Zimmers 1984), and better quality (Suresh and Meredith 1985). However, several studies evidence this (Flynn and Jacobs 1987, Leonard and Rathmill 1977, Morris and Tersine 1990, Rathmill and Leonard 1977, Suresh 1992). Dedicating equipment to families results in a loss of pooling synergy, and thus, poor shop performance (Suresh and Meredith 1994). Also, other possible disadvantages are reduced machine utilization and shop flexibility (Suresh 1992).

Cellular layout demonstrates superiority over a functional configuration (in terms of average time in system and average work-in-process inventory) in a wide variety of operating environments Shafer and Charnes (1993).

The design of cellular manufacturing systems starts with the identification of machine cells and part families. The classification of entities, such as machine and manufactured components, into cells and families respectively, is based on some measure of similarity (Chandrasekharan and Rajagopalan 1986).

The basic data consist of information from the route card arranged in the form of a binary matrix known as a machine-part incidence matrix whose columns and rows represent the components and machines respectively. The (i, j)th element is 1 if the ith machine is used by the jth component. Otherwise it is 0. The problem was originally identified by Burbidge (1975), who developed some heuristic methods suitable for hand computation. Strictly speaking, this simplified approach is valid only when lot sizes of the parts are equal, the required processing times are identical and machine capacities are the same for all the machines. But, because of its simplicity, the machine-part incidence matrix might also be used when the lot sizes are unpredictable, processing times are roughly equal and machine capacities are nearly identical.

In the design of CMS, an 'exceptional' part is the one that needs to be processed in more than one cell and the 'bottleneck' machine is a machine that is needed by many parts from different part families. Bottleneck machines are the source of inter-cell moves and they are inevitable in most practical manufacturing environments. Finding the optimal location of each cell, in the presence of bottleneck machines, is a basic step for solving bottleneck machine problems to improve the material flow. Most approaches to solving the bottleneck machine problem minimize material handling cost (Wang and Sarker 2002).

1.4.1 Similarity coefficients and clustering algorithms in cellular manufacturing

The similarity coefficient method is more flexible in incorporating various types of manufacturing data into the machine cell formation process. This method refers to a set of clustering algorithms developed in the field of numerical taxonomy (orderly classification of plants and animals according to their presumed natural relationships). A similarity measure is defined between pairs of machines or parts, and a clustering algorithm is used to group machines into cells or to group parts into families based on corresponding similarity measures.(McAuley 1972, Seifoddini 1987a, 1987b).

The similarity coefficient has been used to form machine cells for cellular manufacturing applications. McAuley defined the similarity coefficient between two machines as the number of parts visiting both machines divided by the number of parts visiting either machine. Seifoddini (1987a) modified McAuley's similarity coefficient to incorporate the production volume into the machine cell formation process. Gupta (1988) extended this similarity coefficient to incorporate the effect of operational time and operational sequences as well as production volume.

The average linkage clustering algorithm (ALINK) is devised to overcome the deficiencies of the single linkage clustering algorithm, SLINK (Aldenderfer and Blashfield 1984). In ALINK, the similarity coefficient between two clusters is calculated as the average of all pairwise similarity coefficients between members of the two clusters. Such a calculation requires the similarity matrix to be revised every time a new cluster is formed.

1.4.2 Comprehensive machine grouping problem

Comprehensive machine grouping refers to a grouping problem that incorporates processing times, lot sizes and machine capacities. Maximization of the similarities among machines and parts in a cell and minimization of the work imbalance subject to machine capacities are among the objectives of such comprehensive grouping problems. Different versions of comprehensive grouping problems have been studied by a number of researchers (Ballakur and Steudel 1987, Rao and Gu 1995, Zolfaghari and Liang 2003).

The machine/part grouping problems have proven to be NP-complete and cannot be solved in polynomial time. The consideration of processing times, lot sizes and machine capacities will further increase computational complexity of the comprehensive grouping problem. As a result, some emerging search methods have recently been applied to provide approximate solutions. However, the merits of each method and the problems involved in implementation may not be easily apprehended by practitioners, thereby posing difficulties in the selection of an efficient heuristic for industrial applications.

For this reason, Zolfaghari and Liang (2002) performed a comparative study on the performance of the following three important search methods: simulated annealing, genetic algorithms and tabu search for both binary (considering only machines and part families) and comprehensive (involving machine/part types, processing times, lot sizes and machine capacities) machine grouping problems. To test the performance of the three meta-heuristics, two binary performance indices (grouping efficiency and efficacy) and two generalized performance indices (generalized grouping efficiency and efficacy) were respectively used for binary and comprehensive machine/part grouping problems.

The comparisons were made in terms of solution quality, search convergence behaviour, and pre-search effort. The results indicated that simulated annealing outperforms both genetic algorithm and tabu search particularly for large problems.

1.4.3 Group scheduling rules

Given a meaningful classification of parts into families and arrangement of machines into cells, many manufacturing professionals believe that utilizing scheduling techniques which utilize some of the unique features of cellular manufacturing is essential to counter its disadvantages and enhance the likelihood of a successful implementation (Ham et al. 1979, Vaithianathan and McRoberts 1982, Sinha and Hollier 1984, Mosier and Taube 1985, Flynn 1987, Mahmoodi et al. 1990).

The general argument is that such techniques (commonly known as group scheduling heuristics) can enhance the advantages of the similarity among parts in a cellular manufacturing by further reducing overall machine setup times, while at the same time diminish its disadvantages by adapting to more diverse ranges of part families (Mahmoodi 1989, Mahmoodi and Martin 1997). A number of heuristics have been proposed to take advantage of this structure (Ruben et al. 1993).

A typical cell manufactures a number of subfamilies (a subfamily is a collection of parts with similar machining or setup characteristics). Group scheduling heuristics are two-stage procedures that attempt to take advantage of short minor setups while avoiding long major setups (Wirth et al. 1993). Major setups are required between parts belonging to different subfamilies, while minor setups are required between parts within a subfamily. Thus, the first stage involves sequencing jobs within each subfamily, while the second consists of determining which subfamily (queue) to select next for processing at a given workcenter. Dispatching rules are utilized for the first stage, while queue selection rules are used for the second stage.

The research on group scheduling can be classified into two categories: scheduling flow-through cells and job shop cells. In a flow-through cell (in its pure form) all the parts have identical routes. In a job shop cell, parts may arrive and depart at different workcentres and have different routings (Mahmoodi et al. 1990).

Also, the scheduling rules approach can be classified into exhaustive and non exhaustive. The non-exhaustive heuristics allow processing jobs from other subfamily queues after completion of each individual job, while the exhaustive heuristic avoid processing jobs from other subfamilies until all the jobs in the current subfamily are processed (Mahmoodi et al. 1991).

Mahmoodi et al. (1990) developed due date oriented group scheduling heuristics for job shop environments. They compared single-stage traditional job shop heuristics and the two-stage group scheduling heuristics. Group scheduling heuristics that utilized queue selection heuristics such as DDFAM (due-date family heuristic which selects the subfamily queue whose first job has the most imminent due date) in conjunction with SI^x (truncated SPT) and SPT dispatching heuristics were among the best performing under various environments.

Mahmoodi et al. (1992) examined a number of group scheduling heuristics in a flow-through cell environment that included traditional single-stage heuristics and the two-stage group scheduling heuristics that have exhibited superior performance in previous studies under a rigorous set of experimental conditions.

Results varied by experimental condition and performance criteria, but in general, two-stage heuristics outperformed single-stage heuristics under all experimental conditions, as well as being relatively insensitive to changing experimental conditions. In addition, two of the two-stage heuristics displayed superior performance on all performance measures under most experimental conditions. These two heuristics were DDSI (due date truncated shortest processing time) and MSSPT (minimum setup shortest processing time), first and second best performing heuristics respectively. DDSI and MSSPT were selected as the scheduling rules implemented in this thesis.

1.5 Research Scope and Objectives

Extensive research has been done on the performance of traditional production systems; also, new trends in manufacturing systems are being developed and tested for application in different types of manufacturing scenarios. These innovative production systems present constraints and are limited to certain applications within the industry and mostly do not consider setup times either for simplicity or because they are not part of the system.

Limited or almost no analysis has been carried out on the suitability and measurement of a combined manufacturing system. That is, as in the case of this thesis, a flexible manufacturing system that processes not only grouped parts but also non-family parts within a cellular manufacturing layout. An innovative "hybrid" system has been developed in this thesis that attempts to outperform the traditional cellular manufacturing. Both systems (hybrid and cellular) employ group scheduling rules since they are both based on group technology to configure the machine cells and part families.

While reviewing the literature, no research was found that tried to model this type of "hybrid" system,. This thesis performs a simulation study of such a system considering a cellular layout and setup times. Factors such as due date, quantity, scheduling rules and process technologies are considered in the design of the experiment.

1.6 Thesis Organization

This thesis is organized as follows. Chapter 1 is dedicated to the literature review. In Chapter 2, a discussion on the proposed hybrid model and cellular system is presented along with the design of both systems. Chapter 3 presents the simulation model assumptions and equations; and the software package implemented for the running of the experiment as well as the verification and validation of the simulation model. In Chapter 4, the statistical analysis of this work is presented and the experimental design is discussed. Output analysis of the simulation model, along with the truncation analysis and replication method of the initial runs is conducted. The experimental design of the simulation includes analysis of variance (ANOVA) of the results. Finally, Chapter 5 presents conclusions and proposes directions for future research.

CHAPTER 2: PROBLEM DESCRIPTION

This chapter presents a description of the concepts involved in the proposed hybrid manufacturing system. A brief description of the well-known cellular manufacturing system is also provided. The introduction to these two manufacturing systems will lead to a better understanding of the problem. The performance of these two systems will be compared later in Chapter 3.

2.1 Problem Objectives

The purpose of this thesis is to compare the performance of a proposed hybrid manufacturing system with the well-known cellular manufacturing system as described below. The performance measures used for this comparison are mean flowtime, tardiness and earliness. These performance measures were selected based on literature related to this work (Mahmoodi et al. 1991, 1992).

2.2 The Cellular Manufacturing System

Cellular manufacturing is considered as the implementation of group technology (GT). GT is a philosophy that searches for similar processing and design characteristics in the parts to be produced, and divides them into families according to these characteristics.

This grouping of parts into families, leads to minimization of setup times in the system. In order to benefit the most from these part similarities, machines should also be grouped into manufacturing cells. Each of these cells, consisting of a group of different machines, is committed to the manufacturing of particular part families.

When part families are to be processed, a major one-time setup is required for the entire family, and then a minor setup is needed before processing each part. Since this major setup is attributed to the entire family, it is referred to as *family setup*, and since the minor setup is part-specific, we refer to it as *part setup*.

2.3 The Hybrid Manufacturing System

The proposed hybrid model functions within a cellular manufacturing system layout. Its main characteristic is that parts do not necessarily have to belong to a family in order to be processed in the manufacturing cells. The hybrid model consists of family and non-family parts. The setup time of a part family in the hybrid system is assigned in the same way as in the cellular manufacturing system described above. It consists of a family setup and a part setup. For non-family parts, their part setup consists of the *total setup* time of the part. That is, the total setup time of a non-family part is not divided into family setup and part setup. The inclusion of non-family parts in the hybrid system leads to the avoidance of family setups.

2.3.1 Justification of the hybrid manufacturing system

Cellular manufacturing systems have demonstrated superiority over job shops (Shafer and Charnes 1993). Job shops, functional layouts, or process layouts as they are sometimes called, are manufacturing systems designed to accommodate variety in product designs in relatively small batches. Job shops typically use general-purpose machines that can be changed over rapidly to new operations for different product designs. These machines are usually arranged according to the type of process being performed. For example, all machining would be in one department, all assembly in another department, and all labelling in another department.

Simplification of machine tools changeovers, reduction in material handling cost and in-process inventory, minimization of mean flowtime and setup times, as well as easiness of production automation are among the main benefits of implementing cellular manufacturing over a job shop manufacturing system.

All of these cellular manufacturing advantages and its superiority over a job shop were decisive elements in selecting the cellular system configuration as the layout used for the proposed hybrid manufacturing system. In the same manner, these advantages and superiority were key in selecting the cellular system as the manufacturing method being compared with the proposed hybrid model.

2.3.2 Flexibility of the hybrid model

The hybrid model is flexible in several ways. First, the production layout of the hybrid model has the ability to be used either as a hybrid manufacturing system or as a cellular manufacturing system. Second, the layout flexibility of the model allows for a variability of products to be processed, such as part families and independent (i.e., non-family) parts.

Also, the hybrid system allows for more parts (non-family) parts to be processed in the same cellular manufacturing system. In other words, the hybrid system is a combination of the cellular manufacturing system and the job shop. In the following sections, the proposed hybrid model will be briefly compared with some recent manufacturing systems proposed in the literature.

2.3.2.1 Comparison with the hybrid multi-cell flexible model

The hybrid multi-cell flexible manufacturing system is a model developed in 1997 by Kattan. Although his model and the hybrid model discussed in this thesis share the term *hybrid*, they are quite different. The difference relies in the way these two systems deal with congested machines. Kattan's hybrid multi-cell flexible manufacturing system is formed by manufacturing cells that can have similar machines among them. Therefore, when a manufacturing cell is congested, the system sends the parts waiting to be processed to a different cell with a less congested identical machine.

On the other hand, the hybrid model adopted in this thesis, assumes that parts waiting to be processed will remain in the buffer of the cell. These parts will wait there for their next operation, until the machine they need becomes available. This will translate into savings in material handling since no additional movement of parts among cells will be necessary. Also, parts will not interfere with parts being processed in other cells. This will further translate into savings in setup times that could be incurred if parts were to move to less congested machines, where different parts are being processed.

2.3.2.2 Comparison with the virtual cellular manufacturing model

A virtual cellular manufacturing system (VCM) consists of conceptual or virtual cells. These virtual cells are formed in the computer software and do not actually exist on the shop floor. Researches and practitioners of the virtual manufacturing system support not performing the physical separation of the operations into cells. They argued that performing such separations could be constrained by practical, technical and organisational factors.

VCM is flexible in its size and capacity. Virtual cells can expand and contract in size, and can increase capacity by obtaining access to multiple machines of the same type. On the other hand, the hybrid model configuration layout is fixed. However, material handling cost is considerably less in the hybrid model, since family parts do not need to move between cells, in order to be processed. Also, parts do not interfere with parts being processed in other machines. This will represent savings in setup times that could be incurred if parts were to move to a machine where a different part type is being processed.

2.3.3 Design of the hybrid model

In the hybrid model, the configuration of machine cells and part families was considered as a comprehensive machine grouping problem. That is, part types, processing times, lot sizes and machines are part of the data problem, and are considered while searching for the best design for the model. Once the set of family parts was found by the simulated annealing (SA) method (Zolfaghari and Liang 1998), non-family parts were introduced in the system.

Figure 2.1 shows the configuration of the test problem. As it can be seen, the hybrid model consists of three part families (part types 1 to 10) processed in three cells, where each of these cells is enclosed in a rectangle. The hybrid model consists as well of 10 non-family parts that are processed within these three cells (part types 11 to 20). The numbers inside the matrix correspond to the processing times (in minutes) of each part.

Machines	Parts 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	3	5								5					3				
2	3	1	3								2	5			1		5			
3	4		1										2					4		1
4				5	2	3	4				2	1		1	4			3		
5				1		5	2					3				2			3	2
6								1	2	4					3				5	4
7								3	3	2			1	4			3			
8								4	1					5			2			

Figure 2.1 Hybrid model matrix configuration

2.3.4 Design of the cellular manufacturing system

The cellular manufacturing system was configured out of the hybrid model matrix configuration by forcing non-family parts in Figure 2.1 to join the existing cells. For this purpose, we used the average linkage clustering algorithm, ALINK (Seifoddini and Wolfe 1986). ALINK was selected since it was devised to overcome the deficiencies of the single linkage clustering algorithm, SLINK (Aldenderfer and Blashfield 1984). These deficiencies in SLINK include the chaining problem and improper machine assignment originated every time a new part is added to a cell without updating the similarity matrix.

ALINK overcomes this deficiency through an iterative revision of the similarity matrix every time a new cell is formed.

Figure 2.2 shows the solution matrix for the cellular manufacturing system. As it can be seen, now there are only three part families (part types one to twenty) to be processed in the three cells, each cell is enclosed in a rectangle. The numbers inside the matrix correspond to the processing times of each part.

		Parts																			
		1	2	3	11	15	18	4	5	6	7	12	16	19	20	8	9	10	13	14	17
Machines	1	5	3	5	5								3								
	2	3	1	3	2	1						5									5
	3	4		1			4								1				2		
	4				2	4	3	5	2	3	4	1								1	
	5							1		5	2	3	2	3	2						
	6					3								5	4	1	2	4			
	7															3	3	2	1	4	3
	8															4	1			5	2

Figure 2.2 CMS matrix configuration

Notice that the cell size in both manufacturing systems (hybrid and cellular) is well within the norm size of a survey of practitioners conducted by Wemmerlöv and Hyer (1989). They concluded that two thirds of the organizations with cellular manufacturing systems had cells which contained six or less machines.

It is worthwhile to mention that the machine number in figures 2.1 and 2.2 correspond to the type of machine. However, these machine numbers do not necessarily correspond to the order in which parts are processed in these machines. For the operation sequences, setup times and processing times of the part types in this study please refer to Appendix I.

2.4 Deterministic Approach of the Hybrid Model

The purpose of this section is to introduce some of the variables and equations that would arise if the hybrid system were to be considered as a deterministic model. Next, a notation list is given, followed by the description of the variables.

i	index of machines, $i = 1,\ldots,M$
j	index of parts, $j = 1,\ldots,N$
k	index of operations, $k = 1,\ldots,K$
$\bar{t}$	average time per part
t_{ijk}	total time of operation k of part j on machine i
N	number of part types
Q_j	batch size of part j
T_i	total processing time on machine i
L_i	average processing time per part on machine i
U_i	utilization of machine

$$X_{ijk} \begin{cases} 1 \text{ if machine } i \text{ is to process operation } k \text{ of part } j \\ 0 \text{ else} \end{cases}$$

2.4.1 Average time per part type

Family and non-family parts in the hybrid model are processed through the manufacturing cells of the system where the average time per part can be computed as:

$$\bar{t} = \frac{\sum_i \sum_j \sum_k t_{ijk}}{N} \tag{2.1}$$

2.4.2 Total processing time on machine *i*

The total processing time on machine i is the product of the total time of operation k of part j (if machine i is to process operation k of part j) and the batch size of part j that is given as:

$$T_i = \sum_j \sum_k t_{ijk} Q_j X_{ijk} \tag{2.2}$$

2.4.3 Utilization of machine *i*

The hybrid model will contain at least one machine for which T_i will be significant as compared to other machines. This machine will be referred to as the bottleneck machine and is given by:

$$T_i^{max} = max\{T_i | i = 1,2,...M\} \tag{2.3}$$

where $T_i^{\max}$ is employed to calculate the percentage utilization for each machine i; computed as:

$$U_i = \frac{T_i}{T_i^{max}} \tag{2.4}$$

2.4.4 Average processing time per part on machine *i*

An important element in the hybrid manufacturing system is how rapidly parts are processed within a machine. This is given by the ratio of the total processing time on machine i (T_i) to the number of parts produced in machine i, provided that machine i is to process operation k of part j. Thus, the average processing time per part on machine i (L_i) is computed as:

$$L_i = \frac{T_i}{\sum_j \sum_k Q_j X_{ijk}} \tag{2.5}$$

2.4.5 Bottleneck machine for part *j*

The bottleneck machine for part j in the hybrid model is computed as:

$$\text{Bottleneck machine for part } j = \max_i \left\{ L_i \;\middle|\; \sum_k X_{ijk} > 0 \right\} \tag{2.6}$$

Equation (2.6) specifies that the machine having the maximum processing time per part (L_i) is the bottleneck machine for part j, provided that all of the machines considered in this selection are needed for at least one operation of part j.

Equations (2.1) to (2.6) were discussed in this section in order to provide insights into the operation of the hybrid model. However, simulation will be the modeling tool and solving procedure for the comparison of the hybrid and cellular manufacturing systems.

The next chapter provides a comprehensive analysis of the simulation study performed in this thesis.

2.5 Model Scheduling Heuristics

Group scheduling heuristics were developed for implementation in cellular manufacturing systems. They can enhance the advantages of the similarity among parts by further reducing overall machine setup times. At the same time, group scheduling heuristics diminish the cellular system disadvantages by adapting to more diverse ranges of part families.

Since this study compares the performance of a cellular manufacturing system and a proposed hybrid model (that functions within a cellular layout) within a simulation model, group scheduling heuristics were used in each of these manufacturing systems.

Group scheduling heuristics consist of two stages. The first stage involves sequencing jobs within each family, while the second consists of determining which family queue to select next for processing at a given machine. Dispatching rules are utilized for the first stage, while queue selection rules are used for the second stage.

Group scheduling rules can be classified into exhaustive and non-exhaustive. Exhaustive heuristics avoid processing jobs from other families until all the jobs in the current family are processed. Non-exhaustive heuristics allow processing jobs from other family queues after completion of each individual job.

Group scheduling research can also be divided into two other classes: scheduling flow-through cells and job shop cells. In a flow-through cell (in its pure form) all the parts have identical routes. In a job shop cell, parts may arrive and depart at different machines and have different routings.

As can be seen from the operation sequences shown in Appendix I, the proposed hybrid model and the cellular manufacturing system perform flow-through cell and job shop cell movements. Therefore, in order to obtain the best scheduling for the hybrid and cellular manufacturing systems that are being compared in this thesis, flow-through cell and job shop cell group scheduling rules were reviewed from literature.

It was found that DDSI and MSSPT group scheduling rules performed the best in a flow-through cell and in a job shop cell environment (Mahmoodi et al. 1992). For this reason, they were selected for this study. Next, a description of DDSI and MSSPT group scheduling rules is given.

2.5.1 Due date SI^x (DDSI)

This heuristic utilizes the SI^x (truncated shortest processing time) heuristic to sequence jobs within families. SI^x is a two-class truncated SPT heuristic that dynamically assigns priority to jobs with zero or negative slack times and orders by SPT (Oral and Malouin 1973). DDSI attempts to minimize the average number of setups while considering the due date performance by choosing the family queue whose first job has the most imminent due date. This heuristic is dynamic and exhaustive and attempts to minimize the number of very late jobs. While considering due date performance as the main objective, this rule supports what managers consider the most important scheduling criterion, which is meeting the due date (Mahmoodi et al. 1988).

2.5.2 Minimum setup shortest processing time (MSSPT)

This heuristic utilizes the SPT heuristic to sequence jobs within families. SPT processes jobs in non-decreasing order of processing time and performs especially well in terms of flowtime measures. MSSPT focuses on minimizing the number of setups and the total setup time. This is done by exploiting the similarity in setup times among jobs. The family queue which requires the least amount of setup is selected. This heuristic is dynamic and exhaustive.

While considering the SPT rule to sequence jobs within families, this group scheduling rule supports one of the first research conjectures in the scheduling study area performed by Cobham (1954). He concluded that, it is best to keep the servicing times for high-priority units as short as possible since occurrence of long servicing times in units of high priority tends to increase the expected wait for units of all levels.

After having defined the manufacturing systems and group scheduling rules used in this study, the following scheduling-manufacturing scenarios can be defined:

1. MS/HMS: Hybrid manufacturing system with MSSPT group scheduling rule
2. DD/HMS: Hybrid manufacturing system with DDSI group scheduling rule
3. MS/CMS: Cellular manufacturing system with MSSPT group scheduling rule
4. DD/CMS: Cellular manufacturing system with DDSI group scheduling rule

2.6 Chapter Summary

This chapter examined the main aspects related to the description of the problem. The justification and flexibility of the hybrid manufacturing system were discussed. Also, the hybrid model was compared with other innovative manufacturing systems, such as, the multi-cell flexible manufacturing system and the virtual cellular system. The designs of the hybrid and cellular manufacturing systems were conducted through the implementation of clustering algorithms. The group scheduling heuristics implemented in the model for analysis of the system performance were given.

CHAPTER 3: METHODOLOGY

This chapter introduces the concepts used for the simulation of the model, such as the main assumptions of the hybrid and cellular manufacturing systems. The equations and variables of the simulation model are provided along with an explanation of the terms. The characteristics of the simulation package employed in this thesis, AweSim with Visual SLAM, are introduced as well as their implication in the verification and validation of the model.

3.1 Simulation Model

Simulation is defined by Sadoun (2000) as the imitation of the operation of a system or real-world process over time. Manufacturing provides one of the most important applications of simulation. Simulation has been the dominant modelling tool of FMSs, that has been used effectively in design, implementation and operation of FMSs (Barash et al. 1981, Vettin 1977, Jeong and Kim, 1998).

Other studies have used simulation to compare the performance of a variety of group scheduling heuristics in cellular manufacturing systems (e.g., Mosier et al. 1984, Flynn 1987, Wemmerlöv and Vakharia 1991, Ruben et al. 1993). Thus, this study uses simulation as the modelling tool to compare a hybrid and a cellular manufacturing systems. The specified systems used here consist of eight machines grouped into three manufacturing cells. Refer to section 2.3.3 and 2.3.4 for the layout design of these two systems.

The following assumptions apply in developing the simulation model:

1. There are different stations $i = 1,\ldots,M$. Each machine has the ability of performing several operations.
2. Each machine can perform at most one operation at a time.
3. Each part may visit each machine only once.
4. No preemption is allowed.
5. Setup times are sequence-independent
6. Exhaustive policy for families is adopted.
7. Unlimited queue capacity is available.
8. At each machine, there will be a separate queue for the part families and non-family parts.
9. Labour is not considered in the analysis of the system. One worker is assumed to be assigned to each machine and therefore labour cost is constant.
10. Distances between machines and between clusters are assumed to be constant.

3.1.1 Model parameters and equations

This section introduces the variables and parameters formulated for the development and analysis of the hybrid and cellular manufacturing systems in the simulation. First, a notation list is provided and then an explanation for each term is given, when necessary, for further clarification.

i	index of machines, $i = 1,\ldots,M$
j	index of parts, $j = 1,\ldots,N$
h	index of family, $h = 1,\ldots,H$
k	index of operations, $k = 1,\ldots,K$
s_{ij}	total setup time for part j on machine i
s_{ij}^{f}	family setup time for part j on machine i
s_{ij}^{p}	part setup time for part j on machine i
β	ratio of the family setup time to the total setup
f_{ih}	family setup time of family h on machine i
Ω_h	set of parts in family h
$\overline{N}$	average number of part types in an order
P	probability of a part being selected for processing
N	number of part types
$\overline{Q}$	average order size of a part
$\bar{l}$	average lead time per order
γ	variable used in the due date assignment

3.1.1.1 Processing times

Processing times for the hybrid and cellular manufacturing systems are the same, these times are fixed and provided in Appendix I. Major (family) and minor (part) setup times are not sequence-dependent (e.g., the setup time required to switch from family 1 to family 3 is the same as the time required to switch from family 2 to family 3).

3.1.1.2 Setup times for the system

It is essential to understand the implementation of part setup times in both hybrid and cellular manufacturing systems. The reason for this is that setup times are the main focus in developing the differences among these two manufacturing methods.

The hybrid manufacturing system consists of family and non-family parts, where a setup time between parts within the same family is referred as a *part setup* and a setup time between parts in different families is referred as *family setup*. On the other hand, a setup time between non-family parts is referred as a *total setup*.

The cellular manufacturing system consists only of part families in which the same setup time types (family setup and part setup) are implemented between the family parts as in the hybrid manufacturing system.

Family setup times are assigned to parts based on a total setup value. The total setup time of part j on machine i can be divided in family setup and part setup, as represented by the following equation:

$$s_{ij} = s_{ij}^{f} + s_{ij}^{p} \tag{3.1}$$

In order to know which fraction of s_{ij} would be assigned to the family setup and which to the part setup, the β variable, where $0 < \beta < 1$, is multiplied by the total setup time of the part. The product of this multiplication will be the family setup numerical value (s_{ij}^{f}). Whereas the difference between the total setup and the family setup will be the part setup (s_{ij}^{p}).

Therefore, β can be computed as the ratio of the family setup time to the total setup time. This relationship can be expressed by the following formula:

$$\beta = \frac{s_{ij}^{f}}{s_{ij}} \tag{3.2}$$

Given Equation (3.2), the relationship between part setup and β can be examined and the following equation can be obtained:

$$\frac{s_{ij}^{p}}{s_{ij}} = 1 - \beta \tag{3.3}$$

Once the family setup time, s_{ij}^{f}, is computed for each of the family parts through the multiplication of β and total setup time (s_{ij}), a single family setup must be determined for each group or part family. This unique family setup time for family h on machine i is assigned by the maximum family setup time of all the set of parts that belong to that family. This is expressed through the following equation:

$$f_{ih} = \underset{j \in \Omega_h}{Max} \left\{ s_{ij}^{f} \right\} \tag{3.4}$$

A list of the setup times in the hybrid and cellular manufacturing systems is provided in Appendix I.

3.1.1.3 Average number of part types in an order

The manufacturing systems considered in this study are assumed to fabricate a relatively large range of different products. Therefore, orders are presumed to consist of only a fraction of the part types produced by the system. For this purpose, a probability is applied to define the likelihood of a part being included in an order. Thus, the average number of part types in an order is given by:

$$\overline{N} = P.N \tag{3.5}$$

Equation (3.5) implies that only a certain number (determined by a probability) of the total number of part types in the system, will be selected to be part of an order.

3.1.1.4 Average order size of a part

This thesis performs the simulation of a discrete event model. In a discrete event, variables change stochastically at a discrete set of points in time. One of the variables in the model is the average order size of a part. The average order size of a part is given by:

$$\overline{Q} \approx \mathrm{N}\,(\mu\,,\sigma) \tag{3.6}$$

As indicated above, $\overline{Q}$ varies within the system by means of a normal distribution with parameters μ and σ, which correspond to the mean value and standard deviation of the normal distribution.

3.1.1.5 Average lead time per order

The average lead time per order is given as:

$$\bar{l} = \lambda = \overline{N} \,.\, \overline{Q} \,.\, \bar{t} \tag{3.7}$$

where $\overline{N}$ = the average number of part types in an order, $\overline{Q}$ = the average order size of a part and $\bar{t}$ = the average time per part in the simulation system.

The average lead time was fixed to 200 minutes since the reduction of the processing time for parts due to setup savings, would process parts rapidly and put the simulation system in an idle state.

3.1.1.6 Order due dates

The order due dates are taken from a uniform distribution U(a,b) where:

$$a = \bar{l} - \gamma\bar{l} \quad , \quad b = \bar{l} + \gamma\bar{l} \tag{3.8}$$

Equation (3.8) introduces γ. This variable is used to obtain the due date of an order. In doing so, first, the product obtained from multiplying γ by $\bar{l}$ will be reduced from the average lead time per order ($\bar{l}$), in order to obtain the lower bound of the uniform distribution. Then, the product obtained from multiplying γ by $\bar{l}$ will be added to the average lead time per order ($\bar{l}$) in order to obtain the upper bound of the uniform distribution. Finally, these low and high values will be used as the uniform distribution parameters for the assignment of the order due date.

3.1.2 AweSim with Visual SLAM

The simulation was modelled in AweSim with Visual SLAM (Pritsker and O'Reilly 1999). AweSim is a general purpose simulation system that includes the Visual SLAM simulation language to build discrete event and continuous models. This study simulates a discrete event. Discrete event simulation, as described by Banks and Nicol (2001), is the modeling of a dynamic system in which state variables change stochastically at a discrete set of points in time.

In this thesis, Visual SLAM provided the framework for modelling the flow of parts through the cells. This framework consisted of a network and subnetwork graphic structure formed by specialized nodes and branches used to model machines, queues for machines, activities, and part (entity) scheduling decisions. In particular, subnetworks provide for hierarchical modeling as entities (parts) are transferred from a calling network to a subnetwork. The subnetwork model in this study was built in order to avoid repetitive modeling of the processing of parts within both the hybrid and cellular manufacturing systems. Refer to Appendix II for the model network and subnetwork.

Finally, the last step in developing a simulation model with AweSim and Visual SLAM consists in combining the network and subnetwork description statements with the necessary control statements. Control statements provide information about the simulation experiment to be performed, and are also part of the echo report. In this study, control statements were also generated as an array of product data. This array functioned as a reference for AweSim while performing the simulation of the problem. Control statements are provided in the echo output report explained next.

Output reports are generated by the Visual SLAM processor. The output reports include the echo report, summary report and multi-run summary report. The data for each report is available from the AweSim database.

The Network and the Subnetwork are reported in AweSim as a list of statements in an Echo Report. This report provides a listing of input statements, where each statement is assigned a line number, and, if an input error is detected, an error message is printed immediately following the statement where the error occurred. The Visual SLAM processor, as mentioned above, interprets each input statement and performs extensive checks for possible input errors.

The Visual SLAM summary report displays the statistical results for the simulation and is automatically printed at the end of each simulation run. This report consists of a general section followed by the statistical results for the simulation categorized by type. A section for each subnetwork instance then reports on the same statistical results by instance.

If the simulation consists of more than one run, the Visual SLAM multiple run summary report displays the average of all runs for each statistic. Maximum average value, minimum average value, mean value and standard deviation of the average values of each performance measure are examples of some of the statistics given in this type of report.

Refer to Appendix III for samples of the echo and multi-run summary reports of the DD/CMS scenario used for the output analysis in Section 4.1.

3.2 Verification of the model

The purpose of model verification is to assure that the conceptual model is reflected accurately in the computerized representation. This might involve making parameter changes and ascertaining if performance measures change in the expected manner (Banks and Nicol 2001).

One method of verification is to check that each model element is described correctly and that modeling elements are interfaced as specified. AweSim contains the Interactive Executive Environment which allows modelers to watch the running of a model as each status change occurs at defined breakepoints established by the user. At these breakpoints, the modeler can incrementally execute the simulation, examining the changes of the state of the model caused by simulation events.

The Interactive Execution Environment is also used to follow the logical flow described in the model. The modeler can simulate to a decision point, save the system status, and execute the simulation further from that breakpoint. The modeler can then examine the model's response to alternate inputs by restoring the simulation to the previously saved state, changing the appropriate inputs, and then restarting the simulation from the breakpoint. In this way, the verification process is supported by allowing the user to make runs interactively and to access model status variables during an interactive session.

In the verification process of the model, preliminary runs of the model were made with all the randomness removed from the due date and average order size of a part. Counts were made on the model logic response for the processing behaviour of each of the manufacturing-group scheduling scenarios previously described with each of the β and γ levels. These counts were checked against the number expected from manual calculations in order to insure that no discrepancies existed between expected and observed model performance. Then, the necessary changes were performed until the same calculated results were obtained in the simulation model.

The Interactive Executive Environment greatly aided in the verification process of the model. Errors in modeling logic were isolated and solved while using these features available in this debugger.

3.3 Validation of the model

Validation is the process of determining that the simulation model is a useful or reasonable representation of the system. Validation is usually achieved through the calibration of the model, an iterative process of comparing the model to actual system behaviour and using the discrepancies between the two, and the insights gained, to improve the model. This process is repeated until model accuracy is judged to be acceptable (Kleijnen 1995 and Van Horn 1971).

Typically, validation is made with respect to requirements established for the actual performance of each particular type of system. However, for models of new system design such as the case of this thesis, the validation process is more difficult because model outputs cannot be compared to measures of actual systems performance.

Therefore, in order to validate the model, the structure and expected operation of the hybrid and cellular manufacturing system designs was compared with the structure and operation of the model. Each individual component (network and subnetwork) of the model and the interface between components was examined. The second means of validating the manufacturing designs was to review model outputs for reasonableness. For example, examination of the utilization of a machine in a system may have revealed that its utilization was unreasonably low (or high), a possible error probably caused by wrong specification of processing time, or a mistake in model logic that sends too few (or too many) parts to this particular server, or any number of other possible parameter misspecifications or errors in logic. The reasonableness of the model was also evaluated through sensitivity analysis; assessing the sensitivity of outputs to variation of factors in the experimental design of this thesis.

3.4 Chapter Summary

In this chapter, the assumptions of the simulation model were given. The main features and a comprehensive explanation of the simulation modeling software package AweSim with Visual SLAM used for this study was presented. Also, two important stages in the simulation process, verification and validation, were defined. The variables and equations used to generate data for the problem, such as, setup times, order size of a part, order due dates, etc., were given along with an explanation of the equations that assign values to these variables.

CHAPTER 4: STATISTICAL ANALYSIS

This chapter provides an analysis and discussion of the simulation results. The output analysis of the initial simulation runs is conducted. The initial runs of the scenarios are performed in order to obtain enough data to conduct the truncation analysis and replication method, to finally obtain the run length of the experiment. The results of the experimental design section in this chapter are evaluated through analysis of variance (ANOVA).

4.1 Output Analysis

Output analysis is the examination of data generated by simulation. The purpose of the output analysis is to predict the performance of a system or to compare the performance of two or more alternative system designs. In a multi-run simulation, $\hat{\theta}_r$ denotes the performance measure for run *r*. In performing the statistical analysis of the output of all runs two essential variables are to be estimated:

1. The true performance measure θ can be estimated by a point estimator $\hat{\theta}$ such that $\theta = E(\hat{\theta})$.

2. Error in the point estimate, in the form of either a standard error or a confidence interval.

Classical methods of statistics may be used because $\hat{\theta}_1, \hat{\theta}_2,\ldots,\hat{\theta}_R$ constitute a random sample; where $\hat{\theta}_r|r = 1,\ldots R$ is an independent and identically distributed random variable. Since θ is the parameter being estimated, then each $\hat{\theta}_r$ could be considered as an unbiased estimate of the true performance measure.

4.1.1 Initialization bias

According to the length of its duration and initialization time, there are two types of simulation considered in an output analysis, terminating simulation and steady-state simulation. A terminating simulation is one that runs for a period of time T_E , where E corresponds to the event that stops the simulation. If a terminating system simulation is to be modelled, the initial conditions of the system at time 0 must be specified along with the specification of the stopping event E. The other type of simulation is referred to as the steady state simulation, which is the type used in this study. The steady-state simulation objective is to study long-run (steady-state) behaviour of a nonterminating system. Where the nonterminating system can be defined as a system that runs continuously or at least over a long period of time.

To reduce the point estimator bias (caused by artificial and unrealistic initial conditions) in a steady state simulation, the simulation will be run in two phases. An initialization phase from time 0 to time T_0 (obtained through a truncation analysis described in the next subsection), followed by a data collection phase from time T_0 to the stopping time T_0+T_E.

Data collection on the response variables is for period T_E as illustrated in Figure 4.1 (Banks and Nicol 2001).

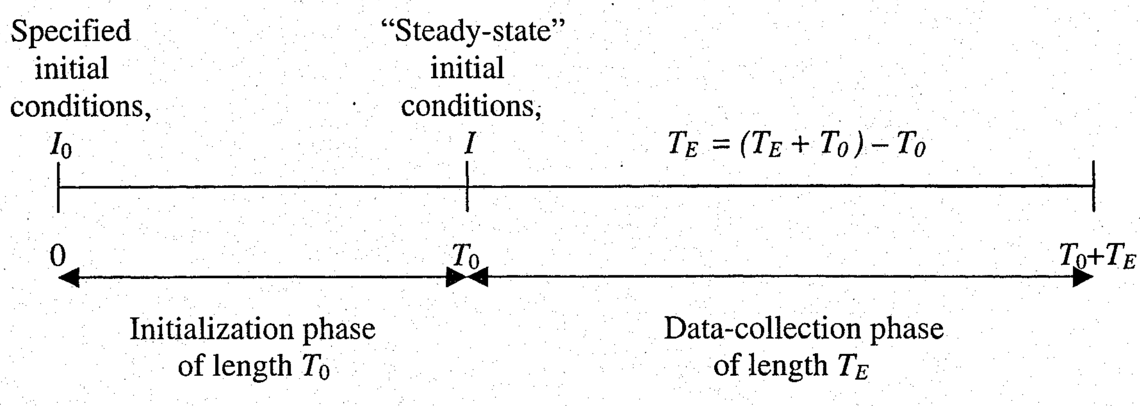


Figure 4.1 Initialization and data-collection phase of a steady-state simulation run

During the initialization phase T_0, an inconsistent trend in the data is observed. To identify the trend in the data and find when it dissipates, a truncation procedure was followed for each of the three performance measures: mean flowtime (F), tardiness (T) and earliness (E). For simplicity, only the mean flowtime (F) performance measure under the cellular manufacturing system model with DDSI group scheduling rule (DD/CMS) will be described. Although, the same procedure was followed for each of the performance measures in the other scheduling-manufacturing scenarios (MS/HMS, DD/HMS, MS/CMS and DD/CMS).

4.1.2 Truncation analysis

Initially, five independent replications were attempted (R=5), each replication beginning in the empty and idle state. The total length of each replication was 1000 minutes (defined arbitrarily), (T_0+T_E). The response variable was the mean flowtime, $\overline{F}(t, r)$ at time t, where the second argument, r, denotes the replication index (r = 1,2,...,5). The output of the simulation was batched in intervals of 100 minutes. The mean value of each replication for batch j=1,...10 can be calculated as follows:

$$Y_{rj} = \frac{1}{100} \int_{(j-1)100}^{j(100)} F(t,r)\,dt \qquad \text{for } r=1,\dots,5 \text{ and } j=1,\dots,10 \tag{4.1}$$

The estimator in Equation 4.1 is simply the time-weighted average flowtime over the time interval [(j-1)100, j(100)). The 10 batch means for the 5 replications are given in Table 4.1.

The ensemble averages are the average of corresponding batch means across replications and are given by:

$$\overline{Y}_j = \frac{1}{R} \sum_{r=1}^{R} Y_{rj} \tag{4.2}$$

where R = is the number of runs or replications.

The ensemble averages $\overline{Y}_j$, j = 1, 2,...,10 are displayed in the last column of Table 4.1. Notice that $\overline{Y}_1$= 52.21 and $\overline{Y}_2$= 148.05 are estimates of mean flowtime over the time periods [0,100] and [101,200], respectively, and they are less than most of the other ensemble averages $\overline{Y}_j$ (j = 3, 4,...,10). This is due to downward bias in these estimators, since the system is empty and idle at time 0. This downward bias of the initial observations is illustrated in Figure 4.2.

Batch, (j)	Run Length, (T)	Replication, r 1	2	3	4	5	Ensemble Average, $\overline{Y}_j$
1	0-100	55.69	46.43	59.42	48.97	50.52	52.21
2	101-200	147.70	147.83	149.88	142.83	152.03	148.05
3	201-300	135.88	126.74	93.37	56.69	162.46	115.03
4	301-400	189.12	227.62	169.66	175.58	189.35	190.27
5	401-500	94.570	248.10	158.65	102.97	110.27	142.91
6	501-600	209.39	277.07	236.84	184.61	201.47	221.88
7	601-700	248.07	330.18	197.97	199.54	174.41	230.03
8	701-800	227.96	376.48	292.53	151.52	242.83	258.27
9	801-900	270.75	393.01	327.68	140.99	278.33	282.15
10	901-1000	294.97	334.03	369.94	310.10	309.34	323.68

Table 4.1 Ensemble average for batch j ($\overline{Y}_j$)

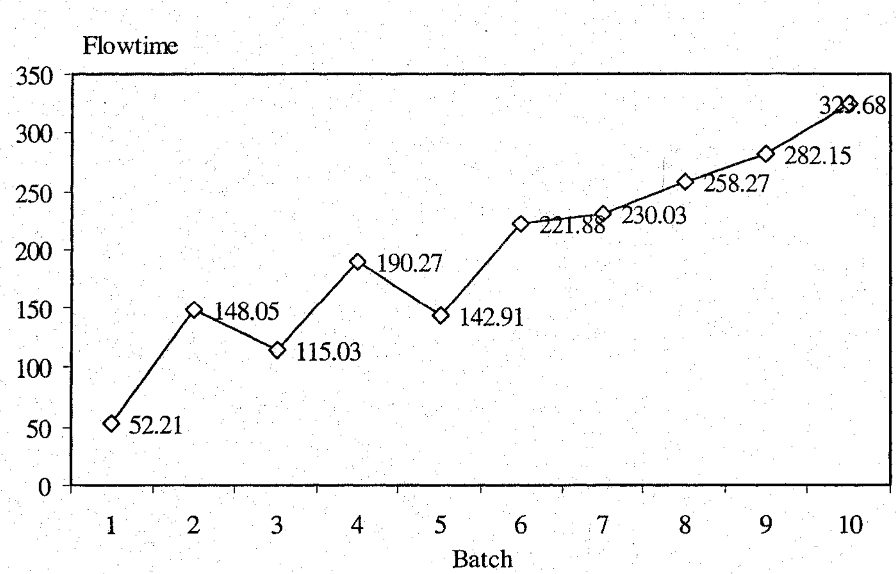


Figure 4.2 Ensemble averages ($\overline{Y}_j$) for the DD/CMS (flowtime) scenario

Figure 4.2 shows that, as time increases, the effect of the initial conditions on later observations reduces and the observations appear to vary around a common mean. Table 4.2 gives the cumulative averages of the sample mean after respectively deleting zero, one, two, and up to five batch means from the beginning. That is, the ensemble average batch means $\overline{Y}_j$ are calculated when deleting $d = 0, 1, \ldots, 4$ observations as:

$$\overline{Y}..(j,d) = \frac{1}{j-d} \sum_{k=d+1}^{j} \overline{Y}_k \tag{4.3}$$

These d-truncate cumulative averages $\overline{Y}..(j,d)$ are listed in Table 4.2 and plotted for comparison purposes in Figure 4.3. From Figure 4.3 it is apparent that downward bias exists, and this initialization bias in the point estimator could be reduced by eliminating some of the initial batch observations. For the 10 ensemble average batch means in Figure 4.3, it appears that the first four observations have considerably more bias than any of the remaining ones. Therefore, a 4-truncation is selected to begin the data-collection phase.

Batch (j)	Cumulative Average (0-Truncate) $\overline{Y}..(j,0)$	Cumulative Average (1-Truncate) $\overline{Y}..(j,1)$	Cumulative Average (2-Truncate) $\overline{Y}..(j,2)$	Cumulative Average (3-Truncate) $\overline{Y}..(j,3)$	Cumulative Average (4-Truncate) $\overline{Y}..(j,4)$
1	52.21	-	-	-	-
2	100.13	148.05	-	-	-
3	105.10	131.54	115.03	-	-
4	126.39	151.12	152.65	190.27	-
5	129.69	149.06	149.40	166.59	142.91
6	145.06	163.63	167.52	185.02	182.39
7	157.20	174.69	180.02	196.27	198.27
8	169.83	186.63	193.06	208.67	213.27
9	182.31	198.57	205.79	220.92	227.05
10	196.45	212.47	220.53	235.60	243.15

Table 4.2 The d-truncate cumulative averages $\overline{Y}..(j,d)$ for the DD/CMS (flowtime) scenario

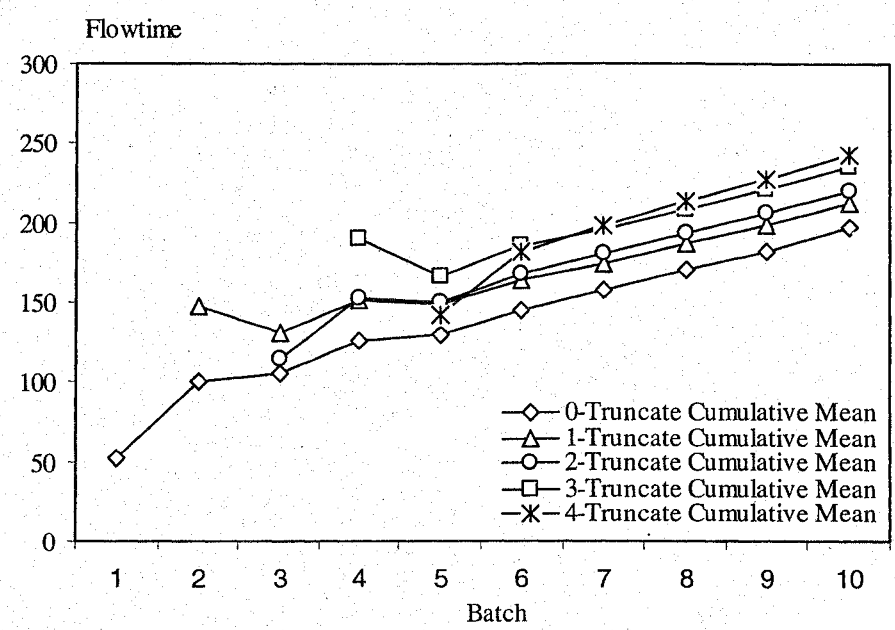


Figure 4.3 The d-truncate cumulative averages $\overline{Y}..(j,d)$ for the DD/CMS (flowtime) scenario

This truncation analysis is repeated for all 3 scheduling-manufacturing scenarios on each of the 3 performance measures. Results on these computations also indicate that a 4-truncation would eliminate most of the initialization bias. Therefore, the 4-truncation was selected for the entire study.

Now that the initialization bias in the point estimator has been reduced to a lower level, the method of independent replications can be used to estimate point-estimator variability and to construct a confidence interval.

4.1.3 Replication method

When using the replication method, each replication is regarded as a single sample for the purpose of estimating mean flow time, F. The sample mean of all remaining observations in replication r, is given by:

$$\overline{Y}_r(n,d) = \frac{1}{n-d} \sum_{j=d+1}^{n} Y_{rj} \tag{4.4}$$

where n is the total number of observations and d is the number of deleted observations.

Since all replications use different random-number streams and all are initialized at time 0 by the same set of initial conditions (I_0), the replications averages $\overline{Y}_1(10,4),\ldots,\overline{Y}_5(10,4)$ are independent and identically distributed random variables, and constitute a random sample from an underlying population having unknown mean flowtime $F_{10,4} = E[\overline{Y}_r(10,4)]$.

The overall point estimator, is also given by:

$$\overline{Y}..(n,d) = \frac{1}{R} \sum_{r=1}^{R} \overline{Y}_r(n,d) = 243.15 \tag{4.5}$$

$\overline{Y}..(n,d)$ is an approximately unbiased estimator of F. For convenience, the values of $\overline{Y}_r(10,4)$ (the mean of the undeleted observations from the rth replication) and $\overline{Y}..(10,4)$ (the mean of $\overline{Y}_1(10,4),\ldots,\overline{Y}_5(10,4)$), where n = 10 and d = 4 as discussed above) will be abbreviated as $\overline{Y}_r$ and $\overline{Y}..$, respectively.

To estimate the standard error of $\overline{Y}..$, the sample variance is computed first by

$$S^2 = \frac{1}{R-1} \sum_{r=1}^{R} (\overline{Y}_r - \overline{Y}..)^2 = 3019.88 \tag{4.6}$$

Thus, the standard error of $\overline{Y}_{..}$ is given by

$$\overline{Y}_{..} = \frac{S}{\sqrt{R}} = 24.58 \tag{4.7}$$

And finally, the 95% confidence interval for *F*, based on the t-distribution, is given by

$$\overline{Y}_{..} - t_{\alpha/2,R-1} \frac{S}{\sqrt{R}} \leqslant F \leqslant \overline{Y}_{..} + t_{\alpha/2,R-1} \frac{S}{\sqrt{R}} \tag{4.8}$$

where α is the level of significance that is selected to be 5%. Thus, $174.918 \leqslant F \leqslant 311.386$.

The values of the two variables initially requested for the output statistical analysis, estimation of the true performance measure, *F*, and the error of this estimate have been calculated. These results were obtained from a simulation consisting of 5 replications with a duration of 1000 minutes each, and a computed initialization period with 400 minutes truncation.

However, if it is desired to estimate *F* within a specified precision and confidence interval, either one of the following two procedures needs to be performed:

1. Increase the number of replications *R* to R^{new}.
2. Increase the run length T_0+T_E to $\frac{R^{new}}{R}(T_0 + T_E)$ as depicted in Figure 4.4.

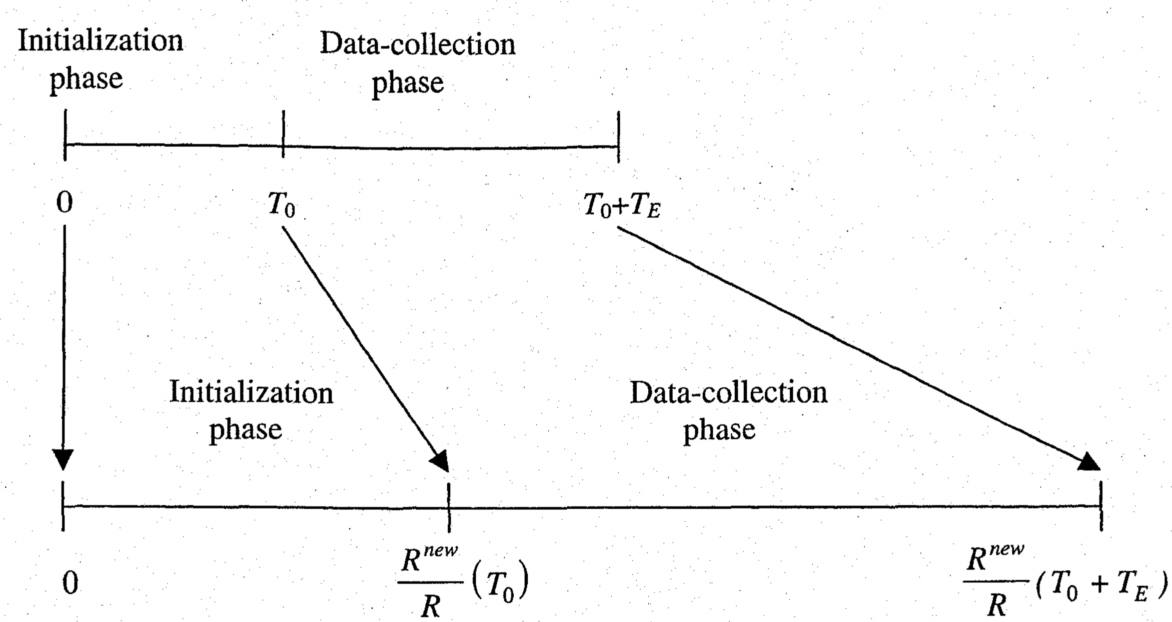


Figure 4.4 Increased run-length to achieve specified accuracy

For this study, the long-run performance measure, F, is desired to be estimated within a level of accuracy, $\pm\varepsilon = 9$ and a 95% confidence interval (α=0.05).

Therefore a new number of replications (R^{new}) and a new run length $\frac{R^{new}}{R}(T_0 + T_E)$ will be calculated. Recall that either one of these solutions is valid. However, in order to estimate a new run length, R^{new} has first to be calculated, whereas the new run length does not need to be known in order to estimate R^{new}. For the purpose of familiarizing the reader with both procedures; the two of them will be computed.

There are two stages for calculating R^{new}. The first one involves obtaining an initial estimate of the variable through an equation that uses a normal distribution. The second stage involves selecting the exact value of R^{new} with a condition that uses student t distribution (Banks and Nicol 2001).

An initial estimate of R^{new} is equal to the smallest integer R greater than or equal to $\left(\frac{z_{\alpha/2}S}{\varepsilon}\right)^2$. Once this value is found, it will be incremented until is satisfies the following condition:

$$R \geqslant \left(\frac{t_{\alpha/2,R\text{-}1}S}{\varepsilon}\right)^2 \tag{4.9}$$

The value of R that satisfies the above condition is the new number of replications R^{new}. Following this procedure R^{new} is found to be equal to 146.

Based on this result, the new run-length for the system can be computed through the following equation:

$$\text{New run length} = \frac{R^{new}}{R}(T_0 + T_E) = 29{,}200 \text{ minutes}, \tag{4.11}$$

$$\text{with a new initialization period, } \frac{R^{new}}{R}(T_0) = 11{,}680 \text{ minutes.} \tag{4.12}$$

These calculations are also computed in each of the other 3 scheduling-manufacturing scenarios for each of the 3 performance measures evaluated, as reported in Table 4.3.

	Scenario			
Performance	MS/HMS	DD/HMS	MS/CMS	DD/CMS
Flowtime, $\overline{F}$	8,400	17,200	16,800	*29,200*
Tardiness, $\overline{T}$	6,400	8,800	7,400	16,200
Earliness, $\overline{E}$	1,000	1,800	4,000	2,800

Table 4.3 New run length (minutes) for each of the performance measures

The *highest* new run length (and its corresponding new initialization period) is selected for the entire study. Results in Table 4.3, indicate that the DD/CMS scenario when the mean flowtime performance measure is used produces the highest value. Therefore, additional data will be deleted, from time 0 to time 11,680 which is the corresponding initialization period for the new run length of 29,200 minutes. This new run length and initialization period will be implemented for the simulation of all the scenarios in the experimental design.

It is worth mentioning that the total amount of simulation effort in increasing the run length is the same as if we had simply increased the number of replication but maintained the original run length.

4.2 Experimental Design

Four experimental factors were considered in this study. The group scheduling rule (GSR): DDSI (due date truncated shortest processing time) and MSSPT (minimum setup shortest processing time; the type of manufacturing system (TMS): HMS (hybrid manufacturing system) and CMS (cellular manufacturing system), β and γ. Where β is a variable used in assigning family setup times, and γ is a variable used in assigning order due dates (refer to section 3.1.1 for further clarification). Table 4.4 shows the different levels chosen for each factor.

Factors	Levels		
	1	2	3
Group scheduling rule (GSR)	DDSI	MSSPT	-
Manufacturing system (TMS)	HMS	CMS	-
β	0.3	0.5	0.8
γ	0.1	0.3	0.5

Table 4.4 Levels chosen for each factor

The sensitivity analysis conducted in this study played an important role in the process of validating the model, as discussed in section 3.3. This is done by determining the expected behaviour of the model input variables. In order to perform the sensitivity analysis values of β and γ are varied in the model, and they are included as factors in our statistical analysis.

The combination of different levels of the four factors provides a total of 36 (i.e., 2×2×3×3) experimental conditions. These 36 combinations were studied to measure the performance of the hybrid and cellular manufacturing systems under different environments. This study considers two dimensions of heuristic performance: how efficiently parts are processed through the cell (i.e., mean flowtime), and how well the schedules adhere with the due dates (i.e., average tardiness and average earliness). Also, given that the parts produced provided interesting results, they were added in the experiment as a fourth performance measure, *P*. It was observed in the output summary report of AweSim that the number of parts produced by the hybrid system consistently outperformed the cellular system in most scenarios.

Since the focus of this thesis is to investigate shop performance under steady-state conditions, a truncation procedure preceded the analysis of the results. A start-up period of 11,680 minutes was truncated from the total data collection of 29,200 minutes to ensure steady-state conditions. It was concluded in the output analysis that the initialization period of 11,680 minutes and the new run length of 29,200 minutes would provide sufficient precision in estimating the mean differences among performance measures, as well as the confidence interval desired.

4.2.1 Analysis of results

Table 4.5 presents the mean values of each of the performance measures for all 36 experiments. Recall that the number of parts produced by the manufacturing system was added as a variable in the experiment.

To aid the reader graphical representations are also presented in figures 4.5 to 4.8. For simplicity, the labels are abbreviated as follows. The first letter and number indicate the γ level, the second letter and number stand for the β level, and the third letter and number indicated whether group scheduling rule MSSPT (1) or DDSI (2) was used. A discussion of the performance of the hybrid and cellular heuristics with respect to each performance measure is presented below.

For example, G1B1S1 in the x axis indicates: γ level of 1 = 0.1, β level of 1 = 0.3 and S level of 1 = MSSPT rule. For the y axis, recall that performance measure units are given in minutes.

Group Scheduling Rule	β	γ	Model	Performance Measures $\overline{F}$	$\overline{T}$	$\overline{E}$	$\overline{P}$
MSSPT	0.3	0.1	HMS	1419.24	1279.94	61.00	16337
			CMS	1962.41	1813.53	51.30	14655
		0.3	HMS	1419.24	1280.62	62.27	16337
			CMS	1962.41	1813.99	52.12	14655
		0.5	HMS	1419.24	1282.19	64.44	16337
			CMS	1962.41	1815.04	53.53	14655
	0.5	0.1	HMS	1535.83	1390.73	55.21	16324
			CMS	1649.72	1493.94	44.50	13635
		0.3	HMS	1535.83	1391.48	56.57	16324
			CMS	1649.72	1494.04	45.18	13635
		0.5	HMS	1535.83	1393.00	58.69	16324
			CMS	1649.72	1494.71	46.43	13635
	0.8	0.1	HMS	1418.35	1274.68	56.86	16323
			CMS	1752.18	1599.95	48.08	14464
		0.3	HMS	1418.35	1274.96	58.21	16323
			CMS	1752.18	1600.26	49.01	14464
		0.5	HMS	1418.35	1276.02	60.33	16323
			CMS	1752.18	1601.10	50.48	14464
DDSI	0.3	0.1	HMS	4751.50	4575.96	24.63	15062
			CMS	1486.30	1305.37	19.59	15038
		0.3	HMS	5263.66	5086.89	24.11	14999
			CMS	1486.39	1306.84	20.85	15422
		0.5	HMS	5113.40	4936.36	23.20	15251
			CMS	1368.23	1188.77	22.25	15947
	0.5	0.1	HMS	5680.19	5494.24	13.93	15431
			CMS	1732.19	1544.01	12.45	14051
		0.3	HMS	5780.10	5593.05	12.78	15401
			CMS	1690.38	1506.63	18.17	14009
		0.5	HMS	5778.60	5594.63	15.90	15491
			CMS	1629.82	1444.25	15.79	13635
	0.8	0.1	HMS	5830.50	5646.67	16.04	15034
			CMS	1296.35	1122.51	26.65	15970
		0.3	HMS	5723.52	5541.08	17.61	15522
			CMS	1532.83	1357.38	25.07	15606
		0.5	HMS	5564.00	5381.77	17.84	15236
			CMS	1486.76	1313.69	29.09	15768

Table 4.5 Summary of experimental results

4.2.1.1 Mean flowtime

From Figure 4.5, it can be seen that the HMS performed better in terms of mean flowtime than the CMS when MSSPT rule was applied. However, it suffered a dramatic increase when the DDSI rule was used. Overall, the CMS showed much more consistent results than the HMS. The CMS outperformed the HMS when the DDSI rule was applied. Figure 4.5 also suggests a small effect from the β and γ factors which will be confirmed later using ANOVA.

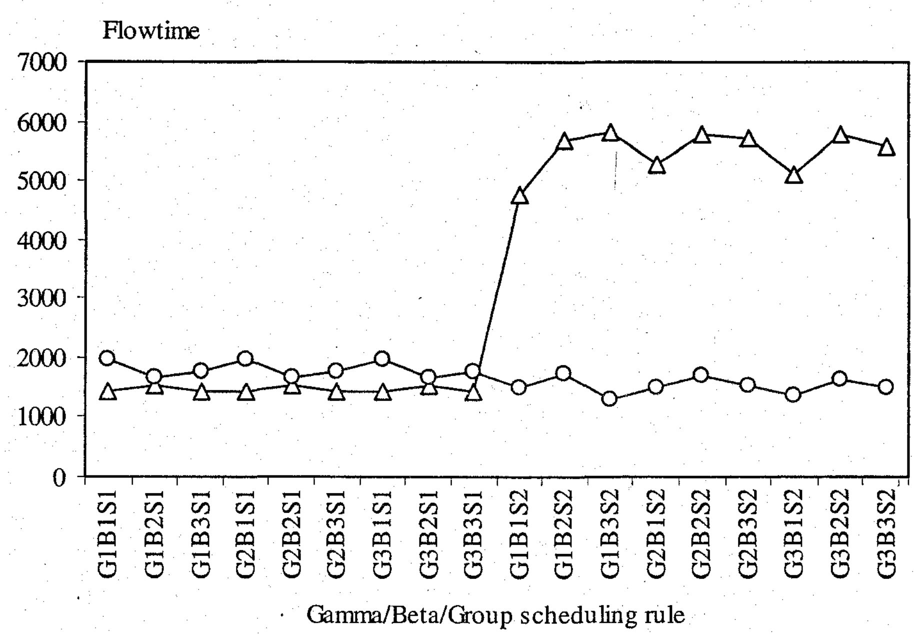


Figure 4.5 Mean flowtime interaction of γ level, β level and GSR

4.2.1.2 Tardiness

Figure 4.6 shows the performance of different scenarios when tardiness is selected as the performance measure. When the DDSI rule was applied, the tardiness performance of the HMS increases in a high range, and it performs the best when MSSPT is applied. Also, when the MSSPT rule is applied, the HMS outperforms the CMS. On the other hand, the

CMS keeps a constant performance regardless of the group scheduling rule, showing a slightly better performance when the DDSI rule is applied. When the DDSI rule is implemented, the HMS is outperformed by the CMS.

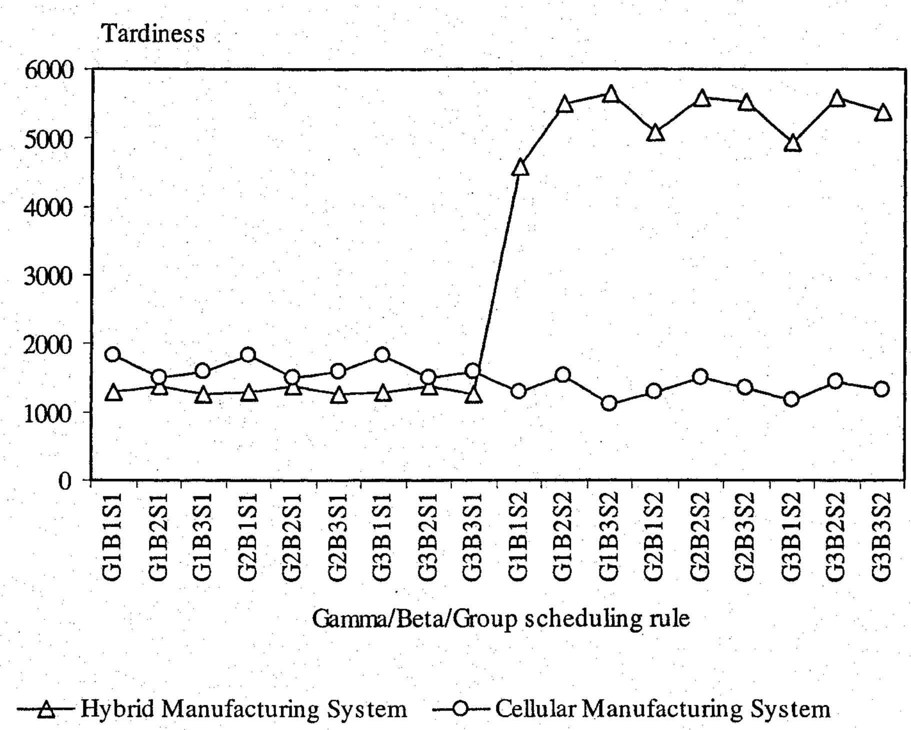


Figure 4.6 Tardiness interaction of γ level, β level and GSR

4.2.1.3 Earliness

This thesis searches for the lowest possible value, bigger than zero, of earliness since just-in-time (JIT) characteristics are taken into consideration. Results from Figure 4.7 show that when this performance measure is evaluated both HMS and CMS exhibit a significant difference in performance as group scheduling rules are changed. When the MSSPT rule is applied the HMS is outperformed by the CMS. When the DDSI rule is applied, the HMS outperforms the CMS, in this case, the difference among the performance of the two manufacturing systems is not as clear as in the previous cases.

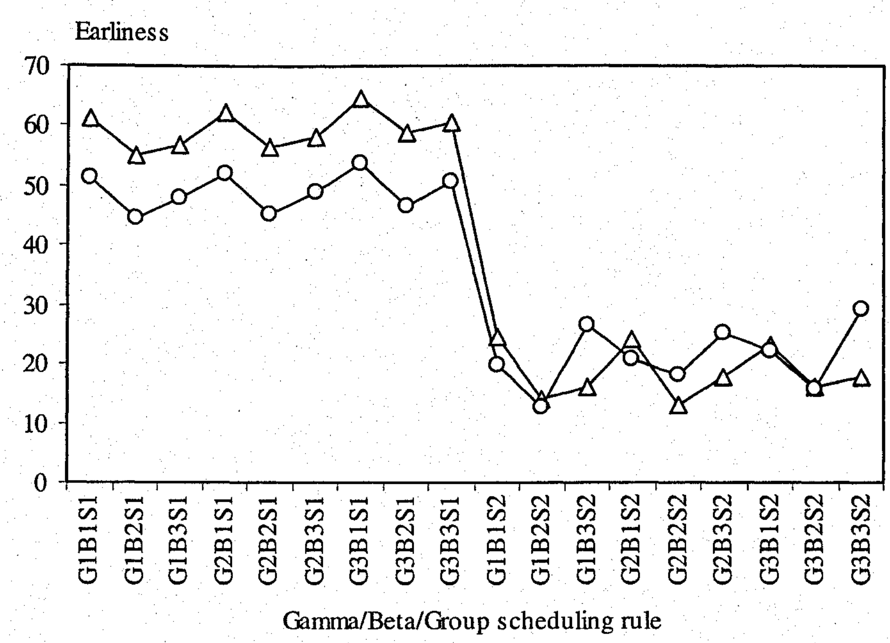


Figure 4.7 Earliness interaction of γ level, β level and GSR

4.2.1.4 Parts produced

The number of parts produced shows constant results in Figure 4.8 for the HMS under the MSSPT rule, and it outperforms the CMS. When the DDSI rule is applied, again the HMS shows a more consistent performance than the CMS. In terms of the number of parts produced, the HMS outperforms the CMS under the MSSPT rule. When the DDSI rule is implemented, the CMS outperforms the HMS when $\beta = 0.3$ and 0.8. When $\beta =$ 0.5 the number of parts produced in the CMS is highly decreased and outperformed by the HMS.

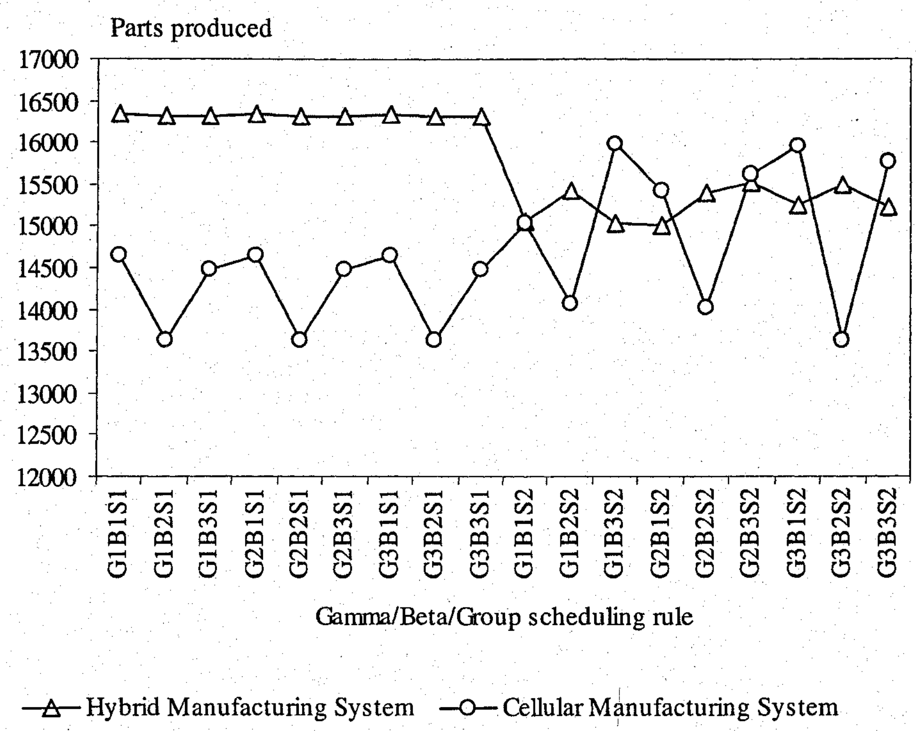


Figure 4.8 Parts produced interaction of γ level, β level and GSR

From figures 4.5 to 4.8 it can be concluded that the HMS outperforms the CMS when the MSSPT rule is implemented. This is due to the nature of the HMS and this scheduling rule. MSSPT focuses on minimization of number of setup and the total setup time. HMS is configured with the objective of minimizing setup times and reducing family setup times. Therefore, the MSSPT rule finds the HMS as the best scenario to be implemented, and therefore, performs as its best.

On the other hand, the CMS outperforms the HMS when DDSI rule is implemented and overall shows a much more consistent performance than the HMS when the scheduling rule is changed. This can be attributed to the consistency of part types that the CMS produces. CMS produces family parts and thus provides a more even scenario for this manufacturing system which leads to a consistency in the results obtained by the MSSPT and DDSI rules. By the same token, since the HMS produces family parts as well as non-family parts the system is more sensitive to changes and therefore a significant variation of results is depicted when switching between scheduling rules.

4.2.2 Analysis of effects

Analysis of variance (ANOVA) was the vehicle for examining the effects of all factors on the experimental results. The results of ANOVA were obtained from the trial version of the Stat-Ease® experimental design software. These results are presented in tables 4.6 and 4.7. The letters A, B, C and D correspond to the γ, β, group scheduling rule (GSR) and the type of manufacturing system (TMS) factors, respectively. Model p-values for all performance measures (flowtime, tardiness, earliness and parts produced) in Table 4.6 and 4.7 indicate that the model is significant, there is only 0.01% chance that a Model F-value this large could occur due to noise. Specifically for flowtime, analysis in Table 4.6 indicates that C, D, BC, BD and CD are significant model terms, for tardiness C, D, BC, BD and CD are significant model terms. Notice that p-values of B for flowtime and tardiness of 0.0561 and 0.0633, respectively, make this factor very close to being significant. For earliness, A, B, C, D, BC, BD, CD and BCD are significant model terms and for number of parts produced, B, D, BD, CD and BCD are significant model terms.

The main effect of each of the factors in the model is discussed in detailed below, followed by discussion of selected higher order (two or more factors) interaction effects. Graphical analysis is presented to argument the discussion of higher order interaction effects.

Source	df	Flowtime $R^2 = 0.9418$ $\mu = 2566.89$ F	p-value	Significance?	Tardiness $R^2 = 0.9419$ $\mu = 2403.06$ F	p-value	Significance?
Model	35	66.52797	<0.0001	Y	66.6589	<0.0001	Y
A	2	0.228236	0.7962	N	0.21733	0.8049	N
B	2	2.938683	0.0561	N	2.812748	0.0633	N
C	1	700.4194	<0.0001	Y	684.6466	<0.0001	Y
D	1	690.5737	<0.0001	Y	703.1026	<0.0001	Y
AB	4	0.643241	0.6325	N	0.644603	0.6316	N
AC	2	0.693623	0.5014	N	0.696147	0.5002	N
AD	2	0.167433	0.8460	N	0.166443	0.8468	N
BC	2	5.581173	0.0046	Y	5.655372	0.0043	Y
BD	2	6.025435	0.0031	Y	5.98147	0.0032	Y
CD	1	918.5274	<0.0001	Y	926.5147	<0.0001	Y
ABC	4	0.147144	0.9640	N	0.144979	0.9650	N
ABD	4	1.473288	0.2133	N	1.473935	0.2132	N
ACD	2	0.390048	0.6777	N	0.3834	0.6822	N
BCD	2	0.420078	0.6578	N	0.405064	0.6677	N
ABCD	4	0.089974	0.9855	N	0.092588	0.9847	N

Table 4.6 ANOVA results for mean flowtime and tardiness

Source	df	Earliness $R^2 = 0.9359$ $\mu = 36.95$ F	p-value	Significance?	Parts Produced $R^2 = 0.6432$ $\mu = 15225.13$ F	p-value	Significance?
Model	35	60.08632	<0.0001	Y	7.415447	<0.0001	Y
A	2	4.814135	0.0095	Y	0.178817	0.8364	N
B	2	27.48814	<0.0001	Y	12.53519	<0.0001	Y
C	1	1913.378	<0.0001	Y	1.666713	0.1988	N
D	1	20.94564	<0.0001	Y	106.3635	<0.0001	Y
AB	4	0.34054	0.8485	N	0.343973	0.8479	N
AC	2	0.664076	0.5163	N	0.204727	0.8151	N
AD	2	0.080569	0.9226	N	0.084232	0.9193	N
BC	2	4.755766	0.0100	Y	1.110062	0.3323	N
BD	2	9.919943	<0.0001	Y	19.29408	<0.0001	Y
CD	1	62.12556	<0.0001	Y	71.71519	<0.0001	Y
ABC	4	0.424917	0.7905	N	0.345658	0.8467	N
ABD	4	0.994099	0.4129	N	0.540021	0.7066	N
ACD	2	1.228143	0.2959	N	0.09723	0.9074	N
BCD	2	3.239034	0.0421	Y	3.087882	0.0486	Y
ABCD	4	0.290988	0.8835	N	0.559498	0.6924	N

Table 4.7 ANOVA results for earliness and number of parts produced

4.2.2.1 Effect of γ

The effect of γ in tables 4.6 and 4.7 reflects its significance in the assignment of due dates (Equation 3.8). γ is not significant for the mean flowtime, tardiness and parts produced performance measures, but it shows significance on the earliness performance measure. The low effect of γ will be verified later when higher order interaction effects are discussed.

4.2.2.2 Effect of β

Tables 4.6 and 4.7 indicate that the effect of β, the variable used in calculating family setup times (Equation 3.2), on the mean flowtime and tardiness performance measure is not significant; however, the value is very close to being significant. With regard to earliness and parts produced performance measures, β is found to be significant.

4.2.2.3 Effect of group scheduling rule

The scheduling rule design factor is significant on the mean flowtime, tardiness and earliness performance measure. The group scheduling rule shows no significance with regard to the number of parts produced; however, its value is relatively close to being significant, as shown in Table 4.7. The significance shown by the group scheduling rule is of great meaning since the experimental design of this thesis aims for it to be a significant factor in the performance of the HMS and CMS.

4.2.2.4 Effect of manufacturing system

The significance of the manufacturing systems is positive on all four performance measures. The overall significance of the manufacturing system term is also of great importance in this study, since it plays an important role in the performance of the HMS and CMS being tested in this experiment (refer to tables 4.6 and 4.7 for reference of the significance of the manufacturing systems).

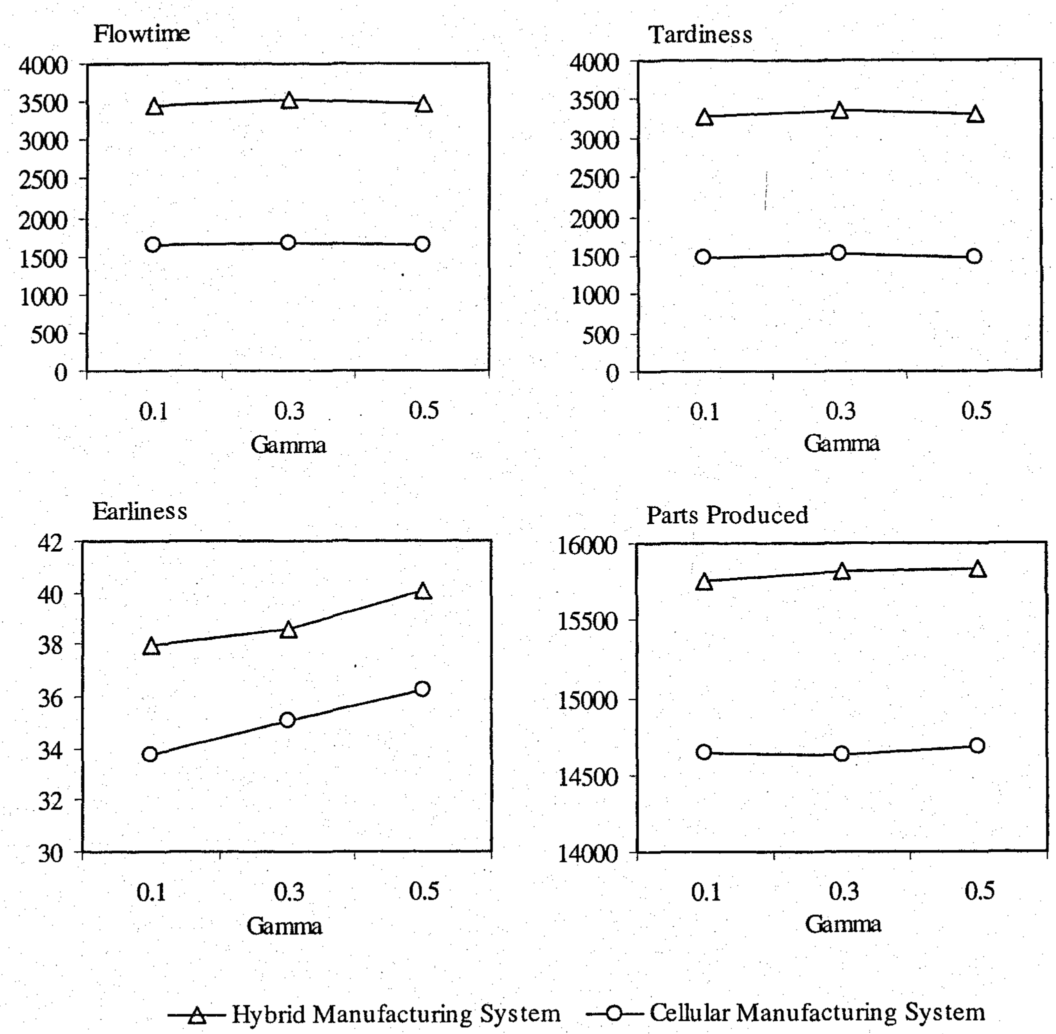


Figure 4.9 γ and TMS interaction

4.2.2.5 γ and manufacturing system interaction

Figure 4.9 presents the interaction between γ (used in the assignment of order due dates) and the type of manufacturing system (TMS) under each of the performance measures. The γ factor shows no effect on mean flowtime for the HMS which is concurrent with its significance in ANOVA. When analyzing tardiness, again, γ has no effect on the CMS and HMS. Clearly the effect of this factor is more pronounced on the earliness performance measure, showing an interesting effect in the same proportion for both manufacturing systems.

Keep in mind that the objective of earliness is to obtain the lowest possible value since just-in-time characteristics are taken into consideration. CMS and HMS are robust to the γ level in the parts produced performance, although there exists a slight decrease in this performance measure of the HMS when $\gamma = 0.1$, and a slight improvement in the CMS when $\gamma = 0.5$. The robustness in these results is consistent with the low single effect of γ previously discussed. Overall there exists no interaction between the manufacturing systems and γ as the lines in all four figures are almost parallel.

4.2.2.6 β and manufacturing system interaction

The interaction between β and manufacturing systems under different performance measures is show in Figure 4.10. With regard to the mean flowtime, the HMS shows a relatively robust behaviour for different levels of β. In particular, when $\beta = 0.3$, the performance in the HMS is reduced. As β increases its value, the HMS becomes stronger. The effect of β on the CMS is negligible as can be appreciated in this figure. When tardiness performance measure is analyzed, it can be observed that the effect of β is very similar as that shown by γ in the previous subsection. Again, when β = 0.3, the HMS reduces its value, whereas the CMS keeps its robustness throughout the levels of β. There is a negligible interaction between the manufacturing systems and β for the mean flowtime and tardiness performance measures.

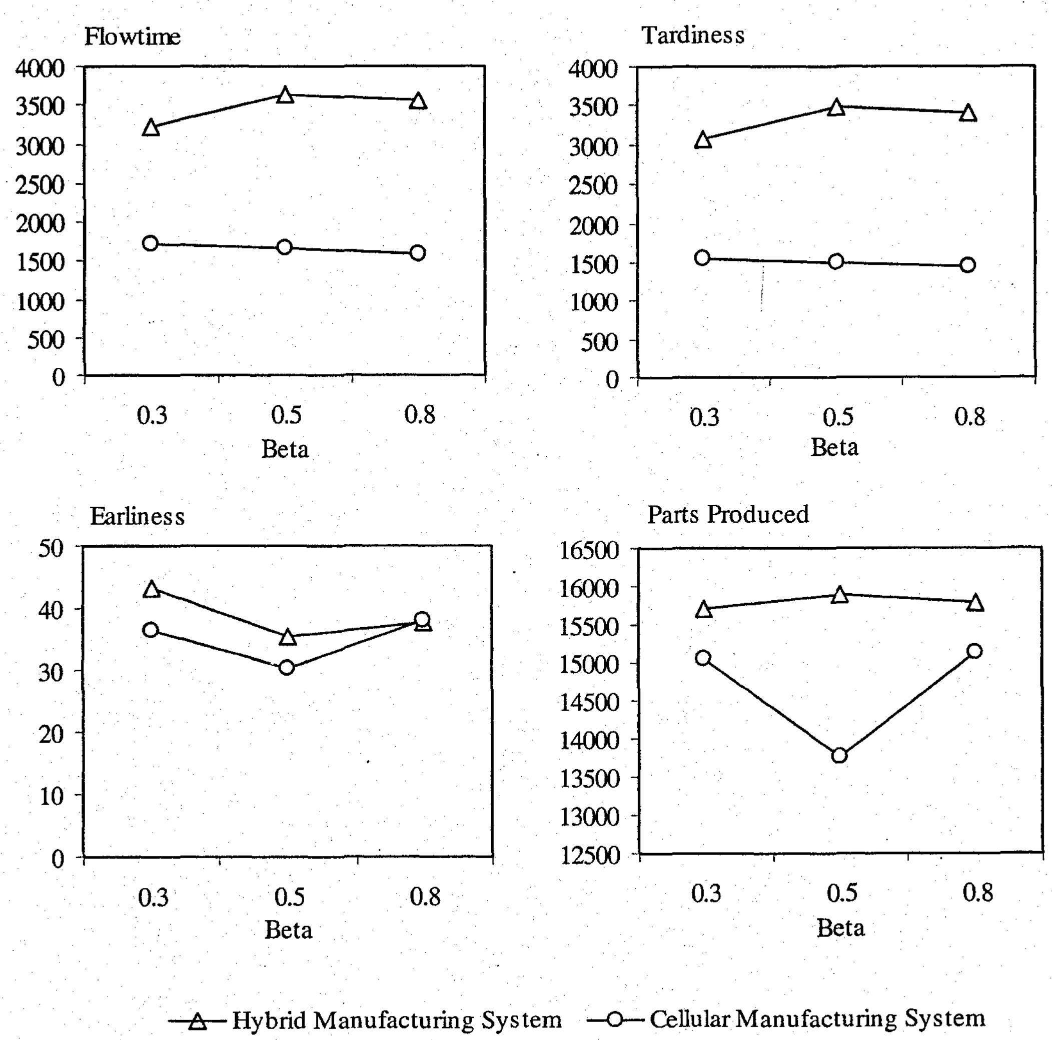


Figure 4.10 β and TMS interaction

The interaction between the two factors increases when earliness performance measure is analyzed. The manufacturing systems exhibit a strong reaction as the level of β increases. With regard to the number of parts produced, HMS outperforms the CMS under all levels of β. The effect of this factor is especially significant for the CMS as it varies from β = 0.3 to β = 0.8. The CMS shows its lowest performance when β = 0.5.

4.2.2.7 Group scheduling rule and manufacturing system interaction

Figure 4.11 presents the interaction between group scheduling rules and manufacturing systems. The significance of these two factors is critical because it will support the structure and methodology used in this simulation study. Very interesting results were obtained from the interaction of these two factors. For mean flowtime the effect of the group scheduling rule is decisive in determining which manufacturing system performs the best. Thus, the HMS outperforms the CMS when MSSPT rule is applied, whereas when DDSI rule is applied, the CMS outperforms the HMS. With regard to tardiness, an almost identical behaviour is found as that in the mean flowtime. The HMS outperforms the CMS under the MSSPT rule; however, it falls significantly behind the CMS under the DDSI scheduling rule.

For earliness, again, there exists a significant effect of the GSR on each of the manufacturing systems. In particular the CMS outperforms the HMS under MSSPT rule while the HMS outperforms the CMS under the DDSI rule. With regard to the number of parts produced, the effect of the GSR is also very significant in both manufacturing systems. In this context, the HMS outperforms the CMS when either the MSSPT or DDSI group scheduling rule is applied. Figure 4.11 showed a very strong interaction between the two factors, the manufacturing systems and GSR, and it provided essential directions for determining the behaviour of the manufacturing systems in this experiment.

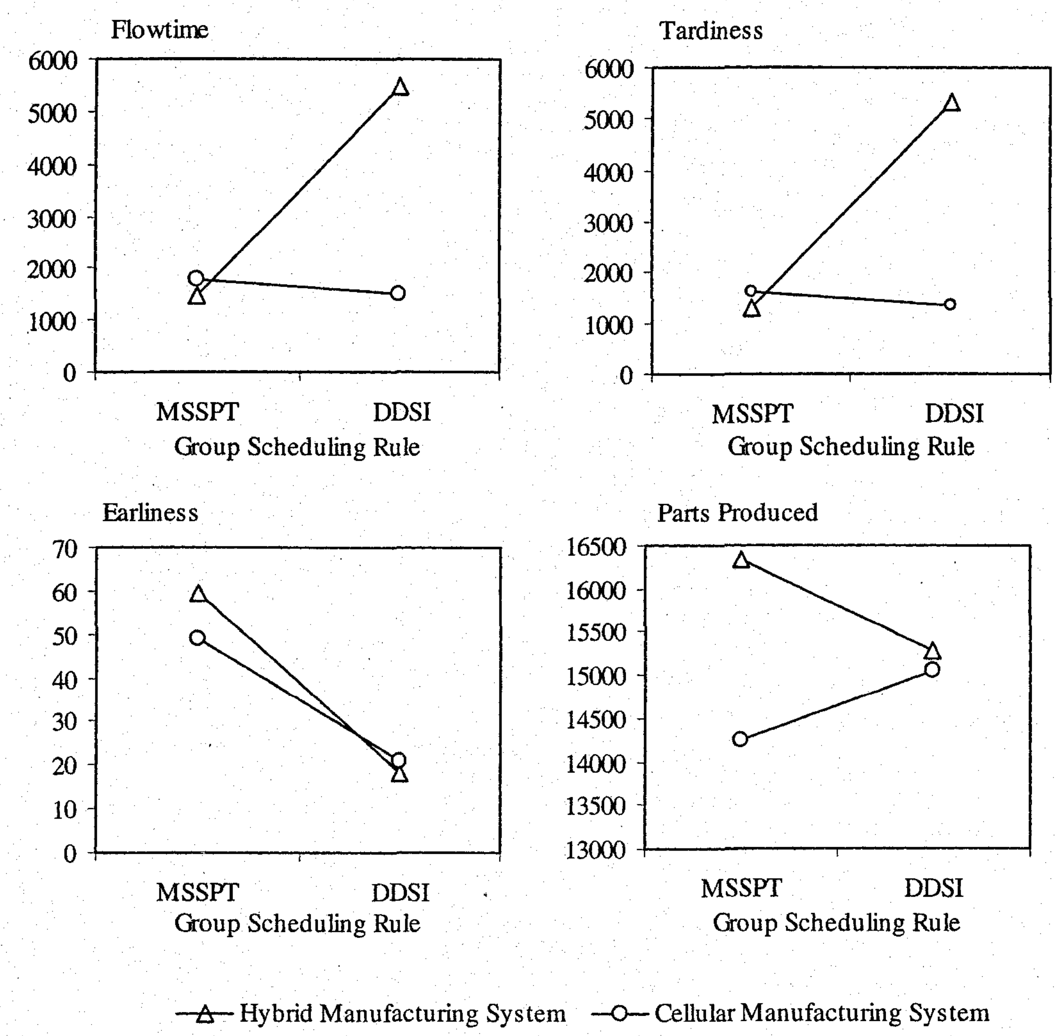


Figure 4.11 GSR and TMS interaction

4.2.2.8 β, γ and manufacturing system interaction under DDSI rule for mean flowtime and tardiness

The combined effects of β, γ and the manufacturing systems on the mean flowtime and tardiness performance measures when the DDSI rule is implemented are shown in Figure

4.12. As shown in the graph, the CMS outperforms the HMS when mean flowtime is considered. The highest mean flowtime and tardiness has occurred when β =0.5.

Also, the HMS shows improvement as the value of β factor is reduced from 0.8 to 0.3. On the other hand, the CMS shows its minimum performance when β = 0.5.

4.2.2.9 β, γ and manufacturing system interaction under DDSI rule for earliness and parts produced

The interaction among β, γ and manufacturing systems under DDSI rule is depicted in Figure 4.13. The HMS outperforms the CMS in terms of earliness when β = 0.8, and the opposite takes place when β = 0.3. When β = 0.5, on the average, both systems exhibit similar performance .

With regard to the parts produced, the CMS outperforms the HMS with β = 0.3 and 0.8, while HMS outperforms CMS with β = 0.5. It is worth noting that in this figure, the value of β = 0.5 considerably changes the earliness performance measure and number of parts produced, as it occurs when analyzing the mean flowtime and tardiness of the previous section.

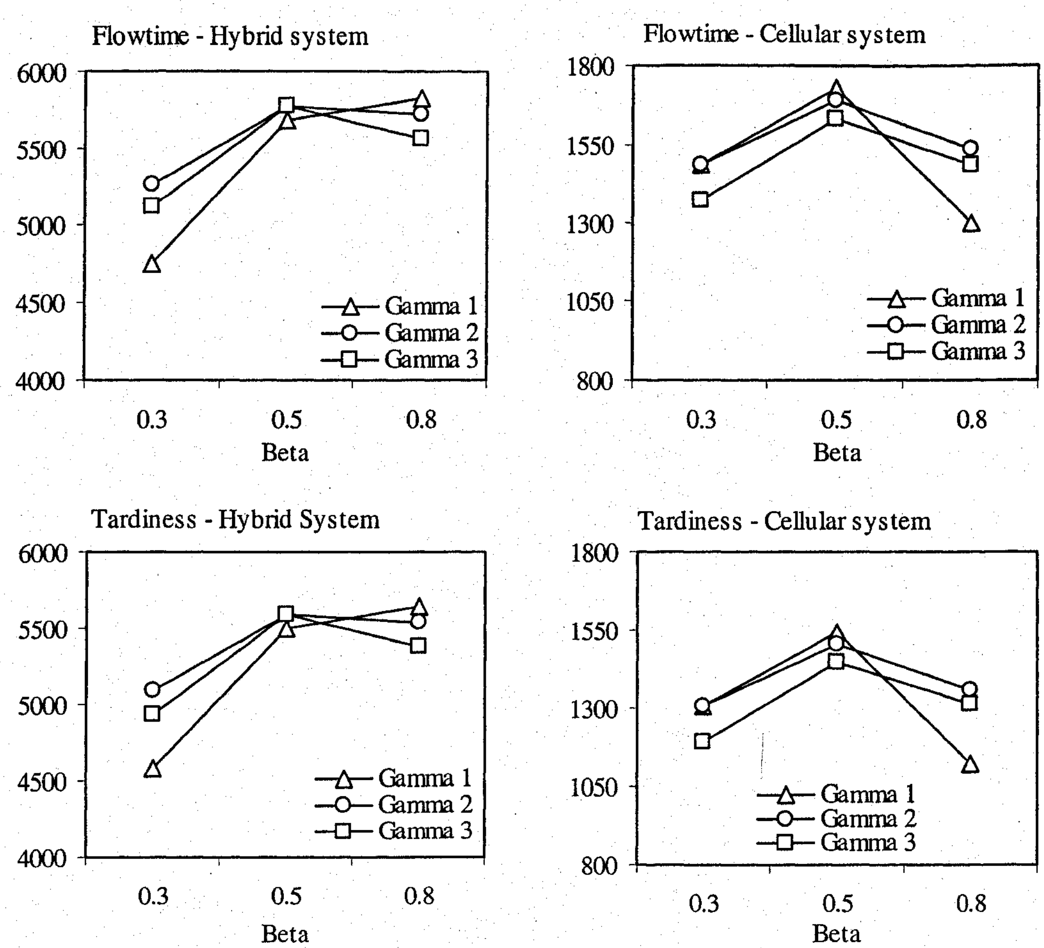


Figure 4.12 β, γ and TMS interaction under DDSI rule for mean flowtime and tardiness performance measures

4.2.3 β, γ and manufacturing system interaction under MSSPT rule

Figure 4.14 shows the results for the higher order interaction among β, γ and manufacturing system when MSSPT rule is implemented. The average performance values was used since as can be seen in Table 4.5, the variation of γ within the same β level provided almost identical performance measure values for each of the manufacturing systems.

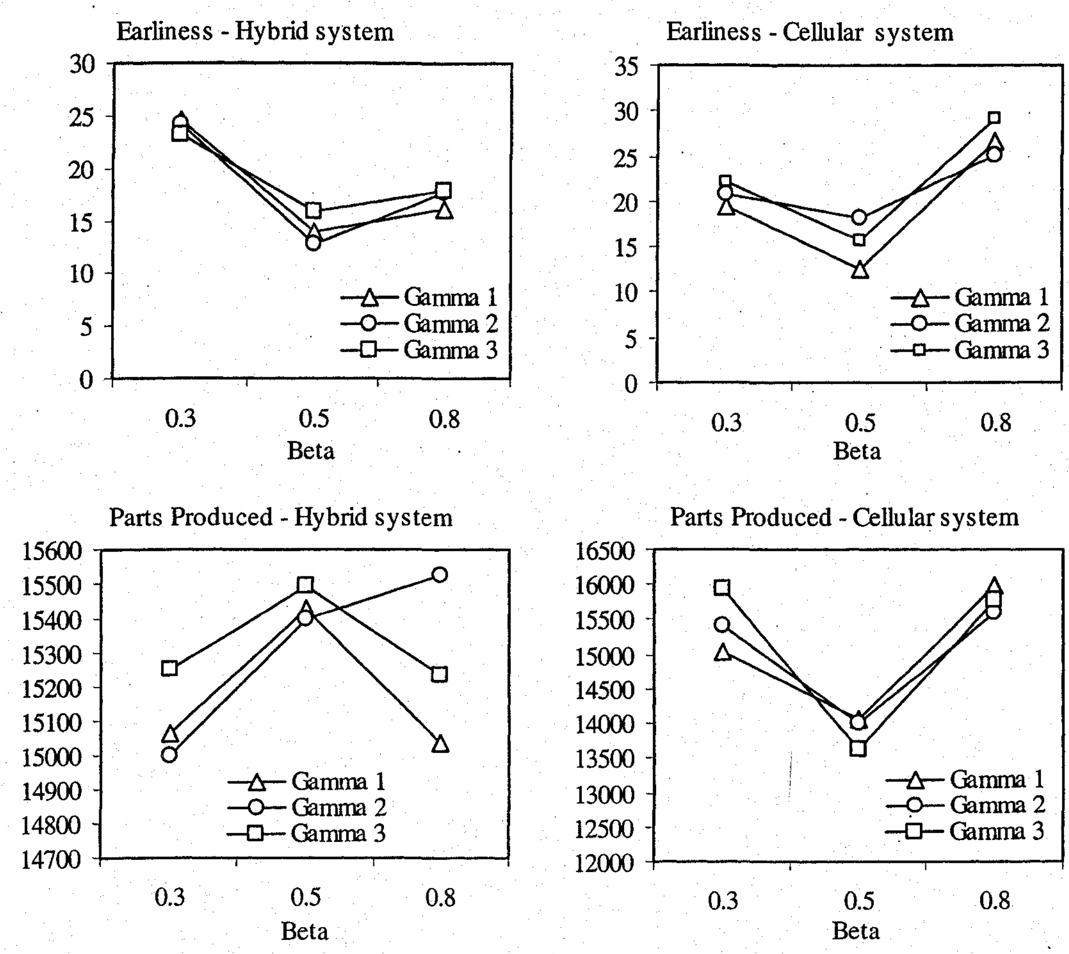


Figure 4.13 β, γ and TMS interaction under DDSI rule for earliness performance measure and number of parts produced

β has a considerable effect on the manufacturing systems with regard to mean flowtime. For tardiness, β produces the same effect in terms of variation as that of the mean flowtime. CMS outperforms HMS when mean flowtime, tardiness and earliness performances are considered. This is consistent with the group scheduling rule and manufacturing system interactions formerly discussed. For the number of parts produced the HMS exhibits a very robust performance for all levels of β, whereas the CMS changes its performance when $\beta = 0.5$. There is a strong interaction among the factors for the mean flowtime and tardiness performance measures, as well as for the parts produces. For earliness, there is no significant interaction among factors.

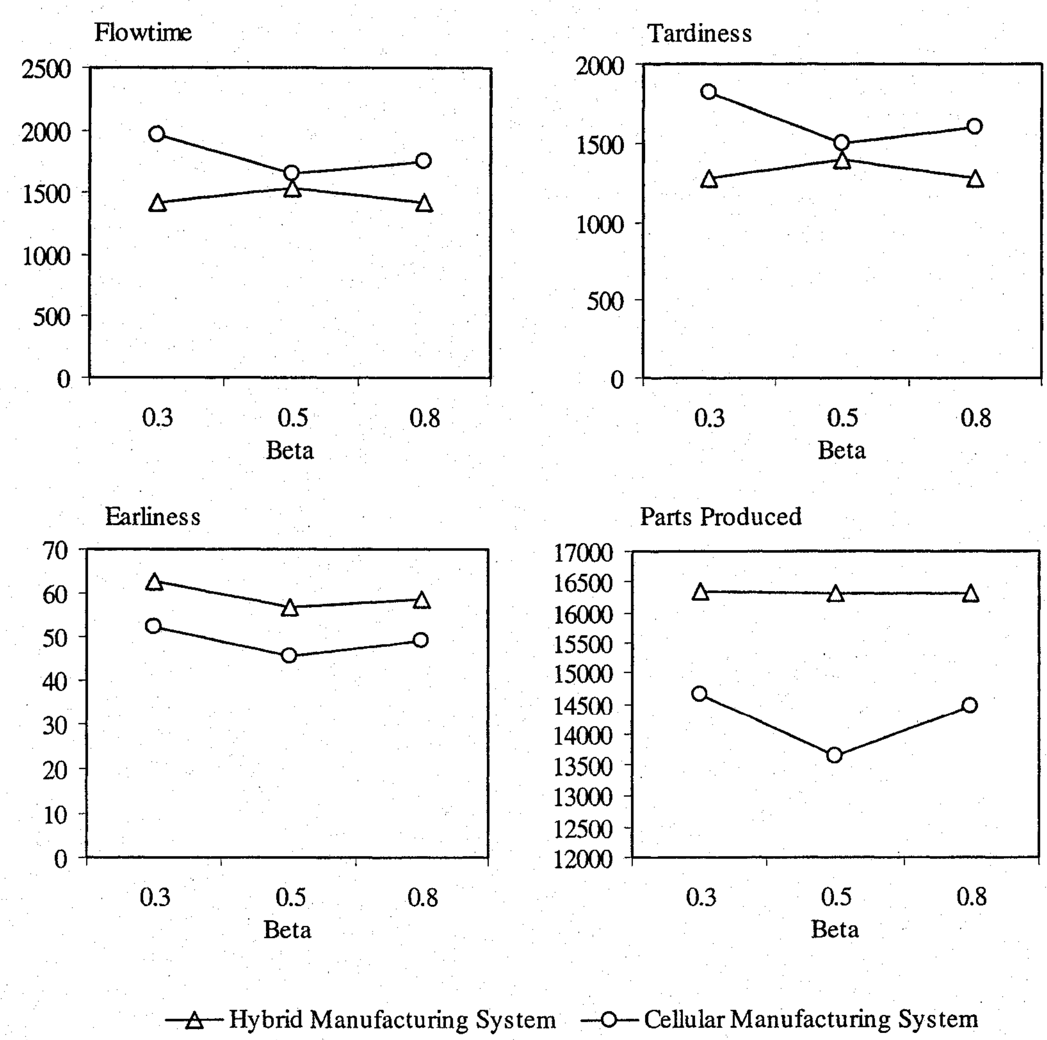


Figure 4.14 β, γ Average and TMS interaction under MSSPT rule

4.3 Experimental Conclusions

The β experimental factor has a significant effect on the model, as it is shown in ANOVA. Its high level of significance in the model is reflected also when discussing main and higher order interaction of this factor with others. This translates to an appropriate value of β to be used in obtaining family setup times.

The effects shown by γ, indicate that this factor is not significant. γ is used in determining the bounds of the uniform distribution that assigns order due dates, therefore, a possible manner to increase the significance of γ might be trying different probability distributions in the assignment of order due dates.

The results of the experiment suggest that for the mean flowtime and tardiness performance measures, the HMS outperforms the CMS when the MSSPT rule is implemented, while the CMS exhibits superior performance when the DDSI rule is applied. With regard to the earliness performance measure, the CMS outperforms the HMS in most scenarios. Finally, the HMS provides a higher number of parts produced than the CMS in most cases.

Thus, if the objective of an industry is either to minimize the number, length and cost of setups or to maximize the throughput rate of the system, then the HMS can be recommended. However, if the objective is to meet order due dates, then the CMS should be implemented.

4.4 Chapter Summary

In this chapter, the output analysis was conducted in order to define the initialization time and run length for the steady-state simulation of this experiment. It consisted of two phases: a truncation analysis and the replication method. Also, the experimental design was developed. The results of the experiment were discussed using ANOVA. The main and higher order interaction effects for all the factors on each of the scenarios in the experiment was discussed. Tables and graphs were provided for a better interpretation and analysis of results.

CHAPTER 5: CONCLUSIONS AND FUTURE RESEARCH

5.1 Concluding Remarks

This thesis numerically compares two manufacturing systems via simulation. These systems are the cellular manufacturing system (CMS) and the proposed hybrid manufacturing system (HMS). Both systems are tested for identical cellular manufacturing layout environment. The main distinction among the hybrid and the cellular manufacturing system is given by the type of parts they can produce, e.g., family or non-family parts. The hybrid system focuses on family and non-family parts while the cellular system manufactures family parts only.

Two group of scheduling rules are considered in this study. The inclusion of group scheduling rules as a design factor provided the sequence for part processing on the machines of each manufacturing system. These scheduling rules, also helped construct the manufacturing-group scheduling scenarios for running the simulation. A truncation analysis and replication method is used to determine the values of the initialization time and run length to reach steady-state.

In order to perform the sensitivity analysis, the values of γ and β are varied in the model, and are included as factors in the statistical analysis.

Several issues arise from the results of this experiment. First, the group scheduling rule and the manufacturing system design factors have a significant impact on the model. In particular, the hybrid system shows its best performance when the MSSPT (minimum setup shortest processing time) group scheduling rule is applied, whereas the cellular system is superior to the hybrid system when DDSI (due date truncated shortest processing time) is implemented.

This study provides evidence of the value of adding flexibility in manufacturing systems. The results demonstrate that, by adding non-family parts to the production schedule of the hybrid manufacturing system, significant benefits in the performance measures can be derived.

The results of this study can be used as a guide in the choice of the appropriate manufacturing system and group scheduling rule heuristics to be selected for a particular objective of an organization. If the objective of an industry is to increase the number of parts produced or minimize the number, length and cost of setups the hybrid manufacturing system is the best alternative but if the objective is to meet order due dates, the cellular system should be implemented.

5.2 Suggestions for Future Research

The hybrid manufacturing system proposed in this thesis showed positive performance in the experimental design. This provides encouragement for extending the study of the hybrid model under different manufacturing conditions. The following are some of the proposed settings and changes to be implemented in the hybrid model to expand its research:

1. Vary the system parameters to include random order interarrival times and processing times.

2. Implement sequence-dependent setup times for the parts in the system. This would allow performing a comparison with the results obtained with sequence-independent setup times in this thesis, to determine the best sequence setup environment for the hybrid system.

3. Investigate other performance measures relevant to manufacturing organizations such as work-in-process, quality, scheduling complexity, tooling investment and the productivity in the manufacturing system.

4. Include system constraints that depict a more realistic picture of the hybrid system such as system breakdowns, limited material handling capability and buffer capacity.

5. Investigate other operating factors that may influence the appropriateness of the hybrid model such as the variability of the product mix, the effect of bottleneck work centers, the density of the machine-part matrix and alternative machine utilization levels.

6. Introduce reassignment process plans that cope with volume variations in the system and that help to distribute the extra load of an overloaded machine.

7. Introduce new dispatching system such as cooperative dispatching and lot-splitting.

8. Implement simulation-based real-time scheduling mechanisms. This mechanism dynamically selects from a set of possible dispatching rules based on information from discrete event simulation information.

9. Compare real-time scheduling algorithms with off-line scheduling to determine under which conditions the hybrid model shows its best performance.

10. Compare the hybrid model with other manufacturing systems such as job and virtual cellular manufacturing.

REFERENCES

Aldenderfer, M. S., & Blashfield, R. K., (1984). *Cluster Analysis.* Sage Publications, Inc., CA.

Babu, A. S., Nandurkar K. N., & Thomas A., (2000). Development of Virtual Cellular Manufacturing Systems for SMEs. *Logistics Information Management*, **13**, 228-242.

Ballakur, A., & Steudel, H. J., (1987). *International Journal of Production Resources*, **25**, 639-665.

Banks, J., & Nicol, D. M., (2001). *Discrete-Event System Simulation.* 3rd ed., Prentice Hall.

Barash, M. M., Leimkuhler, F. F., Solberg, J. J., & Talavage, J. J., (1981). Optimal Planning of Computerized Manufacturing Systems (CMS). *Proc. 8th NSF Grantees' Conf. Production Research and Technology.*

Black, J. T., (1983). Cellular Manufacturing Systems Reduce Set-Up Time, Make Small Lot Production Economical. *Industrial Engineering*, **15**, 36-48.

Black, J. T., (1988). The Design of Manufacturing Cells (Integrated Manufacturing Systems Step 1). *Proceedings of Manufacturing International*, **3**, Atlanta, GA, 143-158.

Browne, J., Dubois, D., Rathmill, K., Sethi, S., & Stecke, K., (1984). Classification of Flexible Manufacturing Systems. *FMS Magazine*, **2**, 114-117.

Burbidge, J. L., (1975). *The Introduction of Group Technology.* Wiley, New York.

Buzacott, J., & Shanthikumar, J., (1980). Models for Understanding Flexible Manufacturing Systems. *AIIE Transactions*, **12**, 84-94.

Chan, F. T. S., Chan H. K., & Lau H.C.W., (2002). The State of the Art in Simulation Study on FMS Scheduling: A Comprehensive Survey. *The International Journal of Advanced Manufacturing Technology*, **19**, 830-849.

Chandrasekharan, M. P., & Rajagopalan, R., (1986). An Ideal Seed Non-Hierarchical Clustering Algorithm for Cellular Manufacturing. *International Journal of Production Research,* **24**, 451-464.

Cobham, A., (1954). Priority Assignment in Waiting Line Problems. *Operations Research*, **1,** 70-76.

De Toni, A., & Tonchia, S., (1998). Manufacturing Flexibility: A Literature Review. *International Journal of Production Research*, **36**, 1587-1617.

Flynn, B. B., & Jacobs, F. R., (1987). An Experimental Comparison of Cellular (GT) Layout with Process Layout. *Decision Sciences*, **18**, 562-581.

Flynn, B. B., (1987). Repetitive Lots: The use of a Sequence Dependent Set-Up Time Scheduling Procedure in Group Technology and Traditional Shops. *Journal of Operations Management*, **7**, 203-216.

Gaither, N., (1996). *Production and Operations Management.* 7th ed., Wadsworth Publishing Company.

Garg, S., Vrat, P., & Kanda A., (2001). Equipment flexibility vs. Inventory: A Simulation Study of Manufacturing Systems. *International Journal of Production Economics*, **70**, 125-143.

Greene, T. J., & Sadowski, R. P., (1984). A Review of Cellular Manufacturing Assumptions, Advantages and Design Techniques. *Journal of Operations Management*, **4**, 85-97.

Groover, M. P., & Zimmers, E. W., (1984). *CAD/CAM Computer-Aided Design and Manufacturing*. Prentice-Hall.

Gupta, T., (1988). A Production Data Based Machine-Component Grouping Model in Cellular Manufacturing. PhD Dissertation (unpublished) (Milwaukee, WI: University of Wisconsin-Milwaukee).

Ham, I., Hitomi, K., Nakamura, N., & Yoshida, T., (1979). Optimal Group Scheduling and Machining-Speed Decision under Due-Date Constraints. *Transactions of the ASME: Journal of Engineering for Industry*, **101**, 123-134.

Huber, V., & Hyer, N., (1985). The Human Impact of Cellular Manufacturing. *Journal of Operations Management,* **5**, 213-228.

Jeong, K.-C., & Kim, Y.-D. (1998). A Real-Time Scheduling Mechanism for a Flexible Manufacturing System: Using Simulation and Dispatching Rules. *International Journal of Production Research*, **36**, 2609-2626.

Kannan, V. R., & Ghosh, S. A., (1996). A Virtual Cellular Manufacturing Approach to Batch Production. *Decision Sciences*, **27**, 519-539.

Kattan, I. A., (1997). Design and scheduling of hybrid multi-cell flexible manufacturing systems. *International Journal of Production Research*, **35,** 1239-1257.

Kleijnen, J. P. C., (1995). Theory and Methodology: Verification and Validation of Simulation Models. *European Journal of Operational Research*, **82**, 145-162.

Leonard, R., & Rathmill, K., (1977). The Group Technology Myths. *Management Today*, 66-69.

MacCarthy, B. L., & Liu, J., (1993). A New Classification Scheme for Flexible Manufacturing Systems. *International Journal of Production Research*, **31**, 299-309.

Mahmoodi, F., Dooley, K. J., & Starr, P. J., (1988). A comparison of dynamic scheduling heuristics for cellular manufacturing. *Proceedings of National Decision Sciences Institute*, 893-895.

Mahmoodi, F., (1989). An Investigation of Dynamic Group Scheduling and Releasing Heuristics in Cellular Manufacturing. Unpublished Ph.D. Dissertation, University of Minnesota.

Mahmoodi, F. Dooley K.J., & Starr P.J., (1990). An Investigation of Dynamic Group Scheduling Heuristics in a Job Shop Manufacturing Cell. *International Journal of Production Research*, **28**, 1695-1711.

Mahmoodi, F., & Dooley, K. J., (1991). A Comparison of Exhaustive and Non-Exhaustive Group Scheduling Heuristics in a Manufacturing Cell. *International Journal of Production Research*, **29**, 1923-1939.

Mahmoodi, F., Tierney, E. J., & Mosier C. T., (1992). Dynamic Group Scheduling Heuristics in a Flow-through Cell Environment. *Decision Sciences*, **23**, 61-85.

Mahmoodi, F., & Martin, G. E., (1997). A New Shop-Based and Predictive Scheduling Heuristic for Cellular Manufacturing. *International Journal of Production Research*, **35**, 313-326.

McAuley, J., (1972). Machine Grouping for Efficient Production. *Production Engineer*, **51**, 53-57.

Millar, H. H., & Yang, T., (1996). Batch Sizes and Lead-Time Performance in Flexible Manufacturing Systems. *International Journal of Flexible Manufacturing Systems*, **8**, 5-21.

Moily, J., Stinson, J., & Vakharia, A. J., (1992). Implementation of Logical Cells: A Simulation Study. Working Paper, University of Florida, Gainesville, FL.

Morris, J.S., & Tersine, R. J., (1990). A Simulation Analysis of Factors Influencing the Attractiveness of Group Technology Cellular Layouts. *Management Science*, **36,** 1567-1578.

Mosier, C. T., Elvers, D. A., & Kelly, D., (1984). Analysis of Group Technology Scheduling Heuristics. *International Journal of Production Research*, **22**, 857-875.

Mosier, C. T., & Taube, L., (1985). The facets of group technology and their impacts on implementation – a state-of-the-art survey. *OMEGA-International Journal of Management Science*, **13,** 381-391.

Mosier, C. T., & Mahmoodi, F., (2002). Work Sequencing in a Manufacturing Cell with Limited Labour Constraints. *International Journal of Production Research*, **40**, 2883-2899.

Oral, M., & Malouin, J.-L., (1973). Evaluation of the Shortest Processing Time Scheduling Rule with Truncation Process. *AIIE Transactions*, **5**, 357-365.

Potts, C.N., & Whitehead, J.D., (2001). Workload Balancing and Loop Layout in the Design of a Flexible Manufacturing System. *European Journal of Operational Research*, **129**, 326-336.

Pritsker, A. A. B., & O'Reilly, J. J., (1999). Simulation with Visual SLAM and AweSim. *John Wiley & Sons*, 2nd Edition.

Ranson, G. M., (1972). *Group Technology*. McGraw-Hill, London.

Rao, H. A., & Gu, P., (1995). *International Journal of Production Resources*, **33**, 1049-1070.

Rathmill, K., & Leonard, R., (1977). The Fundamental Limitations of Cellular Manufacture when Contrasted with Efficient Functional Layout. *Proceedings of the Fourth International Conference on Production Research*, 523-546.

Ruben, R. A., Dossier, C. T., & Mahmoodi, F., (1993). A Comprehensive Analysis of Group Scheduling Heuristics in a Job Shop Cell. *International Journal of Production Research*, **31**, 1343-1369.

Sabuncuoglu, I., (1998). A Study of Scheduling Rules of Flexible Manufacturing Systems: A Simulation Approach. *International Journal of Production Research*, **36**, 527-546.

Sadoun, B., (2000). Applied Systems Simulation: A Review Study. *Information Sciences*. **124**, 173-192.

Salvendy, G., (1982). *Handbook of Industrial Engineering*. John Wiley and Sons, New York.

Seifoddini, H., & Wolfe, P.M., (1986). Application of the Similarity Coefficient Method in Group Technology. *IIE Transactions*, **18**, 271-277.

Seifoddini, H., (1987a). Incorporation of the Production Volume in Machine Cells Formation in Group Technology Applications. *Proceedings of the IXth ICPR*, 2348-2356.

Seifoddini, H., & Wolfe, P.M., (1987b). Selection of a Threshold Value Based on Material Handling Cost in Machine-Component Grouping. *IIE Transactions*, **19**, 266-270.

Shafer, S. M., & Charnes, J. M., (1993). Cellular Versus Functional Layouts Under a Variety of Shop Operating Conditions. *Decision Sciences,* **24**, 665-681.

Sinha, R. K., & Hollier, R. H., (1984). A review of production control problems in cellular manufacturing. *International Journal of Production Research*, **22,** 773-789.

Snead, C. S., (1989). *Group Technology, Foundations for a Competitive Edge*. Nostrand Reinhold, New York

Stecke, K., (1983). Formulation and Solution of Non-Linear Integer Production Planning Problems for Flexible Manufacturing Systems. *Management Science*, **29**, 273-288.

Suresh, N. C., & Meredith, J.R., (1985). Achieving factory automation through group technology principles. *Journal of Operations Management*, **5**, 151-167.

Suresh, N.C., (1992). Partitioning Work Centers for Group Technology: Analytical Extension and Shop Level Simulation Investigation. *Decision Sciences*, **23**, 267-290.

Suresh,, N.C., & Meredith, J.R., (1994). Coping with the Loss of Pooling Synergy in Cellular Manufacturing Systems. *Management Science*, **40**, 466-483.

Vaithianathan, R., & McRoberts, K. L., (1982). On scheduling in a GT environment. *Journal of Manufacturing Systems*, **1**, 149-155.

Van Horn, R. L., (1971). Validation of Simulation Results. *Management Science*, **17**, 247-258.

Vettin, G., (1977). *Materialfluss-Organization in flexiblen Fertigungssystemen.* HGF 77/38, Girardet, Essen.

Wang, S., & Sarker, B. R., (2002). Locating Cells with Bottleneck Machines in Cellular Manufacturing Systems. *International Journal of Production Research*, **40**, 403-424.

Wemmerlöv, U., & Hyer, N. L., (1989). Cellular Manufacturing in the U.S. industry: A survey of users. *International Journal of Production Research*, **27,** 1511-1530.

Wemmerlöv, U., & Vakharia, A. J., (1991). Job and Family Scheduling of a Flow-Line Manufacturing Cell: A Simulation Study. *IIE Transactions*, **23**, 383-393.

Wirth, G. T., Mahmoodi, F. & Mosier, C. T., (1993). An Investigation of Scheduling Policies in a Dual-Constrained Manufacturing Cell. *Decision Sciences*, **24**, 761-788.

Zolfaghari, S., & Liang, M., (1998). Machine Cell/Part Family Formation Considering Processing Times and Machine Capacities: A Simulated Annealing Approach. *Computers and Industrial Engineering*. **34**, 813-823.

Zolfaghari, S., & Liang, M., (2002). A Comparative Study of Simulated Annealing, Genetic Algorithms and Tabu Search for Solving Binary and Comprehensive Machine Grouping Problems. *International Journal of Production Research,* **40,** 2141-2158.

Zolfaghari, S., Liang, M., (2003). A genetic algorithm approach to comprehensive machine cell/part family formation. *Journal of Integrated Manufacturing Systems*, (in press).

APPENDIX I

Test problem data

Operation sequence of parts

Part No.	Operation 1	Operation 2	Operation 3
P1	M1	M2	M3
P2	M1	M2	-
P3	M1	M2	M3
P4	M4	M5	-
P5	M4	-	-
P6	M4	M5	-
P7	M4	M5	-
P8	M6	M7	M8
P9	M6	M7	M8
P10	M6	M8	-
P11	M2	M4	M1
P12	M2	M5	M4
P13	M7	M3	-
P14	M8	M7	M4
P15	M2	M4	M6
P16	M1	M5	-
P17	M7	M8	M2
P18	M4	M3	-
P19	M5	M6	-
P20	M3	M6	M5

M = Machine Type

Processing times for the hybrid and cellular manufacturing systems (units=minutes)

Part No.	Processing time on operation 1	Processing time on operation 2	Processing time on operation 3
P1	5	3	4
P2	3	1	-
P3	5	3	1
P4	5	1	-
P5	2	-	-
P6	3	5	-
P7	4	2	-
P8	1	3	4
P9	2	3	1
P10	4	2	-
P11	2	2	5
P12	5	3	1
P13	1	2	-
P14	5	4	1
P15	1	4	3
P16	3	2	-
P17	3	2	5
P18	3	4	-
P19	3	5	-
P20	1	4	2

Total setup times used for determining the different setups of the parts in the model (units=minutes)

Part No.	Total setup time on operation 1	Total setup time on operation 2	Total setup time on operation 3
P1	3	1	2
P2	1	2	-
P3	2	2	1
P4	1	2	-
P5	1	-	-
P6	2	3	-
P7	2	1	-
P8	2	3	1
P9	1	2	3
P10	3	1	-
P11	1	2	3
P12	2	3	2
P13	1	3	-
P14	2	2	1
P15	2	1	1
P16	1	3	-
P17	3	3	1
P18	3	1	-
P19	3	2	-
P20	1	2	2

Total setup times for the hybrid manufacturing system (units=minutes)

	Total Setup Times			Family Setup Times			Part Setup Times			Part
Part No.	O1	O2	O3	O1	O2	O3	O1	O2	O3	Family
P1							2.1	0.7	1.4	
P2	-	-	-	0.9	0.6	0.6	0.7	1.4	-	1
P3							1.4	1.4	0.7	
P4							0.7	1.4	-	
P5	-	-	-	0.6	0.9	-	0.7	-	-	2
P6							1.4	2.1	-	
P7							1.4	0.7	-	
P8							1.4	2.1	0.7	
P9	-	-	-	0.9	0.9	0.9	0.7	1.4	2.1	3
P10							2.1	0.7	-	
P11	1	2	3							
P12	2	3	2							
P13	1	3	-							
P14	2	2	1							
P15	2	1	1	-	-	-	-	-	-	-
P16	1	3	-							
P17	3	3	1							
P18	3	1	-							
P19	3	2	-							
P20	1	2	2							

O = Operation number

Total setup times for the hybrid manufacturing system (units=minutes)

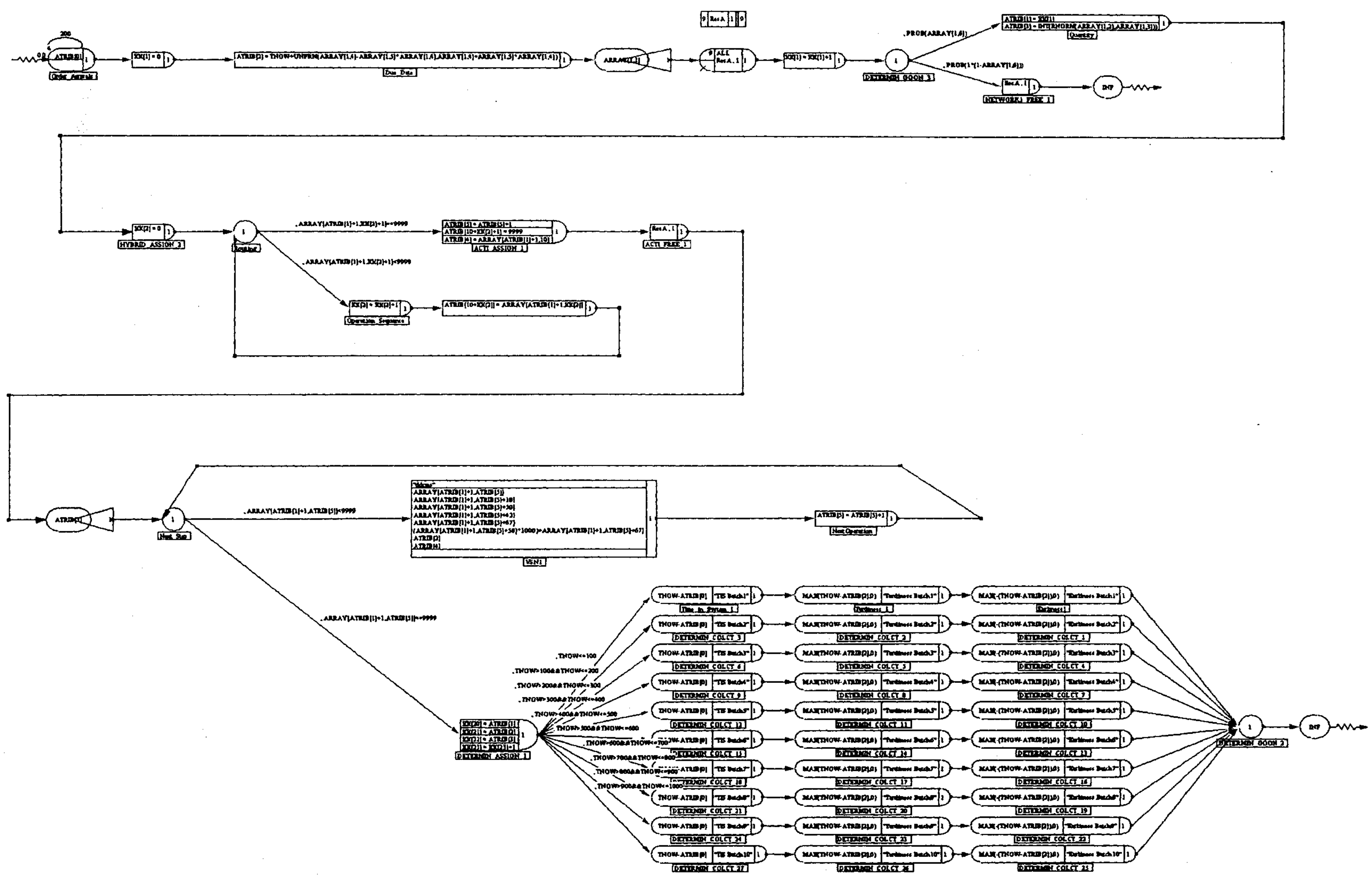
	Family Setup Times			Part Setup Times			Part
Part No.	O1	O2	O3	O1	O2	O3	Family
P1	0.9	0.6	0.6	2.1	0.7	1.4	1
P2	0.9	0.6	-	0.7	1.4	-	
P3	0.9	0.6	0.6	1.4	1.4	0.7	
P11	0.6	0.9	0.9	0.7	1.4	2.1	
P15	0.6	0.9	0.3	1.4	0.7	0.7	
P18	0.9	0.6	-	2.1	0.7	-	
P4	0.6	0.9	-	0.7	1.4	-	2
P5	0.6	-	-	0.7	-	-	
P6	0.6	0.9	-	1.4	2.1	-	
P7	0.6	0.9	-	1.4	0.7	-	
P12	0.6	0.9	0.6	1.4	2.1	1.4	
P16	0.3	0.9	-	0.7	2.1	-	
P19	0.9	0.6	-	2.1	1.4	-	
P20	0.3	0.6	0.9	0.7	1.4	1.4	
P8	0.9	0.9	0.9	1.4	2.1	0.7	3
P9	0.9	0.9	0.9	0.7	1.4	2.1	
P10	0.9	0.9	-	2.1	0.7	-	
P13	0.9	0.9	-	0.7	2.1	-	
P14	0.9	0.9	0.3	1.4	1.4	0.7	
P17	0.9	0.9	0.3	2.1	2.1	0.7	

O = Operation number

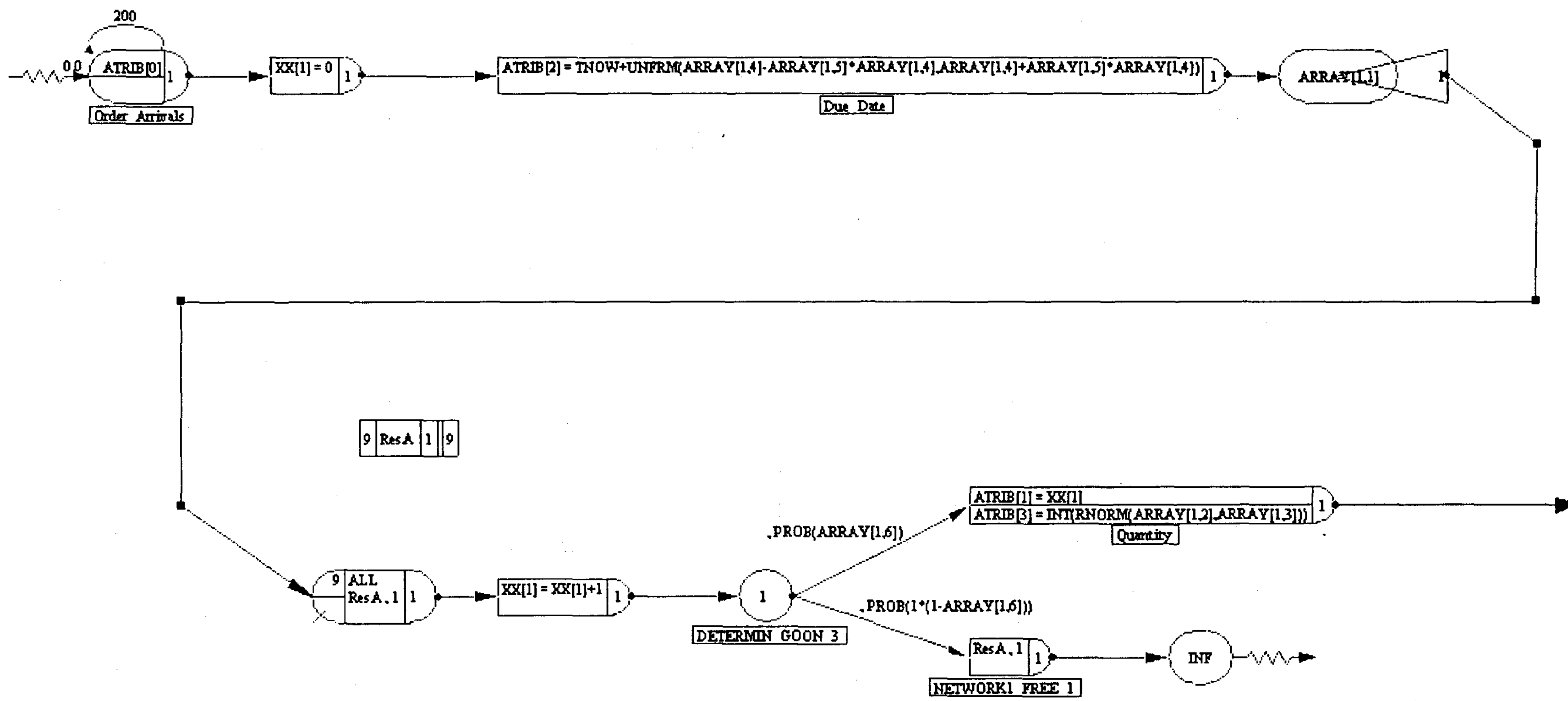
APPENDIX II

Visual SLAM network and subnetwork

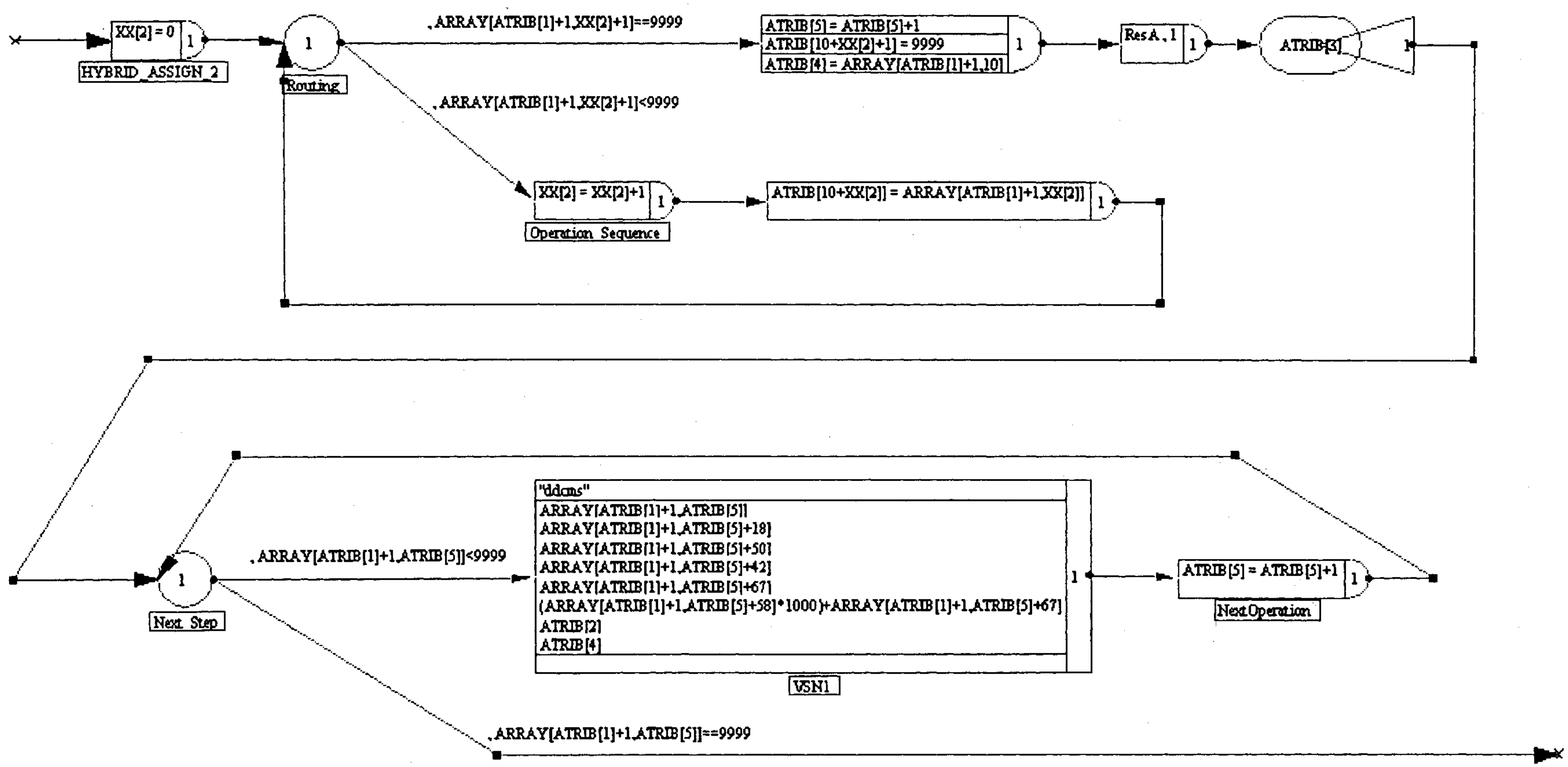
Model Network



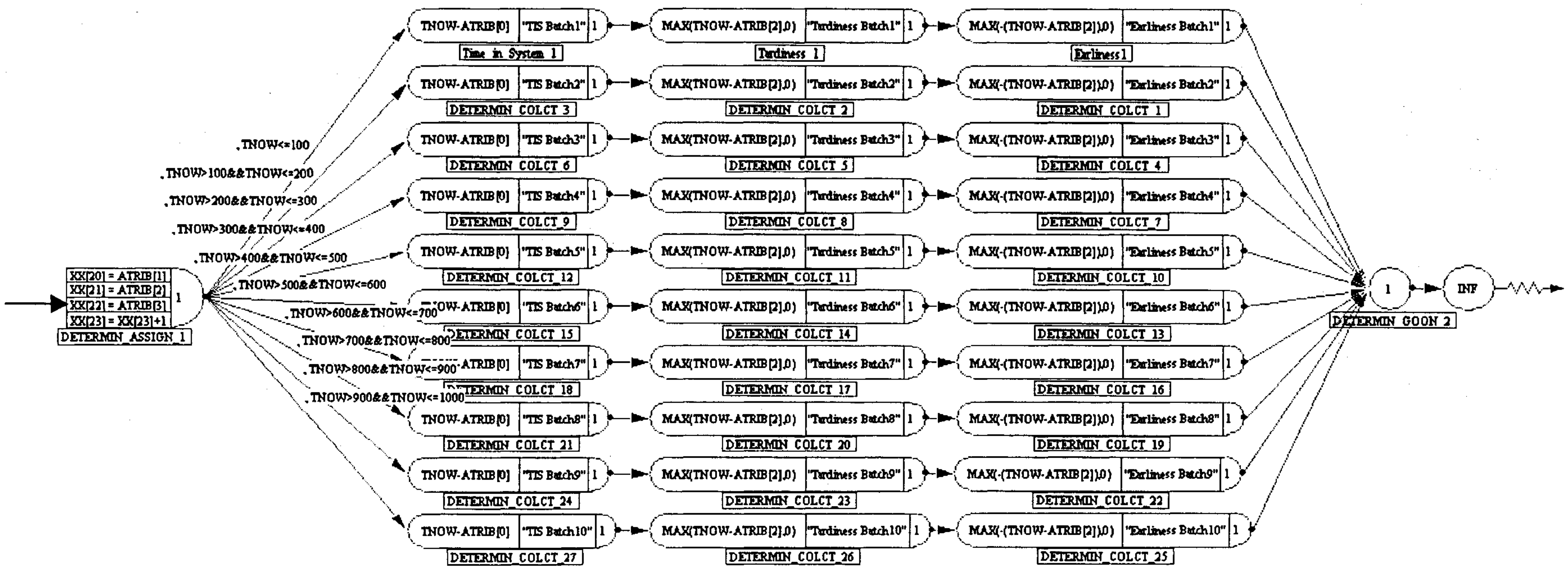
Model Network (enlarged 1/3)



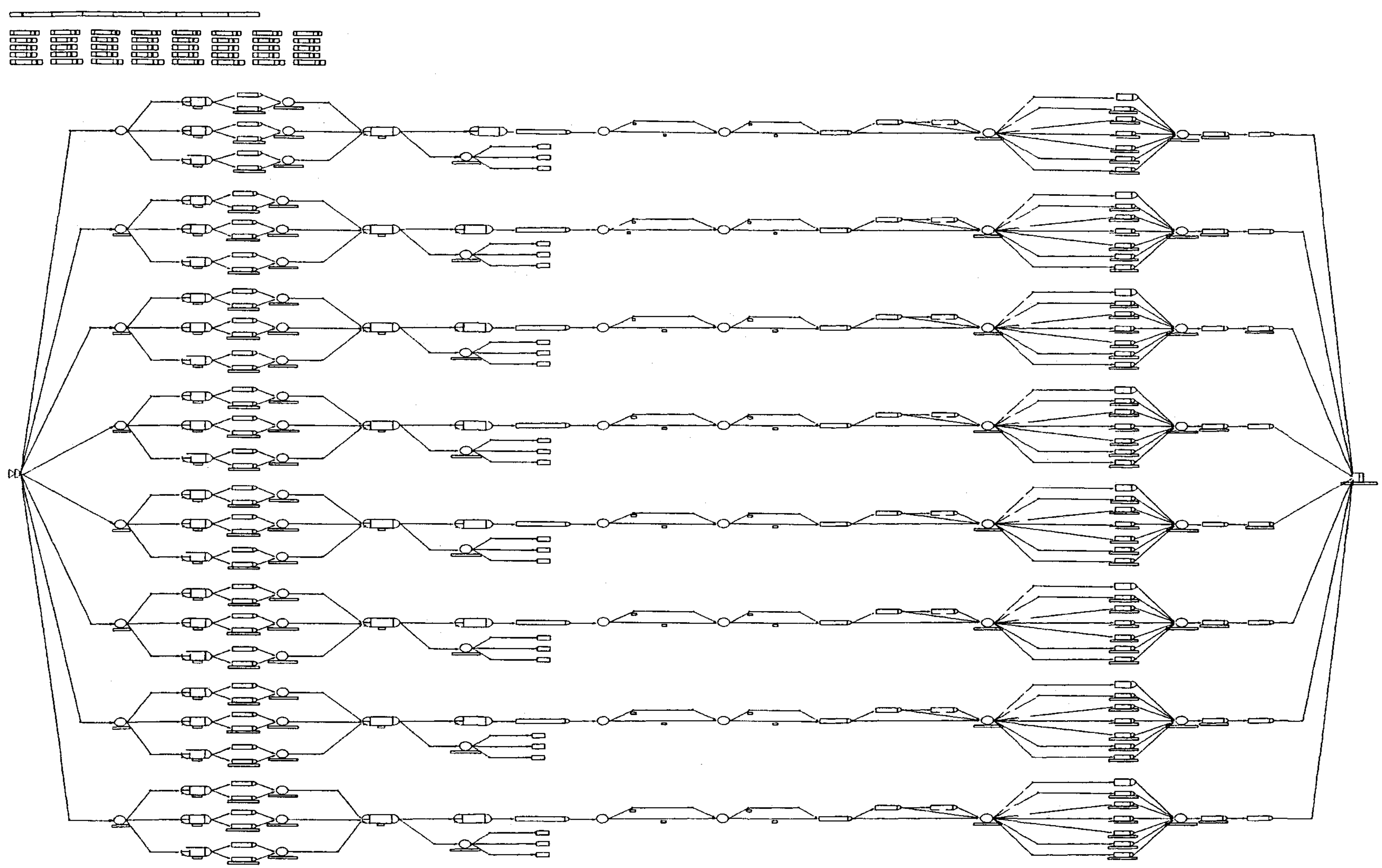
Model Network (enlarged 2/3)



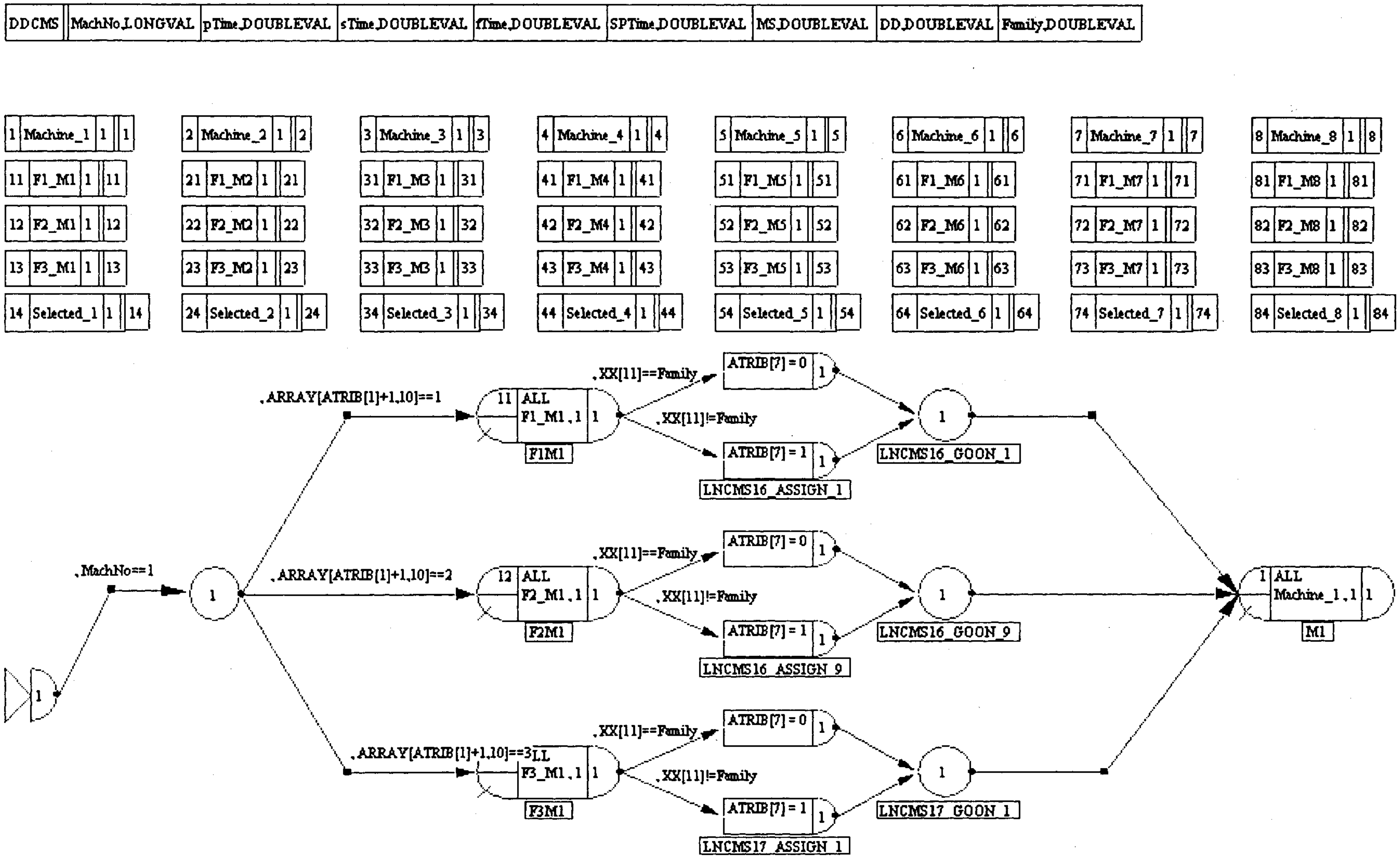
Model Network (enlarged 3/3)



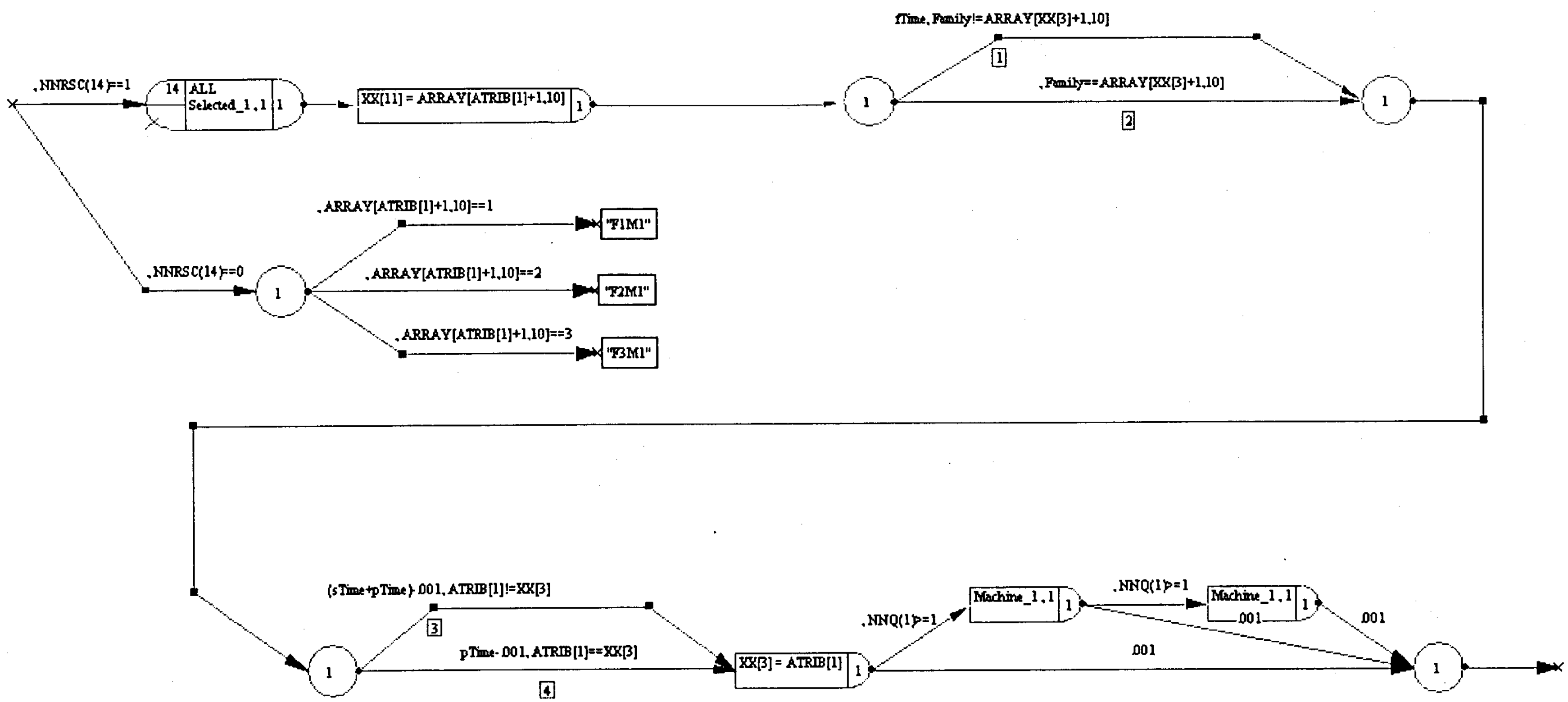
Model Subnetwork



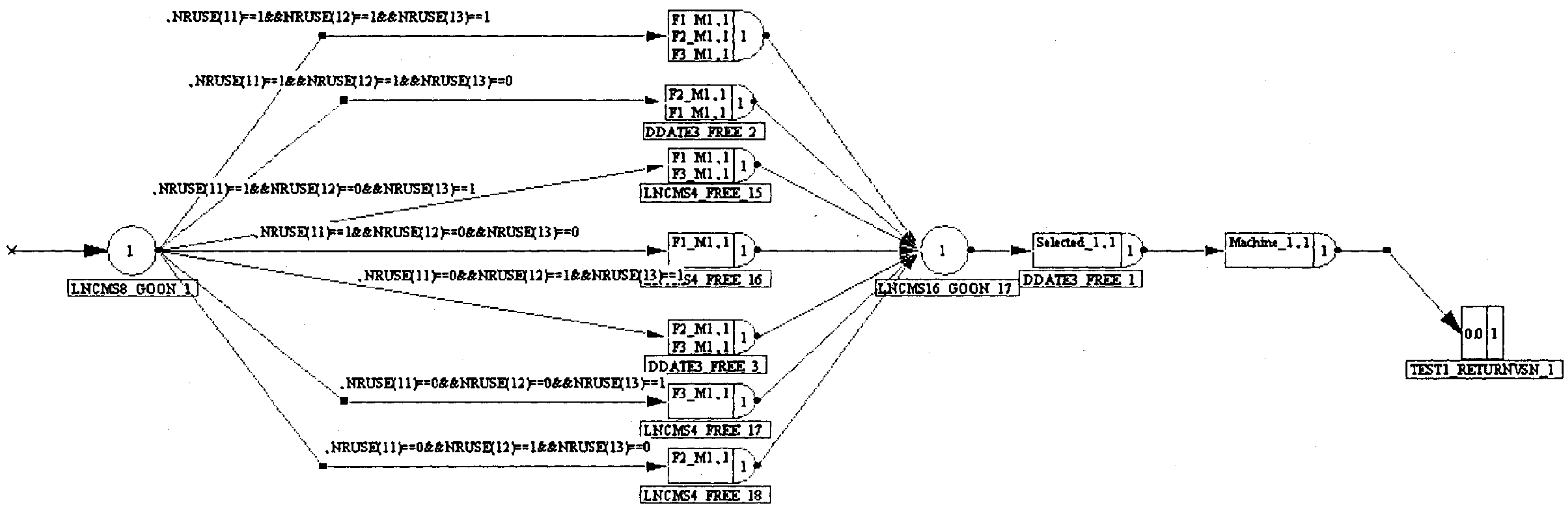
Model Subnetwork (enlarged 1/3)



Model Subnetwork (enlarged 2/3)



Model Subnetwork (enlarged 3/3)



Appendix III

Visual SLAM output reports

Echo report

```
AweSim Input Translator, version 3.0
Copyright (C) 1999 Symix Systems, Inc.

Reading control DDCMS ...

 1 GEN,"Erika","Thesis",September/24/2003,5,YES,YES;
 2 LIMITS,50,10,,40,30;
 3 ARRAY,1,10,{20,25,5,200,0.1,0.5,0,0,0,0};
 4 ARRAY,2,70,
   {1,2,3,9999,9999,9999,9999,9999,9999,1,.1,.1,0,0,0,0,0,0,5,3,4,0,0,0,0
   ,0,3,1,2,0,0,0,0,0,0.9,0.3,0.6,0,0,0,0,0,0.9,0.6,0.6,0,0,0,0,0,2.1,0.7
   ,1.4,0,0,0,0,0,0.9,0.6,0.6,0,0,0,0,0,5,7.1,3.7,5.4};
 5 ARRAY,3,70,
   {1,2,9999,9999,9999,9999,9999,9999,9999,1,.1,0,0,0,0,0,0,0,3,1,0,0,0,0
   ,0,0,1,2,0,0,0,0,0,0,0.3,0.6,0,0,0,0,0,0,0.9,0.6,0,0,0,0,0,0,0.7,1.4,0
   ,0,0,0,0,0,0.9,0.6,0,0,0,0,0,0,6,3.7,2.4,0};
 6 ARRAY,4,70,
   {1,2,3,9999,9999,9999,9999,9999,9999,1,.1,.1,0,0,0,0,0,0,5,3,1,0,0,0,0
   ,0,2,2,1,0,0,0,0,0,0.6,0.6,0.3,0,0,0,0,0,0.9,0.6,0.6,0,0,0,0,0,1.4,1.4
   ,0.7,0,0,0,0,0,0.9,0.6,0.6,0,0,0,0,0,7,6.4,4.4,1.7};
 7 ARRAY,5,70,
   {4,5,9999,9999,9999,9999,9999,9999,9999,2,.1,0,0,0,0,0,0,0,5,1,0,0,0,0
   ,0,0,1,2,0,0,0,0,0,0,0.3,0.6,0,0,0,0,0,0,0.6,0.9,0,0,0,0,0,0,0.7,1.4,0
   ,0,0,0,0,0,0.6,0.9,0,0,0,0,0,0,1,5.7,2.4,0};
 8 ARRAY,6,70,
   {4,9999,9999,9999,9999,9999,9999,9999,9999,2,0,0,0,0,0,0,0,0,2,0,0,0,0
   ,0,0,0,1,0,0,0,0,0,0,0.3,0,0,0,0,0,0,0.6,0,0,0,0,0,0,0.7,0,0,0,0
   ,0,0,0,0.6,0,0,0,0,0,0,2,2.7,0,0};
 9 ARRAY,7,70,
   {4,5,9999,9999,9999,9999,9999,9999,9999,2,.1,0,0,0,0,0,0,0,3,5,0,0,0,0
   ,0,0,2,3,0,0,0,0,0,0,0.6,0.9,0,0,0,0,0,0,0.6,0.9,0,0,0,0,0,0,1.4,2.1,0
   ,0,0,0,0,0,0.6,0.9,0,0,0,0,0,0,3,4.4,7.1,0};
10 ARRAY,8,70,
   {4,5,9999,9999,9999,9999,9999,9999,9999,2,.1,0,0,0,0,0,0,0,4,2,0,0,0,0
   ,0,0,2,1,0,0,0,0,0,0,0.6,0.3,0,0,0,0,0,0,0.6,0.9,0,0,0,0,0,0,1.4,0.7,0
   ,0,0,0,0,0,0.6,0.9,0,0,0,0,0,0,4,5.4,2.7,0};
11 ARRAY,9,70,
   {6,7,8,9999,9999,9999,9999,9999,9999,3,.1,.1,0,0,0,0,0,0,1,3,4,0,0,0,0
   ,0,2,3,1,0,0,0,0,0,0.6,0.9,0.3,0,0,0,0,0,0.9,0.9,0.9,0,0,0,0,0,1.4,2.1
   ,0.7,0,0,0,0,0,0.9,0.9,0.9,0,0,0,0,0,8,2.4,5.1,4.7};
12 ARRAY,10,70,
   {6,7,8,9999,9999,9999,9999,9999,9999,3,.1,.1,0,0,0,0,0,0,2,3,1,0,0,0,0
   ,0,1,2,3,0,0,0,0,0,0.3,0.6,0.9,0,0,0,0,0,0.9,0.9,0.9,0,0,0,0,0,0.7,1.4
   ,2.1,0,0,0,0,0,0.9,0.9,0.9,0,0,0,0,0,9,2.7,4.4,3.1};
13 ARRAY,11,70,
   {6,8,9999,9999,9999,9999,9999,9999,9999,3,.1,0,0,0,0,0,0,0,4,2,0,0,0,0
   ,0,0,3,1,0,0,0,0,0,0,0.9,0.3,0,0,0,0,0,0,0.9,0.9,0,0,0,0,0,0,2.1,0.7,0
   ,0,0,0,0,0,0.9,0.9,0,0,0,0,0,0,10,6.1,2.7,0};
14 ARRAY,12,70,
   {2,4,1,9999,9999,9999,9999,9999,9999,1,2.5,2,0,0,0,0,0,0,2,2,5,0,0,0,0
   ,0,1,2,3,0,0,0,0,0,0.3,0.6,0.9,0,0,0,0,0,0.6,0.9,0.9,0,0,0,0,0,0.7,1.4
   ,2.1,0,0,0,0,0,0.6,0.9,0.9,0,0,0,0,0,11,2.7,3.4,7.1};
15 ARRAY,13,70,
```

```
    {2,5,4,9999,9999,9999,9999,9999,9999,2,2,.1,0,0,0,0,0,0,5,3,1,0,0,0,0,
    0,2,3,2,0,0,0,0,0,0.6,0.9,0.6,0,0,0,0,0,0.6,0.9,0.6,0,0,0,0,0,1.4,2.1,
    1.4,0,0,0,0,0,0.6,0.9,0.6,0,0,0,0,0,12,6.4,5.1,2.4};
 16 ARRAY,14,70,
    {7,3,9999,9999,9999,9999,9999,9999,9999,3,4.5,0,0,0,0,0,0,0,1,2,0,0,0,
    0,0,0,1,3,0,0,0,0,0,0,0.3,0.9,0,0,0,0,0,0,0.9,0.9,0,0,0,0,0,0,0.7,2.1,
    0,0,0,0,0,0,0.9,0.9,0,0,0,0,0,0,13,1.7,4.1,0};
 17 ARRAY,15,70,
    {8,7,4,9999,9999,9999,9999,9999,9999,3,.1,2.5,0,0,0,0,0,0,5,4,1,0,0,0,
    0,0,2,2,1,0,0,0,0,0,0.6,0.6,0.3,0,0,0,0,0,0.9,0.9,0.3,0,0,0,0,0,1.4,1.
    4,0.7,0,0,0,0,0,0.9,0.9,0.3,0,0,0,0,0,14,6.4,5.4,1.7};
 18 ARRAY,16,70,
    {2,4,6,9999,9999,9999,9999,9999,9999,1,2.5,2,0,0,0,0,0,0,1,4,3,0,0,0,0
    ,0,2,1,1,0,0,0,0,0,0.6,0.3,0.3,0,0,0,0,0,0.6,0.9,0.3,0,0,0,0,0,1.4,0.7
    ,0.7,0,0,0,0,0,0.6,0.9,0.3,0,0,0,0,0,15,2.4,4.7,3.7};
 19 ARRAY,17,70,
    {1,5,9999,9999,9999,9999,9999,9999,9999,2,3,0,0,0,0,0,0,0,3,2,0,0,0,0,
    0,0,1,3,0,0,0,0,0,0,0.3,0.9,0,0,0,0,0,0,0.3,0.9,0,0,0,0,0,0,0.7,2.1,0,
    0,0,0,0,0,0.3,0.9,0,0,0,0,0,0,16,3.7,4.1,0};
 20 ARRAY,18,70,
    {7,8,2,9999,9999,9999,9999,9999,9999,3,.1,4.5,0,0,0,0,0,0,3,2,5,0,0,0,
    0,0,3,3,1,0,0,0,0,0,0.9,0.9,0.3,0,0,0,0,0,0.9,0.9,0.3,0,0,0,0,0,2.1,2.
    1,0.7,0,0,0,0,0,0.9,0.9,0.3,0,0,0,0,0,17,5.1,4.1,5.7};
 21 ARRAY,19,70,
    {4,3,9999,9999,9999,9999,9999,9999,9999,1,3,0,0,0,0,0,0,0,3,4,0,0,0,0,
    0,0,3,1,0,0,0,0,0,0,0.9,0.3,0,0,0,0,0,0,0.9,0.6,0,0,0,0,0,0,2.1,0.7,0,
    0,0,0,0,0,0.9,0.6,0,0,0,0,0,0,18,5.1,4.7,0};
 22 ARRAY,20,70,
    {5,6,9999,9999,9999,9999,9999,9999,9999,2,2.5,0,0,0,0,0,0,0,3,5,0,0,0,
    0,0,0,3,2,0,0,0,0,0,0,0.9,0.6,0,0,0,0,0,0,0.9,0.6,0,0,0,0,0,0,2.1,1.4,
    0,0,0,0,0,0,0.9,0.6,0,0,0,0,0,0,19,5.1,6.4,0};
 23 ARRAY,21,70,
    {3,6,5,9999,9999,9999,9999,9999,9999,2,5,2.5,0,0,0,0,0,0,1,4,2,0,0,0,0
    ,0,1,2,2,0,0,0,0,0,0.3,0.6,0.6,0,0,0,0,0,0.3,0.6,0.9,0,0,0,0,0,0.7,1.4
    ,1.4,0,0,0,0,0,0.3,0.6,0.9,0,0,0,0,0,20,1.7,5.4,3.4};
 24 RECORD,1,,TNOW,"TIME",{EXCEL},,TTBEG,TTFIN,,
    {{XX[20],"Part_Index",},{XX[21],"Due Date",},{XX[22],"Quantity",}};
 25 NETWORK,READ;
 26 INITIALIZE,0.0,1200,YES,,NO;
 27 FIN;

CMS successfully read
```

```
Translated file DDCMS successfully written

   Reading network DDCMS - Pass 1...

   1 RESOURCE,9,ResA,1,{9};

   DDCMS - Pass 1 successfully read

   Reading network DDCMS - Pass 2...


   DDCMS - Pass 2 successfully read

   Reading network DDCMS - Pass 3...

   2 ASSIGN,{{XX[1],0}},1;
   3 Order_Arrivals: CREATE,200,0.0,ATRIB[0],,1;
   4 ACTIVITY,1;
   5 ASSIGN,{{XX[1],0}},1;
   6 ACTIVITY;
   7 Due_Date: ASSIGN,
    {{ATRIB[2],TNOW+UNFRM(ARRAY[1,4],ARRAY[1,5]*ARRAY[1,4],
    ARRAY[1,4]+ARRAY[1,5]*ARRAY[1,4])}},1;
   8 ACTIVITY;
   9 UNBATCH,ARRAY[1,1],1;
  10 ACTIVITY,2;
  11 AWAIT,9,{{ResA,1}},ALL,,NONE,1;
  12 ACTIVITY;
  13 ASSIGN,{{XX[1],XX[1]+1}},1;
  14 ACTIVITY,13,,,"DETERMIN_GOON_3";
  15 DETERMIN_GOON_3: GOON,1;
  16 ACTIVITY,3,,PROB(ARRAY[1,6]);
  17 ACTIVITY,4,,PROB(1*(1-ARRAY[1,6])),"NETWORK1_FREE_1";
  18 Quantity:  ASSIGN,{{ATRIB[1],XX[1]},
    {ATRIB[3],INT(RNORM(ARRAY[1,2],ARRAY[1,3]))}},1;
  19 ACTIVITY,11,,,"HYBRID_ASSIGN_2";
  20 HYBRID_ASSIGN_2: ASSIGN,{{XX[2],0}},1;
  21 ACTIVITY;
  22 Routing: GOON,1;
  23 ACTIVITY,5,,ARRAY[ATRIB[1]+1,XX[2]+1]==9999;
  24 ACTIVITY,6,,ARRAY[ATRIB[1]+1,XX[2]+1]<9999,"Operation_Sequence";
  25 ASSIGN,{{ATRIB[5],ATRIB[5]+1},{ATRIB[10+XX[2]+1],9999},
    {ATRIB[4],ARRAY[ATRIB[1]+1,10]}},1;
  26 ACTIVITY;
  27 FREE,{{ResA,1}},1;
  28 ACTIVITY;
  29 UNBATCH,ATRIB[3],1;
  30 ACTIVITY;
  31 Next_Step: GOON,1;
  32 ACTIVITY,8,,ARRAY[ATRIB[1]+1,ATRIB[5]]<9999;
  33 ACTIVITY,10,,ARRAY[ATRIB[1]+1,ATRIB[5]]==9999,"DETERMIN_ASSIGN_1";
  34 VSN1:
    CALLVSN,"ddcms",,{ARRAY[ATRIB[1]+1,ATRIB[5]],ARRAY[ATRIB[1]+1,ATRIB
    [5]+18],ARRAY[ATRIB[1]+1,ATRIB[5]+50],ARRAY[ATRIB[1]+1,ATRIB[5]+42]
    ,ARRAY[ATRIB[1]+1,ATRIB[5]+67],(ARRAY[ATRIB[1]+1,ATRIB[5]+58]*1000)
    +ARRAY[ATRIB[1]+1,ATRIB[5]+67],ATRIB[2],ATRIB[4]},1;
  35 ACTIVITY,9;
```

```
NextOperation: ASSIGN,{{ATRIB[5],ATRIB[5]+1}},1;
ACTIVITY,7,,,"Next_Step";
DETERMIN_ASSIGN_1: ASSIGN,
 {{XX[20],ATRIB[1]},{XX[21],ATRIB[2]},{XX[22],ATRIB[3]},
 {XX[23],XX[23]+1}},1;
ACTIVITY,,,TNOW<=100;
ACTIVITY,,,TNOW>100&&TNOW<=200,"DETERMIN_COLCT_3";
ACTIVITY,,,TNOW>200&&TNOW<=300,"DETERMIN_COLCT_6";
ACTIVITY,,,TNOW>300&&TNOW<=400,"DETERMIN_COLCT_9";
ACTIVITY,,,TNOW>400&&TNOW<=500,"DETERMIN_COLCT_12";
ACTIVITY,,,TNOW>500&&TNOW<=600,"DETERMIN_COLCT_15";
ACTIVITY,,,TNOW>600&&TNOW<=700,"DETERMIN_COLCT_18";
ACTIVITY,,,TNOW>700&&TNOW<=800,"DETERMIN_COLCT_21";
ACTIVITY,,,TNOW>800&&TNOW<=900,"DETERMIN_COLCT_24";
ACTIVITY,,,TNOW>900&&TNOW<=1000,"DETERMIN_COLCT_27";
ACTIVITY,,,TNOW>1000,"DETERMIN_COLCT_30";
Time_in_System_1: COLCT,,TNOW-ATRIB[0],"TIS Batch1",,,,1;
ACTIVITY;
Tardiness_1: COLCT,,MAX(TNOW-ATRIB[2],0),"Tardiness Batch1",,,,1;
ACTIVITY;
Earliness1: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
 "Earliness Batch1",,,,1;
ACTIVITY;
DETERMIN_GOON_2: GOON,1;
ACTIVITY;
TERMINATE,INF;
DETERMIN_COLCT_3: COLCT,,TNOW-ATRIB[0],"TIS Batch2",,,,1;
ACTIVITY;
DETERMIN_COLCT_2: COLCT,,MAX(TNOW-ATRIB[2],0),
 "Tardiness Batch2",,,,1;
ACTIVITY;
DETERMIN_COLCT_1: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
 "Earliness Batch2",,,,1;
ACTIVITY,,,,"DETERMIN_GOON_2";
DETERMIN_COLCT_6: COLCT,,TNOW-ATRIB[0],"TIS Batch3",,,,1;
ACTIVITY;
DETERMIN_COLCT_5: COLCT,,MAX(TNOW-ATRIB[2],0),
 "Tardiness Batch3",,,,1;
ACTIVITY;
DETERMIN_COLCT_4: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
 "Earliness Batch3",,,,1;
ACTIVITY,,,,"DETERMIN_GOON_2";
DETERMIN_COLCT_9: COLCT,,TNOW-ATRIB[0],"TIS Batch4",,,,1;
ACTIVITY;
DETERMIN_COLCT_8: COLCT,,MAX(TNOW-ATRIB[2],0),
 "Tardiness Batch4",,,,1;
ACTIVITY;
DETERMIN_COLCT_7: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
 "Earliness Batch4",,,,1;
ACTIVITY,,,,"DETERMIN_GOON_2";
DETERMIN_COLCT_12: COLCT,,TNOW-ATRIB[0],"TIS Batch5",,,,1;
ACTIVITY;
DETERMIN_COLCT_11: COLCT,,MAX(TNOW-ATRIB[2],0),
 "Tardiness Batch5",,,,1;
ACTIVITY;
DETERMIN_COLCT_10: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
 "Earliness Batch5",,,,1;
```

```
82 ACTIVITY,,,,"DETERMIN_GOON_2";
83 DETERMIN_COLCT_15: COLCT,,TNOW-ATRIB[0],"TIS Batch6",,,,1;
84 ACTIVITY;
85 DETERMIN_COLCT_14: COLCT,,MAX(TNOW-ATRIB[2],0),
   "Tardiness Batch6",,,,1;
86 ACTIVITY;
87 DETERMIN_COLCT_13: COLCT,,MAX(-(TNOW-ATRIB[2]),0),"Earliness
   Batch6",,,,1;
88 ACTIVITY,,,,"DETERMIN_GOON_2";
89 DETERMIN_COLCT_18: COLCT,,TNOW-ATRIB[0],"TIS Batch7",,,,1;
90 ACTIVITY;
91 DETERMIN_COLCT_17: COLCT,,MAX(TNOW-ATRIB[2],0),
   "Tardiness Batch7",,,,1;
92 ACTIVITY;
93 DETERMIN_COLCT_16: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
   "Earliness Batch7",,,,1;
94 ACTIVITY,,,,"DETERMIN_GOON_2";
95 DETERMIN_COLCT_21: COLCT,,TNOW-ATRIB[0],"TIS Batch8",,,,1;
96 ACTIVITY;
97 DETERMIN_COLCT_20: COLCT,,MAX(TNOW-ATRIB[2],0),
   "Tardiness Batch8",,,,1;
98 ACTIVITY;
99 DETERMIN_COLCT_19: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
   "Earliness Batch8",,,,1;
100 ACTIVITY,,,,"DETERMIN_GOON_2";
101 DETERMIN_COLCT_24: COLCT,,TNOW-ATRIB[0],"TIS Batch9",,,,1;
102 ACTIVITY;
103 DETERMIN_COLCT_23: COLCT,,MAX(TNOW-ATRIB[2],0),
    "Tardiness Batch9",,,,1;
104 ACTIVITY;
105 DETERMIN_COLCT_22: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
    "Earliness Batch9",,,,1;
106 ACTIVITY,,,,"DETERMIN_GOON_2";
107 DETERMIN_COLCT_27: COLCT,,TNOW-ATRIB[0],"TIS Batch10",,,,1;
108 ACTIVITY;
109 DETERMIN_COLCT_26: COLCT,,MAX(TNOW-ATRIB[2],0),
    "Tardiness Batch10",,,,1;
110 ACTIVITY;
111 DETERMIN_COLCT_25: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
    "Earliness Batch10",,,,1;
112 ACTIVITY,,,,"DETERMIN_GOON_2";
113 DETERMIN_COLCT_30: COLCT,,TNOW-ATRIB[0],"TIS BatchX",,,,1;
114 ACTIVITY;
115 DETERMIN_COLCT_29: COLCT,,MAX(TNOW-ATRIB[2],0),
    "Tardiness BatchX",,,,1;
116 ACTIVITY;
117 DETERMIN_COLCT_28: COLCT,,MAX(-(TNOW-ATRIB[2]),0),
    "Earliness BatchX",,,,1;
118 ACTIVITY,,,,"DETERMIN_GOON_2";
119 Operation_Sequence: ASSIGN,{{XX[2],XX[2]+1}},1;
120 ACTIVITY;
121 ASSIGN,{{ATRIB[10+XX[2]],ARRAY[ATRIB[1]+1,XX[2]]}},1;
122 ACTIVITY,,,,"Routing";
123 NETWORK1_FREE_1: FREE,{{ResA,1}},1;
124 ACTIVITY;
125 TERMINATE,INF;
```

```
;ATRIB[2]=Due Dates (randomly assigned)= UNIFORM(1-
 (alfa)l,l+(alfa)l).
;l=ARRAY[1,4]=Average lead time per order=N*Q*t=10*120*7.7=9240
;Alfa can be any percentage (e.g.10%) defined by ARRRAY[1,5]
;N=Average order of parts (batch size)=P*N=Probability*Total
NoParts=0.5*20=10
;Q=ARRAY[1,2]=Order size of a part=miu symb.=120
;ARRAY[1,1]=20 Parts of the mfg system
;t=Average lead time per
part=(Sum(i)*Sum(j)*Sum(k)*tijk)/N=154/20=7.7
;t(i,j,k)=Time for processing operation k of part j on machine i
;Probability of a part being selected for processing or not
;ATRIB[5]=Next operation
;can be assigned one entity at a time.
;ATRIB[1]=Part No
;ResA is assigned so the operation sequence (XX[2]), machine
number (ATRIB10+XX[2]) and family number (ATRIB[4]) ,
;ATRIB[3]=Quantity to be produced of each selected part
;ARRAY[1,2]=Miu. ARRAY[1,3]=Sigma

DDCMS - Pass 3 successfully read
```

Reading subnetwork DDCMS ...

```
 1 VSN,DDCMS,{{MachNo,LONGVAL,MachineNo.},{pTime,DOUBLEVAL,Processing
   Time},{sTime,DOUBLEVAL,Part Setup Time},{fTime,DOUBLEVAL,Family
   Setup Time},{SPTime,DOUBLEVAL,Shortest Processing Time
   (pTime+sTime)},{MS,DOUBLEVAL,Minimum Setup},{DD,DOUBLEVAL,Slack
   Time = Due Date - TNOW},{Family,DOUBLEVAL, }};
 2 RESOURCE,11,F1_M1,1,{11};
 3 RESOURCE,21,F1_M2,1,{21};
 4 RESOURCE,31,F1_M3,1,{31};
 5 RESOURCE,41,F1_M4,1,{41};
 6 RESOURCE,4,Machine_4,1,{4};
 7 RESOURCE,51,F1_M5,1,{51};
 8 RESOURCE,5,Machine_5,1,{5};
 9 RESOURCE,61,F1_M6,1,{61};
10 RESOURCE,6,Machine_6,1,{6};
11 RESOURCE,71,F1_M7,1,{71};
12 RESOURCE,7,Machine_7,1,{7};
13 RESOURCE,8,Machine_8,1,{8};
14 RESOURCE,81,F1_M8,1,{81};
15 RESOURCE,1,Machine_1,1,{1};
16 RESOURCE,3,Machine_3,1,{3};
17 RESOURCE,2,Machine_2,1,{2};
18 RESOURCE,72,F2_M7,1,{72};
19 RESOURCE,73,F3_M7,1,{73};
20 RESOURCE,62,F2_M6,1,{62};
21 RESOURCE,63,F3_M6,1,{63};
22 RESOURCE,52,F2_M5,1,{52};
23 RESOURCE,53,F3_M5,1,{53};
24 RESOURCE,42,F2_M4,1,{42};
25 RESOURCE,43,F3_M4,1,{43};
26 RESOURCE,32,F2_M3,1,{32};
27 RESOURCE,33,F3_M3,1,{33};
28 RESOURCE,23,F3_M2,1,{23};
29 RESOURCE,22,F2_M2,1,{22};
30 RESOURCE,12,F2_M1,1,{12};
31 RESOURCE,13,F3_M1,1,{13};
32 RESOURCE,83,F3_M8,1,{83};
33 RESOURCE,82,F2_M8,1,{82};
34 RESOURCE,74,Selected_7,1,{74};
35 RESOURCE,64,Selected_6,1,{64};
36 RESOURCE,54,Selected_5,1,{54};
37 RESOURCE,44,Selected_4,1,{44};
38 RESOURCE,34,Selected_3,1,{34};
39 RESOURCE,24,Selected_2,1,{24};
40 RESOURCE,14,Selected_1,1,{14};
41 RESOURCE,84,Selected_8,1,{84};
42 PRIORITY,{{8,LVF((ATRIB[7]*10000)+DD)}};
43 PRIORITY,{{7,LVF((ATRIB[7]*10000)+DD)}};
44 PRIORITY,{{6,LVF((ATRIB[7]*10000)+DD)}};
45 PRIORITY,{{5,LVF((ATRIB[7]*10000)+DD)}};
46 PRIORITY,{{4,LVF((ATRIB[7]*10000)+DD)}};
47 PRIORITY,{{3,LVF((ATRIB[7]*10000)+DD)}};
48 PRIORITY,{{2,LVF((ATRIB[7]*10000)+DD)}};
49 PRIORITY,{{1,LVF((ATRIB[7]*10000)+DD)}};
50 PRIORITY,{{83,LVF((((DD-TNOW-SPTime)/
   (ABS(DD-TNOW- SPTime)))*100)+SPTime)}};
```

```
PRIORITY,{{82,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{81,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{73,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{72,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{71,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{63,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{62,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{61,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{53,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{52,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{51,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{43,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{42,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{41,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{33,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{32,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{31,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{23,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{22,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{21,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{13,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{12,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
PRIORITY,{{11,LVF((((DD-TNOW-SPTime)/(ABS(DD-TNOW-SPTime)))*100)+SPTime)}};
EQUIVALENCE,{{SPTime,ARRAY[ATRIB[1]+1,ATRIB[5]+67]}};
EQUIVALENCE,{{DD,ATRIB[2]}};
;* IN ORDER TO CHANGE FROM MS RULE TO DD, WE JUST HAVE TO GO TO THE EDIT ACTION BAR, SELECT CONTROLS AND EDIT THE CHANGE IN THE FILE OF EACH MACHINE
;pTime = Processing Time of the Part
;Activity 1, 5, 9, 13 : "fTime" if, First Family Part to be Processed or Different Family Part as the Last Part Processed
;sTime = Setup Time of the Part
;Activity 2, 6, 10, 14 : Non-Family Part or Same Family Part as the Last Part Processed
```

```
;fTime = Family time of the (Family) Part
;Activity 3, 7, 11, 15 : "sTime + p Time" if, Different Part
 Number as the Last Part Processed
;SPTime = Shortest Processing Time of the Part = sTime + pTime
;Activity 4, 8, 12, 16 : "pTime" if, Same Part Number as the Last
 Part Processed
;MSTime = Minimum Setup Rule. (Stime*M)+SPTime
;DDTime =  Due Date Rule. (Due Date of the Part*M)+SPTime
;M = Maximum Number
ENTERVSN,,1;
ACTIVITY,,,MachNo==1;
ACTIVITY,,,MachNo==2,"Machine_2";
ACTIVITY,,,MachNo==3,"Machine_3";
ACTIVITY,,,MachNo==4,"Machine_4";
ACTIVITY,,,MachNo==5,"Machine_5";
ACTIVITY,,,MachNo==6,"Machine_6";
ACTIVITY,,,MachNo==7,"Machine_7";
ACTIVITY,,,MachNo==8,"Machine_8";
GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M1";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M1";
F1M1: AWAIT,11,{{F1_M1,1}},ALL,,NONE,1;
ACTIVITY,,,XX[11]==Family;
ACTIVITY,,,XX[11]!=Family,"LNCMS16_ASSIGN_1";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_1: GOON,1;
ACTIVITY;
M1: AWAIT,1,{{Machine_1,1}},ALL,,NONE,1;
ACTIVITY,,,NNRSC(14)==1;
ACTIVITY,,,NNRSC(14)==0,"DDATE3_GOON_1";
AWAIT,14,{{Selected_1,1}},ALL,,NONE,1;
ACTIVITY;
ASSIGN,{{XX[11],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,1,fTime,Family!=ARRAY[XX[3]+1,10],,,"Family Setup M1";
ACTIVITY,2,,Family==ARRAY[XX[3]+1,10],,,"NF&sameF M1";
GOON,1;
ACTIVITY,3,(sTime+pTime)-.001,ATRIB[1]!=XX[3],,,
 "Setup+Process M1";
ACTIVITY,4,pTime-.001,ATRIB[1]==XX[3],,,"Process M1";
ASSIGN,{{XX[3],ATRIB[1]}},1;
ACTIVITY,,,NNQ(1)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_1";
FREE,{{Machine_1,1}},1;
ACTIVITY,,,NNQ(1)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_1";
FREE,{{Machine_1,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_1: GOON,1;
ACTIVITY,,,NRUSE(11)==1&&NRUSE(12)==1&&NRUSE(13)==1;
ACTIVITY,,,NRUSE(11)==1&&NRUSE(12)==1&&NRUSE(13)==0,
 "DDATE3_FREE_2";
ACTIVITY,,,NRUSE(11)==1&&NRUSE(12)==0&&NRUSE(13)==1,
 "LNCMS4_FREE_15";
```

```
ACTIVITY,,,NRUSE(11)==1&&NRUSE(12)==0&&NRUSE(13)==0,
"LNCMS4_FREE_16";
ACTIVITY,,,NRUSE(11)==0&&NRUSE(12)==1&&NRUSE(13)==1,
"DDATE3_FREE_3";
ACTIVITY,,,NRUSE(11)==0&&NRUSE(12)==0&&NRUSE(13)==1,
"LNCMS4_FREE_17";
ACTIVITY,,,NRUSE(11)==0&&NRUSE(12)==1&&NRUSE(13)==0,
"LNCMS4_FREE_18";
FREE,{{F1_M1,1},{F2_M1,1},{F3_M1,1}},1;
ACTIVITY;
LNCMS16_GOON_17: GOON,1;
ACTIVITY;
DDATE3_FREE_1: FREE,{{Selected_1,1}},1;
ACTIVITY;
FREE,{{Machine_1,1}},1;
ACTIVITY,,,,"TEST1_RETURNVSN_1";
TEST1_RETURNVSN_1: RETURNVSN,0.0,1;
DDATE3_FREE_2: FREE,{{F2_M1,1},{F1_M1,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_17";
LNCMS4_FREE_15: FREE,{{F1_M1,1},{F3_M1,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_17";
LNCMS4_FREE_16: FREE,{{F1_M1,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_17";
DDATE3_FREE_3: FREE,{{F2_M1,1},{F3_M1,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_17";
LNCMS4_FREE_17: FREE,{{F3_M1,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_17";
LNCMS4_FREE_18: FREE,{{F2_M1,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_17";
DDATE3_GOON_1: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1,"F1M1";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M1";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M1";
LNCMS16_ASSIGN_1: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_1";
F2M1: AWAIT,12,{{F2_M1,1}},ALL,,NONE,1;
ACTIVITY,,,XX[11]==Family;
ACTIVITY,,,XX[11]!=Family,"LNCMS16_ASSIGN_9";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_9: GOON,1;
ACTIVITY,,,,"M1";
LNCMS16_ASSIGN_9: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_9";
F3M1: AWAIT,13,{{F3_M1,1}},ALL,,NONE,1;
ACTIVITY,,,XX[11]==Family;
ACTIVITY,,,XX[11]!=Family,"LNCMS17_ASSIGN_1";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_1: GOON,1;
ACTIVITY,,,,"M1";
LNCMS17_ASSIGN_1: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS17_GOON_1";
Machine_2: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1,,10;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M2",10;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M2";
```

```
F1M2: AWAIT,21,{{F1_M2,1}},ALL,,NONE,1;
ACTIVITY,,,XX[12]==Family;
ACTIVITY,,,XX[12]!=Family,"LNCMS16_ASSIGN_2";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_2: GOON,1;
ACTIVITY;
M2: AWAIT,2,{{Machine_2,1}},ALL,,NONE,1;
ACTIVITY,,,NNRSC(24)==1;
ACTIVITY,,,NNRSC(24)==0,"DDATE3_GOON_2";
AWAIT,24,{{Selected_2,1}},ALL,,NONE,1;
ACTIVITY;
ASSIGN,{{XX[12],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,5,fTime,Family!=ARRAY[XX[4]+1,10],,,"Family Setup M2";
ACTIVITY,6,,Family==ARRAY[XX[4]+1,10],,,"NF&sameF M2";
GOON,1;
ACTIVITY,7,(sTime+pTime)-.001,ATRIB[1]!=XX[4],,,
"Setup+Process M2";
ACTIVITY,8,pTime-.001,ATRIB[1]==XX[4],,,"Process M2";
ASSIGN,{{XX[4],ATRIB[1]}},1;
ACTIVITY,,,NNQ(2)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_2";
FREE,{{Machine_2,1}},1;
ACTIVITY,,,NNQ(2)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_2";
FREE,{{Machine_2,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_2: GOON,1;
ACTIVITY,,,NRUSE(21)==1&&NRUSE(22)==1&&NRUSE(23)==1;
ACTIVITY,,,NRUSE(21)==1&&NRUSE(22)==1&&NRUSE(23)==0,
"LNCMS4_FREE_22";
ACTIVITY,,,NRUSE(21)==1&&NRUSE(22)==0&&NRUSE(23)==1,
"LNCMS4_FREE_21";
ACTIVITY,,,NRUSE(21)==1&&NRUSE(22)==0&&NRUSE(23)==0,
"LNCMS4_FREE_20";
ACTIVITY,,,NRUSE(21)==0&&NRUSE(22)==1&&NRUSE(23)==1,
"LNCMS4_FREE_19";
ACTIVITY,,,NRUSE(21)==0&&NRUSE(22)==0&&NRUSE(23)==1,
"LNCMS4_FREE_2";
ACTIVITY,,,NRUSE(21)==0&&NRUSE(22)==1&&NRUSE(23)==0,
"LNCMS4_FREE_1";
FREE,{{F2_M2,1},{F3_M2,1},{F1_M2,1}},1;
ACTIVITY;
LNCMS16_GOON_18: GOON,1;
ACTIVITY;
DDATE3_FREE_7: FREE,{{Selected_2,1}},1;
ACTIVITY;
FREE,{{Machine_2,1}},1;
ACTIVITY,,,,"TEST1_RETURNVSN_1";
LNCMS4_FREE_22: FREE,{{F2_M2,1},{F1_M2,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_18";
LNCMS4_FREE_21: FREE,{{F3_M2,1},{F1_M2,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_18";
LNCMS4_FREE_20: FREE,{{F1_M2,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_18";
```

```
LNCMS4_FREE_19: FREE,{{F2_M2,1},{F3_M2,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_18";
LNCMS4_FREE_2: FREE,{{F3_M2,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_18";
LNCMS4_FREE_1: FREE,{{F2_M2,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_18";
DDATE3_GOON_2: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1,"F1M2";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M2";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M2";
LNCMS16_ASSIGN_2: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_2";
F2M2: AWAIT,22,{{F2_M2,1}},ALL,,NONE,1;
ACTIVITY,,,XX[12]==Family;
ACTIVITY,,,XX[12]!=Family,"LNCMS16_ASSIGN_10";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_10: GOON,1;
ACTIVITY,,,,"M2";
LNCMS16_ASSIGN_10: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_10";
F3M2: AWAIT,23,{{F3_M2,1}},ALL,,NONE,1;
ACTIVITY,,,XX[12]==Family;
ACTIVITY,,,XX[12]!=Family,"LNCMS17_ASSIGN_2";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_2: GOON,1;
ACTIVITY,,,,"M2";
LNCMS17_ASSIGN_2: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS17_GOON_2";
Machine_3: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M3";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M3";
F1M3: AWAIT,31,{{F1_M3,1}},ALL,,NONE,1;
ACTIVITY,,,XX[13]==Family;
ACTIVITY,,,XX[13]!=Family,"LNCMS16_ASSIGN_3";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_3: GOON,1;
ACTIVITY;
M3: AWAIT,3,{{Machine_3,1}},ALL,,NONE,1;
ACTIVITY,,,NNRSC(34)==1;
ACTIVITY,,,NNRSC(34)==0,"DDATEC_GOON_22";
AWAIT,34,{{Selected_3,1}},ALL,,NONE,1;
ACTIVITY;
ASSIGN,{{XX[13],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,9,fTime,Family!=ARRAY[XX[5]+1,10],,,"Family Setup M3";
ACTIVITY,10,,Family==ARRAY[XX[5]+1,10],,,"NF&sameF M3";
GOON,1;
ACTIVITY,11,(sTime+pTime)-.001,ATRIB[1]!=XX[5],,,
    "Setup+Process M3";
ACTIVITY,12,pTime-.001,ATRIB[1]==XX[5],,,"Process M3";
ASSIGN,{{XX[5],ATRIB[1]}},1;
ACTIVITY,,,NNQ(3)>=1;
```

```
ACTIVITY,,.001,,"LNCMS8_GOON_3";
FREE,{{Machine_3,1}},1;
ACTIVITY,,NNQ(3)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_3";
FREE,{{Machine_3,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_3: GOON,1;
ACTIVITY,,NRUSE(31)==1&&NRUSE(32)==1&&NRUSE(33)==1;
ACTIVITY,,NRUSE(31)==1&&NRUSE(32)==1&&NRUSE(33)==0,
 "LNCMS4_FREE_4";
ACTIVITY,,NRUSE(31)==1&&NRUSE(32)==0&&NRUSE(33)==1,
 "LNCMS4_FREE_23";
ACTIVITY,,NRUSE(31)==1&&NRUSE(32)==0&&NRUSE(33)==0,
 "LNCMS4_FREE_24";
ACTIVITY,,NRUSE(31)==0&&NRUSE(32)==1&&NRUSE(33)==1,
 "LNCMS4_FREE_3";
ACTIVITY,,NRUSE(31)==0&&NRUSE(32)==0&&NRUSE(33)==1,
 "LNCMS4_FREE_25";
ACTIVITY,,NRUSE(31)==0&&NRUSE(32)==1&&NRUSE(33)==0,
 "LNCMS4_FREE_26";
FREE,{{F1_M3,1},{F2_M3,1},{F3_M3,1}},1;
ACTIVITY;
LNCMS16_GOON_19: GOON,1;
ACTIVITY;
FREE,{{Selected_3,1}},1;
ACTIVITY;
DDATE3_FREE_13: FREE,{{Machine_3,1}},1;
ACTIVITY,,,"TEST1_RETURNVSN_1";
LNCMS4_FREE_4: FREE,{{F1_M3,1},{F2_M3,1}},1;
ACTIVITY,,,"LNCMS16_GOON_19";
LNCMS4_FREE_23: FREE,{{F1_M3,1},{F3_M3,1}},1;
ACTIVITY,,,"LNCMS16_GOON_19";
LNCMS4_FREE_24: FREE,{{F1_M3,1}},1;
ACTIVITY,,,"LNCMS16_GOON_19";
LNCMS4_FREE_3: FREE,{{F2_M3,1},{F3_M3,1}},1;
ACTIVITY,,,"LNCMS16_GOON_19";
LNCMS4_FREE_25: FREE,{{F3_M3,1}},1;
ACTIVITY,,,"LNCMS16_GOON_19";
LNCMS4_FREE_26: FREE,{{F2_M3,1}},1;
ACTIVITY,,,"LNCMS16_GOON_19";
DDATEC_GOON_22: GOON,1;
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==1,"F1M3";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==2,"F2M3";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==3,"F3M3";
LNCMS16_ASSIGN_3: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,"LNCMS16_GOON_3";
F2M3: AWAIT,32,{{F2_M3,1}},ALL,,NONE,1;
ACTIVITY,,XX[13]==Family;
ACTIVITY,,XX[13]!=Family,"LNCMS16_ASSIGN_11";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_11: GOON,1;
ACTIVITY,,,"M3";
LNCMS16_ASSIGN_11: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,"LNCMS16_GOON_11";
F3M3: AWAIT,33,{{F3_M3,1}},ALL,,NONE,1;
ACTIVITY,,XX[13]==Family;
```

```
ACTIVITY,,,XX[13]!=Family,"LNCMS17_ASSIGN_3";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_3: GOON,1;
ACTIVITY,,,,"M3";
LNCMS17_ASSIGN_3: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS17_GOON_3";
Machine_4: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M4";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M4";
F1M4: AWAIT,41,{{F1_M4,1}},ALL,,NONE,1;
ACTIVITY,,,XX[14]==Family;
ACTIVITY,,,XX[14]!=Family,"LNCMS16_ASSIGN_4";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_4: GOON,1;
ACTIVITY;
M4: AWAIT,4,{{Machine_4,1}},ALL,,NONE,1;
ACTIVITY,,,NNRSC(44)==1;
ACTIVITY,,,NNRSC(44)==0,"DDATEC_GOON_24";
AWAIT,44,{{Selected_4,1}},ALL,,NONE,1;
ACTIVITY;
ASSIGN,{{XX[14],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,13,fTime,Family!=ARRAY[XX[6]+1,10],,,"Family Setup M4";
ACTIVITY,14,,Family==ARRAY[XX[6]+1,10],,,"NF&sameF M4";
GOON,1;
ACTIVITY,15,(sTime+pTime)-.001,ATRIB[1]!=XX[6],,,
 "Setup+Process M4";
ACTIVITY,16,pTime-.001,ATRIB[1]==XX[6],,,"Process M4";
ASSIGN,{{XX[6],ATRIB[1]}},1;
ACTIVITY,,,NNQ(4)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_4";
FREE,{{Machine_4,1}},1;
ACTIVITY,,,NNQ(4)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_4";
FREE,{{Machine_4,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_4: GOON,1;
ACTIVITY,,,NRUSE(41)==1&&NRUSE(42)==1&&NRUSE(43)==1;
ACTIVITY,,,NRUSE(41)==1&&NRUSE(42)==1&&NRUSE(43)==0,
 "LNCMS4_FREE_6";
ACTIVITY,,,NRUSE(41)==1&&NRUSE(42)==0&&NRUSE(43)==1,
 "LNCMS4_FREE_27";
ACTIVITY,,,NRUSE(41)==1&&NRUSE(42)==0&&NRUSE(43)==0,
 "LNCMS4_FREE_28";
ACTIVITY,,,NRUSE(41)==0&&NRUSE(42)==1&&NRUSE(43)==1,
 "LNCMS4_FREE_5";
ACTIVITY,,,NRUSE(41)==0&&NRUSE(42)==0&&NRUSE(43)==1,
 "LNCMS4_FREE_29";
ACTIVITY,,,NRUSE(41)==0&&NRUSE(42)==1&&NRUSE(43)==0,
 "LNCMS4_FREE_30";
FREE,{{F1_M4,1},{F2_M4,1},{F3_M4,1}},1;
ACTIVITY;
LNCMS16_GOON_20: GOON,1;
```

```
ACTIVITY;
DDATE3_FREE_19: FREE,{{Selected_4,1}},1;
ACTIVITY;
FREE,{{Machine_4,1}},1;
ACTIVITY,,,,"TEST1_RETURNVSN_1";
LNCMS4_FREE_6: FREE,{{F1_M4,1},{F2_M4,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_20";
LNCMS4_FREE_27: FREE,{{F3_M4,1},{F1_M4,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_20";
LNCMS4_FREE_28: FREE,{{F1_M4,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_20";
LNCMS4_FREE_5: FREE,{{F2_M4,1},{F3_M4,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_20";
LNCMS4_FREE_29: FREE,{{F3_M4,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_20";
LNCMS4_FREE_30: FREE,{{F2_M4,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_20";
DDATEC_GOON_24: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1,"F1M4";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M4";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M4";
LNCMS16_ASSIGN_4: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_4";
F2M4: AWAIT,42,{{F2_M4,1}},ALL,,NONE,1;
ACTIVITY,,,XX[14]==Family;
ACTIVITY,,,XX[14]!=Family,"LNCMS16_ASSIGN_12";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_12: GOON,1;
ACTIVITY,,,,"M4";
LNCMS16_ASSIGN_12: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_12";
F3M4: AWAIT,43,{{F3_M4,1}},ALL,,NONE,1;
ACTIVITY,,,XX[14]==Family;
ACTIVITY,,,XX[14]!=Family,"LNCMS17_ASSIGN_4";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_4: GOON,1;
ACTIVITY,,,,"M4";
LNCMS17_ASSIGN_4: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS17_GOON_4";
Machine_5: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M5";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M5";
F1M5: AWAIT,51,{{F1_M5,1}},ALL,,NONE,1;
ACTIVITY,,,XX[15]==Family;
ACTIVITY,,,XX[15]!=Family,"LNCMS16_ASSIGN_5";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_5: GOON,1;
ACTIVITY;
M5: AWAIT,5,{{Machine_5,1}},ALL,,NONE,1;
ACTIVITY,,,NNRSC(54)==1;
ACTIVITY,,,NNRSC(54)==0,"DDATEC_GOON_26";
AWAIT,54,{{Selected_5,1}},ALL,,NONE,1;
ACTIVITY;
```

```
ASSIGN,{{XX[15],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,17,fTime,Family!=ARRAY[XX[7]+1,10],,,"Family Setup M5";
ACTIVITY,18,,Family==ARRAY[XX[7]+1,10],,,"NF&sameF M5";
GOON,1;
ACTIVITY,19,(sTime+pTime)-.001,ATRIB[1]!=XX[7],,,
 "Setup+Process M5";
ACTIVITY,20,pTime-.001,ATRIB[1]==XX[7],,,"Process M5";
ASSIGN,{{XX[7],ATRIB[1]}},1;
ACTIVITY,,NNQ(5)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_5";
FREE,{{Machine_5,1}},1;
ACTIVITY,,NNQ(5)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_5";
FREE,{{Machine_5,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_5: GOON,1;
ACTIVITY,,NRUSE(51)==1&&NRUSE(52)==1&&NRUSE(53)==1;
ACTIVITY,,NRUSE(51)==1&&NRUSE(52)==1&&NRUSE(53)==0,
 "LNCMS4_FREE_8";
ACTIVITY,,NRUSE(51)==1&&NRUSE(52)==0&&NRUSE(53)==1,
 "LNCMS4_FREE_31";
ACTIVITY,,NRUSE(51)==1&&NRUSE(52)==0&&NRUSE(53)==0,
 "LNCMS4_FREE_32";
ACTIVITY,,NRUSE(51)==0&&NRUSE(52)==1&&NRUSE(53)==1,
 "LNCMS4_FREE_7";
ACTIVITY,,NRUSE(51)==0&&NRUSE(52)==0&&NRUSE(53)==1,
 "LNCMS4_FREE_33";
ACTIVITY,,NRUSE(51)==0&&NRUSE(52)==1&&NRUSE(53)==0,
 "LNCMS4_FREE_34";
FREE,{{F1_M5,1},{F2_M5,1},{F3_M5,1}},1;
ACTIVITY;
LNCMS16_GOON_22: GOON,1;
ACTIVITY;
FREE,{{Selected_5,1}},1;
ACTIVITY;
DDATE3_FREE_25: FREE,{{Machine_5,1}},1;
ACTIVITY,,,,"TEST1_RETURNVSN_1";
LNCMS4_FREE_8: FREE,{{F1_M5,1},{F2_M5,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_22";
LNCMS4_FREE_31: FREE,{{F1_M5,1},{F3_M5,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_22";
LNCMS4_FREE_32: FREE,{{F1_M5,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_22";
LNCMS4_FREE_7: FREE,{{F2_M5,1},{F3_M5,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_22";
LNCMS4_FREE_33: FREE,{{F3_M5,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_22";
LNCMS4_FREE_34: FREE,{{F2_M5,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_22";
DDATEC_GOON_26: GOON,1;
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==1,"F1M5";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==2,"F2M5";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==3,"F3M5";
LNCMS16_ASSIGN_5: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_5";
```

```
F2M5: AWAIT,52,{{F2_M5,1}},ALL,,NONE,1;
ACTIVITY,,,XX[15]==Family;
ACTIVITY,,,XX[15]!=Family,"LNCMS16_ASSIGN_13";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_13: GOON,1;
ACTIVITY,,,,"M5";
LNCMS16_ASSIGN_13: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_13";
F3M5: AWAIT,53,{{F3_M5,1}},ALL,,NONE,1;
ACTIVITY,,,XX[15]==Family;
ACTIVITY,,,XX[15]!=Family,"LNCMS17_ASSIGN_5";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_5: GOON,1;
ACTIVITY,,,,"M5";
LNCMS17_ASSIGN_5: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS17_GOON_5";
Machine_6: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M6";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M6";
F1M6: AWAIT,61,{{F1_M6,1}},ALL,,NONE,1;
ACTIVITY,,,XX[16]==Family;
ACTIVITY,,,XX[16]!=Family,"LNCMS16_ASSIGN_6";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_6: GOON,1;
ACTIVITY;
M6: AWAIT,6,{{Machine_6,1}},ALL,,NONE,1;
ACTIVITY,,,NNRSC(64)==1;
ACTIVITY,,,NNRSC(64)==0,"DDATE3_GOON_3";
AWAIT,64,{{Selected_6,1}},ALL,,NONE,1;
ACTIVITY;
ASSIGN,{{XX[16],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,21,fTime,Family!=ARRAY[XX[8]+1,10],,,"Family Setup M6";
ACTIVITY,22,,Family==ARRAY[XX[8]+1,10],,,"NF&sameF M6";
GOON,1;
ACTIVITY,23,(sTime+pTime)-.001,ATRIB[1]!=XX[8],,,
"Setup+Process M6";
ACTIVITY,24,pTime-.001,ATRIB[1]==XX[8],,,"Process M6";
ASSIGN,{{XX[8],ATRIB[1]}},1;
ACTIVITY,,,NNQ(6)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_6";
FREE,{{Machine_6,1}},1;
ACTIVITY,,,NNQ(1)>=6;
ACTIVITY,,.001,,"LNCMS8_GOON_6";
FREE,{{Machine_6,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_6: GOON,1;
ACTIVITY,,,NRUSE(61)==1&&NRUSE(62)==1&&NRUSE(63)==1;
ACTIVITY,,,NRUSE(61)==1&&NRUSE(62)==1&&NRUSE(63)==0,
"LNCMS4_FREE_10";
ACTIVITY,,,NRUSE(61)==1&&NRUSE(62)==0&&NRUSE(63)==1,
"LNCMS4_FREE_35";
```

```
ACTIVITY,,,NRUSE(61)==1&&NRUSE(62)==0&&NRUSE(63)==0,
    "LNCMS4_FREE_36";
ACTIVITY,,,NRUSE(61)==0&&NRUSE(62)==1&&NRUSE(63)==1,
    "LNCMS4_FREE_9";
ACTIVITY,,,NRUSE(61)==0&&NRUSE(62)==0&&NRUSE(63)==1,
    "LNCMS4_FREE_37";
ACTIVITY,,,NRUSE(61)==0&&NRUSE(62)==1&&NRUSE(63)==0,
    "LNCMS17_FREE_1";
FREE,{{F1_M6,1},{F2_M6,1},{F3_M6,1}},1;
ACTIVITY;
LNCMS16_GOON_23: GOON,1;
ACTIVITY;
DDATE3_FREE_31: FREE,{{Selected_6,1}},1;
ACTIVITY;
FREE,{{Machine_6,1}},1;
ACTIVITY,,,,"TEST1_RETURNVSN_1";
LNCMS4_FREE_10: FREE,{{F1_M6,1},{F2_M6,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_23";
LNCMS4_FREE_35: FREE,{{F1_M6,1},{F3_M6,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_23";
LNCMS4_FREE_36: FREE,{{F1_M6,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_23";
LNCMS4_FREE_9: FREE,{{F2_M6,1},{F3_M6,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_23";
LNCMS4_FREE_37: FREE,{{F3_M6,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_23";
LNCMS17_FREE_1: FREE,{{F2_M6,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_23";
DDATE3_GOON_3: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1,"F1M6";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M6";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M6";
LNCMS16_ASSIGN_6: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_6";
F2M6: AWAIT,62,{{F2_M6,1}},ALL,,NONE,1;
ACTIVITY,,,XX[16]==Family;
ACTIVITY,,,XX[16]!=Family,"LNCMS16_ASSIGN_14";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_14: GOON,1;
ACTIVITY,,,,"M6";
LNCMS16_ASSIGN_14: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS16_GOON_14";
F3M6: AWAIT,63,{{F3_M6,1}},ALL,,NONE,1;
ACTIVITY,,,XX[16]==Family;
ACTIVITY,,,XX[16]!=Family,"LNCMS17_ASSIGN_6";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_6: GOON,1;
ACTIVITY,,,,"M6";
LNCMS17_ASSIGN_6: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS17_GOON_6";
Machine_7: GOON,1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==1;
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==2,"F2M7";
ACTIVITY,,,ARRAY[ATRIB[1]+1,10]==3,"F3M7";
F1M7: AWAIT,71,{{F1_M7,1}},ALL,,NONE,1;
```

```
ACTIVITY,,,XX[17]==Family;
ACTIVITY,,,XX[17]!=Family,"LNCMS16_ASSIGN_7";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_7: GOON,1;
ACTIVITY;
M7: AWAIT,7,{{Machine_7,1}},ALL,,NONE,1;
ACTIVITY,,,NNRSC(74)==1;
ACTIVITY,,,NNRSC(74)==0,"DDATEC_GOON_30";
AWAIT,74,{{Selected_7,1}},ALL,,NONE,1;
ACTIVITY;
ASSIGN,{{XX[17],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,25,fTime,Family!=ARRAY[XX[9]+1,10],,,"Family Setup M7";
ACTIVITY,26,,Family==ARRAY[XX[9]+1,10],,,"NF&sameF M7";
GOON,1;
ACTIVITY,27,(sTime+pTime)-.001,ATRIB[1]!=XX[9],,,
 "Setup+Process M7";
ACTIVITY,28,pTime-.001,ATRIB[1]==XX[9],,,"Process M7";
ASSIGN,{{XX[9],ATRIB[1]}},1;
ACTIVITY,,,NNQ(7)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_7";
FREE,{{Machine_7,1}},1;
ACTIVITY,,,NNQ(7)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_7";
FREE,{{Machine_7,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_7: GOON,1;
ACTIVITY,,,NRUSE(71)==1&&NRUSE(72)==1&&NRUSE(73)==1;
ACTIVITY,,,NRUSE(71)==1&&NRUSE(72)==1&&NRUSE(73)==0,
 "LNCMS4_FREE_12";
ACTIVITY,,,NRUSE(71)==1&&NRUSE(72)==0&&NRUSE(73)==1,
 "LNCMS4_FREE_39";
ACTIVITY,,,NRUSE(71)==1&&NRUSE(72)==0&&NRUSE(73)==0,
 "LNCMS4_FREE_40";
ACTIVITY,,,NRUSE(71)==0&&NRUSE(72)==1&&NRUSE(73)==1,
 "LNCMS4_FREE_11";
ACTIVITY,,,NRUSE(71)==0&&NRUSE(72)==0&&NRUSE(73)==1,
 "LNCMS4_FREE_41";
ACTIVITY,,,NRUSE(71)==0&&NRUSE(72)==1&&NRUSE(73)==0,
 "LNCMS4_FREE_42";
FREE,{{F1_M7,1},{F2_M7,1},{F3_M7,1}},1;
ACTIVITY;
LNCMS16_GOON_24: GOON,1;
ACTIVITY;
DDATE3_FREE_37: FREE,{{Selected_7,1}},1;
ACTIVITY;
FREE,{{Machine_7,1}},1;
ACTIVITY,,,,"TEST1_RETURNVSN_1";
LNCMS4_FREE_12: FREE,{{F1_M7,1},{F2_M7,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_24";
LNCMS4_FREE_39: FREE,{{F1_M7,1},{F3_M7,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_24";
LNCMS4_FREE_40: FREE,{{F1_M7,1}},1;
ACTIVITY,,,,"LNCMS16_GOON_24";
LNCMS4_FREE_11: FREE,{{F2_M7,1},{F3_M7,1}},1;
```

```
ACTIVITY,,,"LNCMS16_GOON_24";
LNCMS4_FREE_41: FREE,{{F3_M7,1}},1;
ACTIVITY,,,"LNCMS16_GOON_24";
LNCMS4_FREE_42: FREE,{{F2_M7,1}},1;
ACTIVITY,,,"LNCMS16_GOON_24";
DDATEC_GOON_30: GOON,1;
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==1,"F1M7";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==2,"F2M7";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==3,"F3M7";
LNCMS16_ASSIGN_7: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,"LNCMS16_GOON_7";
F2M7: AWAIT,72,{{F2_M7,1}},ALL,,NONE,1;
ACTIVITY,,XX[17]==Family;
ACTIVITY,,XX[17]!=Family,"LNCMS16_ASSIGN_15";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_15: GOON,1;
ACTIVITY,,,"M7";
LNCMS16_ASSIGN_15: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,"LNCMS16_GOON_15";
F3M7: AWAIT,73,{{F3_M7,1}},ALL,,NONE,1;
ACTIVITY,,XX[17]==Family;
ACTIVITY,,XX[17]!=Family,"LNCMS17_ASSIGN_7";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_7: GOON,1;
ACTIVITY,,,"M7";
LNCMS17_ASSIGN_7: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,"LNCMS17_GOON_7";
Machine_8: GOON,1;
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==1;
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==2,"F2M8";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==3,"F3M8";
F1M8: AWAIT,81,{{F1_M8,1}},ALL,,NONE,1;
ACTIVITY,,XX[18]==Family;
ACTIVITY,,XX[18]!=Family,"LNCMS16_ASSIGN_8";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_8: GOON,1;
ACTIVITY;
M8: AWAIT,8,{{Machine_8,1}},ALL,,NONE,1;
ACTIVITY,,NNRSC(84)==1;
ACTIVITY,,NNRSC(84)==0,"DDATEC_GOON_32";
AWAIT,84,{{Selected_8,1}},ALL,,NONE,1;
ACTIVITY;
ASSIGN,{{XX[18],ARRAY[ATRIB[1]+1,10]}},1;
ACTIVITY;
GOON,1;
ACTIVITY,29,fTime,Family!=ARRAY[XX[10]+1,10],,,"Family Setup M8";
ACTIVITY,30,,Family==ARRAY[XX[10]+1,10],,,"NF&sameF M8";
GOON,1;
ACTIVITY,31,(sTime+pTime)-.001,ATRIB[1]!=XX[10],,,
 "Setup+Process  M8";
ACTIVITY,32,pTime-.001,ATRIB[1]==XX[10],,,"Process M8";
ASSIGN,{{XX[10],ATRIB[1]}},1;
ACTIVITY,,NNQ(8)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_8";
```

```
FREE,{{Machine_8,1}},1;
ACTIVITY,,NNQ(8)>=1;
ACTIVITY,,.001,,"LNCMS8_GOON_8";
FREE,{{Machine_8,1}},1;
ACTIVITY,,.001;
LNCMS8_GOON_8: GOON,1;
ACTIVITY,,NRUSE(81)==1&&NRUSE(82)==1&&NRUSE(83)==1;
ACTIVITY,,NRUSE(81)==1&&NRUSE(82)==1&&NRUSE(83)==0,
 "LNCMS4_FREE_14";
ACTIVITY,,NRUSE(81)==1&&NRUSE(82)==0&&NRUSE(83)==1,
 "LNCMS4_FREE_43";
ACTIVITY,,NRUSE(81)==1&&NRUSE(82)==0&&NRUSE(83)==0,
 "LNCMS4_FREE_44";
ACTIVITY,,NRUSE(81)==0&&NRUSE(82)==1&&NRUSE(83)==1,
 "LNCMS4_FREE_13";
ACTIVITY,,NRUSE(81)==0&&NRUSE(82)==0&&NRUSE(83)==1,
 "LNCMS4_FREE_45";
ACTIVITY,,NRUSE(81)==0&&NRUSE(82)==1&&NRUSE(83)==0,
 "LNCMS4_FREE_46";
FREE,{{F1_M8,1},{F2_M8,1},{F3_M8,1}},1;
ACTIVITY;
LNCMS16_GOON_25: GOON,1;
ACTIVITY;
DDATE3_FREE_43: FREE,{{Selected_8,1}},1;
ACTIVITY;
FREE,{{Machine_8,1}},1;
ACTIVITY,,,"TEST1_RETURNVSN_1";
LNCMS4_FREE_14: FREE,{{F1_M8,1},{F2_M8,1}},1;
ACTIVITY,,,"LNCMS16_GOON_25";
LNCMS4_FREE_43: FREE,{{F1_M8,1},{F3_M8,1}},1;
ACTIVITY,,,"LNCMS16_GOON_25";
LNCMS4_FREE_44: FREE,{{F1_M8,1}},1;
ACTIVITY,,,"LNCMS16_GOON_25";
LNCMS4_FREE_13: FREE,{{F2_M8,1},{F3_M8,1}},1;
ACTIVITY,,,"LNCMS16_GOON_25";
LNCMS4_FREE_45: FREE,{{F3_M8,1}},1;
ACTIVITY,,,"LNCMS16_GOON_25";
LNCMS4_FREE_46: FREE,{{F2_M8,1}},1;
ACTIVITY,,,"LNCMS16_GOON_25";
DDATEC_GOON_32: GOON,1;
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==1,"F1M8";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==2,"F2M8";
ACTIVITY,,ARRAY[ATRIB[1]+1,10]==3,"F3M8";
LNCMS16_ASSIGN_8: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,"LNCMS16_GOON_8";
F2M8: AWAIT,82,{{F2_M8,1}},ALL,,NONE,1;
ACTIVITY,,XX[18]==Family;
ACTIVITY,,XX[18]!=Family,"LNCMS16_ASSIGN_16";
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS16_GOON_16: GOON,1;
ACTIVITY,,,"M8";
LNCMS16_ASSIGN_16: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,"LNCMS16_GOON_16";
F3M8: AWAIT,83,{{F3_M8,1}},ALL,,NONE,1;
ACTIVITY,,XX[18]==Family;
ACTIVITY,,XX[18]!=Family,"LNCMS17_ASSIGN_8";
```

```
ASSIGN,{{ATRIB[7],0}},1;
ACTIVITY;
LNCMS17_GOON_8: GOON,1;
ACTIVITY,,,,"M8";
LNCMS17_ASSIGN_8: ASSIGN,{{ATRIB[7],1}},1;
ACTIVITY,,,,"LNCMS17_GOON_8";

DDCMS successfully read


Translated network file DDCMS.TRN successfully written
```

Multiple run summary report

** AweSim! MULTIPLE RUN SUMMARY REPORT **
Thu Sep 04 00:19:38 2003

Simulation Project : Thesis
Modeler : Erika
Date : September/24/2003
Scenario: DDCMS
Number of runs 5

** OBSERVED STATISTICS for scenario DDCMS **

Label	Mean Value	Standard Deviation	Standard Error	Minimum Average Value	Maximum Average Value
TIS Batch1	52.205	5.262	2.353	46.433	59.418
Tardiness Batch1	0.000	0.000	0.000	0.000	0.000
Earliness Batch1	140.520	6.381	2.854	135.525	148.313
TIS Batch2	148.053	3.413	1.526	142.830	152.026
Tardiness Batch2	0.475	0.678	0.303	0.000	1.594
Earliness Batch2	45.147	7.424	3.320	35.718	56.163
TIS Batch3	115.027	40.908	18.295	56.692	162.458
Tardiness Batch3	16.009	11.550	5.165	1.104	27.252
Earliness Batch3	96.874	29.174	13.047	61.634	140.733
TIS Batch4	190.266	22.571	10.094	169.664	227.621
Tardiness Batch4	35.212	18.002	8.051	18.475	66.027
Earliness Batch4	40.792	7.585	3.392	29.889	49.553
TIS Batch5	142.910	63.856	28.557	94.569	248.101
Tardiness Batch5	35.094	23.493	10.506	11.925	71.494
Earliness Batch5	88.963	39.415	17.627	20.616	115.706
TIS Batch6	221.875	36.165	16.173	184.607	277.071
Tardiness Batch6	57.354	24.996	11.179	32.732	94.990
Earliness Batch6	35.240	10.826	4.842	19.033	49.202
TIS Batch7	230.033	62.064	27.756	174.408	330.184
Tardiness Batch7	93.963	60.936	27.251	32.754	191.276
Earliness Batch7	66.794	23.984	10.726	46.797	107.257
TIS Batch8	258.266	83.218	37.216	151.521	376.484
Tardiness Batch8	86.428	68.691	30.720	0.000	187.329
Earliness Batch8	32.054	16.304	7.291	9.704	51.304
TIS Batch9	282.152	92.779	41.492	140.991	393.005
Tardiness Batch9	116.764	68.140	30.473	23.855	208.064
Earliness Batch9	37.526	29.636	13.254	15.441	85.230
TIS Batch10	323.675	29.417	13.155	294.967	369.937
Tardiness Batch1	146.038	35.941	16.074	111.415	201.244
Earliness Batch1	23.387	9.818	4.391	12.499	36.792

** FILE STATISTICS for scenario DDCMS **

File Number	Label or Input Location	Average Length	Standard Deviation	Standard Error	Maximum Average Length
9	RES. RESA	0.000	0.000	0.000	0.000

File Number	Average Wait Time
9	0.000

** RESOURCE STATISTICS for scenario DDCMS **

Resource Number	Resource Label	Average Util.	Standard Deviation	Standard Error	Average Available
9	RESA	0.000	0.000	0.000	1.000

Resource Number	Maximum Average Utilization	Minimum Average Utilization	Maximum Average Available	Minimum Average Available
9	0.000	0.000	1.000	1.000

** AweSim! MULTIPLE RUN SUMMARY REPORT **

Subnetwork : DDCMS
Instance : (null)

** FILE STATISTICS for DDCMS (null) **

File Number	Label or Input Location	Average Length	Standard Deviation	Standard Error	Maximum Average Length
1	RES. MACHINE_1	0.601	0.211	0.094	0.935
2	RES. MACHINE_2	1.156	0.345	0.154	1.516
3	RES. MACHINE_3	0.122	0.107	0.048	0.258
4	RES. MACHINE_4	1.044	0.142	0.063	1.201
5	RES. MACHINE_5	0.000	0.000	0.000	0.000
6	RES. MACHINE_6	8.235	9.746	4.359	20.784
7	RES. MACHINE_7	0.000	0.000	0.000	0.000
8	RES. MACHINE_8	0.000	0.000	0.000	0.000
11	RES. F1_M1	29.158	21.053	9.415	67.251
12	RES. F2_M1	22.902	6.888	3.081	33.383
13	RES. F3_M1	0.000	0.000	0.000	0.000
14	RES. SELECTED_1	0.000	0.000	0.000	0.000
21	RES. F1_M2	7.923	1.979	0.885	10.711
22	RES. F2_M2	25.849	11.984	5.360	45.449
23	RES. F3_M2	9.722	4.455	1.992	15.786
24	RES. SELECTED_2	0.000	0.000	0.000	0.000
31	RES. F1_M3	1.149	0.922	0.412	2.982
32	RES. F2_M3	2.273	1.542	0.690	3.783
33	RES. F3_M3	0.569	0.296	0.132	0.803
34	RES. SELECTED_3	0.000	0.000	0.000	0.000
41	RES. F1_M4	100.837	31.615	14.139	139.996
42	RES. F2_M4	36.766	28.660	12.817	92.339
43	RES. F3_M4	0.147	0.136	0.061	0.374
44	RES. SELECTED_4	0.000	0.000	0.000	0.000
51	RES. F1_M5	0.000	0.000	0.000	0.000
52	RES. F2_M5	24.282	18.640	8.336	57.503
53	RES. F3_M5	0.000	0.000	0.000	0.000
54	RES. SELECTED_5	0.000	0.000	0.000	0.000
61	RES. F1_M6	0.604	0.804	0.360	2.007
62	RES. F2_M6	18.995	16.898	7.557	51.265
63	RES. F3_M6	18.515	16.906	7.561	50.988
64	RES. SELECTED_6	0.000	0.000	0.000	0.000
71	RES. F1_M7	0.000	0.000	0.000	0.000
72	RES. F2_M7	0.000	0.000	0.000	0.000
73	RES. F3_M7	16.899	7.029	3.143	26.012
74	RES. SELECTED_7	0.000	0.000	0.000	0.000
81	RES. F1_M8	0.000	0.000	0.000	0.000
82	RES. F2_M8	0.000	0.000	0.000	0.000
83	RES. F3_M8	13.645	4.975	2.225	21.468
84	RES. SELECTED_8	0.000	0.000	0.000	0.000

File Number	Average Wait Time
1	1.718
2	1.457
3	0.317
4	1.494
5	0.000
6	16.341
7	0.000
8	0.000
11	127.085
12	109.573
13	0.000
14	0.000
21	25.929
22	83.670
23	37.547
24	0.000
31	7.594
32	19.684
33	12.374
34	0.000
41	200.338
42	98.148
43	1.390
44	0.000
51	0.000
52	79.544
53	0.000
54	0.000
61	24.896
62	77.473
63	64.660
64	0.000
71	0.000
72	0.000
73	55.331
74	0.000
81	0.000
82	0.000
83	48.304
84	0.000

** ACTIVITY STATISTICS for DDCMS (null) **

Activity Number	Label or Input Location	Average Utilization	Standard Deviation	Standard Error	Maximum Average Utilization
1	Family Setup M1	0.002	0.001	0.000	0.003
2	NF&sameF M1	0.000	0.000	0.000	0.000
3	Setup+Process M1	0.053	0.004	0.002	0.058
4	Process M1	0.832	0.075	0.034	0.946
5	Family Setup M2	0.017	0.005	0.002	0.026
6	NF&sameF M2	0.000	0.000	0.000	0.000
7	Setup+Process M2	0.284	0.082	0.037	0.421
8	Process M2	0.620	0.117	0.052	0.797
9	Family Setup M3	0.006	0.004	0.002	0.012
10	NF&sameF M3	0.000	0.000	0.000	0.000
11	Setup+Process M3	0.061	0.031	0.014	0.102
12	Process M3	0.380	0.088	0.039	0.525
13	Family Setup M4	0.035	0.007	0.003	0.043
14	NF&sameF M4	0.000	0.000	0.000	0.000
15	Setup+Process M4	0.276	0.060	0.027	0.336
16	Process M4	0.660	0.022	0.010	0.683
17	Family Setup M5	0.001	0.000	0.000	0.001
18	NF&sameF M5	0.000	0.000	0.000	0.000
19	Setup+Process M5	0.422	0.064	0.029	0.481
20	Process M5	0.446	0.059	0.026	0.515
21	Family Setup M6	0.025	0.024	0.011	0.060
22	NF&sameF M6	0.000	0.000	0.000	0.000
23	Setup+Process M6	0.205	0.184	0.082	0.473
24	Process M6	0.634	0.130	0.058	0.847
25	Family Setup M7	0.001	0.000	0.000	0.001
26	NF&sameF M7	0.000	0.000	0.000	0.000
27	Setup+Process M7	0.129	0.079	0.035	0.281
28	Process M7	0.668	0.087	0.039	0.812
29	Family Setup M8	0.001	0.000	0.000	0.001
30	NF&sameF M8	0.000	0.000	0.000	0.000
31	Setup+Process M8	0.305	0.099	0.044	0.466
32	Process M8	0.546	0.091	0.041	0.693

Activity Number	Minimum Average Utilization
1	0.001
2	0.000
3	0.047
4	0.727
5	0.011
6	0.000
7	0.189
8	0.455
9	0.001
10	0.000
11	0.019
12	0.273
13	0.021

14	0.000
15	0.163
16	0.625
17	0.001
18	0.000
19	0.298
20	0.351
21	0.005
22	0.000
23	0.045
24	0.467
25	0.001
26	0.000
27	0.051
28	0.597
29	0.001
30	0.000
31	0.181
32	0.418

** RESOURCE STATISTICS for DDCMS (null) **

Resource Number	Resource Label	Average Util.	Standard Deviation	Standard Error	Average Available
1	MACHINE_1	0.887	0.078	0.035	0.113
2	MACHINE_2	0.922	0.075	0.034	0.078
3	MACHINE_3	0.447	0.120	0.054	0.553
4	MACHINE_4	0.971	0.058	0.026	0.029
5	MACHINE_5	0.870	0.058	0.026	0.130
6	MACHINE_6	0.865	0.142	0.063	0.135
7	MACHINE_7	0.798	0.104	0.047	0.202
8	MACHINE_8	0.853	0.077	0.034	0.147
11	F1_M1	0.764	0.137	0.061	0.236
12	F2_M1	0.724	0.163	0.073	0.276
13	F3_M1	0.000	0.000	0.000	1.000
14	SELECTED_1	0.887	0.078	0.035	0.113
21	F1_M2	0.696	0.092	0.041	0.304
22	F2_M2	0.734	0.169	0.076	0.266
23	F3_M2	0.649	0.194	0.087	0.351
24	SELECTED_2	0.922	0.075	0.034	0.078
31	F1_M3	0.344	0.127	0.057	0.656
32	F2_M3	0.125	0.081	0.036	0.875
33	F3_M3	0.101	0.056	0.025	0.899
34	SELECTED_3	0.447	0.120	0.054	0.553
41	F1_M4	0.962	0.074	0.033	0.038
42	F2_M4	0.844	0.127	0.057	0.156
43	F3_M4	0.209	0.020	0.009	0.791
44	SELECTED_4	0.971	0.058	0.026	0.029
51	F1_M5	0.000	0.000	0.000	1.000
52	F2_M5	0.870	0.058	0.026	0.130
53	F3_M5	0.000	0.000	0.000	1.000
54	SELECTED_5	0.870	0.058	0.026	0.130
61	F1_M6	0.108	0.133	0.059	0.892
62	F2_M6	0.724	0.212	0.095	0.276
63	F3_M6	0.584	0.240	0.107	0.416
64	SELECTED_6	0.865	0.142	0.063	0.135
71	F1_M7	0.000	0.000	0.000	1.000
72	F2_M7	0.000	0.000	0.000	1.000
73	F3_M7	0.798	0.104	0.047	0.202
74	SELECTED_7	0.798	0.104	0.047	0.202
81	F1_M8	0.000	0.000	0.000	1.000
82	F2_M8	0.000	0.000	0.000	1.000
83	F3_M8	0.853	0.077	0.034	0.147
84	SELECTED_8	0.853	0.077	0.034	0.147

Resource Number	Maximum Average Utilization	Minimum Average Utilization	Maximum Average Available	Minimum Average Available
1	1.000	0.776	0.224	0.000
2	1.000	0.797	0.202	0.000
3	0.634	0.294	0.706	0.366
4	1.000	0.854	0.146	0.000

5	0.948	0.803	0.197	0.052
6	1.000	0.632	0.368	0.000
7	0.925	0.654	0.346	0.075
8	0.917	0.708	0.292	0.083
11	0.935	0.552	0.448	0.065
12	1.000	0.545	0.455	0.000
13	0.000	0.000	1.000	1.000
14	1.000	0.776	0.224	0.000
21	0.816	0.547	0.453	0.184
22	1.000	0.503	0.497	0.000
23	0.826	0.401	0.599	0.174
24	1.000	0.797	0.202	0.000
31	0.585	0.235	0.765	0.415
32	0.224	0.021	0.979	0.776
33	0.159	0.000	1.000	0.841
34	0.634	0.294	0.706	0.366
41	1.000	0.814	0.186	0.000
42	1.000	0.627	0.373	0.000
43	0.238	0.177	0.823	0.762
44	1.000	0.854	0.146	0.000
51	0.000	0.000	1.000	1.000
52	0.948	0.803	0.197	0.052
53	0.000	0.000	1.000	1.000
54	0.948	0.803	0.197	0.052
61	0.285	0.000	1.000	0.715
62	0.998	0.399	0.601	0.002
63	1.000	0.363	0.637	0.000
64	1.000	0.632	0.368	0.000
71	0.000	0.000	1.000	1.000
72	0.000	0.000	1.000	1.000
73	0.925	0.654	0.346	0.075
74	0.925	0.654	0.346	0.075
81	0.000	0.000	1.000	1.000
82	0.000	0.000	1.000	1.000
83	0.917	0.708	0.292	0.083
84	0.917	0.708	0.292	0.083