Adaptive Scheduling and Tool Flow Control in Automated Manufacturing Systems

Jie Chen

Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirement for the degree of

Doctor of Philosophy
in
Industrial and Systems Engineering

F. Frank Chen, Chair
Ralph D. Badinelli
Michael P. Deisenroth
C. Patrick Koelling
John P. Shewchuk

April 2, 2003
Blacksburg, Virginia

Keywords: Dynamic Scheduling, Flexible Tooling, Routing Flexibility, and Machine Availability

Adaptive Scheduling and Tool Flow Control in Automated Manufacturing Systems

Jie Chen

(ABSTRACT)

The recent manufacturing environment is characterized as having diverse products due to mass customization, short production lead-time, and unstable customer demand. Today, the need for flexibility, quick responsiveness, and robustness to system uncertainties in production scheduling decisions has increased significantly. In traditional job shops, tooling is usually assumed as a fixed resource. However, when tooling resource is shared among different machines, a greater product variety, routing flexibility with a smaller tool inventory can be realized. Such a strategy is usually enabled by an automatic tool changing mechanism and tool delivery system to reduce the time for tooling setup, hence allows parts to be processed in small batches. In this research, a dynamic scheduling problem under flexible tooling resource constraints is studied. An integrated approach is proposed to allow two levels of hierarchical, dynamic decision making for job scheduling and tool flow control in Automated Manufacturing Systems. It decomposes the overall problem into a series of static sub-problems for each scheduling window, handles random disruptions by updating job ready time, completion time, and machine status on a rolling horizon basis, and considers the machine availability explicitly in generating schedules.

Two types of manufacturing system models are used in simulation studies to test the effectiveness of the proposed dynamic scheduling approach. First, hypothetical models are generated using some generic shop flow structures (e.g. flexible flow shops, job shops, and single-stage systems) and configurations. They are tested to provide the empirical evidence about how well the proposed approach performs for the general automated manufacturing systems where parts have alternative routings. Second, a model based on a real industrial flexible manufacturing system was used to test the effectiveness of the proposed approach when machine types, part routing, tooling, and other production parameters closely mimic to the real flexible

manufacturing operations. The study results show that the proposed scheduling approach significantly outperforms other dispatching heuristics, including Cost Over Time (COVERT), Apparent Tardiness Cost (ATC), and Bottleneck Dynamics (BD), on due-date related performance measures under both types of manufacturing systems models. It is also found that the performance difference between the proposed scheduling approach and other heuristics tend to become more significant when the number of machines is increased. The more operation steps a system has, the better the proposed method performs, relative to the other heuristics. This research also investigates in what conditions (e.g. the number of machines, the number of operation steps, and shop load conditions) the proposed approach works the best, and how the performance of this proposed approach changes when these conditions change.

When tooling resource is shared, parts can be routed to machines that do not have all the required tools. This may result in higher routing flexibility. However, research work to date in sharing of tooling resources often places more emphasis on the real-time control and manipulation of tools, and pays less attention to the loading of machines and initial tool allocation at the planning stage. In this research, a machine-loading model with shared tools is proposed to maximize routing flexibility while maintaining minimum resident tools. The performance of the proposed loading heuristic is compared to that of a random loading method using hypothetically generated single stage system models. The study result indicates that better system performances can be obtained by taking into account the resident tooling ratio in assigning part types and allocating tools to machines at the initial planning stage.

Acknowledgement

First, I would like to thank my advisor Dr. F. Frank Chen for his prudent guidance and invaluable advice in my graduate studies. I greatly appreciate his support and encouragement in conducting of this dissertation research and throughout my graduate studies at Virginia Tech.

I would also like to thank my committee members for allowing me the opportunity to pursue this research, and being patient in guiding me in completion of this dissertation. In particular I thank Dr. Ralph D. Badinelli for his insightful comments and important suggestions during many discussions of my research methodology. I thank Dr. John P. Shewchuk for his helpful advice in improving the experimental studies, Dr. C. Patrick Koelling for his sound advise in simulation and statistical analyses, and Dr. Michael P. Deisenroth for offering constructive suggestions and helps.

Finally, I must thank my wife, Jin Zhang, for her great support and patient throughout my studies in the graduate school. Without her help, this dissertation would not have been completed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Dynamic Scheduling under Flexible Tooling

In general, scheduling involves decisions of allocating resources to tasks over time, and optimizing one or more objectives. Scheduling models can be either deterministic or stochastic (Pinedo 1995). Deterministic models assume that all job data are known exactly in advance. In stochastic models, not all job data but their distributions are known. Static scheduling problems assume that a list of n jobs are available at the beginning of the scheduling period, while the dynamic scheduling problems deal with an ongoing situation, in which new jobs are continuously added to the system.

Machine scheduling plays an important role to the overall operational control of many manufacturing systems by efficiently allocating various resources to competing activities. In the past, many researchers concentrated on analytical modeling of the deterministic scheduling problem in search of optimal solutions. However, the system complexities and constraints on real-time responses that arise from increasing global competition have forced industry to pursue good production schedules that have greater adaptability and flexibility, instead of optimal schedules.

In batch production systems, the limited tool magazine capacity constrains the number of tools available at one time on a machine for continuous processing of parts (Chen et al.1993). In traditional manufacturing systems, tooling is usually considered as a secondary issues and the tooling resource is usually allocated at the pre-release planning stage statically before the scheduling of jobs. Consequently, early studies of scheduling under tooling constraints generally addressed this problem by using the decomposition approach such as loading then sequencing. In scheduling, tooling resource decided in the loading stage is usually assumed as a fixed constraint in most early research.

Tool sharing (or flexible tooling) is considered as an effective method to achieve reduction in tool inventory and associated tooling costs. After the tool completes its operation on a part, it is returned to the tool magazine where it remains idle until another operation requiring the same tool is initiated by the system. In such a case, utilization of tools can be very low. These

idle tools may be dynamically shared among machines, which increases their utilization while reducing the number of tools in the system (Kashyap and Khator, 1996).

When tools are assigned to operations dynamically and different jobs compete for the same type of tools, dynamic tool scheduling, which involves decisions of sequencing tooling requests and assigning tools to different operations, also becomes an important component of overall operational control of such systems, in addition to dispatching and machine routing decision. Grieco et al. (2001) pointed out that one of the major reasons that limit the current application of tool handling systems is the lack of efficient algorithm to control tool flow, and to coordinate it with part flow to minimize unnecessary delays.

1.2 Statement of the Research Problem

In a dynamic job shop environment, each job can have several operations with a set of tools used in each operation and alternative routings (i.e. each operation can be processed at alternative workstations). Jobs of varying part types may arrive at different time periods to be processed on machines for a given amount of time with shop floor disturbances, such as machine random failures. However, when tooling resources are being shared among different machines, a greater product variety, routing flexibility with a smaller tool inventory can be realized. Such a strategy is usually enabled by an automatic tool changing mechanism and tool delivery system to reduce the time for tooling setup, hence allows parts to be processed in small batches. The dynamic decisions for part routing and dispatching, and tool assignments are important components of the overall operational control of such systems. Dynamic scheduling decisions must be computationally efficient and to be able to handle system uncertainties such as jobs arriving with tight due-date and machine breakdown.

Myopic dispatching rules decompose the part/machine (or operation/tool) assignment into smaller sub-problems, namely machine routing (or tool selection) and part dispatching (or operation tool request prioritizing). Resources and tasks are assigned sequentially while the interaction between sub-problems due to jobs (or operations) and machines (or tools) becoming available at different time periods are not considered. Furthermore, frequent rescheduling to react to disruptions such as machine breakdowns can make the behavior of the system hard to predict, hence reduce the effectiveness of the dynamic scheduling.

Therefore, there is a need for a shop floor scheduling system to be able to generate dynamic job schedules and coordinate with the flow of the flexible tooling resource, while accommodating shop floor disturbances such as machine failures and rush orders.

When tooling resources are shared, parts can be routed to machines that do not have all the required tools, resulting in higher routing flexibility. Most research to date on dynamic tool sharing often ignores the tool loading decisions and relies mainly on the real-time control of tools during the scheduling stage. However, this may result in congested tool handling systems especially in the Single-Stage Systems where tool-sharing activities are more frequent and all operations of a part are performed on a single machine. Consequently, the initial tool allocation

decision at the machine loading stage should explore the trade-off between the routing flexibility and the tool handling traffic. It needs to be closely coordinated with the real-time assignment of tools at the scheduling stage.

1.3 Research Objective, Approach, and Hypothesis

The objective of this research is to develop an integrated approach to coordinate part/machine scheduling and operation/tool assignment. Specifically, the framework proposed in this research (1) allows two levels (cell and machine levels) of hierarchical, dynamic decision making for resource and task assignment on a rolling horizon basis, (2) formulates the assignment decision at each level as a minimum-cost flow problem during each short-term window and solves it by an efficient network optimization algorithm, (3) takes both the criticalities of jobs and machine reliabilities into account in decision making, (4) implements only the decision for some period of each window, rescheduling resources and tasks for the remaining periods with a certain degree of short-term look ahead, and (5) handles uncertainty of machine down time by estimating the operation completion time based on the transient analysis of machine availability for a two-state Markov process.

One hypothesis tested in this research is that the performance of the proposed dynamic scheduling approach is superior to that of different dispatching heuristics when the machine availability is explicitly considered in generating adaptive schedules. The proposed approach is tested together with other dispatching heuristics, including Cost Over Time (COVERT), Apparent Tardiness Cost (ATC), and Bottleneck Dynamics (BD) on two types of manufacturing system models. First, hypothetical models, generated using some generic shop flow structures (e.g. flexible flow shops, job shops, and single-stage systems) and configurations. They are studied to provide empirical results about the performance of the proposed approach under more general shop floor configurations. Second, a model based on a real industrial flexible manufacturing system is experimented to see how it performs when machine types, part routing, tooling, and other parameters are representative of flexible manufacturing settings.

Furthermore, one of the objectives of this research is to investigate conditions (or factors) under which the proposed approach works the best, and how the performance of the proposed approach changes when these conditions change. Examples of these factors include the number of machines, the number of operation steps, and shop load conditions.

When tooling resource is shared, parts can be routed to machines that do not have all the required tools, resulting in higher routing flexibility. Research to date in scheduling with shared tooling resources often places more emphasis on the real-time control and manipulation of tools, and pay less attention on the loading of machines and initial tool allocation at the planning stage. Another research hypothesis tested in this research is that better system performances can be obtained by considering the resident tool ratio in assigning part types and allocating tools to machines initially. A machine-loading model with shared tools is proposed to maximize routing flexibility while maintaining a minimum resident tool ratio. The proposed loading heuristic is compared to a random loading method under hypothetically generated single stage system models.

1.4 Performance Measures of Interests

The major objective of shop floor control under a dynamic order environment is to satisfy customer demand of parts with varying due dates. Since the part orders can be received randomly over time, and may not be available at the beginning of the planning horizon, a make-span based decision is not adequate in such a dynamic system. Furthermore, since a make-to-order environment is considered in this study, the total inventory is expected to be much lower than that of make-to-stock systems. Therefore, job earliness will not be major concerns for the system in this study. Instead, mean flow time, mean job tardiness, and percentage of tardy job, will be major indicators of the system performance. The definitions of these performance measures are described as follows.

<u>Mean Flow Time</u>: $\sum F_i / n$ is defined as the average amount of time a job spends in the system, from its arrival to its completion. ($F_i = C_i - r_i$, C_i is the completion time of job i, r_i is the arrival time of job i, and n is the number of jobs completed during a specific scheduling period.)

<u>Mean Tardiness</u>: $\sum T_i / n$ is the average tardiness of all jobs. ($T_i = max\ \{0,\ C_i - d_i\}$ is the tardiness of a job i. d_i is the due date of part i.)

<u>Percentage of Tardy Jobs</u>: ($\sum \delta(T_i) / n$)*100% is the percentage of jobs that are tardy.

Where $\delta(T_i) = \begin{cases} 1, & if\ T_i > 0 \\ 0, & otherwise \end{cases}$

1.5 Organization of the Dissertation

This dissertation is divided into five chapters. Chapter 1 has provided an overview of the problem studied, research objectives and approaches, and performance measures of interests. Chapter 2 reviews and summarizes the relevant literature in the area of machine loading, tool allocation, static and dynamic scheduling under tooling resource constraints, and routing flexibility.

Chapter 3 describes the proposed dynamic scheduling approach. It begins with the overview of the machine control scheme and the rolling horizon decomposition. Then it describes how to formulate minimum-cost flow models in the cell-level and machine-level sub-problem heuristics. Furthermore, it illustrates the approach used to handle the floor uncertainties by generating adaptive schedules. The reminder of Chapter 3 presents a machine-loading method in the tool-shared environment. An integer-programming optimization model and a heuristic solution procedure are proposed.

Chapter 4 validates the performance of the proposed dynamic scheduling approach by comparing it with other approaches based on various dispatching heuristics under two types of manufacturing system models in simulation studies. The effect of rolling horizon length is also investigated. The performance of the proposed loading heuristic is compared to a random loading method under hypothetically generated single-stage system models. The chapter also describes in detail the experimental factors and models, statistical analysis procedures, and simulation input data employed in these studies.

Finally Chapter 5 summarizes the research results and contributions, and provides recommendations for future research.

Chapter 2 Review of Literature

This chapter reviews the relevant literature in the area of tooling related problems, setup planning, static scheduling under tooling constraints, dynamic scheduling, dispatching heuristics and other related issues to provide a foundation for this research.

2.1 Tooling Problems and Setup Planning

There are generally two categories of tooling related problems, (1) the tool planning problems, and (2) the tool control problems. Tool planning problems include design of tool handling systems, tool allocation strategies, and tool requirement planning to ensure that the appropriate tools are available with right quantities. Tool control problems include tool flow control, information system support, and tool monitoring to coordinate the tool transfer between machines and tool storage areas. In traditional manufacturing systems, tooling is usually considered as a secondary issue, and tools are allocated to machines statically before scheduling jobs. Planning problems of tool requirement, tool loading and static allocation strategies are often addressed in this environment. Since the performance of a system is often affected by the availability of tools, job scheduling under tooling constraints becomes an important research problem, and has gained more attentions from researchers in recent years.

Sections 2.1.1 gives some background information about static tool allocation strategies. Section 2.1.2 reviews the definitions of flexibility, and routing flexibility. In section 2.1.3, existing literature in the area of machine loading and grouping is reviewed.

2.1.1 Tool Allocation Strategies

Tool allocation involves decisions of when and which tools are to be brought to or to be taken from machine tool magazines considering the part-mix batch size, process plan, tool magazine capacity, and tooling requirements (Mason 1986). Various tool-allocation strategies including bulk exchange, tool sharing, migration at the completion of a part type, and resident tooling (Hankins and Rovito 1984) are described as following.

Bulk exchange This strategy provides a complete dedicated set of tools for each different part type. At the completion of a part type, all tools are returned to the tool room, to be replaced with a different dedicated set of tools for the next part type. With bulk exchange, easy tool control is achieved at the expense of excessive tooling inventory. This strategy is good for applications that involve high volume and low variety of parts. Bulk exchange is also referred as Tool Batching (Amoako-Gyampah and Meredith 1996).

Tool sharing *(in* a frozen production window). This is an improvement over the bulk exchange strategy with respect to tool inventory. Commonly used tools are shared among the various part types being manufactured within a fixed production window. Tools are loaded only once at the beginning of the production window so that all tools to be used must be accommodated in the tool matrix at the machine. At the end of the production window, a new set of tools for the next production window is loaded. The part mix rather than the length of the production window dictates the tool matrix capacity required. This strategy can be considered as a bulk exchange for a production window (not for a part type).

Migration at the completion of a **part type** - It is also named as Flexible Tooling by Amoako-Gyampah (1994) and Amoako-Gyampah and Meredith (1996). This strategy uses more shared tools than the previous strategies. There is no complete changeover of tools. When the production of a particular part type is completed at a machine, only those tools that are unique to the part type are removed from the tool magazine, allowing new tools for other parts to be loaded for the next machining cycle. While this strategy allows a further reduction in tool inventory through sharing of common tools between part types, it requires the application of sophisticated decision logic to determine, for example, which tools should be removed from the tool magazine and which should stay for new operations.

Resident tooling This strategy identifies the high usage tools for the entire production mix and forms clusters of different combinations of tools representing similar processing requirements of parts with the objective of keeping them residing at various machines. Tool changes occur only when a particular tool reaches the end of its predetermined life. This strategy

does not minimize tool inventory. However, it provides routing flexibility and therefore allows quick response to changes in the production schedule. Furthermore, ease of tool condition monitoring and easy identification of tools for replacement are also the benefits of this policy.

In order to provide greater product varieties and quick responsiveness, a recent trend in flexible manufacturing systems is to use a flexible tooling strategy. With flexible tool strategy (or tool-sharing), tools can be borrowed from other machines, then the configuration of a tool magazine is evolving continuously, and tool flows will change based on the processing requirements of parts at different machines. Such a strategy is often enabled by an automatic tool changing mechanism and a fast tool delivery system to reduce the time for tooling setup, hence allows parts to be processed in small batches. The scheduling problem under flexible tooling strategy will involve dynamic decisions about scheduling of both jobs and tool requests over time. See section 2.3.4 for the existing literature in this area.

2.1.2 Routing Flexibility

Routing flexibility is usually addressed in the system and process design stage. However, when flexible tooling strategy is used in the manufacturing system, routing flexibility can be further improved since jobs can be routed to machines that do not have all the required tools at the beginning of the operation. Therefore, how to determine the alternative routing for a part type is also an important operational planning decision when tools are assigned to machines in a tool-shared environment. This section gives a brief review of the definition of flexibility.

Manufacturing flexibility can be defined as being able to reconfigure manufacturing resources so as to efficiently produce different products of acceptable quality (Sethi and Sethi 1990). Flexibility is required for a manufacturing system to cope with both internal and external environmental uncertainties. Especially, routing flexibility allows for easier scheduling of jobs in real-time and increases the machine availability and utilization. This section gives a brief review of definitions and measures for routing flexibility.

Routing flexibility is a capability of system to produce a given part mix on alternate machines, or with alternate resources (Carter 1986). This definition is consistent with Browne, et al. (1984), Gerwin (1982), Chung and Chen (1989), Sethi and Sethi (1990), and Chandra and

Tombak (1992). The Routing flexibilities mentioned above are concerned with the operational flexibility dimension (Shewchuk and Moodie 1998), which is the ability to utilize the elements of a manufacturing system in alternative ways, at a time.

An obvious measure of routing flexibility is the number (n) of possible routes in which a part can be processed as it is suggested in Chatterjee at al. (1984) and Chung and Chen (1989). The above routing flexibility measure increases as the number of machines capable of processing the part increases. Yao (1985) presented a measure, which takes the machine reliability into consideration. Another factor neglected in the above measure, is the machine capacities, as machines with differing capacities are weighted equally under the measure. Chandra and Tombak (1992) proposed a suitable measure that for measuring routing flexibility, which takes both of these factors into account.

Most previous research efforts have shown that having alternative routing can significantly reduce parts waiting time, better balance the workload of machines, hence increases the system throughput.

2.1.3 Pre-release Planning

Due to the complex nature of operational decision problems associated with manufacturing systems, a hierarchical approach has often been applied to address the interrelated problems in early research of FMS planning decisions.

Pre-release planning problems involve setup decisions of a manufacturing system, which are made and implemented before the system begins to manufacture parts. Stecke (1983) identifies five production-planning problems to be addressed in the FMS environment, which are as follows:

(1) Part selection problem: The part selection problem determines a subset of parts out of a large set (either forecasted production demand or customer orders) to be produced simultaneously over a pre-defined planning horizon.

(2) Machine grouping problem: The machine-grouping problem partitions the machines of similar types into identically tooled machine groups. Each machine in a particular group is then able to perform the same operations.

(3) Production ratio problem: The production ratio problem determines the part mix ratio at which the part types selected in (1) should be produced during a planning horizon.

(4) Resource allocation problem: The resource allocation problem allocates the required number of pallets and fixtures to the selected part types based on the part ratio.

(5) Machine loading problem: The loading problem allocates all the operations of the selected parts and required tools to the machines. Since the operation assignment is dependent on the tool assignment, the tool allocation is usually determined simultaneously with operation assignment.

The pre-release planning problems are addressed and implemented periodically before the start of production of a new part mix (or planning horizon). Once the planning problems are solved, the types and ratio of parts to be produced are determined, the machines are grouped, and tools are assigned to machines. A part can then travel through the system by visiting the machines, which hold the tools that are required for a series of operations on the part. Objectives that are important for loading problem include minimizing work-in-process inventory, cost of tools, or variable machining cost. There are a number of financial and technical objectives in the machine-loading problem reported in the literature, such as minimization of the relevant cost, minimization of the number of tools, and balance of machine workload (Tempelmeier and Kuhn 1993).

2.1.4 Machine Loading with Shared Tools

Most previous research efforts have shown that having alternative routing in FMS can significantly reduce parts waiting time, better balance the workload of machines, hence increases the system throughput. Higher routing flexibility can be realized through machine grouping, which can provide alternative machines for processing a part type. In total grouping, one or more groups of machines are tooled identically, while in partial grouping, each operation is assigned to multiple machines but no two machines are tooled identically. Stecke and Raman (1994) demonstrated through a simulation study that partial grouping with state dependent routing performs well across a range of different system utilization level and variation among operation time. Lee and Kim (2000) also studied the loading problem associated with the partial

grouping configuration. The objective is to minimize the maximum workload of machines. They argued that research on the loading problem under partial grouping configuration may be more important since the partial grouping not only gives a better performance than the total grouping but also is a more practical or realistic alternative for obtaining pooling effects with a small number of machines. When flexible tooling strategy is adopted, parts can be routed to machines, which may not have all the required tools. This can not only result in a further improvement in the routing flexibility but also increase the part type variety produced simultaneously in the system. However, the critical constraint on tool traffic must be introduced in the loading and routing decision because tool transport system itself is a physical device with finite capacity (Grieco, et al. 2001). Although there is a research project (Hahn and Sanders, 1994) of producing a prototype of a high-speed tool transport system equipped with linear induction motors having peak speed of over 200 meters per minute, the capacity of commercially available tool transport system with conventional material handling devices is still rather limited. Therefore, efficient planning and controlling tool flows, and coordinating with the part routing remain critical to the overall system performance in a tool-shared environment.

There are a few attempts to address the loading problem under a tool-shared environment with an automatic tool transport system, according to the literature. Han et al. (1989) developed a loading method to minimize the amount of tool traffic among machines under a 'reasonable' workload imbalance level in a tool-shared FMS. They concluded from a simulation experiment that no significant difference among workload imbalance levels is found. Song et al. (1995) studied loading problem under tool movement policy and presented a heuristic algorithm to group parts and tools based on similarities among parts, and a workload imbalance factor was also used to limit the workload of each machine. System throughput is often used as a shop performance measure in evaluating the effectiveness of various loading methods. Roh and Kim (1997) suggested three heuristic approaches namely *list scheduling, sequential,* and *iterative* approaches for part and tool loading, and part sequencing problems in FMS with an automatic tool transporter. In the list scheduling approach, loading and schedules of parts are determined simultaneously based on part priorities. In the sequential, and iterative approach, a parallel machine scheduling problem is solved to determine the loading plan and the sequence of parts, and then the tool loading problem is solved using the part loading plan. A static order environment is assumed, where a set of parts to be produced and their quantities and due-dates

are defined. Most of previous loading algorithms seek to balance workload of each machine. However, when it comes to scheduling and controlling part flows, it is difficult to adhere to a loading plan obtained by the suggested loading algorithm for partially grouped machines. In other words, planned workloads for the machines determined at the planning stage may not be maintained or realized at the scheduling stage even in the static order environment (Lee and Kim, 2000). In a dynamic order environment, with due-date based shop performance measure, machine workload balance can be achieved through scheduling with alternative routing and coordination with efficient tool loading and scheduling.

2.2 Static Scheduling with Tooling Constraints

In traditional manufacturing systems, tooling is usually considered as a secondary issue, and the tooling resource is normally allocated at the pre-release planning stage statically before scheduling jobs. Consequently, early studies of scheduling under tooling constraints generally addressed this problem by using the decomposition approach such as loading then sequencing. The loading step assigns operations and tools to machines and solves the sequencing step in a manner similar to the conventional job shop problem.

However, the performance of the schedule can often be affected by the delay due to the absence of required tools. There were some early research efforts to address the job scheduling while taking into account the tool availability at the same time. Carrie and Petsopoulos (1985) evaluated several part launch sequences in FMS to minimize the tool changes due to tool product variety but found that none makes any noticeable difference to system performance. They concluded that this was due to two factors. First, the operation sequences require parts to return to the load/unload area and the roughing and semi finishing machines several times during processing. So that very soon after launching parts, the initial priorities have little influence on the progress of parts. Second, since only one set of fixture of each type was available, after the first part of each type had been launched on the FMS, the launching of subsequent parts depends on fixtures being released by previous parts of same type, rather than by some externally determined priority. Carrie and Perera (1986) showed that tool changes due to product varieties are only a small part of the total number of changes. Therefore, it is better to devote attention to efficiently performing tool changes for whichever causes, rather than evolve a scheduling method, which assumes that they (tools) are an unchangeable constraint defined by the loading step.

Han, et al. (1989) developed a mathematical formulation for tool loading to minimize the amount of tool traffic among the machines and the tool crib. Job scheduling in their study is not constrained on the initial location of tools, since the required tools can be borrowed from other machines.

Chandra et al. (1993) studied a single-machine job and tool-sequencing problem. They provided a nonlinear programming model to find an optimal sequence of jobs that minimizes the

total time of changing tools and fixtures and guarantees that jobs will be processed before their due-dates. A dynamic programming procedure was developed to solve the problem optimally and a randomization heuristic procedure was used to obtain good solutions for large problems with a high computational efficiency.

Shewchuk and Chang (1995) classified the tools as a recyclable resource, and developed generalized problem formulation for resource-constrained job scheduling problem. The trade-off between total job completion time and quantities of assigned recyclable resources (tools) was studied for the single machine job-scheduling case. Two solution approaches (minimize total completion time first then minimize the quantities of assigned resources, and minimize resource quantities first then minimize total completion time) are developed. They indicated that it might be possible to attain a given level of performance with few tools when jobs and tools are scheduled simultaneously. Alfieri and Brandimarte (1997) studied job shop scheduling with "delay-renewable" resources. Two decomposition heuristic approaches based on a disjunctive graph representation were developed to solve the generalized job shop scheduling problem with tooling constraints.

The scheduling problem under the flexible tooling strategy will involve dynamic scheduling decisions of both jobs and tooling requests over time. Section 2.3.5 gives more detail review of literature about the dynamic tool scheduling problems.

2.3 Dynamic Scheduling and Control

Static scheduling assumes that jobs of different types arrive at the same time, and decisions are made before jobs enter the system. A dynamic scheduling approach schedules jobs as they arrive continuously. In this section, dynamic scheduling issues such as part-release control, various approaches for heuristic dispatching and next machine selection rules are reviewed.

2.3.1 FMS Scheduling and Control Problems

Production processes can only begin after the pre-release planning decisions are made, and the cutting tools are loaded on the tool magazines. The scheduling function involves the assignment of resources (e.g. machines and tools) to tasks (e.g. jobs and operations) for the system to perform operations over a planning horizon. The scheduling problem can be divided into two sub-problems, part-release and detailed resource assignment (Melnyk et al. 1985). Part-release decisions must be made before jobs are released to the shop floor. The release control decides which jobs are released to the system and when they are released. Once jobs are released to the shop floor, the required resources to complete all job operations, such as machines and tools, need to be assigned to the jobs over time. The scheduling function must follow the decisions like routing and tooling decided from the FMS pre-release planning. Different scheduling methods range from simple dispatching rules to sophisticated artificial intelligence based techniques.

The FMS control problems involve monitoring of the shop floor status such as machine and tool status, and job status. The information collected by the shop floor monitoring system can be fed back to the planning and scheduling systems. The difference between the targeted and actual performances of the system can then be identified. A corrective action can be taken when the measured difference exceed a predefined range or the system resources are subject to significant disturbance such as a machine breakdown and/or tool breakage.

In the hierarchical approach described above, the routing of a part through the machines within a system is usually determined at the pre-release planning stage. They are usually addressed under a static manufacturing environment where the amount of the parts to be produced is known and the product mix remains the same throughout the planning horizon. After the pre-release decision is made, part-release and dispatching functions at lower levels must follow the decisions made at the planning stage. This will reduce the routing flexibility of the system by restricting the future assignment of jobs to a limited set of machines, which have the required tools. Furthermore, pre-release planning decisions do not consider unexpected environment changes and are therefore not robust when there exist some system disturbances such as machine breakdowns.

2.3.2 Part-release Control

Part-release decisions must be made before a job is released to the shop floor. The release control decides which jobs are released to the system and when they are released. In an FMS, since a closed flow system is often assumed, a job is simply released when a job of the same part type leaves the system.

Once a job is introduced into the system, the flow of the job on the shop floor is determined by dispatching rules and next machine selection rules. The dispatching is a decision process of selecting a part to be processed from a pool of parts at a machine buffer when it becomes available, while the next machine selection is a decision process of choosing a route among alternative machines for a part when it completes its current operation. The relationship between the part-release and dispatching has been studied by Philipoom and Fry (1992) and Melnyk, et al. (1994). Some research shows that part-release control plays an important role in the shop floor control (Garetti et al. (1990), Roderick, et al. (1992), Hendry and Wong (1994)). However, most researchers believe that the part-release control could reduce possible dispatching opportunities in the shop floor, and it maybe less important than dispatching and next machine selection decision in an FMS if a closed flow system is assumed.

2.3.3 Dispatching and Next Machine Selection

The dispatching is a decision process of selecting a part to be processed next when a machine becomes available, while next machine selection involves selecting a route among multiple alternative routes. Due to their computational efficiency, the heuristic rules are often employed in the dispatching problem. Most dispatching rules can be classified into four classes (Blackstone et al. 1982, and Montazeri and Van Wassenhove 1990): (1) rules based on arriving time, (2) rules based on processing time, (3) rules based on due-dates, and (4) composite rules. However, no single dispatching rule dominates all other rules under various shop floor configurations and operating conditions. Melnyk, et al. (1989) and Ghosh, et al. (1992) argued that shop floor performance quickly deteriorates when the dispatching rule does not consider tooling constraints. A comprehensive review of definitions of various dispatching heuristics reported in the literature is given in section 2.4.5.

As the routing flexibility is realized in FMS by machine grouping and tool loading, an operation can select a route (next machine) among several alternative machines. Khator and Nof (1985), Yao and Pei (1990), Ro and Kim (1990), Zeestraten (1990), Raju and Chetty (1993) have presented and evaluated some machine selection rules. The next machine selection rules are generally based on (1) work in process (WIP) level, (2) processing time, and (3) utilization of candidate machine. O'Keefe and Kasirajan (1992) found that a combination of dispatching rules and next machine rules performs better than others.

Myopic dispatching rules make the decisions of machine routing and part dispatching separately from each other. Machines and jobs are assigned sequentially. The interaction between sub-problems due to jobs and machines becoming available at different time periods is not taken into account in these rules. In recognizing that routing flexibility is not only alternative routing for parts but also alternative jobs for machines, Chandra (1990) proposed an optimization based "multiple dispatching" method to make routing decisions for all parts simultaneously. The decision process was modeled as a maximal weight perfect matching problem and solved by the Hungarian method. However, in his work, tooling required for the operations are assumed available all the time at respective machines. The failed machines are simply excluded from the decision process, which might not be the optimal decision for machine selection, since some machine failures can be repaired in a short period of time.

Since information about the state of the system is of great importance in determining dynamics underlying manufacturing systems, many adaptive scheduling policies using

information obtained from snapshots of the system at various points in time for dynamic scheduling are developed. Chandra and Talvage (1991) proposed an opportunistic dispatching approach in FMS environment, which selects a dispatching rule based on shop and part information. They introduced a "pool" concept in dispatching decisions. Ishii and Talavage (1991, 1994) proposed algorithms by using simulation to select a dispatching rule among candidate conventional dispatching rules dynamically for a next short time period based on the system status. More recently, different scheduling rules are selected on the basis of dynamically changing the manufacturing system pattern, which is the combination of system attributes (Piramuthu et al. 2000), and these adaptive scheduling approaches are often referred as *Pattern Directed Scheduling (PDS).* Artificial intelligence techniques, such as fuzzy logic, neural networks and expert systems are often employed in this type of scheduling approaches. While such an adaptive policy is conceptually appealing, its effectiveness highly depends upon three critical elements (Park et al. 1997): 1) an efficient characterization of any given manufacturing state; 2) the completeness of the set of scheduling rules considered; and 3) the correctness of the decision that maps the manufacturing state into the appropriate dispatching rule. All three elements of the adaptive policy are often problem specific.

Some researchers (Shaw 1989, Upton et al. 1991, Lin 1993, and Yang 1994) have worked on dispatching/machine selection problems under distributed manufacturing architecture by applying a negotiation-based bidding approach. The decisions are made independently based on the local information and communication by individual autonomous entities to achieve their individual goals. This approach may be applicable for some large-scale futuristic systems where each entity has its own intelligence and decision process.

2.3.4 Rolling Time Horizon Approach

Given the computational impossibility of using an exact method procedure and the poor solution quality of myopic dispatching rules, some researchers tried to seek intermediate methods such as the rolling time horizon approach. It decomposes the dynamic scheduling problem into a number of static problems with individual scheduling horizon. This approach solves the scheduling problems for multiple production periods, but only implements the schedule for the first period. One period later, the scheduling for the next planning horizon is

reinitiated by taking the new information gathered from the current shop floor status into account (Shafaei and Brunn 1999a). The solution to the overall problem is approximated by segments of the solutions of these static sub-problems. The static sub-problems can be solved by either heuristics methods or by exact method such as Branch and Bound.

The rolling time horizon concept was used by Baker (1977) to solve the multi-period dynamic lot-sizing problem, and only the first period's decision is implemented; one period later the multi-period model is updated and the process repeated. Baker and Petersen (1979) provided a general framework for analyzing rolling schedules and examine analytically a fundamental quadratic cost model for the effects of such factors as the length of the planning interval, the uncertainty in forecasts and the period of demand.

Muhlemman et al. (1982) used the similar rolling schedule concept for the dynamic job shop problem to reschedule the jobs with different frequencies according to the actual status of the shop at each time. A number of heuristic rules are evaluated for the rescheduling of jobs at different level of uncertainty surrounding the machine breakdowns and process times.

Church and Uzsoy (1992) addressed scheduling problems with identical parallel machines through an event-driven rescheduling approach. The dynamic problem is decomposed into a series of static problems that are implemented on a rolling horizon basis. Sun and Lin (1994) adopted the rolling time window approach to decompose the scheduling problem in time dimension and accommodate the dynamic conditions in a job shop. A rolling time window is partitioned in two periods whose sizes are not necessarily equal. The scheduling is over the entire rolling time window but the resulting schedule is only carried out over the first time period. The rescheduling is over the entire rolling horizon but the resulting schedule is only carried out over the first time period. The rescheduling over the next rolling time window starts after the first time period, the second time period in the current rolling time window is overlapped with the next rolling time window.

Ovacik and Uzsoy (1994) also used the Rolling Horizon Procedure (RHP) to solve single machine problems. A branch and bound algorithm was developed to solve the sub-problem optimally. They extended their procedure to identical parallel machine problems with dynamic job arrivals and sequence-dependent setup times (Ovacik and Uzsoy 1995).

Table 1.1 lists the characteristics of various applications of the rolling time horizon approaches reviewed in this section. Generally speaking, despite the fact that the rolling horizon approach has been recognized as an appropriate approach for the dynamic scheduling problems, one of the main problems in using this approach is the lack of an efficient scheduling method (Shafaei and Brunn 1999b).

Table 1.1. Characteristics of Various Rolling Time Horizon Applications

Authors	Problem type studied	When to reschedule	Rescheduling solution type	Period /horizon length
Baker (1977) Baker and Petersen (1979)	Multi-period lot sizing	Reschedule every period for multi-periods; Overlapped with next horizon.	Optimal solution	Depends on the periodical demand; Multiple equal length periods per horizon;
Muhlemman et al. (1982)	Dynamic job shop	Reschedule every horizon;	Non-optimal /dispatching rule	Continuous time space. Length depends on computational efficiency.
Church and Uzsoy (1992)	Identical parallel machine	Reschedule every period; Overlapped with next horizon.	Non-optimal /Heuristic rule	Same as above
Sun and Lin (1994)	Dynamic job shop	Same as above.	Non-optimal /iterative improvement	Continuous time space. 2 periods every horizon; Length depends on computational efficiency.
Ovacik and Uzsoy (1994)	Single machine	Same as above	Optimal / Branch and Bound	Continuous time space. Length depends on computational efficiency.
Ovacik and Uzsoy (1995)	Identical parallel machine	Same as above	Non-optimal / Heuristic + Branch and Bound	Same as above
Shafaei and Brunn (1999a, b)	Dynamic job shop	Same as above	Non-optimal / SPT+CR rule	Same as above

2.3.5 Dynamic Scheduling with Flexible Tooling

In a tool shared environment, machine idle time due to non-availability of required tools occurs because these tools are either available on the other machine magazines or are in use. To reduce this idle time, it is essential to initiate tool movement for the next operation of a job even while its current operation is being completed. Once the required tool has been identified, the machine issues a request for the required tool to a central tool dispatcher. The tool dispatcher collects these tool requests, and uses a control rule to select a tool request for processing. A tool selection rule is then applied when a tool is available at more than one machine. The various rules reported in the literature are:

Request Selection Rule: Select a request from a pool of requests issued by various machines.

- FCFS First Come First Serve
- SPT Shortest Processing Time
- LOR least number of operation remaining

Tool Selection Rule: select the machine from which tool is to be transported to fulfill a selected request.

- SDT Shortest distance traveled by tool transporter
- HVTL High value of tool life

Early simulation studies (Gaalman, et al. 1987) showed that the system under the tool sharing policy could be operated with considerably less investment in tools while maintaining a small fraction of machine idle times. Recent research has been done to evaluate the tool sharing policy under Single-Stage Multi-machine Systems. Xu and Randhawa (1998) analyzed the effect of different job scheduling rules and tool request selection rules under tool sharing policy. Kashyap and Khator (1996) investigated the effect of request selection rule and tool selection rule in an SSMS under tool sharing environment. Their work shows that the system performance

(make-span of the system) is significantly affected by the request selection rules especially when the tool duplication level is low such that frequent tool competition occurs. A "look-ahead" method used to determine both the requirement of a tool at a machine center and the availability of tools before the actual operation tool place is illustrated in their study. Grieco et al. (1995) used a simulation to determine how many copies of various types of tools should be maintained. They suggested that it is possible to reduce the tooling cost while preserving the system performance by a proper management of tool delivery system. Gargeya and Deane (1999) proposed an approach to schedule jobs on a contingency basis depending upon the criticality of each job with regard to the critical resource (machine or tool). The critical resource in a given time period is identified as the resource with highest ratio of units of resource required to units of resource available. They use a CRCR (critical resource-critical ratio) rule to prioritize jobs requiring the critical resource, and the standard critical ratio rule to prioritize the jobs not requiring the critical resource.

Koo and Tanchoco (1999) used an optimization-based approach to address operation/tool assignment in single-stage Multi-machine manufacturing environment. In their work the operations and tools are selected to formulate the problem as minimum-cost flow problem dynamically while parts are being processed.

The previous comparative studies of the dynamic tool sharing control generally assume that all part's operations can be completed on only one machine, which is the characteristic of Single-Stage Multi-Machine System (SSMS). However, in actual industrial systems, parts still need to visit more than one machine (i.e. Wash and Debur, and CMM, etc.) to complete all the operations, even though versatile identical machines are equipped on the shop floor. To our knowledge, no effort has been made to study dynamic tool sharing control in Multi-stage Multiple Machine Systems.

2.4 Dispatching Heuristics

The dispatching heuristics can be classified in two categories: single-pass and multi-pass heuristic rules (Kutanoglu and Sabuncuoglu 1999). In single-pass heuristic rules, a single complete solution is built up based on a certain priority rule in one step. Most priority dispatching rules proposed in the literature can be considered in this category. In multi-pass heuristics, an initial sequence is generated in the first pass according to one rule, and then in consecutive passes, a search is conducted for the performance improvement using local search heuristics such as simulated annealing, tabu search, etc. In this section, the definitions of some well-known dispatching heuristic rules as well as those recently developed with tardiness objectives are reviewed in detail.

Following are some notations used in this section for various definitions of dispatching heuristics.

n	Number of jobs in a scheduling horizon.
m_i	Number of operations of a job i.
w_i	Weight or tardiness cost of a job i.
r_i	Release date of a job i.
d_i	Due-date of a job i.
C_i	Completion time of a job i.
T_i	Tardiness of a job i, $max\ \{0, C_i - d_i\}$.
a_i	Initial Flow allowance of a job i is the time between the release date and the due-date, $d_i - r_i$.
$A_i\ (t)$	Flow allowance of a job i at time t, is the time between the release date and the current time t, $d_i - t$.
a_{ij}	Arrival time of a job i for operation j to the current machine.
p_{ij}	Processing time of operation j of a job i.
P_{ij}	Total remaining processing time of a job i from operation j.
p_{avg}	Average processing time of jobs waiting for a resource.

$S_{ij}(t)$ Slack a job i waiting for operation j at time t, is the time after the remaining work is deducted from allowance at time t, $A_i(t) - P_{ij}$.

m_{ij} Remaining number of operations of a job i from operation j.

W_{ij} Estimated waiting time for operation j of a job i.

K, h Look ahead parameter to adjust the expected waiting time to worst case.

b waiting time estimation parameter

2.4.1 Due-date and Processing Time Based Rules

Many traditional dispatching heuristics are based on the part due-dates and processing time. FCFS ((First Come First Serve), EDD (Earliest Due-Date), and SLACK (Least Slack) are commonly used rules based on the part due-dates.

FCFS gives the priority to the job with the earliest arrival time a_{ij}. It is generally used as a benchmark to compare with other rules.

EDD rule is the simplest version of allowance based priority rule. It selects the job with the smallest due-date d_i.

SLACK rule sequences jobs in increasing order of their slack index, which is defined as,

$$SLACK_{ij}(t) = d_i - t - \sum_{q=j}^{m_i} p_{iq} \qquad (1)$$

The heuristics rules based on the part processing time often use a ratio to determine a priority index for dispatching.

WSPT (Weighted Shortest Processing Time) rule selects the job based on the descending order of the priority index

$$WSPT_{ij} = \frac{w_i}{p_{ij}} \qquad (2)$$

S/RPT (Slack per Remaining Process Time) is a ratio-based rule that gives the priority to the job with longer remaining processing time and smaller slack. Its priority index is defined as

$$S/RPT_{ij}\ (t) = \frac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{\sum_{q=j}^{mi} p_{iq}} \qquad (3)$$

Other ratio-based rules include CR (Critical Ratio) and S/OPN (Slack per remaining Operation). Their indexes are defined as $CR_{ij}(t)$ *and* $S/OPN_{ij}(t)$ in equations (4) and (5).

$$CR_{ij}\ (t) = \frac{d_i - t}{\sum_{q=j}^{m_i} p_{iq}} \qquad (4)$$

$$S/OPN_{ij}\ (t) = \frac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{m_i - j + 1} \qquad (5)$$

When the remaining allowance or slack time is negative, the ratio-based rules will behave contrary to their original intents. Kanet (1982) proposed another rule called MDSPRO (Modified Dynamic Slack per Remaining Operation) to resolve this problem.

$$MDSPRO_{ij}\ (t) = \begin{cases} \dfrac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{m_i - j + 1}, & if\ d_i - t - \sum_{q=j}^{m_i} p_{iq} > 0 \\ (m_i - j + 1)\left(d_i - t - \sum_{q=j}^{m_i} p_{iq}\right), & otherwise \end{cases} \qquad (6)$$

The relative performances of the rules described above are very sensitive to system conditions such as shop-load level, due-date tightness, balanced or bottleneck shop.

2.4.2 Operation Due-date Based Rules

There are some other heuristics rules that use the operation due-dates instead of the part due-dates. The operation due-dates are often determined by allocating the initial flow allowance of a job to the operations proportional to their processing time. ODD (Operation Due-Date), OSLACK (Operation Slack), and OCR (Operation critical ratio) are the operation due-date versions of EDD, SLACK, and CR. Their priority indexes are defined in equations (7), (8), and (9). It seems that operation due-date based rules performs better than those job due-date based rules (Kanet and Hayya 1982).

$$ODD_{ij} = r_i + \frac{d_i - r_i}{\sum_{q=1}^{m_i} p_{iq}} \times \sum_{q=1}^{j} p_{iq} \qquad (7)$$

$$OSLACK_{ij}(t) = r_i + \frac{d_i - r_i}{\sum_{q=1}^{m_i} p_{iq}} \times \sum_{q=1}^{j} p_{iq} - t - p_{ij} \qquad (8)$$

$$OCR_{ij}(t) = \frac{r_i + \frac{d_i - r_i}{\sum_{q=1}^{m_i} p_{iq}} \times \sum_{q=1}^{j} p_{iq} - t}{p_{ij}} \qquad (9)$$

Some heuristics are based on both the processing time and due-date information. For example, MDD (Modified Due-Date) uses a job's original due-date as the due-date when the job's slack is greater than zero. When the job's slack becomes zero, the earliest finish time act as the modified due-date (Baker and Bertrand 1982). MOD (Modified Operation Due date) rule is the operation due-date version of MDD. Equations (10) and (11) give the priority indexes for MDD and MOD.

$$MDD_{ij}(t) = \max\left\{ d_i, t + \sum_{q=j}^{m_i} p_{iq} \right\} \qquad (10)$$

$$MOD_{ij}(t) = \max\left\{ r_i + \frac{d_i - r_i}{\sum_{q=1}^{m_i} p_{iq}} \times \sum_{q=1}^{j} p_{iq}, t + p_{ij} \right\} \qquad (11)$$

2.4.3 Heuristics for Tardiness Objective

There are several heuristics recently developed for the tardiness objective due to the fact that the due-date related performance measures are frequently used in the applications of dynamic scheduling. COVERT (Cost OVER Time) rule is a popular heuristic rule for tardiness objective. COVERT priority index (equation 12) represents the expected incremental tardiness cost per unit of processing time.

$$COVERT_{ij}(t) = \frac{w_i}{p_{ij}} \times \max\left\{ 0, \frac{h\sum_{q=j}^{m_i} W_{iq} - \max\left\{ 0, d_i - t - \sum_{q=j}^{m_i} p_{iq} \right\}}{h\sum_{q=j}^{m_i} W_{iq}} \right\} \qquad (12)$$

COVERT sequences jobs in the descending order of its priority index. The expected waiting time, W_{iq}, is generally estimated as proportional to its processing time using a parameter b, and it also needs to be adjusted to the worst case by a look-ahead parameter h. If job i queuing for operation j has zero or negative slack then its expected priority is w_i/p_{ij}. If its slack exceeds some worst-case estimates of the remaining waiting time over remaining operations, its expected cost is set to zero. If the slack time is between these two extremes, then the priority goes up linearly as the slack decreases.

Anderson and Nyirenda (1990) developed two new rules. One rule W(CR+SPT) combines CR and SPT, while the other rule W(S/RPT+SPT) combines the S/RPT and SPT rule. Both rules sequence the jobs in descending order of their priority indexes defined in equations (13) and (14).

$$W\,(CR+SPT)_{ij}\,(t) = \frac{w_i}{p_{ij} \times \max\left\{1.0, \dfrac{d_i - t}{\sum_{q=j}^{m_i} p_{iq}}\right\}} \qquad (13)$$

$$W\,(S/RPT+SPT)_{ij}\,(t) = \frac{w_i}{p_{ij} \times \max\left\{1.0, \dfrac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{\sum_{q=j}^{mi} p_{iq}}\right\}} \qquad (14)$$

ATC (Apparent Tardiness Cost, Vepsalainen and Morton 1987) is very similar to COVERT, with two main differences. First, the slack is local resource constrained slack (equation 15), which takes into account the waiting time on downstream machines.

$$SS_{ij}\,(t) = d_i - \sum_{q=j+1}^{m_i} (W_{iq} + p_{iq}) - p_{ij} - t \qquad (15)$$

Second, the decay function for weight/processing time is exponential instead of linear. Jobs are also sequenced by the descending order of ATC priority index (16) defined as,

$$ATC_{ij}(t) = \frac{w_i}{p_{ij}} \exp\left(-\frac{\max\left\{0, d_i - \sum_{q=j+1}^{m_i}\left(W_{iq} + p_{iq}\right) - p_{ij} - t\right\}}{Kp_{avg}} \right) \quad (16)$$

Morton and Pentico (1993) proposed a new dispatching heuristic called BD (Bottleneck Dynamics), which is similar to ATC. BD priority index (equation 17) uses the same enumerator called *activity price* as ATC does. The denominator is replaced with *total remaining resource usage* instead of the current processing time. The resource usage of job i for operation q at machine k(q) at time t is calculated as the resource price of the machine times the processing time of the operation, i.e. $R_{k(q)}(t)p_{iq}$, where $R_{k(q)}(t)$ is the resource price of the machine k(q).

$$BD_{ij}(t) = \frac{w_i \exp\left(-\frac{\max\left\{0, d_i - \sum_{q=j+1}^{m_i}(W_{iq} + p_{iq}) - p_{ij} - t\right\}}{Kp_{avg}} \right)}{\sum_{q=j}^{m_i} R_{k(q)}(t)p_{iq}} \quad (17)$$

The price of a machine in the resource usage calculation can be determined by either *myopic, uniform* or *bottleneck* pricing method (Lawrence and Morton 1993). The myopic pricing scales the current machine price to one, and all others to zero. The uniform pricing assumes that all resources are of equal importance and assigns them prices of one. The bottleneck pricing identifies the bottleneck machine with highest utilization, and gives it a scaled price of one, while other resources are assigned prices of zero.

It is found that no single heuristic dispatching rule performs the best under all possible conditions and the relative performances of the rules are affected by factors such as shop-load levels, due-date tightness, and scheduling criteria. Since all the studies of dispatching rules have been done in different operational settings, the results sometimes give conflicting reports. According to the recently study by Kutanoglu and Sabuncuoglu (1999), heuristics rules

developed more recently like COVERT, ATC, S/RPT+SPT, BD are reported to perform well in the dynamic job shop scheduling with tardiness objective.

Myopic dispatching rules decompose the part/machine (or operation/tool) assignment into smaller sub-problems, namely machine routing (or tool selection) and part dispatching (or operation tool request prioritizing). Resources and tasks are assigned sequentially. The interaction between sub-problems due to jobs (or operation) and machines (or tools) becoming available at different times is not considered in these dispatching rules.

2.5 Summary of Literature Review

From the preceding review, several important issues in the tooling and dynamic scheduling problems are summarized as follows:

1) After the pre-release decision is made, part-release and dispatching functions at lower levels must follow the decisions made at the planning stage. This will reduce the routing flexibility of the system by restricting the future assignment of jobs to a limited set of machines that have the required tools. Furthermore, pre-release planning decisions do not consider unexpected environment changes and are, therefore, not robust with respect to possible system disturbances like machine breakdowns.

2) Myopic dispatching rules decompose the part/machine (or operation/tool) assignment into smaller sub-problems, namely machine routing (or tool selection) and part dispatching (or operation tool request prioritizing). Resources and tasks are assigned sequentially while the interaction between sub-problems due to jobs (or operations) and machines (or tools) becoming available at different times are not considered.

3) It is found that no single heuristic dispatching rule is the best under all possible conditions and the relative performances of the rules are affected by factors such as shop-load level, due-date tightness, and scheduling criteria. Since all the studies of dispatching rules have been done in different operational settings, the results sometimes give conflicting reports.

4) The previous comparative studies of the dynamic tool sharing control generally assume that all operations of each part can be completed on only one machine, which is the characteristic of Single-Stage Multi-Machine System (SSMS). However, in real industrial systems, parts usually need to visit more than one machine (e.g. Wash and Debur, and CMM, etc.) to complete all the operations, even though versatile identical machines are equipped on the shop floor.

5) While the adaptive policy of *Pattern directed scheduling* approaches is conceptually appealing, its effectiveness highly depends upon three critical elements: 1) an efficient characterization of any given manufacturing state; 2) the completeness of the set of scheduling rules considered; and 3) the correctness of the decision that maps the manufacturing state into the appropriate dispatching rule. All three elements of the adaptive policy are often problem specific.

6) Despite the fact that the rolling horizon approach has been recognized as an appropriate approach for the dynamic scheduling problems, one of the main problems in using this approach is the lack of an efficient scheduling method.

7) Machine availability is not accounted for in each of the static sub-problems of existing rolling horizon approaches. The "optimal" decision of a sub problem can easily become unfavorable due to random disruptions, such as machine failures.

8) Many existing research work on dynamic tool sharing often ignore the tool loading decisions and rely mainly on the real-time control of tools during the scheduling stage. This may result in a higher tool handling traffic. There is a need for the closer coordination of tool flow control between the tool allocation at the machine loading stage and the real-time assignment of tools at the scheduling stage.

(9) There are a few attempts to address the loading problem under the tool-shared environment. However, the trade-off between high routing flexibility and tool handling traffic is not recognized.

Chapter 3 Research Methodology

3.1 Introduction

As stated in previous chapters, myopic dispatching rules decompose the part/machine (or operation/tool) assignment into smaller sub-problems, namely machine routing (or tool selection) and part dispatching (or operation tool request prioritizing). Resources and tasks are assigned sequentially while the interaction between sub-problems due to jobs (operations) and machines (tools) becoming available at different time periods are not considered. Furthermore, the dynamic scheduling needs to be computationally efficient and to be able to handle system uncertainties such as urgent job arrivals and machine breakdowns.

Most previous research in the dynamic scheduling of tools assumes that all operations of a part can be completed in one machine. However, in most machining systems, one part usually requires multiple pallet-fixture combinations to complete all operations in the same machine or in different machines. So, multiple setups for machining of a part are not uncommon in discrete part manufacturing industries.

In this research, an integrated approach to make coupled decisions about part/machine scheduling and operation/tool assignments on a rolling window basis is proposed. Specifically, a five-fold framework is proposed in this research (1) allows two levels (cell level and machine level) of hierarchical, dynamic decision making for resource and task assignment; (2) formulates the assignment decision at each level as a minimum-cost flow (MCF) problem during each short-term window, and solves it by an efficient network optimization algorithm; (3) takes both the criticalities of jobs and machine reliabilities into account in decision making; (4) implements the decision for first period of each window, and reschedule resources and tasks for the remaining periods with a short-term look ahead; and (5) handles uncertainty of machine failure down time by estimating the operation completion time based on the transient analysis of machine availability for a two-state Markov process.

The rest of this chapter is divided into 5 sections. Section 3.2 gives an overview of the proposed scheduling approach. Section 3.3 describes a network flow based model for the assignment of tasks and resources during each short-term scheduling window. Section 3.4

discusses how the proposed rolling window approach handle the uncertainties in the shop floors such as machine breakdowns and arriving orders with tight due dates. A machine loading model with shared tools in single-stage systems and a heuristic solution procedure are proposed in Section 3.5. Section 3.6 summarizes some of the assumptions and implementation aspects of the proposed methods in this research.

3.2 Overview of the Proposed Scheduling Approach

This research proposes a framework of a shop floor control system, which consists of a two level hierarchical scheduler for the dynamic decision of job dispatching/next machine selection and tooling schedules and a machine control scheme for operational control of jobs and tools. Most of the pre-release function is not performed before the start of a production period, so that machines, parts and tools are not assigned to each other at the planning stage. The dynamic operational decisions with minimal commands from the higher level allow the system to respond quickly to disturbances such as machine failures or demand changes.

Job and workstation scheduling decisions are usually made at the cell level. On the other hand, operation and tooling resource scheduling are made at the machine level. The proposed scheduler first decomposes the dynamic scheduling problems into static job and tool assignment during each short-term rolling window. It optimizes weighted completion time of tasks for each short-term window by formulating the task and resource assignment problem as a minimum-cost network flow (MCF) problem during each short-term scheduling window. Section 3.2.1 and Section 3.2.2 will present the proposed integrated dynamic scheduler and corresponding machine control framework, respectively. Decomposition of the dynamic job scheduling at the cell level using rolling window approach is illustrated through an example in Section 3.2.3.

3.2.1 Proposed Machine Control Scheme

This section describes the corresponding cell control functions used together with the integrated dynamic scheduler framework and some assumptions made in this research. Machines may be subject to breakdown randomly during an operation on a part (between two operation sequences). The mean time between failure and mean time to repair are known. Each machine has its own buffer for local storage of parts in addition to the central part buffer at the load and unload station. Jobs are released immediately after arriving at the shop floor. Figure 3.1 illustrates the overall procedure of the proposed cell control scheme. The parts arrive at the system according to the customer orders following a specific distribution. The due date of parts is assigned by the well-known Total Work Content Rule (Conway, et al. 1967). The combination

of machine, parts, and tools are not decided in advance. Each part type has a fixed process plan which determines the required machine types and tool types.

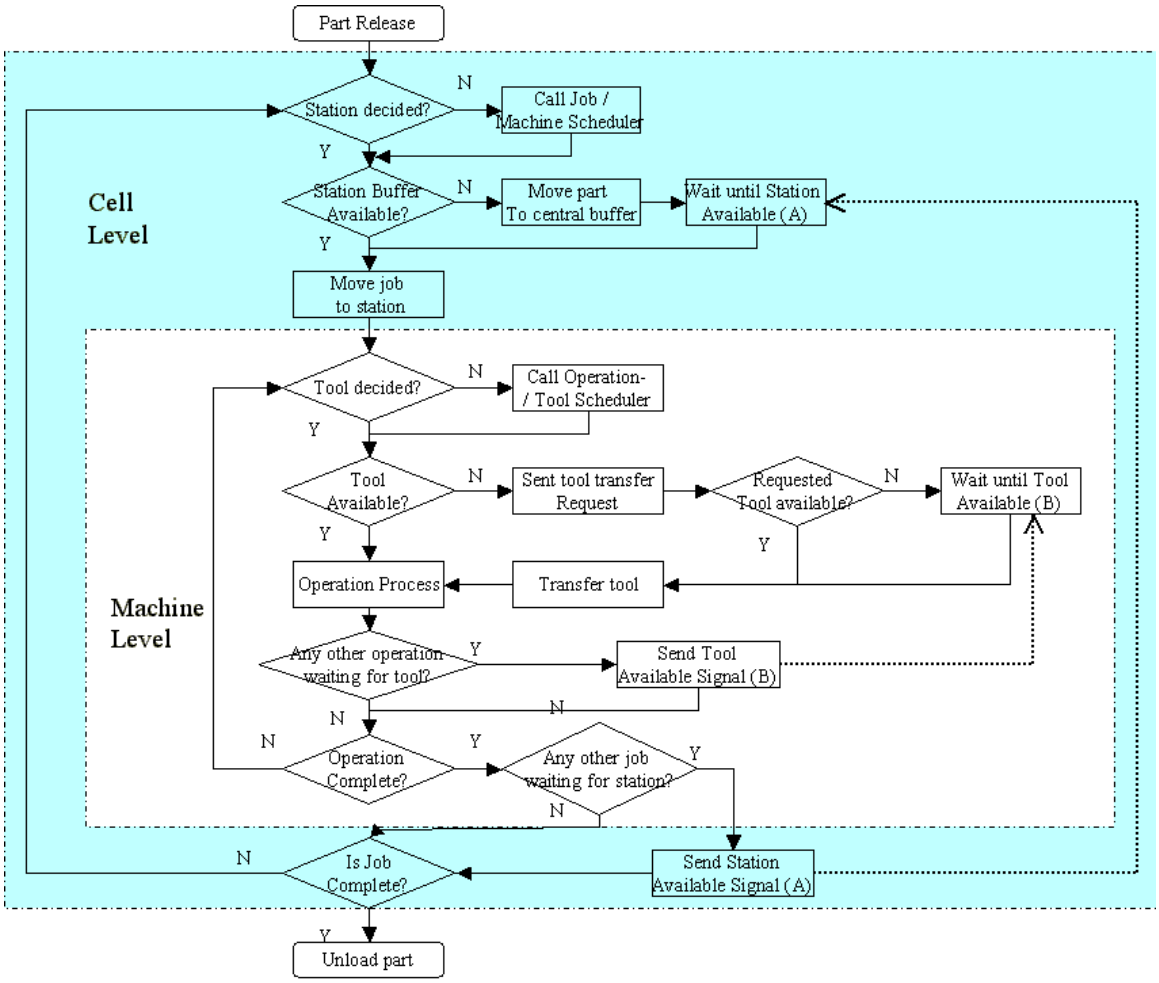


Figure 3.1. Proposed Machine Control Scheme

The dynamic scheduler determines the job-machine schedule during each short-term scheduling window as described in previous section. After completing an operation, a part will check if the next operation requires a different type of pallet. If yes, it will be sent to load and unload station to secure a new pallet, otherwise it will be transported to its scheduled machine for next operation if buffer space is available at that machine. In the case that a part completes a job earlier than the expected time and its next operation station is still unknown or not committed

by the dynamic scheduler during previous window, job/machine scheduler will be initiated to decide the route for that part.

The newly arrived parts may either wait in a system central buffer at the load and unload station, or go to directly the scheduled machine directly if machine buffer space is available. The loading time is assumed to be the same for all part types. Similar to the job-machine schedule, the operation-tool assignment is determined during each short-term scheduling window. Tool switch time is included in the processing time and it is assumed that a fast-automated tool changer is used to perform the tool switch. Whenever an operation is completed, the used tool is checked for remaining tool life. It will be replaced if its tool life is expired. Overhead tool carriers are used to transport tools between central tool room and workstation tool magazines.

3.2.2 Overview of Integrated Dynamic Scheduler

Figure 3.2 shows the overview of the dynamic scheduler. At the cell level, once a job is released into the shop floor, it will become active for the job/station scheduling decision process. During each short-term scheduling window, a set of candidate jobs and a set of candidate machines are selected to formulate into a minimum-cost network flow problem. Only those jobs, whose current operation routes (machine) are not committed or are about to complete their current operations within the scheduling window while next operation route is not yet decided, are selected for the decision process.

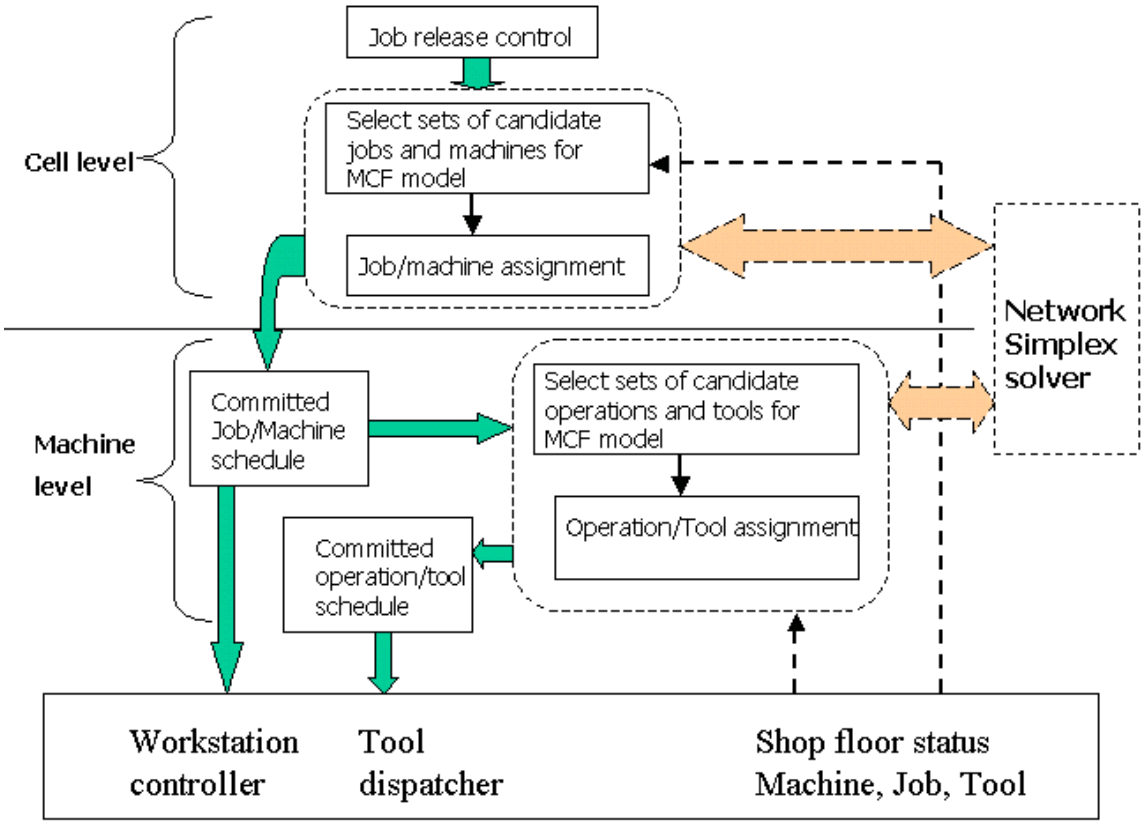


Figure 3.2. Overview of Integrated Dynamic Scheduler

The estimated job operation start time and finish time will be calculated based on the current status of each candidate job and current schedule of each candidate machine. Section 3.3 will discuss the modeling detail of this process. The model will be solved by using the network simplex algorithm, and the solution will be converted into temporary routes for candidate jobs. If a job temporary operation start time is within the current scheduling window, then that route will be committed and the job will be added to the corresponding machine schedule. The available time of that machine will be updated accordingly.

It is assumed that during each operation, a job may require more than one type of tool for a process. Each process step of a job with a different type of tool is called an operation sequence. Once the current operation route and its machine schedule are committed, the required tool for the first operation sequence will enter the decision process for operation/tool scheduling. Similarly, during each short-term scheduling window of operation and tool assignment, a set of candidate operation sequences and a set of candidate tools are selected to formulate into a

minimum-cost network flow (MCF) problem. Only those operation sequences, whose current required tool are not committed or their current operation sequences are about to complete within the scheduling window while next required tool is not yet decided, are selected for the decision process. The estimated operation sequence start time and finish time will be calculated based on the current status of each candidate operation sequence and current schedule of each candidate tool. The model will be solved by using the network simplex algorithm, and the solution will be converted into temporary tools for candidate operation sequences.

If a temporary operation sequence start time is within the current scheduling window of operation and tool assignment, then that tool schedule will be committed and the operation sequence will be added to the corresponding tool schedule. The available time of that tool will be updated accordingly.

3.2.3 Decomposition using Rolling Time Horizon Approach

The proposed rolling window approach decomposes the dynamic scheduling problem into a series of static sub-problems for each short time window. Due to the dynamic job arrivals and uncertainties in machine and tooling availabilities, early commitment of the resources to tasks may become unfavorable to the overall system performance, and cause the system to be inflexible to the disruptions. Therefore only one job's operation is scheduled to a selected candidate machine during each decision process, and only the next unscheduled operation of a job is considered as a candidate task to be scheduled. In case there are more candidate tasks than available resources, the scheduling decision for unscheduled candidate tasks will be made during next short-term window, which is overlapped with the current scheduling window. Figure 3.3 shows an example of decomposition at the cell level using the proposed rolling window approach.

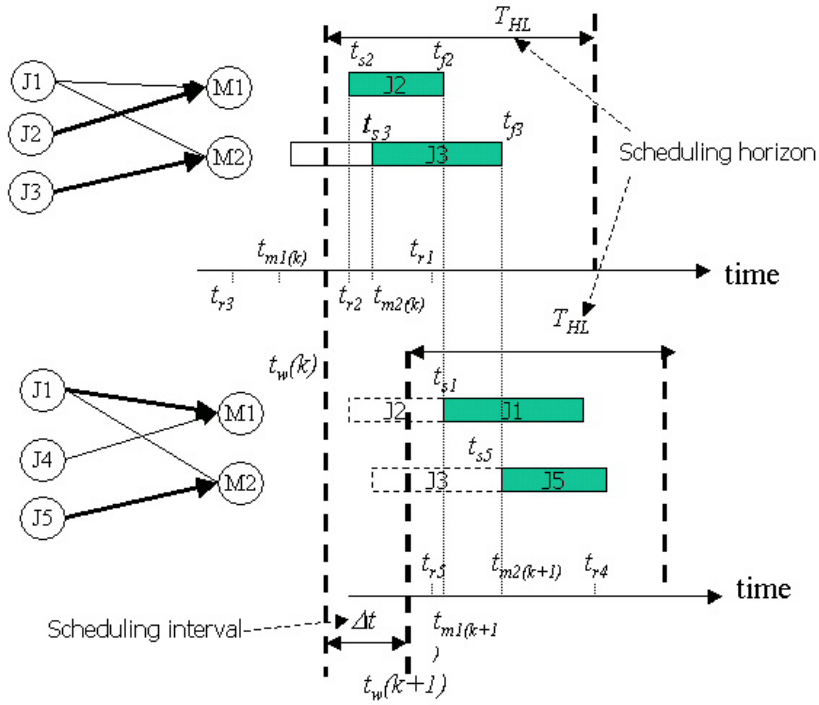


Figure 3.3 Decomposition by the Rolling Window Approach

Suppose, at time $t_w(k)$, beginning of k_{th} scheduling horizon, three candidate tasks (J1, J2, and J3, which are unscheduled operations of various jobs) are ready within the k_{th} scheduling horizon of length T_{HL}. Only two machines *M1* and *M2* are available at $t_{m1(k)}$, and $t_{m2(k)}$. t_{r1}, t_{r2}, and t_{r3} are ready time for these three tasks. After the MCF model for the static sub-problems is formulated and solved, tasks 2 and 3 are scheduled at *M1* and *M2* respectively. If the estimated task start time of task 2 and 3 (t_{s2} and t_{s3} are smaller than the next rescheduling time, $t_{now}(k+1)$, tasks 2 and 3 will be implemented. Tf2 and tf3 are estimated task finish times, which are also times when machine 1 and 2 become available again at $t_{m1(k+1)}$ and $t_{m2(k+1)}$ in the next horizon. The unscheduled task 1 will be considered as a candidate task together with tasks 4 (J4) and 5 (J5), which become ready at t_{r4} and t_{r5}, during the $(k+1)_{th}$ scheduling horizon of length T_{HL}. The decision process for static sub-problem at cell level for the new scheduling window will repeat, and the scheduling horizon will roll forward every Δt time period. Δt is referred as the scheduling interval and is smaller or equal to the scheduling horizon length ($\Delta t = t_w(k+1) - t_w(k)$ and $\Delta t \leq T_{HL}$).

3.3 Heuristics for Static Sub-problems

In this section, the network flow based models for task-resource assignments are presented first in sections 3.3.1 and 3.3.2. The scheduling procedures for static sub-problems at both the cell level and machine level are then illustrated in sections 3.3.3, and 3.3.4. Finally, a different scheduling procedure for the cell-level sub-problem taking into account tool availabilities is proposed in section 3.3.5.

3.3.1 MCF Model Formulation

During each short-term scheduling window, a set of candidate tasks and resources are selected at each level from the shop floor to be formulated as a Minimum Cost Flow (MCF) problem. The task ready time and resource (machine and tool) available time are considered as if they were static. The task/resource assignment is further formulated as a minimum network flow problem illustrated in Figure 3.4.

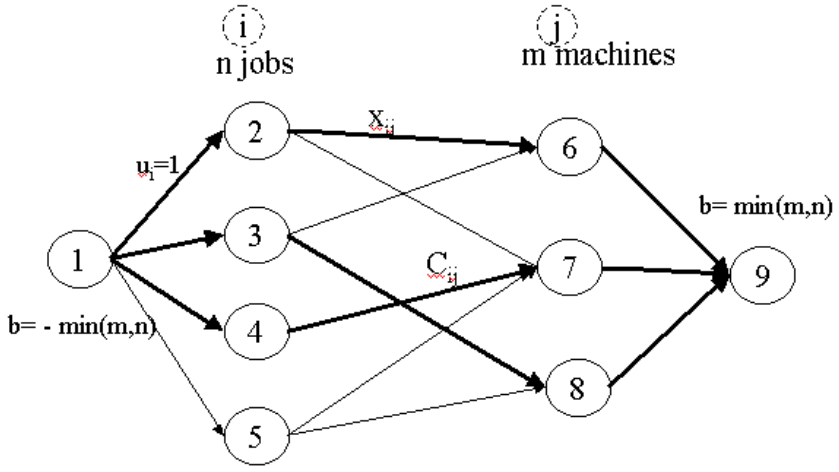


Figure 3.4. Network Flow Model of Task and Resource Assignment

Nodes 2, 3, 4, 5 are candidate tasks (e.g. jobs at the cell level or operation sequences at the machine level.) Nodes 6, 7, 8 correspond to required resources (e.g. machines, tools) by those candidate tasks. Node 1 and node 9 are dummy nodes, which serve as start node and end node. The arcs between candidate tasks and resources represent alternative assignment of the resources to tasks. All arcs are unit capacity. The net flow at the start node is minimum value of the number of candidate tasks and the number of candidate resources. The net flow at end node is the negative value of net flow at start node. The net flow values for all the remaining nodes are zero. The cost coefficients for arcs between the start node and candidate task nodes are set to zero. The cost coefficients (C_{ij}) for arcs between candidate resource nodes and end node are all zero as well. The cost coefficients corresponding to alternative assignment of resource to task is determined to optimize weighted completion time of tasks. The detail derivations of these cost coefficients are explained in following sections. The formulated minimum network flow problem can be summarized as follows:

X_{ij}	Flow of arc from node i to node j
U_{ij}	Capacity of the arc from node i to node j
C_{ij}	Cost associated with assigning task i to resource j
V	Set of nodes in the networks
b_i	Net flow at node i
A	Set of directed arcs connecting nodes in the networks
$A_{in}(i)$	Set of arcs that are immediate predecessors to node i
$A_{out}(i)$	Set of arcs that are immediate successors to node i.

Min

$$\sum_{(i,j)\in A} C_{ij} X_{ij} \qquad (18)$$

Subject to:

$$\sum_{j\in A_{in}(i)} X_{ji} - \sum_{j'\in A_{out}(i)} X_{ij'} = b_i \ \forall i \in V \qquad (19)$$

$$0 \le X_{ij} \le U_{ij} \qquad \forall (i, j) \in A \qquad (20)$$

The above-formulated model can be solved efficiently by the network simplex algorithm. For the detail of network simplex algorithm please refer to Bazara et al. (1990).

3.3.2 Data Required for MCF Model

t_w	the current system time
t_{di}	the due time of part i.
t_{rpi}	the remaining processing time of part i.
t_{pi}	the processing time of task i
t_{ri}	the ready time at which the last task of part i is completed
t_{mj}	the available time at which the last scheduled task on resource j is completed
m_{ij}	the expected travel time delay of part (or tool) i from its current location to machine j
t_{si}	the estimated start time of task i
t_{fi}	the estimated finish time of task i

The start time of task i is determined by the ready time of the task, available time of the resource, and the expected travel time delay. The expected travel time delay is the time to transfer a job from its previous operation station to its subsequent operation station at the cell-level job scheduling. At the machine level, it stands for the expected time to transport the required tool from its previous magazine to the station where the next job requests it. When t_{ri} is less than t_w, it means job *i* is ready before the current time. If the time job *i* becomes ready plus the expected travel time delay m_{ij} is greater than the available time of machine t_{mj} , then the machine j will be idle waiting for the job. Otherwise, job i will arrive at machine j early and wait for machine j to become available. Therefore, the task start time can be estimated by the following expression.

$$t_{si} = max\{ (max\{t_{ri} , t_w\} + m_{ij}), t_{mj} \} \qquad (21)$$

Due to the dynamic nature of shop floor and random disruptions like machine failures, the estimates of the task ready times and the resource available times need to be updated frequently. Section 3.4.1 gives more detail discussion about this issue.

The finish time of a task will be the start time plus the expected processing time of the task, i.e.

$$t_{fi} = t_{si} + E(t_{pi}) . \qquad (22)$$

Since the machines are subject to random breakdowns, the processing time of the task can be estimated based on the analysis of a machine's availability. (See section 3.4 for detail.)

3.3.3 Cell-level Sub-problem Decisions

At the cell level, the information about both the part urgency and machine workload should be included into the job and machine assignment decision process. Two types of decisions made in static sub problems at the cell level are:

1) When m, the number of candidate machines, and n, the number candidate jobs, are not equal, Select m jobs from n candidates if n > m. (Jobs with lower weight (w_i), smaller slack time or larger process time will be preferred) Select n machines from m candidates if m > n.

2) After k candidate jobs and k candidate machines being selected, where k = min (m, n), assign selected k jobs to k machines.

Consequently, the total weighted completion time $w_i t_{fi}$ is used as the cost coefficient, where w_i is the ratio of job slackness to remaining processing time.

$$w_i = \begin{cases} \dfrac{d_i' - t_{rpi}}{t_{rpi}}, & d_i' > t_{rpi} \\ \dfrac{1}{t_{rpi}}, & otherwise \end{cases} \qquad (23)$$

where, $d_i' = d_i - t_w$ and t_{rpi} is the remaining process time of task i

Furthermore, machines are subject to breakdowns during an operation. Hence, high variation in machine availability will result in significant deviation of the finish time estimation from the actual time. To reduce this uncertainty, the job start time t_{si} is used instead of t_{fi}, since the objective of total weighted completion time can be approximated by total weighted task start time. Then, the cost coefficient for job/machine assignment becomes

$$C_{ij} = \left(\frac{\max\{1, t_{di} - t_w - t_{rpi}\}}{t_{rpi}} \right) * t_{si} \qquad (24)$$

In order to balance the machine workload and prioritize the urgent jobs, the machines with earlier start time, which means a lower workload, should be chosen in the dynamic decision process. The first term in the bracket is the job urgency defined by job slackness over remaining processing time. If the job is already late, behind the due-date, the urgency of job will be 1 over remaining processing time. The second term, job operation start time serves the purpose of balancing the workload of machines, since the operation start time is determined by the available time of the candidate machine and the possible travel time delay of the job. Therefore, the machine with an earlier start time, which means a lower workload, will be more desirable in the dynamic decision process.

At the cell level, tasks that need to be scheduled are operations of jobs. The required machines are major binding resources to be assigned to each job. Only those jobs, whose current operation machines are not committed or their current operations are about to complete within the scheduling window while their next operation route (machine) is undecided, are selected as candidate jobs.

If the required tools are not available at the machine level, it may cause extra time delay in transporting tools from the remote magazine to the station that requires them, hence affect the overall resource (machine and tool) availability. Therefore it is highly desirable if the decisions of scheduling jobs to machines at the cell level can consider the availabilities of required tools at each candidate machine. However, the exact locations of all tools in the system are difficult to predict, and the time and sequence at which the tool is transferred to the different stations are subject to changes due to random machine failures. On the other hand, when a fast tool delivery system is used, the expected tool transport time is much smaller than a job's operation processing time and machine down times, and therefore job urgencies (in term of slack or flow allowance) and machine availabilities will have much larger impact on the overall system performance than the availabilities of tools at the cell level. Therefore, a sequential decision procedure at the cell level is proposed in this section, which considers job urgencies and machine availabilities with a higher priority in selecting candidate jobs and machines when their quantities are not equal.

When the number of candidate machines and the number candidate jobs become equal after k candidate jobs and k candidate machines being selected, where k = min (m, n), a MCF model (18)-(20) can be formulated to minimize the total adjusted start time, using the cost coefficient C'_{ij} *defined* in equation (25). The candidate operations will be assigned to the machines with earlier adjusted task start times, which correspond to high percentage of tools that are available at the machines.

$$C'_{ij} = (t_{si} + \sum_{k=1}^{n_i} \delta_{ijk} * W_k) \qquad (25)$$

Where, n_i is the number of tool types required by an operation i of a job

And δ_{ijk} is defined as

$$\delta_{ijk} = \begin{cases} 1, \; if \; tool \; k \; is \; not \; at \; machine \; j \; before \; time \; t_{si} + \sum_{l=1}^{k-1} (p_{ijl} + \delta_{ijl} \cdot W_l) \\ 0, \qquad otherwise \end{cases}$$

W_k is the expected tool waiting time for the k_{th} required tool type for a job's operation. It is estimated by using a constant multiplier (lead time constant) times processing time as a waiting time estimate (Vepsalainen and Morton 1987). When the k_{th} required tool is not available by the estimated start time ($t_{si} + \sum_{l=1}^{k-1}(p_{ijl} + \delta_{ijl} \cdot W_l)$) of the k_{th} sequence with processing time of p_{ijl}, a waiting time of W_k will be incurred.

Following shows the steps of the heuristic procedure for static sub-problems at the cell level. T_{HL} is the scheduling horizon length. $t_w(k)$ is the time epoch at the beginning of k_{th} scheduling horizon.

<u>Cell-level Sub-problem Heuristic</u>

Step 1. Select the candidate tasks whose unscheduled operation sequences become ready during k_{th} scheduling window between time $t_{now}(k)$ and $t_{now}(k) + T_{HL}$. If no candidate task is selected, stop. Otherwise, continue.

Step 2. Select the candidate machines (required by candidate operations), formulate the MCF model P1 (18)-(20) using coefficient (24).

Step 3. Solve the model P1 formed in step 2. Reformulate another MFC model P2 (18)-(20) using all selected jobs and machines in the solution of model P1 using the coefficients based on the equation (25).

Step 4. Solve the model P2 formed in step 3,

Loop: If the estimated start time of selected operation i is less than time $t_{now}(k) + \Delta_t$, implement the operation i, otherwise continue

Until all of k selected operations are checked.

Step 5. Wait until time $t_{now}(k) + \Delta_t$, then k=k+1, go to step 1.

As stated before, since job urgencies and machine availabilities have larger impact on the overall system performance at the cell level, they are considered with higher priorities in selecting jobs and machines when the number of jobs and machines are not equal in the step 2.

3.3.4 Machine-level Sub-problem Decisions

Once the higher-level job schedule is completed, it is necessary to assign tooling resources to operation sequences in a way to reduce the tool waiting time of all jobs due to tooling shortage at the machine level. Since it is assumed that machines do not fail during each operation sequence, the sequence finish time is determined by the sequence process time, the waiting time and expected tool transfer time of the required tool once the sequence is started. Consequently, the network flow model of operation and tool assignment will minimize $\sum_i t_{fi}$, the total actual completion time of selected operation sequences during each short-term window.

So, the operation sequences finish time will be used as the cost coefficient for tool assignment, i.e.

$$C_{ij} = t_{fi} \qquad (26)$$

At the machine level, tasks that need to be scheduled are operation sequences of different jobs. The required tools are major binding resources to be assigned to different operation sequences. Operation sequences are selected as candidates for the decision process if required tools for their current processing steps are not committed or the required tools for their subsequent process steps are not yet decided. The heuristic procedure for static sub-problems at the machine level is described as follows.

Machine-level Sub-problem Heuristic

Step 1. Select the candidate operations whose unscheduled operation sequence becomes ready during k_{th} scheduling window between time $t_w(k)$ and $t_w(k) + T_{HL}$. If no candidate tasks selected, stop. Otherwise, continue.

Step 2. Select the candidate tools (required by candidate operations), formulate the MCF model (18-20) using coefficient defined in Equation *(26)*.

Step 3. Solve the model formed in Step 2,

Loop: If the estimated start times of selected operation i are greater than time $t_w(k)+\Delta_t$, carry out the operation i. Otherwise, continue

Until all of k selected operations are checked.

Step 4. Wait until time $t_w(k)+\Delta_t$, then $k=k+1$, go to Step 1.

3.4 Handling Shop Floor Uncertainties

Dynamic scheduling must be computationally efficient and be able to handle system uncertainties such as arriving jobs with urgent due dates and machine breakdowns. Frequent rescheduling to react to such disruptions can make the behavior of the system hard to predict, hence reduce the effectiveness of the dynamic scheduling. Another approach to handling disruptions is to update job ready time and completion time, and machine status on a rolling horizon basis, and consider the machine availability explicitly in generating schedules dynamically. When machine downtime has a small variation, the operation completion time is estimated by using limiting (steady-state) machine availability. However, steady-state analysis sometimes is unlikely to provide a complete picture of the system when there is a large variation in machine downtime and repair time, and frequent disruptions (e.g. tool failures) exist. Transient analysis of machine availability will be more meaningful in such situation during a finite observation period.

It is assumed that there are two types major disturbances causing the uncertainties on the shop floors. They are machine random breakdowns and random job arrivals. Since a machine can fail during the process of an operation, the actual processing time of a job is greater than the required processing time defined by the part process plan. This disturbance will affect the accuracy of estimation for operation start time and finish time in previous section, hence deteriorate the performance of the dynamic scheduler. Furthermore, random jobs arriving with tight due-dates should be considered with higher priority in the scheduling process. The predictions about task urgencies and resources availabilities are only relatively accurate in the short-term window. So, a rolling scheduling window approach is used to make the proposed scheduler respond to these disturbances quickly. During each scheduling window, if the estimated start time of a selected task is after the scheduling window, the schedule for this task is not committed in the current window. It will be rescheduled in the next window together with possible new job-arrivals and resources that become available in that window.

Since the machine can break down during the operation process, estimates of task finish time in Section 3.3.3 should take into account the reliability of the machines. If the availability of a machine is defined as MTBF/(MTBF+MTTR), then the task finish time can be estimated as follows,

$$t_{fi} = t_{si} + t_{proci} * (MTBF+MTTR)/MTBF \quad (27)$$

Whenever a task is completed, the available time of a corresponding resource, the ready time, start time, and finish time of all tasks scheduled on that resource should be updated accordingly. The following section provides more detail information about this.

3.4.1 Updating Schedules of Machines and Tools

Because of the dynamic nature of shop floor with unreliable resources, the quality of a schedule within each individual scheduling window and system performance over the long term depends greatly on the accuracy of estimates about resource availabilities and job (operation sequence) ready times and completion times. For example, there are n jobs scheduled for machine j at time t. If machine j fails at time t_{downj}, the corresponding operation start time t_{si} and the completion time t_{fi} of all jobs, which are scheduled after time t_{downi}, will be affected. Therefore, the estimation of operation start time, finish time, and machine next available time t_{mj} for new jobs needs to be updated during each short-term scheduling time window. Figure 3.5 shows the effect of the machine failure on all jobs scheduled for the machine.

In this research, it is assumed that only normal or minor machine failures exist and jobs will not be rerouted to the other stations in the event that the scheduled machine fails. When the machine fails, all scheduled jobs at that machine will wait until the machine is up again. The operation sequences of the job will be re-started after the machine is up. Sometimes it could improve the system performance by rerouting jobs to other stations when major machine breakdowns occur and the down machines stay in the same state for more than one shift or even days. In that case, the failed machine will not be selected as a candidate station for each scheduling decision process.

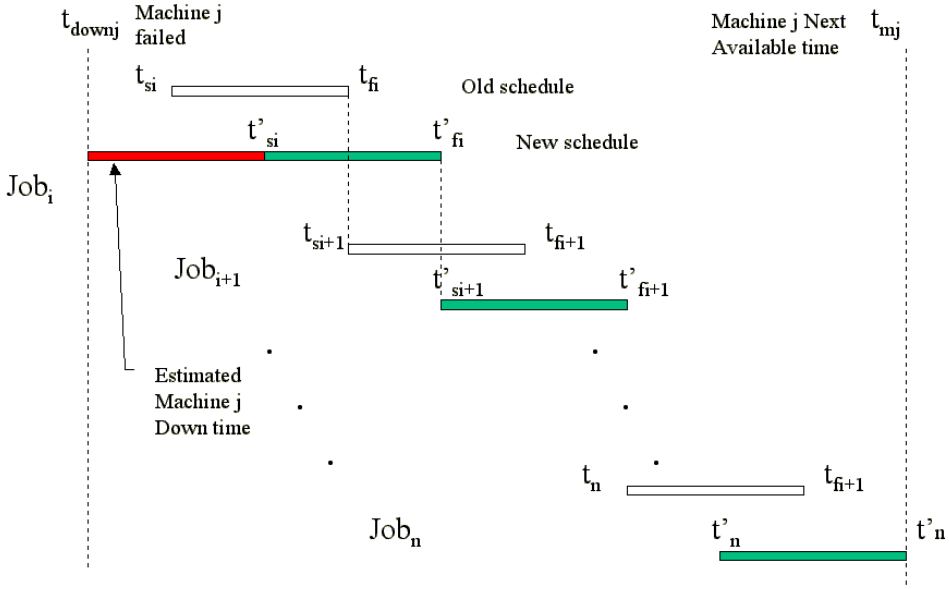


Figure 3.5. Effects of Machine Breakdown on Scheduled Jobs at Machine j

At the beginning of each short-term scheduling window at the cell level, each workstation status is checked. If the machine j is down, the new start time of the first job is estimated as

$$t'_{s1} = Max(t_{now}, t_{downj} + MTTR_j) \qquad (28)$$

Where t_{downj} is the time when machine j fails, and $MTTR_j$ is mean time to repair of machine j. The new start time of remaining jobs are estimated as

$$t'_{si} = t'_{f,i-1}, \ \forall i = 2,3, \ldots n \qquad (29)$$

When exponential distribution is assumed for the machine failure, the new finish time of the job i is determined by

$$t'_{fi} = t'_{si} + t_{proci} * (MTBF_j + MTTR_j)/MTBF_j \qquad (30)$$

Where $MTBF_j$ is mean time between failure of machine j, and t_{proci} is the remaining process time of the job i. The next available time of machine j becomes t'_{fn} as shown in Figure 3.5.

Similarly, during each short-term scheduling decision process at the machine level, all scheduled operation sequences for each tool also need to be updated accordingly based on the status of machines where the scheduled operation sequences are performed. Figure 3.6 illustrates the effects of machine failure and tool breakage on scheduled sequence time and the available time of tool. For example, there are n operation sequences scheduled to use tool k at time t. If the machine j is down, the new start time of the first operation sequence is estimated by equation (31). The start time of remaining scheduled sequence is defined as

$$t'_{si} = max(t'_{f,i-1} + t_{worn} + m_{i-1,\, i},\ t_{si}) ,\ \forall i = 2,3, \ldots n \qquad (31)$$

Where t_{worn} is the reconditioning time of a tool if it becomes worn after the operation sequence i-1. $m_{i-1,\, i}$ is the expected travel time of tool transporter to transfer tool k from the station of operation sequence i - 1 to the station performing operation sequence i. The new finish time of the operation sequence i can be determined by equation (31).

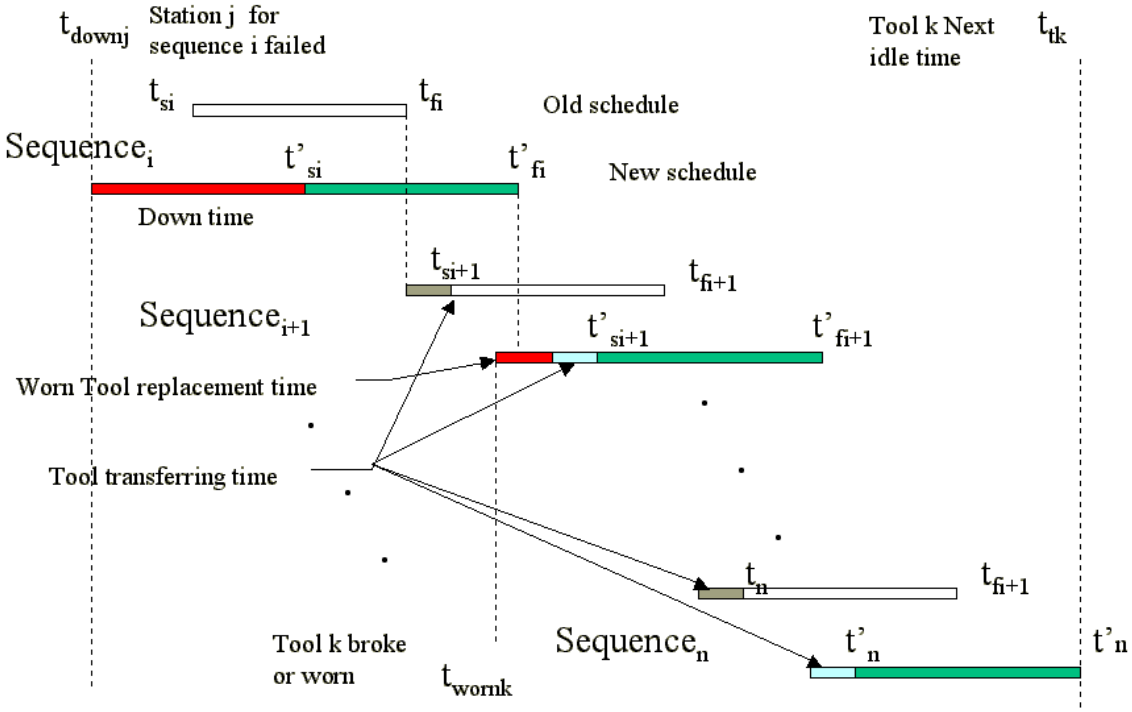


Figure 3.6. Effects of Machine Failure and Tool Breakage on Scheduled Sequence

3.4.2 Transient Analysis of Machine Availability

In previous section, the operation finish time is estimated by defining the availability of the workstation as it is in the equation (30), which is based on the steady-state analysis of machine availability. However, steady-state analysis sometimes is unlikely to provide a complete picture of system when there is a large variation in machine downtime and repair time, and frequent disruptions exist. Transient analysis of machine availability will be more meaningful in such situation during a finite observation period. In this section, machine downtime and up time are assumed to be exponentially distributed. An unreliable machine will be modeled as a two-state continuous time Markov chain.

Let $P_{ij}(t)$ be the stationary transition probability for the system to change to state j in time t if it is in state i at time zero. λ is the machine failure rate, which is 1/MTBF. μ is the repair rate, which has the value of 1/MTTR. Machine is up when i = 1 otherwise it is down (i = 0). For a two-state Markov chain, the following *Kolmogorov's forward* equations (35, 36) hold (Osaki 1992):

$$P'_{i0}(t) = \lambda P_{i1}(t) - \mu P_{i0}(t) \qquad (i = 0, 1) \qquad (32)$$

$$P'_{i1}(t) = -\lambda P_{i1}(t) + \mu P_{i0}(t) \qquad (i = 0, 1) \qquad (33)$$

By assuming the initial conditions be $P_{11}(0) = 1$ and $P_{00}(0) = 0$, and equations (15) and (16) can be solved by using a *Laplace* transform. If the initial condition is $P_{11}(0) = 0$ and $P_{00}(0) = 1$, then equations (32) and (33) can be solved as

$$P_{11}(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t} \qquad (34)$$

$$P_{01}(t) = \frac{\mu}{\lambda + \mu} - \frac{\mu}{\lambda + \mu} e^{-(\lambda+\mu)t} \qquad (35)$$

Let $A_k(t) = P_{k1}(t)$ (see Figure 3.7) denote the <u>*point-wise availability*</u> of the machine given that was in state k at time 0 (Osaki 1992).

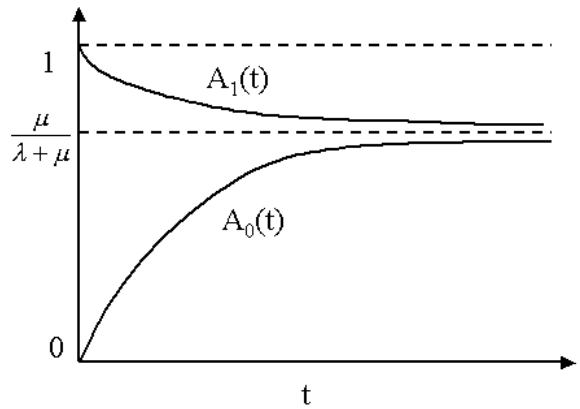


Figure 3.7. Point-wise Availability of Machine

$$A = \lim_{t->\infty} A_k(t) = \frac{\mu}{\lambda+\mu} = \frac{1/MTTR}{1/MTBF + 1/MTTR} \qquad (36)$$

$$A = \frac{MTBF}{MTBF + MTTR} \qquad (37)$$

Equation (36) gives us the limiting availability, which is the same as the machine availability used in the equation (30) previously.

Let t_s be the start time of an operation, t_f be the finish time of an operation, and t_{proc} be the operation process time. Assuming that machine is up at time 0, then the amount of time the machine is up within the period between t_s and t_f is determined by $A_1(t)$. Therefore,

$$\int_{t_s}^{t_f} A_1(t) * 1 dt = t_{proc}$$

$$\int_{t_s}^{t_f} \left(\frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t}\right) dt = t_{proc}$$

$$\frac{\mu}{\lambda+\mu}(t_f - t_s) - \frac{\lambda}{(\lambda+\mu)^2}\left(e^{-(\lambda+\mu)t_f} - e^{-(\lambda+\mu)t_s}\right) = t_{proc}$$

Let

$$G(x) = \frac{\mu}{\lambda+\mu}(x - t_s) - \frac{\lambda}{(\lambda+\mu)^2}(e^{-(\lambda+\mu)x} - e^{-(\lambda+\mu)t_s}) - t_{proc} \quad (38)$$

And $G(t_f) = 0$

Since $G(x)$ is increasing function over x, the value of t_f can be found by minimizing the absolute value of $G(x)$. The new estimate of the operation finish time can be obtained by solving this unconstrained optimization problem using Golden Section Search algorithm (Bazaraa et al. 1993).

Appendix D gives the implementation of Golden Section Method in Visual Basic.

3.5 Tool Loading in Single Stage Systems

In previous sections of this chapter, it is assumed the overall system performance in the long run is not sensitive to the initial tool loading (which tool is loaded on which machine), since the alternative routings exist for all part types and tools are dynamically shared among machines. Consequently, tools are loaded arbitrarily among identical machines. Machines are not grouped to allow maximum routing flexibility. However, this may result in congested too handling systems especially in Single Stage Systems where tool-sharing activities are more frequent and all operations of a part are performed on a single machine. Another research hypothesis tested in this research is that better performance can be achieved by planning tool flow in the machine loading stage.

In Section 3.5.1, the machine-loading problem in the tool-shared environment is first introduced. Then, Section 3.5.2 presents an integer programming formulation of the loading model in the single-stage systems. The objective is to maximize routing flexibility and maintain a Minimum Resident Tool Ratio for assigning part types and tools to machines. A heuristic algorithm for generating loading plans is proposed in Section 3.5.3. In Chapter 4, the performance of the proposed loading method is compared empirically to that of a random loading method through simulation studies.

3.5.1 Machine Loading with Shared Tools

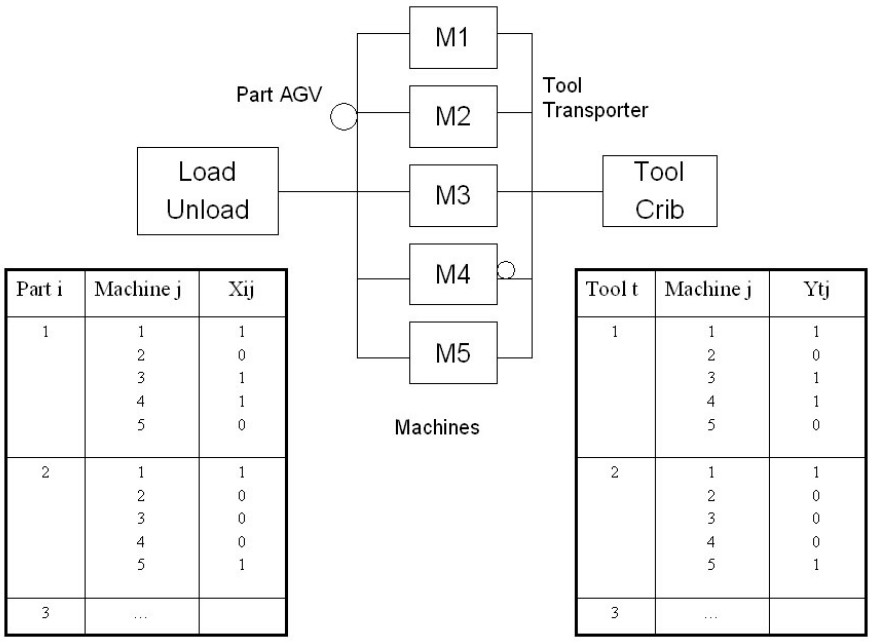
The characteristics of grouping and loading problems depend greatly on the strategy of managing the tools within the system. Generally three methods, namely *batching, flexible,* and *hybrid* tool strategy can be used depending on the structure of the physical system (Grieco et al. 2001). With the batching strategy, the configuration of each tool magazine is " frozen" for a given length of time during which each machine can only use the tools loaded on its tool magazine. After this period, the configuration of the tool magazine is changed and a new period starts. With the flexible tool strategy (or tool-sharing), tools can be borrowed from other machines. In this case, the configuration of a tool magazine is evolving continuously, and tool flows will change on the processing requirements of parts at different machines. The hybrid

approach is a combination of both flexible and batching strategies with some limitation introduced. Some tools may be assigned to the machines for given periods as "resident" tools. The batching approach is considered more appropriate in a high volume/low variety environment with static batch demands. However, in low volume/high variety order environment, part types are usually produced to order. The primary manufacturing concern in such systems is meeting order due-dates and producing a large variety of part types in small quantities. Flexible or hybrid tool strategies should be the preference because they provide greater product variety and quick responsiveness. They are enabled by an automatic tool changing mechanism and tool delivery systems to reduce the time for tooling setup, and hence enable system resources to be utilized continuously. Grieco et al. (2001) pointed out that one of the major reasons that limit the current application of tool handling system is the lack of efficient algorithm to control tool flow, and coordinate it with part flow to minimize unnecessary delays. Existing research work in dynamic tool scheduling often ignores the tool loading decision, and rely mainly on high-speed tool handling systems for the real-time tool assignment decisions. This may result in a higher tool handling traffic. There is a need for the closer coordination between the tool allocation at the machine loading stage and real-time assignment of tools at the scheduling stage.

The manufacturing system modeled in this section consists of several identical machines, each of which has a tool magazine. An automatic tool transporter is available to move the tools between machines and/or central tool room. Before a new production period starts, the tools are partitioned into groups and loaded into the tool magazines of machines. All machines can perform all the operations on any part type given necessary tools are provided. It is assumed that the number of tool copies of each tool type is predetermined. There are versatile machines in the system, and each machine can perform all the operations of parts. The parts don't move between machines but only travel between the load/unload station and the machines. It can be viewed as a group of independent flexible machining modules (Koo and Tanchoco, 1999). While minimal part movement makes the part routing problem relatively simple, dynamic tool assignment is often used for large variety of different part types with a limited tool inventory level.

Figure 3.8 illustrates an example system under study. Machines are subject to random breakdowns. Parts arrive dynamically over time at the load /unload station. One operator works

at the load/unload station placing parts on some types of fixtures before releasing them into the system. When all machines are busy, parts are transported to a central buffer. Each machine can process all the operations of any part type if required tools are available. After a machine completes processing of an operation, it checks whether tools required for the next operations are on its tool magazine. If they are not available at local tool magazines, a request to borrow tools from other machines will be sent to the central tool dispatcher. If more than one machine request a same tool, the dispatching decision will be made using the proposed rolling horizon approach at the machine-level scheduling.



Part i	Machine j	Xij
1	1	1
	2	0
	3	1
	4	1
	5	0
2	1	1
	2	0
	3	0
	4	0
	5	1
3	…	

Tool t	Machine j	Ytj
1	1	1
	2	0
	3	1
	4	1
	5	0
2	1	1
	2	0
	3	0
	4	0
	5	1
3	…	

Figure 3.8. Tool Loading Problem under Flexible Tooling Strategy

The decisions need to be made in the loading stage are:

1. Allocate part types to multiple machines and generate a list of alternative routing for each part type.
2. Assign copies of each tool type to multiple machines that require these tools to process part types allocated to them.

The overall performance measure of the system under studied is due-date related measures such as Mean Tardiness, Percent of Tardy Job, and Mean Flow Time. These performance measures cannot be optimized through the loading decision since they are further affected by the real-time scheduling and control decisions due to the dynamic nature of the shop floor. However, the machine loading and initial tool allocation decisions still have a significant impact on the routing flexibility and the tooling handling traffic, and hence the overall scheduling performance of the system. Maximizing routing flexibility is used as a surrogate objective at the loading stage to improve the overall system performance. The trade-off between routing flexibility and the tool handling traffic is explicitly considered by maintaining a minimum resident tool ratio (MRTR) for each part type's alternative routing. The measure used for routing flexibility is the total number of alternative routes and weighted by the part mix ratio. The part type with a higher mix ratio should is expected to have more alternative routes than the low volume part type when tooling resource is limited.

3.5.2 Loading Model Formulation

The following notation for indices, parameters, and decision variables is introduced for the purpose of model formulation.

Indices:

i – index of part type $i = 1, 2, \ldots N$

j – index of machines $j = 1, 2, \ldots M$

t – index of tool types $t = 1, 2, \ldots T$

Parameters:

a_{tj} = 1, if tool type t already loaded at machine j; 0, otherwise.

b_{it} = 1, if tool type t is required by part type i; 0, otherwise.

m_t - number of tool copies of type t available.

c - magazine capacity.

r_i - product mix ratio of part type i, which equals to the percentage of the total volume of part type i.

β - Minimum resident tool ratio, ratio of number of tools loaded to number of tools required

Variables:

x_{ij} = 1, if part type i is assigned to machine j; 0, otherwise.

y_{tj} = 1, if tool type t is to be loaded at machine j; 0, otherwise.

<u>*Loading Model*</u>

Maximize

$$\sum_{i=1}^{N}\sum_{j=1}^{M} r_i x_{ij} \qquad (39)$$

Subject to:

$$\sum_{t=1}^{T} x_{ij}(y_{tj} + a_{tj})b_{it} \geq \beta \sum_{t=1}^{T} x_{ij} b_{it}, \forall i, j \qquad (40)$$

$$y_{tj} \leq 1 - a_{tj}, \ \forall j, t \qquad (41)$$

$$\sum_{j=1}^{M} y_{tj} \leq m_t, \forall t \qquad (42)$$

$$\sum_{t=1}^{T} y_{tj} \leq c - \sum_{t=1}^{T} a_{tj}, \forall j \qquad (43)$$

$$x_{ij} \in \{0, 1\}, \ \forall i, j \qquad (44)$$

$$y_{tj} \in \{0, 1\}, \ \forall t, j \qquad (45)$$

Parameter a_{tj} is introduced for the purpose of applying a loading method in the event of machine breakdowns or preventive maintenance. Some tools might already be on the tool magazines from a previous loading. The tools on the down machines are partitioned and loaded into the tool magazines of machines while the location of existing tools is taken into account. The objective function is to minimize the routing flexibility as defined in (39). Constraint set (40) denotes the restriction that each part should be assigned to machines, where a minimum

resident tool ratio can be maintained in the tool magazines. The left side of (40) is the number of tools, which are either already on the machine or to be loaded on the machine when part type i is assigned to the machine j. The right side of (40) is the number of tools required to maintain a minimum resident tool ratio (MRTR) β. Constraint set (41) simply fixes the value of y_{tj} to 0 if there is already a tool type t available on machine j. Constraint set (42) denotes limitation of available tool copies of each tool type. Constraint set (43) specifies the tool magazine capacity. Nonlinear constraint *(40)* can be linearized to constraint *(46)* by adding an additional 0-1 variable z_{ijt} *(49)* and two additional linear constraints *(47)* and *(48):*

$$\sum_{t=1}^{T} z_{ijt} b_{it} \geq \sum_{t=1}^{T} x_{ij} (\beta - a_{tj}) b_{it}, \forall i, j \qquad (46)$$

$$2z_{ijt} \leq x_{ij} + y_{tj}, \quad \forall i, j, t \qquad (47)$$

$$z_{ijt} \geq x_{ij} + y_{tj} - 1 \quad \forall i, j, t \qquad (48)$$

$$z_{ijt} \in \{0, 1\}, \quad \forall i, j, t \qquad (49)$$

Minimum Resident Tool Ratio (MRTR) depends on the actual tool transport system capacity. When tool transport speed is low, and the tool-handling system is congested, a higher MRTR should be used. When transport speed is high, a lower MRTR can be used to generate better machine pooling effect.

3.5.3 Heuristic Solution Procedure

The integer-programming model formulated in the previous section can be implemented in the commercial optimization package such as ILOG/CPLEX, and the optimal loading solution can be obtained using the built-in Branch & Bound algorithm in the solver.

However, the above formulation involves a large number of constraints and binary variables, and it is very likely that an optimal solution for a large size problem will require excessive computation time. In order to implement the loading repeatedly in short-term for large size problems, a heuristic solution method is proposed in this study.

The following heuristic method is motivated by the MIMU (Minimum Intersection Maximum Union) rule developed for job and tool grouping (Tang and Denardo 1988). It attempts to assign part types and load tools sequentially, and check minimum resident tool ratio (MRTR) for each part type assignment after each step. The part type with largest number of common tools is selected first, and assigned to the machines with the smallest number of tools loaded. Workload balance is considered implicitly by assigning part types with higher part mix ratio first.

Figure 3.9 show the flow diagram of proposed heuristic procedure.

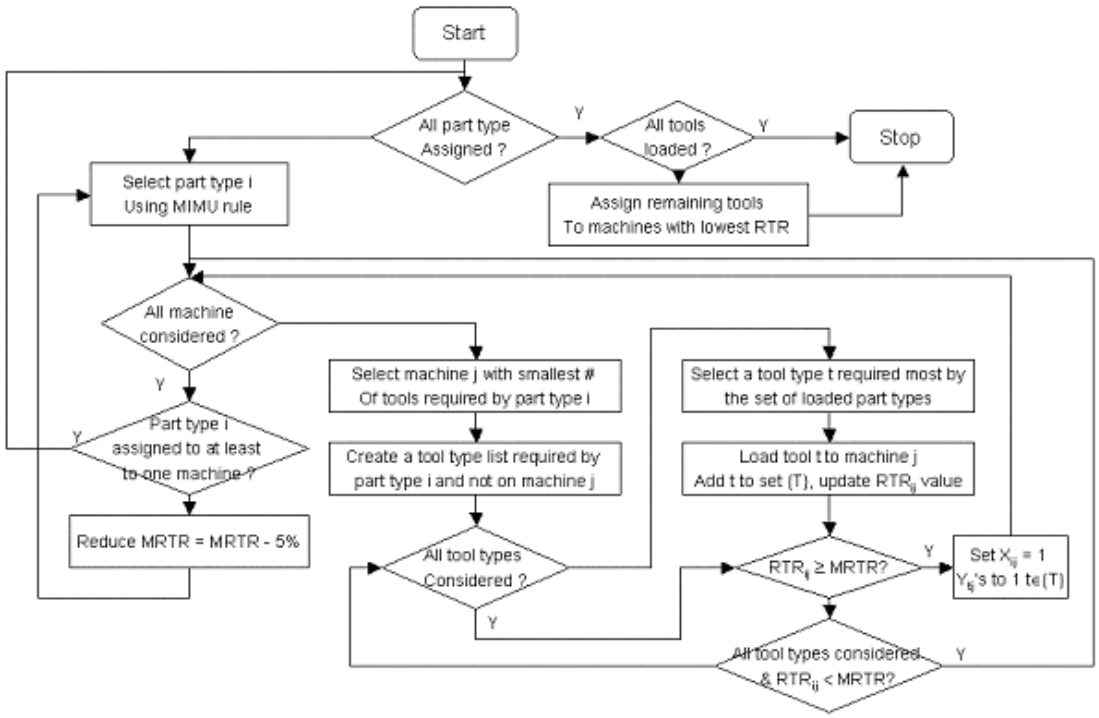


Figure 3.9. Flow Diagram of Proposed Loading Heuristic

The detail steps of the proposed heuristic are described as follows.

The Loading Heuristic

Step 0. Create a list of unassigned part types $\{P_0\}$ and a list of assigned part types $\{P_1\}$. If $\{P_1\} = \varnothing$, then select a part i which requires largest number of tools, break tie by selecting the part type with highest product mix ratio r_i. Create a list of unassigned machines $\{M_0\}$, an empty list of assigned machines $\{M_1\}= \varnothing$, and an empty list of machines considered $\{M_2\} = \varnothing$. Go to step 2.

Step 1. 1-a) Check if all part types in $\{P_0\}$ have been assigned to machines. If $\{P_0\} = \varnothing$, go to step 4. Otherwise continue.

1-b) Select a part type i (based on MIMU rule) which requires largest number of common tools with part types already assigned in $\{P_1\}$, break tie by selecting the part type with highest product mix ratio r_i.

Step 2. 2-a) Check if all machines in $\{M_0\}$ have been considered. If $\{M_0\}= \varnothing$, check if part type i has been assigned to at least one machine. If yes, go to Step 1-a), otherwise reduce the MRTR ratio by 5% and go to Step 1-c). If $\{M_0\}\neq \varnothing$ continue.

2-b) Select a machine j from $\{M_0\}$ with smallest number of tools required by part type i, break tie by selecting the machine with smallest number of tools. Remove machine j from $\{M_0\}$.

2-c) Create a list of tool types $\{T_0\}$ needed by part type i that have not been loaded on machine j. The available number of tools is considered when $\{T_0\}$ is created. Let $\{T_1\} = \varnothing$, a list of tools already pre-selected for machine j.

Step 3. 3-a) Check if all tools in $\{T_0\}$ have been considered. If yes, go to Step 3-c), otherwise continue.

3-b) Select a tool t type from $\{T_0\}$, which is required most by part types in $\{P_0\}$, and break tie by selecting the tool type required most by the part types in $\{P_1\}$ list. Add tool type t to $\{T_1\}$, a list of tools that are already pre-selected for machine j.

3-c) Check if the resident tool ratio (RTR) will be greater than the MRTR after all the pre-selected added to machine j. If yes, assign part type i to machine j (x_{ij} = 1), and load all pre-selected tool types in $\{T_1\}$ to machine j (y_{tj} =1, $\forall t \in \{T_1\}$). Go to Step 2-a). If not, go to Step 3-b) if $\{T_0\} \neq \varnothing$; Otherwise, go to Step 2-a)

Step 4. Assign each of remaining tool types to machines with an assigned operation i requiring tool types the resident tool ratio is lowest. Break tie by loading tools to machines with smallest number of tools.

Appendix C4 shows the Visual Basic Implementation of the proposed loading heuristic.

Events such as machine breakdowns or preventive maintenance could make the existing loading plans impracticable (thus requiring a revision of the plan). So a new loading plan should be constructed after such a disruption. The new loading plan should take into account not only the active capacity of machines, but also the current location of tools at various machines. The proposed heuristic algorithm can be implemented to generate new loading plans online, and reassign part types to machines and reload tools (from the machine that is down) to other active machines. This requires the loading method to be computationally efficient. Tool transport system can be used to migrate tools in a 'bulk exchange' manner to other active machines. When machine becomes up again, the tools can be migrated back to the original machine in batches.

3.5.4 A Numeric Example

The following is a numerical example of the loading problem described before. There are total 10 part types, 5 machines, and 50 tool types in this example. Table 3.1 gives the part mix ratio of each type. The tooling requirement of each part type is shown in Table 3.2.

Table 3.1. Part Mix Ratio for the Example Loading Problem

Part Type No	Num Of Operations	Process Time	Part Mix Ratio r_i
1	12	14.9	0.12
2	22	30.46	0.113
3	18	20.95	0.057
4	12	14.61	0.095
5	15	18.57	0.067
6	12	16.06	0.114
7	16	17.19	0.085
8	15	17.39	0.132
9	17	19.3	0.091
10	18	21.1	0.126

Table 3.3 and 3.4 show the solution obtained using the proposed loading heuristic for part type allocation and tool loading.

Table 3.2. Part Tool Matrix for the Example Loading Problem

Tool Type t	Part Type I											Tool Type t	Part Type I										
	m_t	1	2	3	4	5	6	7	8	9	10		m_t	1	2	3	4	5	6	7	8	9	10
1	2	0	1	0	1	1	0	0	1	0	1	26	2	0	1	1	0	1	0	0	1	1	1
2	3	0	0	0	0	0	0	0	0	0	1	27	2	0	1	0	1	0	0	0	0	0	0
3	3	0	0	1	0	0	1	0	0	1	1	28	2	0	0	0	1	1	0	0	0	1	0
4	3	0	0	0	1	0	1	0	1	0	1	29	2	0	1	1	0	1	0	0	0	0	0
5	2	0	1	0	0	1	0	0	0	1	1	30	2	0	0	1	0	0	0	1	0	0	0
6	2	0	0	1	0	0	0	1	0	0	1	31	3	1	0	0	0	0	0	0	1	0	0
7	2	1	0	0	0	0	0	0	0	0	0	32	3	0	1	0	0	0	0	1	0	1	0
8	2	1	1	0	0	0	1	0	0	0	1	33	3	1	1	0	0	0	0	1	1	0	0
9	2	0	0	0	0	1	0	0	0	0	0	34	2	0	0	0	0	0	0	0	1	1	1
10	2	1	1	0	0	0	0	0	1	0	0	35	2	0	0	0	0	1	0	0	0	0	0
11	2	0	0	1	1	0	0	0	0	1	0	36	2	0	0	0	0	1	0	0	0	0	0
12	3	0	0	1	0	0	0	0	0	1	1	37	2	0	1	0	1	0	0	0	0	0	0
13	3	0	0	1	0	0	0	1	0	1	1	38	2	0	0	0	0	1	0	0	0	0	0
14	2	0	0	0	0	0	1	1	0	0	0	39	2	0	0	0	0	0	0	0	0	1	1
15	2	0	1	1	1	0	0	1	1	0	0	40	2	1	0	1	1	0	0	1	0	0	1
16	2	1	1	1	0	0	1	0	0	0	0	41	2	1	1	1	0	0	0	0	0	0	0
17	3	0	0	1	0	0	1	1	1	0	0	42	3	0	1	0	0	0	1	0	0	1	1
18	3	1	0	0	1	0	0	0	0	1	0	43	3	0	1	0	0	1	0	1	0	0	0
19	2	0	0	1	0	0	1	1	1	1	1	44	2	0	1	1	1	0	0	0	0	0	0
20	2	0	0	0	0	1	0	0	0	0	1	45	3	0	1	0	0	0	1	0	1	1	1
21	3	0	1	0	0	0	1	1	0	1	0	46	2	0	1	0	0	1	1	0	0	0	1
22	3	0	0	0	0	1	0	0	0	0	0	47	3	1	1	0	0	0	0	1	1	0	0
23	2	1	1	0	0	0	0	0	1	0	0	48	2	0	1	1	1	0	1	1	0	0	0
24	2	1	0	0	0	1	0	0	0	0	0	49	2	0	0	1	0	0	0	1	0	0	0
25	2	0	0	0	0	0	0	0	1	1	0	50	3	0	0	1	1	1	0	1	1	1	0

Table 3.3. Part Type Allocation Solution for the Example Loading Problem

Part Type #	Station #	Xij	BETAij	Part Type #	Station #	Xij	BETAij
	1	1	0.833		1	0	0.417
	2	0	0.333		2	1	0.75
	3	0	0.167		3	1	0.833
	4	0	0.25		4	0	0.333
1	5	1	0.75	6	5	0	0.167
	1	1	0.5		1	0	0.375
	2	1	0.545		2	1	0.875
	3	0	0.409		3	1	0.688
	4	0	0.318		4	0	0.25
2	5	1	0.545	7	5	0	0.312
	1	0	0.444		1	1	0.867
	2	1	0.556		2	1	0.8
	3	1	0.5		3	0	0.267
	4	0	0.444		4	0	0.2
3	5	0	0.333	8	5	0	0.333
	1	0	0.417		1	0	0.294
	2	0	0.417		2	1	0.529
	3	0	0.25		3	1	0.529
	4	1	0.667		4	1	0.706
4	5	1	0.5	9	5	0	0.471
	1	1	0.6		1	0	0.278
	2	0	0.467		2	0	0.444
	3	0	0.4		3	1	0.778
	4	0	0.4		4	1	0.722
5	5	0	0.333	10	5	0	0.167

BETAij is actual resident tool ratio for assigning part type i to machine j. Xij is equal to one when part type i is assigned to machine j.

Table 3.4. Tool Assignment Solution for an Example Problem

ToolNo	Tool Type #	Station No	ToolNo	Tool Type #	Station No	ToolNo	Tool Type #	Station No	ToolNo	Tool Type #	Station No
1	1	1	31	13	4	61	26	2	91	40	3
2	1	2	32	14	2	62	27	4	92	40	4
3	2	3	33	14	3	63	27	5	93	41	1
4	2	4	34	15	1	64	28	4	94	41	5
5	2	5	35	15	2	65	28	5	95	42	2
6	3	2	36	16	1	66	29	1	96	42	3
7	3	3	37	16	5	67	29	5	97	42	4
8	3	4	38	17	1	68	30	2	98	43	2
9	4	1	39	17	2	69	30	3	99	43	3
10	4	2	40	17	3	70	31	1	100	43	5
11	4	3	41	18	1	71	31	2	101	44	4
12	5	3	42	18	4	72	31	5	102	44	5
13	5	4	43	18	5	73	32	2	103	45	1
14	6	3	44	19	1	74	32	3	104	45	2
15	6	4	45	19	2	75	32	5	105	45	3
16	7	1	46	20	3	76	33	1	106	46	3
17	7	5	47	20	4	77	33	2	107	46	4
18	8	3	48	21	2	78	33	5	108	47	1
19	8	4	49	21	3	79	34	3	109	47	2
20	9	2	50	21	5	80	34	4	110	47	5
21	9	5	51	22	1	81	35	1	111	48	2
22	10	1	52	22	2	82	35	3	112	48	3
23	10	2	53	22	3	83	36	1	113	49	2
24	11	4	54	23	1	84	36	4	114	49	3
25	11	5	55	23	5	85	37	4	115	50	1
26	12	3	56	24	1	86	37	5	116	50	2
27	12	4	57	24	5	87	38	1	117	50	4
28	12	5	58	25	4	88	38	2			
29	13	2	59	25	5	89	39	4			
30	13	3	60	26	1	90	39	5			

3.6 Summary of Major Assumptions

The following are some of the assumptions and implementation aspects of the proposed method in this research.

1) Tool life is assumed to be deterministic even though actual tool life could be stochastic. This assumption is justified by the fact that many companies are using in-process probing device to prevent the catastrophic tool breakage, and adopting policies that tools are reconditioned or replaced when they still has small amount of tool life remaining.

2) The batch size of jobs is assumed to be one. This should be interpreted as the minimum batch size allowed. The purpose of this assumption is to provide the system more Process or Mix flexibility without setting up the machines for a specific batch of jobs under dynamic tool sharing environment. In the actual situation, the size of a customer order can consist of more that one part.

3) All jobs should be released into the shop floor if the required pallets are available in this case. Otherwise, certain pre-release planning (e.g. flexible batching concept,) should be performed dynamically to prioritize the jobs released next. Discussion of part-release control is considered outside the scope of this research.

4) Random machine failure can occur during an operation of a job, between two consecutive sequences. Each machine's time before failure and time to repair are exponentially distributed. Jobs are not rerouted in case of a normal or minor machine breakdown. A job's operation is resumed after the machine becomes up again.

5) In the Single Stage Systems, each machine can process all the operations of any part type if required tools are available. After a machine completes processing of an operation, it checks whether tools required for the next operations are on its tool magazine. If they are not available at local tool magazines, a request to borrow tools from other machines will be sent to the central tool dispatcher. If more than one machine requesting a same tool,

the dispatching decision will be made using the proposed rolling horizon approach at the machine-level scheduling.

Chapter 4 Simulation Experiments and Interpretation of Results

Three types of experimental studies were performed in this research in order to

1) Test the effectiveness of the proposed scheduling approach compared to the conventional dispatching heuristics under different manufacturing environments.
2) Investigate the effects of the rolling horizon length on the performance of the proposed scheduling approach.
3) Test the performance of a proposed tool loading method under Single Stage Multi-machine systems.

For different experiments, manufacturing model instances were created by specifying some of the parameters, while other parameters were chosen randomly from given ranges of values. The proposed dynamic scheduling approach was tested on two types of manufacturing system models. First, hypothetical models, which were generated based on generic shop flow structure and configurations, were studied in order to provide more general empirical results about the applicability and the performance of the proposed approach under various production environments. The effect of rolling horizon length was also investigated using the hypothetical models. Second, an industrial example model based on an actual industrial flexible manufacturing system is studied in the experiment to see how it performs when machine types, part routing, tooling, and other parameters are closer to the real metal-cutting operations. Lastly, the performance of the proposed tool loading method was tested under hypothetically generated single stage system models, Tool Loading Model.

Section 4.1 describes what types of models are selected for the experiments and what factors are chosen at various pre-determined levels. Statistical analysis procedures used in these studies are explained next. Section 4.3 provides detail description of the simulation models and input parameters. Experiment results are analyzed and interpreted in Section 4.4.

4.1 Experimental Models and Factors

4.1.1 Hypothetical Models

The hypothetical system models are based on three types of generic shop flow structures:

- Flexible Job Shop

Job shop flow structure is very common not only in flexible manufacturing systems but also in many other production systems. Parts are produced in multiple processing stages. There are alternative machines at each stage. The directions of parts' process flow in the system can be different.

- Flexible Flow Shop

In the flexible flow shop, parts are also produced in multiple processing stages. There are alternative machines at each stage. However, the parts' process flow is usually in the same direction. Many flexible-machining systems with U-layouts, and some group technology based layouts can be characterized by this type of shop flow structure.

- Single Stage System

There are versatile machines in the system, and each machine can perform all the operations of parts. The parts don't move between machines but only travel between the load/unload station and the machines. It can be viewed as a group of independent flexible machining modules (Koo and Tanchoco, 1999). While minimal part movement makes the part routing problem relatively simple, dynamic tool assignment is often used for large variety of different part types with a limited tool inventory level.

Three performance measures, mean tardiness, average flow time, and percentage of tardy jobs are used. Their definition can be found in Section 1.4. Five experimental factors are considered for hypothetical models:

- Shop flow structure,
- Number of machines,
- Number of operation steps,
- Shop-load levels.

- Scheduling approaches

Three types of shop flow structures as described above: Job Shop, Flow Shop, and Single-Stage Systems are used in the experiment. Number of machines usually determines the size and production volume of a system. A low level of 6 machines and a high level of 9 machines are used for this factor. It is assumed that parts are processed in 3 stages. For each stage, there are 2 alternative machines for the six-machines systems, and 3 alternative machines for the nine-machines system. Parts are assumed to arrive in the shop floor based on the poisson process.

Operation step refers to a processing step of a part with a different tool type. It is related to the processing time of a part and the number of tool types used. A low level of 10 and a high level of 15 operation steps is chosen at each stage of job shop and flow shop models. For single stage models, a low level of 20 steps and a high level of 30 steps are used to make the overall processing time of parts comparable to those of job shop and flow shop systems. Processing time of each step is randomly generated using a uniform distribution of (1, 5) minutes, which include the tool-changing time. Tools are shared among the various part types being manufactured. They are randomly loaded on each workstation magazine at the beginning of the production.

Shop-load level is selected as one of the experimental factors for testing the robustness of the proposed approach under demand changes. The shop-load level is determined by the maximum utilization of all machine types. The scheduling approach is the treatment factor with 10 levels, which include the proposed approach (denoted ORW) and 9 other dispatching heuristics as follows. Their definitions can be found in Section 2.4.

EDD	Earliest Due-Date
SPT	Shortest Process Time
CR	Critical Ratio rule
SLACK	Minimal Slack time
SRPT+SPT	Shortest Remaining Process Time plus Shortest Process Time
CR+SPT	Critical Ratio plus Shortest Process Time
COVERT	Cost OVER Time
ATC	Apparent Tardiness Cost
BD	Bottleneck Dynamics

Table 4.1 summarizes all the experimental factors and levels used for different hypothetical models.

Table 4.1. Experimental Factors and Levels for Hypothetical System Models

Experimental Factors	**Number of levels**	**Levels**
A. Shop flow structure	3	Job shop (JS), Flow shop (FS), Single-stage (SS).
B. Number of Machines	2	6 (2 machines each stage) 9 (3 machines each stage)
C. Number of operation steps	2	10 and 15 for job shop and flow shop 20 and 30 steps for single stage system
D. Shop-load level	2	75% and 90%
E. Scheduling approaches	10	ORW (the proposed approach), EDD, SPT, SLACK, CR, SRPT/SPT, CR/SPT, COVERT, ATC, BD

There were total of 3x2x2x2x10 = 240 cases for experimental studies using hypothetical system models. It is assumed that machine downtime level is at 10%. There are 10 part types

produced in each scenario. Analysis of variance (ANOVA) F-test, multiple comparison, and interaction plots were used to investigate the significance of different experiment factors (A, B, C, D, and E), and interactions between them. The investigation of interactions between the scheduling approach factor and other factors helps to identify under what conditions the proposed approach works best. Since the interactions between more than two factors are often insignificant, they are therefore assumed negligible.

Table 4.2 shows all the experimental factors and levels used for testing horizon length effects.

Table 4.2. Experimental Factors and Levels for Testing Horizon Length Effect

Experimental Factors	**Number of levels**	**Levels**
A. Shop flow structure	3	Job shop (JS), Flow shop (FS), Single-stage (SS).
B. Number of Machines	2	6 (2 machines each stage) 9 (3 machines each stage)
C. Number of operation steps	2	10 and 15 for job shop and flow shop 15 and 30 steps for single stage system
D. Rolling horizon length	7	3, 6, 9, 12, 15, 100, and 200 minutes.

For each of 3x2x2 = 12 combinations of shop flow structures, number of machines, and number of operation steps in hypothetical systems models, seven levels of the rolling horizon lengths will be used in simulations to study the effect on the performance of the proposed scheduling approach. Shop-load is set at the 90% level. Machine downtime level is 10%. ANOVA F-test, and multiple comparisons were used to test the significance of the rolling horizon length on the system performance.

4.1.2 Industrial Example Model

A system model based on an actual industrial Flexible Manufacturing System at the Caterpillar East Peoria Plant is used to compare the proposed scheduling approach with other dispatching heuristics. Detail system description can be found in Section 4.31.

Table 4.3. Experimental Factors and Levels for Industrial Example Model

Experimental Factors	**Number of levels**	**Levels**
A. Shop-load level	3	60%, 75% and 90%
B. Machine breakdowns	3	3%, 6%, and 10%
C. Scheduling approaches	10	ORW (proposed approach), EDD, SPT, SLACK, CR, SRPT/SPT, CR/SPT, COVERT, ATC, BD

Table 4.3 summarizes all the experimental factors and levels used for industrial example models. There are total of 3x3x10 = 90 cases for experimental studies. Analysis of variance (ANOVA) F-test, and multiple comparisons were used to test the significance of the treatment factor (scheduling approaches). The interaction plot will be used to investigate the interactions between the shop-load levels and machine breakdowns and the scheduling approaches.

4.1.3 Tool Loading Models

Table 4.4 lists the experiment factors and levels for testing the performance of the proposed tool loading method under the single stage systems. Tool Carrier Speed, number of machines, and number of operation steps are used as experimental factors in additional to the loading methods. Song et al. (1995) studied loading problem under the tool movement policy and presented a heuristic algorithm to group parts and tools based on similarities among parts. However, preliminary simulation studies showed that their approach is dominated by the proposed loading method and the random approach under the system being studied. Table 4.5 lists the average flow time of three loading methods under single stage models with a tool carrier speed of 180 m/min. The average flow time of the approach based on Song et al. (1995) (denoted SONG) is 3 to 4 times of those in the models with the proposed approach (denoted MRF) and

random loading approach (RAND). Therefore, only the proposed loading approach and random loading are used as in the simulation studies.

Table 4.4. Experimental Factors and Levels for Tool Loading Model

Experimental Factors	**Number of levels**	**Levels**
A. Tool Carrier Speed	2	100 and 180 meter/min
B. Number of Machines	2	6 and 9
C. Number of operation steps	2	20 and 30 steps
D. Loading methods	2	Proposed heuristic (MRF) Random loading (RAN)

There are total of 2x2x2x2 = 16 cases for experimental studies for the performance of the proposed loading method under the single stage systems. Analysis of Variance (ANOVA) F-tests are used to test the significance of the treatment factor (loading approaches) and the interaction between factors. Paired-t tests are used to test if there is any significant difference between the loading approaches.

Table 4.5. Comparison on Average Flow Time of Different Loading Methods in a Preliminary Study

Number of Machines	Number of Operation		Average Flow Time		
			RAND	**MRF**	**SONG**
6	10	Mean Std Dev	149.99 56.63	133.51 39.42	652.92 95.75
6	15	Mean Std Dev	131.88 16.52	129.28 14.08	423.53 32.74
9	10	Mean Std Dev	104.66 13.30	99.07 11.23	423.53 32.74
9	15	Mean Std Dev	132.57 5.85	103.35 6.75	481.19 40.99

4.2 Statistical Analysis Procedures

4.2.1 Simulation Approach

The type of simulation study used in this study is non-terminating system simulation. There are two reasons for this choice. First, the type of manufacturing environment is characterized with large part-type varieties and low volume. Second, it is an automated production system, and machine setup (especially tooling) is done in a continuous and automated manner. Therefore, the production system under study can produce parts in a "non-terminating" manner without events such as batch machine setups. Non-terminating simulation is used for the experiments in this research.

For non-terminating system simulation, both Replication and Batch-Mean approaches can be used to collect simulation statistics. Replication approach can ensure the independence of sample observations by using different random number seeds. However, the observations during the warm-up period need to be removed for all replications, which may be a waste of computer time if the warm-up period is very long. Batch-Mean approach uses a single long simulation run and divides it into batches or sub-intervals. Each sub-interval can then be used as an observation. The assumption being made here is that if the sub-intervals are sufficiently large then the observations obtained from each sub-interval will be uncorrelated. To ensure that this is indeed the case, it is necessary to test for independence between the batches. It should be noted that Batch-Mean approach is preferred only when an analyst wants to reduce the waste of total simulation time due to the warm-up period. Replication approach is used in this research because it can ensure independent observations by using different random number seeds.

4.2.2 Warm-up Period

A simulation model may have some parameters that become stable after a certain period of time. For example, if the shop is empty at the beginning of the simulation, the number of jobs in the shop may become relatively stable after a warm up period. Some approximation formula to estimate this transient period length is available for only simple queuing systems. But, for most

general systems there is no approximate formula for the length of warm-up period. Usually a graphical procedure (Welch's) based on observations from a number of simulation trial runs can be used to determine it. Figure 4.1 shows the moving average value of the Work-In-Process (WIP) level of a hypothetical system model during the simulation trial run. The WIP observations first were grouped into batches of 100 minutes long each. Then the moving average values of the WIP batch means are plotted with a moving period length of 10 batches.

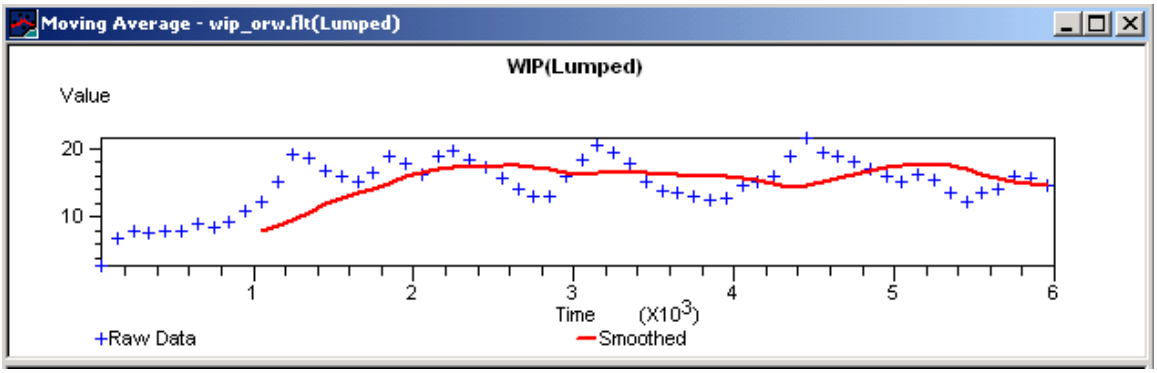


Figure 4.1. Moving Average of Work-In-Process (WIP) Level of the System during a Simulation Trial Run

From the plot, it can be noticed that the smoothed WIP level becomes relatively stable after 2000 minutes. So, a warm-up period of 2000 minutes is used in the simulation experiments, to reduce the impacts of the initial shop floor condition on the experimental results. Since the average length of cutting tool life and the mean time between machine breakdowns (Section 4.3.2) are smaller than the warm-up period, a significant number of disruptions (such as tool replacement and machine breakdowns) have also occurred during the warm-up period.

4.2.3 Number of Replications and Run Length

When the purpose of an experiment is to obtain an accurate confidence interval for a performance measure, a pilot run is carried out with an initial number of replications (n_0) and the half width of the confidence interval (h_0) is calculated. The required number of replications for a desired half width of the confidence interval (h) can often be approximated by the following formula (Kelton, et al. 1998.)

$$n \cong n_0 \frac{h_0^{\,2}}{h^2} \qquad (50)$$

However, when the purpose of an experiment is to test the significance of factors using Analysis of Variance (ANOVA), the number of replications is usually determined by the smallest detectable difference and the mean square of experimental errors to ensure the power ($1-\beta$) of ANOVA F-test, given the probability of type I error is α. The smallest detectable difference is the minimum difference between two extreme treatments (best and worst) that is worth detecting, if such a difference exists. Some pilot experiments are usually conducted to obtain estimates of error mean squares. A table (developed by Bowman and Kastenbaum 1975) in Hinkelmann and Kempthorne's textbook (Design and Analysis of Experiments, 1994) can be used to determine the number of replications based on smallest detectable difference, the estimated error mean square and the values of α and $1-\beta$.

The following is an example of determining the number of replications during the simulation run. First, an arbitrary number of 10 replications were used to conduct a trial run for the hypothetical job shop with nine machines and 15 operation steps. From Figure 4.2, the maximum ($\tau_{\max}$) and minimum ($\tau_{\min}$) values of Average Flow Time of all scheduling approaches are found to be 519.22 and 308.46 minutes respectively. An estimate of the error mean square ($\sigma_e^{\,2}$) is 3153. A standardized minimum difference can be obtained as

$$\Delta^* = \frac{\tau_{\max} - \tau_{\min}}{\sigma_e} = \frac{519.22 - 308.46}{\sqrt{3135}} = 3.76$$

```
Analysis of Variance on Average Flow Time
Source     DF        SS        MS        F        p
Schedule    9    303255     33695    10.69    0.000
Error      90    283797      3153
Total      99    587052
                                   Individual 95% CIs For Mean
                                   Based on Pooled StDev
 Level      N      Mean     StDev  ------+---------+---------+---------+
     1     10    308.46     20.66  (----*---)
     2     10    456.02     58.54                    (---*---)
     3     10    480.70     46.88                       (---*---)
     4     10    497.18     56.57                         (---*----)
     5     10    519.22     70.12                           (----*---)
     6     10    452.59     42.58                    (----*---)
     7     10    454.99     55.30                    (----*---)
     8     10    493.55     80.99                         (----*---)
     9     10    488.48     57.78                         (---*---)
    10     10    455.92     50.93                    (---*---)
                                   ------+---------+---------+---------+
Pooled StDev =    56.15                 320       400       480       560
```

Figure 4.2. One-way ANOVA of Average Flow Time for Hypothetical Job Dhops with Nine Machines and Fifteen Operation Steps

When type I error $\alpha = 0.05$, and the power ($1-\beta$) of F-test is selected as 0.9, a minimum of 4 replications is required to detect the standardized smallest difference of 3.76 for ten treatments (scheduling approaches) based on the table in Hinkelmann and Kempthorne (1994). Therefore, the initial selection of 10 replications is sufficient for the ANOVA F-tests. Ten replications are used for all simulation experiments.

Since the replication mean is treated as a single sample observation, the run length of each replication is determined to ensure that the rare events such as machine breakdowns and tool replacements occur during each replication. In this research a replication run length of 4000 minutes after the warm-up period was used for the experiments.

4.2.4 Analysis of Variance and Comparisons

To test the performance of the proposed scheduling approach, one-way Analysis of Variance (ANOVA) is used to test if there is any significant difference among scheduling approaches for each case in both hypothetical models and industrial example models. Then, Tukey's multiple comparisons with family error rate of 0.05 are used to obtain the confidence intervals for all pair-wise differences between level means. For each shop flow type in the hypothetical models and the industrial example models, multi-way ANOVAs based on General

Linear Model (GLM) are used to test the significance of the interactions between factors. Interactions between more than two factors are assumed to be negligible.

To test the performance of the proposed loading method, Paired-t tests were used to test if there is any significant difference between the proposed loading method and the random loading method.

4.3 Simulation Models and Assumptions

4.3.1 Model Descriptions

This section describes the systems that were modeled and some assumptions made in the simulation experiments.

4.3.1.1 Hypothetical Models

Figure 4.3 shows a layout of the hypothetical models for the simulation experiments, which typically consist of

- Three types workstations (Vertical Machining Center, Vertical Turret Lathe, and Shaper).
- One load/unload station
- One part delivery system with three AGVs
- One tool transport system with three tool carriers
- One central tool room for tool reconditioning

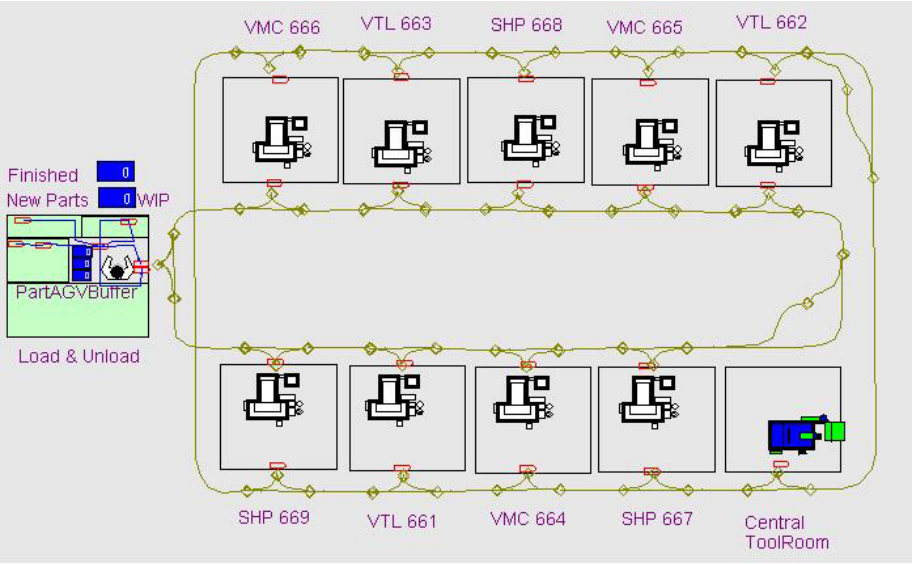


Figure 4.3. Layout of the Hypothetical Job Shop with Nine Machines

Each workstation has an input and output buffer for temporarily storing parts before and after receiving its required operations, respectively. The part delivery AGV system is used to transport parts on three types of fixtures between the loaded/unloaded station and machines. Dedicated tool carriers are used to transport the tools between workstations and the central tool room for the replacement of worn tools. The vehicle path is assumed to be unidirectional. A zone control method is used for the vehicle traffic controls. The operations and processing steps of parts follows the routings and part-tool requirements specified in Section 4.3.2 for three types of shop flow configurations (i.e. flow shop, job shop, and single stage). Tools are randomly loaded on each workstation magazine at the beginning.

There are 10 part types with a fixed part mix ratio simultaneously produced in the systems. The parts arrive at the system according to the customer orders with their own due-dates set by the total work content rule (Conway et al. 1967). The parts scheduling and tool flow control logic follow the proposed cell control scheme and the scheduling framework described in the Chapter 3.

4.3.1.2 Industrial Example Model

The layout of an industrial example model, whose configuration is based on the data from an industrial Flexible Manufacturing Cell at a Caterpillar Peoria Plant, is shown in Figure 4.4. There are five identical vertical turning centers (VTL), four identical vertical machining centers (VMC), two identical gear shaper stations, one wash and debur station, and one coordinate measuring machine (CMM). In the load/unload area, there are four identical L/UL stations and thirteen bi-level storage buffers including four active buffer stands which are the only passages allowing parts to exit to workstations from L/UL area, or return from workstations to L/UL stations. Two identical rail guided vehicles (RGV) serve the load/unload stations and the bi-level storage. Three identical self-guided vehicles (SGV) provide part-handling services between workstations and four active buffer stands. Three overhead tool carriers are used to transport tool between central tool room and workstation tool magazines

The parts are processed at different workstations while they are secured on chuck-pallets and are transported among stations within the Cell. All processing and transporting operations on the palletized parts are completed automatically.

Any manually performed operations on the parts within the Cell occur at the Load/Unload stations where parts are loaded and unloaded to/from the chuck-pallets. That means parts always enter the Cell after manual loading. Parts also may be removed from one type of chuck-pallet, and placed onto another type of chuck-pallet for further processing. Some manual assembly and deburring operations may also be performed at Load/unload stations.

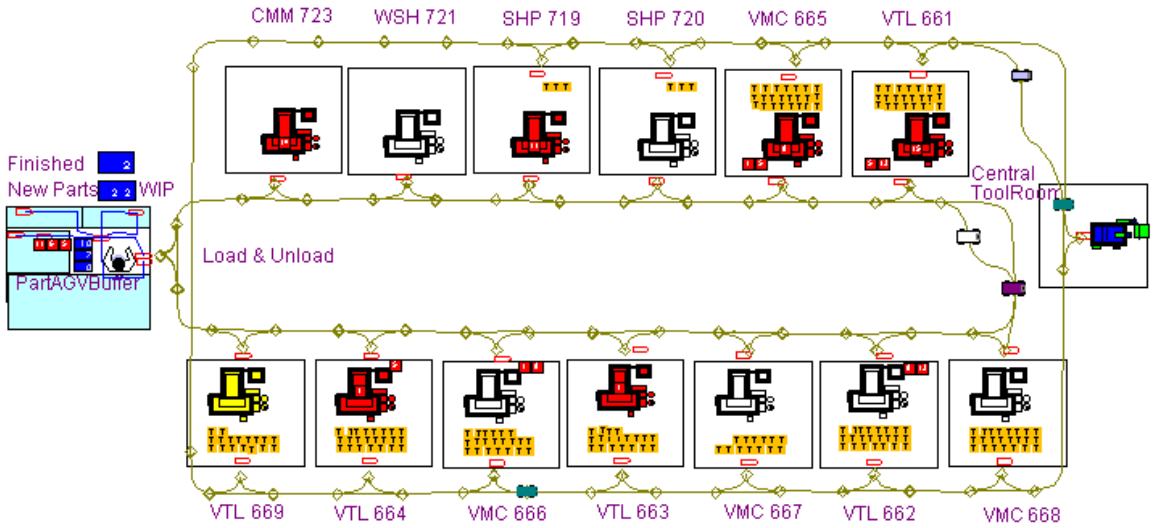


Figure 4.4. Layout of a Flexible Manufacturing System at Caterpillar, East Peoria, IL

Chuck-pallets not currently in use are stored in the central storage system, from where a chuck-pallet is dispatched for part loading at a load/unload station. After the part is loaded on a chuck-pallet, it is queued in the central buffer storage cubicles if its next machine is not immediately available. Once an appropriate workstation is available, the chuck-pallet with part will be sent via an SGV system to the machining station. After completing the machining operation, the chuck-pallet with part leaves the SGV system and returns to a L/UL station, queued in the storage system, if necessary, to wait for the load/unload station to become available. When the part is unloaded, and if the chuck-pallet is to be used for further part processing, it is immediately reloaded at the station and the cycle repeats, otherwise another type of chuck-pallet is requested. If this type of chuck-pallet is not available, the unloaded part will be sent to WIP racks.

There are three kinds of chuck-pallets used in the example system, including thirteen self-centering pallets (named Pallet A), ten compensating pallets (named Pallet B), and nine flat pallets (named Pallet C). A total 9 different part types are concurrently processed in this system. Part routings and processing and tooling requirement of each operation are based on the actual production data in that manufacturing cell. Like in hypothetical models, the parts scheduling and the tool flow control logic follow the proposed scheduling and control framework described in the Chapter 3.

4.3.1.3 Tool Loading Models

The simulation models used for testing the tool loading methods are similar to that of the single stage shop configurations in the hypothetical models. The number of machines and the number of operations steps of parts, and tool loading methods are experimental factors. At the loading stage, the proposed tool loading method and a random loading method is used to 1) allocate part types to multiple machines and generate a list of alternative routing for each part type, and 2) assign copies of each tool type to multiple machines that require these tools to process part types allocated to them. The proposed rolling horizon approach is used for the job-machine assignment and the tool flow control at the scheduling stage.

4.3.2 Simulation Input data

In addition to the experimental factors and levels, this section summarizes four types of input data for the simulation experiments. They include 1) part types and process routings, 2) machine downtime data, 3) Tool types and tool life, 4) material handling systems data.

4.3.2.1 Part Types and Routings

In the hypothetical models, there are ten part types generated for each of three shop flow structures. For the job shops and flow shops, parts are assumed to be processed in three operation stages. Parts are processed only through one operation in the single stage systems. All part types follow the same routing through three types of machines in the flow shops. The routings of parts in job shops are randomly generated among the six alternative sequences of three machine types. As indicated in section 4.1, the numbers of operation steps at each stage are pre-specified.

Part types and routing data in the hypothetical job shops with nine machines are shown in the Appendix A (Additional Simulation Input Data) Section.

In the industrial example model, the part types and routings are based on the actual production data. Table 4.6 lists the routing data for various part types processed in the original system. The following code is used for workstation types:

VTL Vertical Turret Lathe
VMC Vertical Machining Center
SHP Shaper
WSH Wash and Debur Station
CMM Coordinate Measuring Machine

Table 4.6. Routing Data of Various Part Types in the Industrial Example Model

Part Type	Operation No	Station Type	Pallet Type	Num Of Sequence	Operation Time
A1	1	VTL	1	10	61.98
	2	VMC	1	4	11.89
	3	CMM	1	1	3.76
	4	VMC	2	12	66.92
	5	WSH	2	1	8.45
	6	CMM	2	1	9.93
A2	1	VTL	1	12	75.78
	2	VMC	1	4	16.87
	3	CMM	1	1	6.09
	4	VMC	2	11	63.48
	5	WSH	2	1	7.79
	6	CMM	2	1	2.68
A3	1	VTL	1	12	74.24
	2	VMC	1	4	24.34
	3	CMM	1	1	8.93
	4	VMC	2	11	75.61
	5	WSH	2	1	3.17
	6	CMM	2	1	2.3
B1	1	VTL	1	10	59.57
	2	CMM	1	1	4.6
	3	SHP	2	1	4.4
	4	VMC	2	7	45
	5	WSH	2	1	6.13
	6	CMM	2	1	4.05
B2	1	VTL	1	10	75.53
	2	CMM	1	1	8.8
	3	SHP	2	1	6.46
	4	VMC	2	7	46.65
	5	WSH	2	1	5.01
	6	CMM	2	1	4.64
C1	1	VTL	1	11	79.46
	2	VMC	1	6	33.36
	3	CMM	1	1	8.27
	4	SHP	2	1	7.33
	5	VMC	2	6	37.93
	6	WSH	2	1	7.43
	7	CMM	2	1	9.43
C2	1	VTL	1	12	83.6
	2	VMC	1	6	30.25
	3	CMM	1	1	6.33
	4	SHP	2	1	6.94
	5	VMC	2	6	38.17
	6	WSH	2	1	2.66
	7	CMM	2	1	6.61
C3	1	VTL	1	10	54.89
	2	VMC	1	6	41.77
	3	CMM	1	1	5.98
	4	SHP	2	1	2.88
	5	VMC	2	6	39.79
	6	WSH	2	1	6.36
	7	CMM	2	1	8.46
C4	1	VTL	1	11	58.5
	2	VMC	1	8	43.75
	3	CMM	1	1	9.58
	4	SHP	2	1	4.01
	5	VMC	2	6	45.51
	6	WSH	2	1	9.53
	7	CMM	2	1	4.08

4.3.2.2 Machine Downtime

The machine downtime and up time are assumed to be exponentially distributed. The machine breakdown level is defined by the ratio of MTTR (Mean Time To Repair) over the sum of MTTR and MTBF (Mean Time Between Repair). In the hypothetical system machine downtime level are assumed to be 10%. Three levels of machine breakdown percentage (3%, 6%, and 10%) as shown in the tables 4.7, 4.8, and 4.9 are used in the experiments of the industrial example models. Appendix A.5 shows the machine downtime data for the hypothetical models.

Table 4.7. 3% Machine Downtime Level in the Industrial Example Model

StationNo	StnNo	StnTypeID	MTBF	MTTR
1	661	VTL	829.20	25.65
2	662	VTL	660.41	20.43
3	663	VTL	957.89	29.63
4	664	VTL	1139.46	35.24
5	665	VMC	1130.77	34.97
6	666	VMC	1175.08	36.34
7	667	VMC	608.70	18.83
8	668	VMC	844.45	26.12
9	669	VTL	1117.95	34.58
10	719	SHP	683.15	21.13
11	720	SHP	747.02	23.10
12	721	WSH	627.28	19.40
13	723	CMM	619.43	19.16

Table 4.8. 6% Machine Downtime Level in the Industrial Example Model

StationNo	StnNo	StnTypeID	MTBF	MTTR
1	661	VTL	829.20	52.93
2	662	VTL	660.41	42.15
3	663	VTL	957.89	61.14
4	664	VTL	1139.46	72.73
5	665	VMC	1130.77	72.18
6	666	VMC	1175.08	75.01
7	667	VMC	608.70	38.85
8	668	VMC	844.45	53.90
9	669	VTL	1117.95	71.36
10	719	SHP	683.15	43.61
11	720	SHP	747.02	47.68
12	721	WSH	627.28	40.04
13	723	CMM	619.43	39.54

Table 4.9. 9% Machine Downtime Level in the Industrial Example Model

StationNo	StnNo	StnTypeID	MTBF	MTTR
1	661	VTL	829.20	92.13
2	662	VTL	660.41	73.38
3	663	VTL	957.89	106.43
4	664	VTL	1139.46	126.61
5	665	VMC	1130.77	125.64
6	666	VMC	1175.08	130.56
7	667	VMC	608.70	67.63
8	668	VMC	844.45	93.83
9	669	VTL	1117.95	124.22
10	719	SHP	683.15	75.91
11	720	SHP	747.02	83.00
12	721	WSH	627.28	69.70
13	723	CMM	619.43	68.83

The shop-load level will be the maximum utilization of all machine types. The following (Table 4.10) are the inter-arrival times of jobs with corresponding shop-load levels in the industrial example model. Jobs are assumed to arrive in the shop floor based on the Poisson process.

Table 4.10. Job Inter-arrival Time at Various Shop-load Levels in the Industrial Example Model

Shop-load level	Job inter-arrival time
60%	32 min
75%	28 min
90%	24 min

4.3.2.3 Tool Types and Life

In the hypothetical models, there are total of 100 tool types. Ten (or fifteen) different tool types are randomly chosen for each operation of a part type. Process cutting time required is randomly generated using a uniform distribution between 1 and 5 minutes. The numbers of tool copies are also randomly generated between 2 and 4.

In Appendix A (Additional Simulation Input Data), part-tool requirement matrix for the hypothetical flow shops, job shops, and the single stage systems are listed.

Tool life is assumed to be deterministic and finite and worn tools can be recovered through re-sharpening at the central tool room. From tool life statistics from the existing literature (Kumaran, 1995), average tool life is around 120 minutes for milling and turning tools made of carbide or high-speed steel. Based on the fact that the average cutting time of each tool is about 6 minutes in the industrial example models, the tool life is assumed to 20 times of the average cutting time in this research.

4.3.2.4 Material Handling

The part delivery system consists of three AGV transporters with a velocity of 60 meter/minute and an acceleration/deceleration rate of 3 m/s^2. Three overhead tool transporters are used to exchange tools between the tool room and machines during the production. In both hypothetical models and the industrial example model, the tool transports have a velocity of 180 meters/minute and an acceleration /deceleration rate of 3 m/s^2. Two tool carrier speeds: 100 and 180 meter/minute are used for testing the proposed loading method.

Table 4.11 shows the from-to travel distance (in meters) of part AGVs in the industrial example model. The station numbers 1 through 13 are the same of those in Tables 4.7-7.9 for machine downtime levels. The 14th station represents a load-unload station.

Table 4.11. From-to Travel Distance Chart of Part AGVs in the Industrial Example Model

From /To	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	54	46	38	6	42	50	58	34	14	10	18	22	28
2	10	0	54	46	14	50	58	4	42	22	18	26	30	36
3	18	8	0	54	22	58	4	12	50	30	26	34	38	44
4	26	16	8	0	30	4	12	20	58	38	34	42	46	52
5	58	48	40	32	0	36	44	52	28	8	4	12	16	22
6	22	12	4	58	26	0	8	16	54	34	30	38	42	48
7	14	4	58	50	18	54	0	8	46	26	22	30	34	40
8	6	58	50	42	10	46	54	0	38	18	14	22	26	32
9	30	20	12	4	34	8	16	24	0	42	38	46	50	56
10	50	40	32	24	54	28	36	44	20	0	58	4	8	14
11	54	44	36	28	58	32	40	48	24	4	0	8	12	18
12	46	36	28	20	50	24	32	40	16	58	54	0	4	10
13	42	32	24	16	46	20	28	36	12	54	50	58	0	6
14	36	26	18	10	40	14	22	30	6	48	44	52	56	0

Table 4.12 shows the from-to travel distance of tool carriers in the industrial example modes. The 12th station represents the central tool room.

Table 4.12. From-to Travel Distance Chart of Tool Transporters in the Industrial Example Model

From /To	1	2	3	4	5	6	7	8	9	10	11	12
1	0	58	50	42	4	46	54	62	38	12	8	70
2	20	0	70	62	24	66	74	4	58	32	28	12
3	28	8	0	70	32	74	4	12	66	40	36	20
4	36	16	8	0	40	4	12	20	74	48	44	28
5	72	54	46	38	0	42	50	58	34	8	4	64
6	32	12	4	74	36	0	8	16	70	44	40	24
7	24	4	74	66	28	70	0	8	62	36	32	16
8	16	74	66	58	20	62	70	0	54	28	24	8
9	40	20	12	4	44	8	16	24	0	52	48	32
10	66	46	38	30	70	34	42	50	26	0	74	58
11	70	50	42	34	74	38	46	54	30	4	0	62
12	8	66	58	50	12	54	62	70	46	20	16	0

4.3.3 Experimental Structure

The shop floor control system is modeled under the Arena simulation environment. The integrated dynamic scheduler is interfaced with the virtual shop floor control system in the Arena simulation model through Visual Basic programming. Figure 4.5 illustrates the framework of simulation used in this study. During each scheduling window, minimum-cost flow models for task and resource assignment are built dynamically in MPL modeling language, and then the scheduler is linked to CPLEX solver callable library to solve the network model through windows DLL calls. The jobs and tool schedule decisions are passed back to dynamic scheduler module after the model is solved and the job and machines schedule databases are changed accordingly if these decisions are confirmed. The maximum time used for each decision process is about 9.68 second on a Pentium III 450 MHz PC including time for selecting candidate tasks and resources and formulating and solving the MPL model through CPLEX.

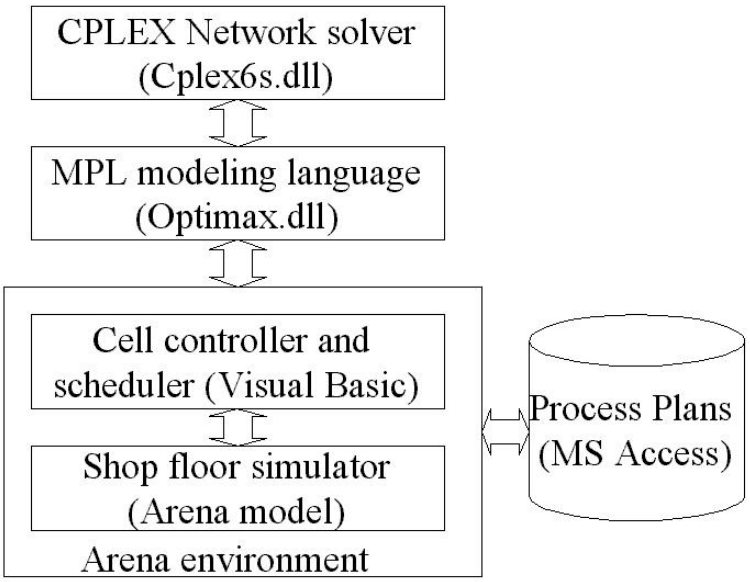


Figure 4.5. Simulation Experiment Framework

Appendix C lists the major modules in the Arena simulation model used in this study. Appendix D lists Visual Basic Codes for Dynamic Scheduler.

4.4 Experimental Results and Analyses

In this Section, the experimental results for testing the performance of the proposed scheduling under both hypothetical models and industrial example models, and the performance of proposed tool loading methods in the single stage systems are reported respectively. For each of the experimental models described in previous Sections, ten replications, with 4,000 minutes of production time in each replication after a warm-up period of 2000 minutes, were performed. From a number of simulation trials runs, it is noticed that the length of rolling horizon of 6 minutes tends to perform relatively better, and the system performance is not sensitive to the changes of the rolling horizon length when it is relatively small. Consequently, the rolling horizon length of 6 minutes is chosen for the proposed scheduling approach in the simulation experiments. In Section 4.4.1.4, this behavior of the rolling horizon length is further validated by the experimental results based on different lengths.

4.4.1 Hypothetical Models

Three performance measures, mean tardiness, average flow time, and percentage of tardy jobs were used in the simulation experiments. A rolling horizon length of 6 minutes is used for both cell-level and machine-level scheduling. For each of the three shop flow structures (Flow shops, Job shops, and Single Stage Systems), One-way Analysis of Variance on three performance measures are conducted at each level of two factors, number of machines and number of operation steps. Tukey's multiple comparisons are used to test any significant difference between the proposed scheduling approach and other dispatching heuristics. A multi-way ANOVA of four factors and interaction plots are then employed to test the significance of each factor and the interactions.

4.4.1.1 Flexible Flow Shops

Figure 4.6 shows the one-way ANOVA on Average Flow Time for Flow shop with nine machine and 15 steps of each operation at 90% shop-load level. Table 4.13 shows the levels of the scheduling approach factor. Level 1 represents the proposed approach.

Table 4.13. Levels of the Scheduling Approach Factor

Scheduling approaches	ORW	EDD	SPT	SLACK	CR	SRPT/SPT	CR/SPT	COVERT	ATC	BD
Level	1	2	3	4	5	6	7	8	9	10

```
Analysis of Variance on Average Flow Time
Source     DF        SS        MS        F        p
Schedule    9    238578     26509     4.63    0.000
Error      90    515254      5725
Total      99    753832
                                  Individual 95% CIs For Mean
                                  Based on Pooled StDev
 Level      N      Mean     StDev  -+---------+---------+---------+-----
     1     10    285.49     41.90   (-----*-----)
     2     10    443.74     67.39                       (----*-----)
     3     10    421.03     40.49                     (-----*-----)
     4     10    453.31     89.52                        (-----*-----)
     5     10    448.53    120.92                       (-----*-----)
     6     10    453.42     91.17                        (-----*-----)
     7     10    446.22     58.83                       (-----*-----)
     8     10    443.56     47.51                       (----*-----)
     9     10    446.44     99.42                       (-----*-----)
    10     10    453.57     52.56                        (-----*-----)
                                  -+---------+---------+---------+-----
Pooled StDev =    75.66           240       320       400       480
Tukey's pairwise comparisons
Family error rate = 0.0500
Individual error rate = 0.00165
Critical value = 4.59
Intervals for (column level mean) - (row level mean)
               1         2         3         4         5         6         7
     2    -268.1
           -48.4
     3    -245.4     -87.1
           -25.7     132.5
     4    -277.6    -119.4    -142.1
           -58.0     100.3      77.5
     5    -272.9    -114.6    -137.3    -105.0
           -53.2     105.0      82.3     114.6
     6    -277.8    -119.5    -142.2    -109.9    -114.7
           -58.1     100.1      77.4     109.7     104.9
     7    -270.6    -112.3    -135.0    -102.7    -107.5    -102.6
           -50.9     107.3      84.6     116.9     112.1     117.0
     8    -267.9    -109.6    -132.4    -100.1    -104.8    -100.0    -107.2
           -48.2     110.0      87.3     119.6     114.8     119.7     112.5
     9    -270.8    -112.5    -135.2    -102.9    -107.7    -102.8    -110.0
           -51.1     107.1      84.4     116.7     111.9     116.8     109.6
    10    -277.9    -119.7    -142.4    -110.1    -114.9    -110.0    -117.2
           -58.3     100.0      77.3     109.6     104.8     109.7     102.5
               8         9
     9    -112.7
           106.9
    10    -119.8    -117.0
            99.8     102.7
```

Figure 4.6. One-way Analysis of Variance and Tukey's Test on the Average Flow Time for Hypothetical Flow Job with Nine Machines and Fifteen Operations Steps

The result of Analysis of Variance on average flow time is shown at the top of Figure 4.6. Individual 95% level confidence intervals fro mean based on pooled standard deviation are plotted next. The proposed approach (level 1) has a mean flow time of 285.49 minutes and a standard deviation of 41.90. As the results of Tukey's pairwise comparison, confidence intervals of differences between the mean from the column levels and the mean from the row levels are shown at the bottom of Figure 16. A P value of less than 0.001 for F-test shows the difference between among scheduling approaches. In the Tukey's comparison, the confidence interval of the difference between the proposed approach (level 1) and the best (level 3) of other dispatching heuristics is (-245.4, -25.7), which indicates the better performance of the proposed approach.

One-way ANOVA and Tukey's test were conducted for all levels of four factors for Flow Shop models.

Table 4.14. One-way ANOVA and Tukey's Test for Hypothetical Flow Shops

			P value of ANOVA F-test			Tukey's Confidence Interval *		
Number of Machines	Number of Operations Steps	Shop-Load	Average Flow Time	Tardiness	Percent of Tardy Job	Average Flow Time	Tardiness	Percent of Tardy Job
6	10	75%	<0.001	0.458	0.026	(-37.81, -8.73)	(-10.003, 3.84)	(-0.13, 0.011)
6	10	90%	<0.001	0.336	0.038	(-53.21, -14.51)	(-15.6 ,5.78)	(-0.20 , 0.016)
6	15	75%	<0.001	0.04	<0.001	(-99.4 , -5.9)	(-44.23 , 16.79)	(-3.2 , 0.0417)
6	15	90%	0.126	0.188	<0.001	(-143.5 ,82.7)	(-137.0 , 82)	(-0.1257 ,0.1542)
9	10	75%	<0.001	0.006	<0.001	(-55.38 , -27.80)	(-16.17 , 0.68)	(-0.1915 , -0.0467)
9	10	90%	<0.001	0.001	<0.001	(-71.95 , -30.99)	(-25.02 , 1.96)	(-0.283 , -0.0649)
9	15	75%	<0.001	0.002	<0.001	(-129.7 , -55.2)	(-44.86 , 0.44)	(-0.406 , -0.103)
9	15	90%	<0.001	0.032	<0.001	(-245.4 , -25.7)	(-172.9 ,26.4)	(-0.7048 , 0.2710)

* Tukey's confidence interval of the difference between the proposed method and the next best dispatching heuristics. (Minutes)

Table 4.14 shows the P-value of the ANOVA F-test on Average Flow Time, Tardiness, and Percent of Tardy Jobs for all levels of experimental factors under flow shop environments.

Tukey's confidence intervals of the difference between the proposed scheduling method and the next best dispatching heuristics are also shown in the table. For the case of six machine, 10 operation steps, and 90% shop load level, the P-value for the average flow time is less than 0.001. A confidence interval of (-53.21, -14.51) on average flow time is found for this case, which shows that the proposed approach generates significantly smaller flow time than the next best dispatching rule. The results indicate that the differences among various scheduling approaches are significant on Average Flow Time and Percent of Tardy Jobs in most cases except in the case of six machines with fifteen operation steps and the shop-load is 90%. According to the Tukey's multiple comparison, the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time in most cases. Percent of Tardy Jobs of the proposed approach is significantly better than that of the next best dispatching heuristics when there are nine machines.

Factor	Levels	Values									
NumMach	2	1	2								
NumOptn	2	1	2								
ShopLoad	2	1	2								
Schedule	10	1	2	3	4	5	6	7	8	9	10

Analysis of Variance for Average Flow Time

Source	DF	Seq SS	Adj SS	Adj MS	F	P
NumMach	1	33322	33322	33322	18.24	0.000
NumOptn	1	8671934	8671934	8671934	4746.89	0.000
ShopLoad	1	1414630	1414630	1414630	774.35	0.000
Schedule	9	355675	355675	39519	21.63	0.000
NumMach*NumOptn	1	176632	176632	176632	96.69	0.000
NumMach*ShopLoad	1	71254	71254	71254	39.00	0.000
NumMach*Schedule	9	37159	37159	4129	2.26	**0.017**
NumOptn*ShopLoad	1	822720	822720	822720	450.34	0.000
NumOptn*Schedule	9	68827	68827	7647	4.19	**0.000**
ShopLoad*Schedule	9	12955	12955	1439	0.79	**0.628**
Error	757	1382938	1382938	1827		
Total	799	13048047				

Figure 4.7. Analysis of Variance on Average Flow Time for Flow Shop Models with All Factors

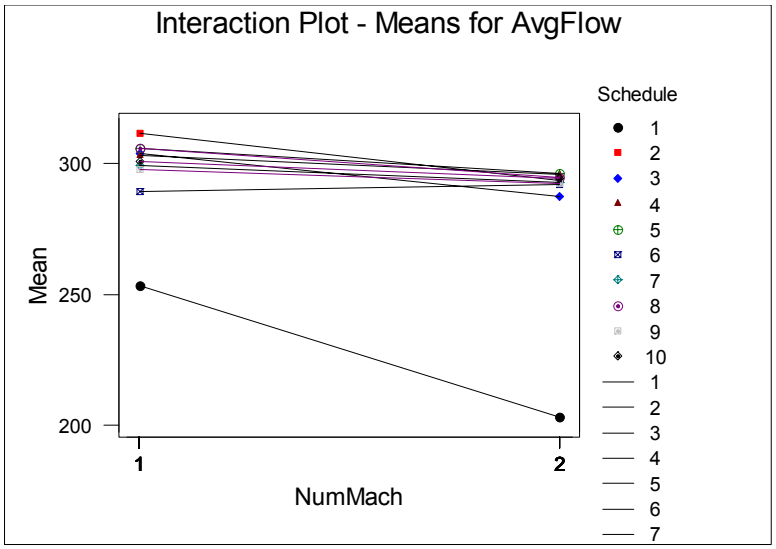


Figure 4.8. Interaction Plot between Number of Machines and the Scheduling Approach on Average Flow Time for Flow Shop Models

Figure 4.7 shows the Analysis of Variance on Average Flow time for all factors in the flow shop models. A P-value of 0.017 indicates that there is a significant interaction between the number of machines factor and the scheduling approach. From the interaction plot in Figure 4.8, it can be noticed that the performance gap between the proposed scheduling approach and the other heuristics tends to become larger when the number of machines is higher.

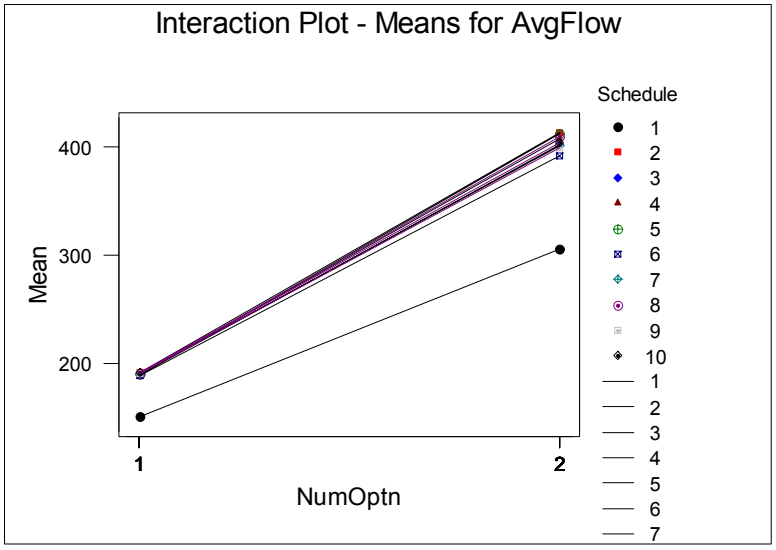


Figure 4.9. Interaction Plot between Number of Operation Steps and the Scheduling Approach on Average Flow Time for Flow Shop Models

The ANOVA also indicates a significant interaction between the number of operations steps and the scheduling approach. From Figure 4.9, it is be found that the higher the number of operation steps, the better the proposed method performs relative to the other heuristics.

The interaction between the scheduling approach and the shop-load is not significant on the average flow time. Although the interactions between the number of machines and the number of operation steps, and the shop load level are also found significant, investigation of those interactions is not the major objective of this research.

4.4.1.2 Flexible Job Shops

Table 4.15 shows the P-value of the ANOVA F-test and Tukey's confidence interval on Average Flow Time, Tardiness, and Percent of Tardy Jobs for all levels of experimental factors under job shop environments. It is found that the differences among various scheduling approaches are significant on Average Flow Time in most cases except in the case of six machines with fifteen operation steps and the shop-load is 90%. According to the Tukey's multiple comparison, the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time 5 out 8 cases. Percent of Tardy Jobs of the proposed approach is significantly smaller than that of the next best dispatching heuristics when there are nine machines. It is similar to the flow shop models in that when the number of machines is higher, the proposed scheduling approach tends to perform relatively better. Analysis of Variance on average flow time for all factors is conducted again to validate this interaction.

Table 4.15. One-way ANOVA and Tukey's Test for Hypothetical Job Shops

			P value of ANOVA F-test			Tukey's Confidence Interval *		
Number of Machines	Number of Operations Steps	Shop-Load	Average Flow Time	Tardiness	Percent of Tardy Job	Average Flow Time	Tardiness	Percent of Tardy Job
6	10	75%	<0.001	0.176	0.003	(-36.95, -0.76)	(-11.11, 7.26)	(-0.14, 0.0426)
6	10	90%	<0.001	0.184	0.021	(-50.41, 5.95)	(-20.27, 14.95)	(-0.2058, 0.0695)
6	15	75%	0.035	0.114	0.063	(-75.6, 32.3)	(-39.8, 41.6)	(-2447, 0.1133)
6	15	90%	0.224	0.246	<0.001	(-182.7, 78.5)	(-180.1, 77.0)	(-0.0665, 0.1384)
9	10	75%	<0.001	0.004	<0.001	(-47.5, -20.92)	(-13.3, 2.09)	(-0.1639, -0.0297)
9	10	90%	<0.001	0.001	<0.001	(-57.99, -23.51)	(-20.29, 3.08)	(-0.2441, -0.07219)
9	15	75%	<0.001	<0.001	<0.001	(-105.2, -30.7)	(-39.08, 7.21)	(-0.33658, -0.04856)
9	15	90%	<0.001	<0.001	<0.001	(-225.6, -62.6)	(-177.4, -27.0)	(-0.5663, -0.2778)

* Tukey's confidence interval of the difference between the proposed method and the next best dispatching heuristics. (Minutes)

Figure 4.10 shows the Analysis of Variance on Average Flow time for all factors in the job shop models. The F-test for the interactions between the number of machines factor and the scheduling approach, the number of operations and the scheduling approach, and the shop-load and the scheduling approach are all significant. Figures 4.11-4.13 show the interaction plots between these three factors. The performance differences between the proposed scheduling approach and the other heuristics tend to become more significant when there are more machines and operation steps and at the higher shop-load.

Factor	Levels	Values									
NumMach	2	1	2								
NumOptn	2	1	2								
ShopLoad	2	1	2								
Schedule	10	1	2	3	4	5	6	7	8	9	10

Analysis of Variance for Average Flow Time

Source	DF	Seq SS	Adj SS	Adj MS	F	P
NumMach	1	316089	316089	316089	162.11	0.000
NumOptn	1	11819149	11819149	11819149	6061.70	0.000
ShopLoad	1	2006054	2006054	2006054	1028.85	0.000
Schedule	9	311766	311766	34641	17.77	0.000
NumMach*NumOptn	1	409166	409166	409166	209.85	0.000
NumMach*ShopLoad	1	135219	135219	135219	69.35	0.000
NumMach*Schedule	9	35024	35024	3892	2.00	**0.037**
NumOptn*ShopLoad	1	1343097	1343097	1343097	688.84	0.000
NumOptn*Schedule	9	71703	71703	7967	4.09	**0.000**
ShopLoad*Schedule	9	41727	41727	4636	2.38	**0.012**
Error	757	1476003	1476003	1950		
Total	799	17964998				

Figure 4.10. Analysis of Variance on Average Flow Time for Job Shop Models with All Factors

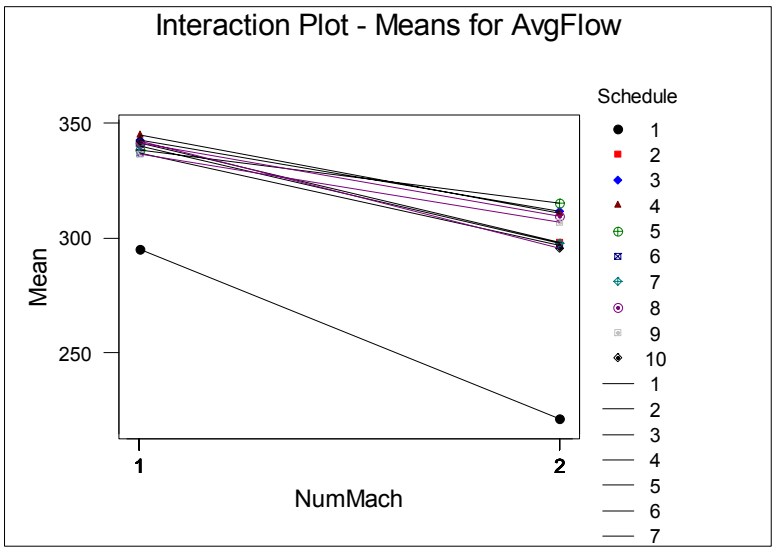


Figure 4.11. Interaction Plot between Number of Machines and the Scheduling Approach on Average Flow Time for Job Shop Models

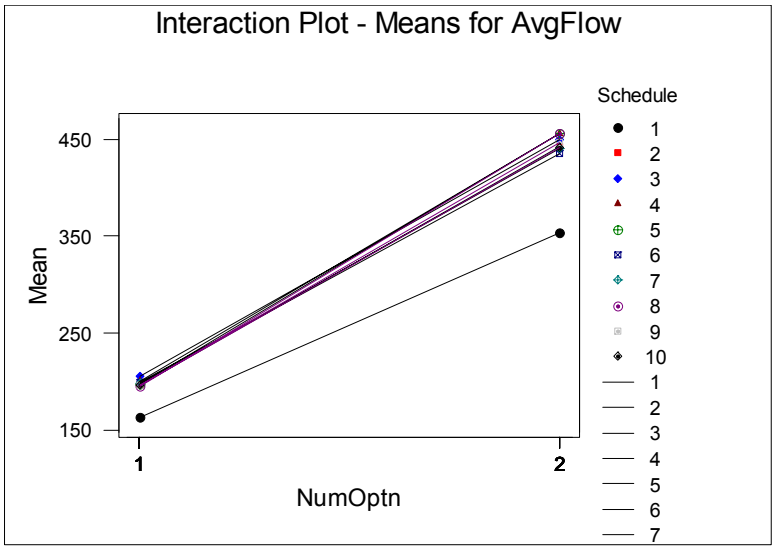


Figure 4.12. Interaction Plot between Number of Operation Steps and the Scheduling Approach on Average Flow Time for Job Shop Models

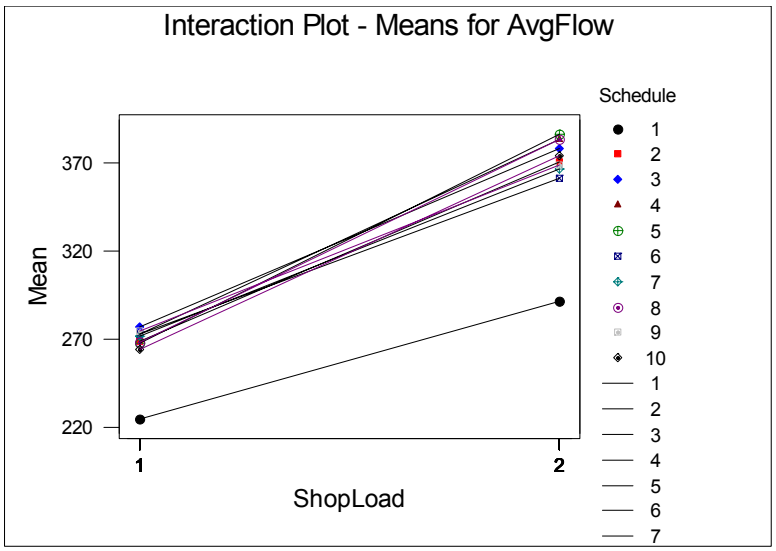


Figure 4.13. Interaction Plot between the Shop-Load and the Scheduling Approach on Average Flow Time for Job Shop Models

4.4.1.3 Single Stage Systems

The P-value of the ANOVA F-test and Tukey's confidence interval on Average Flow Time, Tardiness, and Percent of Tardy Jobs in single-stage systems are shown in Table 4.16. It is

found that the differences between various scheduling approaches are significant on three performance measures in all cases. From the Tukey's multiple comparison, the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time, Tardiness, and Percent of Tardy Jobs in all cases.

Table 4.16. One-way ANOVA and Tukey's Test for Hypothetical Single Stage Models

			P value of ANOVA F-test			Tukey's Confidence Interval *		
Number of Machines	Number of Operations Steps	Shop-Load	Average Flow Time	Tardiness	Percent of Tardy Job	Average Flow Time	Tardiness	Percent of Tardy Job
6	10	75%	<0.001	<0.001	<0.001	(-45.18, -32.87)	(-8.24, -1.37)	(-0.1461, -0.0546)
6	10	90%	<0.001	<0.001	<0.001	(-50.15, -38.78)	(-11.08, -2.7)	(-0.17, -0.0887)
6	15	75%	<0.001	<0.001	<0.001	(-80.55, -64.67)	(-17.96, -8.18)	(-0.2621, -0.1578)
6	15	90%	<0.001	<0.001	<0.001	(-78.89, -68.02)	(-17.05, -10.49)	(-0.2525, -0.1882)
9	10	75%	<0.001	<0.001	<0.001	(-85.43, -51.63)	(-15.57, -9.48)	(-0.2350, -0.1728)
9	10	90%	<0.001	<0.001	<0.001	(-82.17, -48.33)	(-14.94, -9.58)	(-0.2253, -0.1705)
9	15	75%	<0.001	<0.001	<0.001	(-83.01, -51.45)	(-15.787, -10.242)	(-0.2358, -0.1796)
9	15	90%	<0.001	<0.001	<0.001	(-86.07, -55.43)	(-18.05, -11.37)	(-0.25169, -0.1918)

* Tukey's confidence interval of the difference between the proposed method and the next best dispatching heuristics. (minutes)

Figure 4.14 shows the Analysis of Variance on Average Flow time for all factors in the single stage models. Like in the job shop models, the F-test for the interactions between the number of machines factor and the scheduling approach, the number of operations and the scheduling approach, and the shop-load and the scheduling approach are all significant. Figures 4.15-4.17 show the interaction plots between these three factors. The performance gaps between the proposed scheduling approach and the other heuristics tend to become larger when the number of machines is higher, the number of operation steps is higher, and the shop-load is higher.

Factor	Levels	Values									
NumMach	2	1	2								
NumOptn	2	1	2								
ShopLoad	2	1	2								
Schedule	10	1	2	3	4	5	6	7	8	9	10

Analysis of Variance for Average Flow Time

Source	DF	Seq SS	Adj SS	Adj MS	F	P
NumMach	1	34557	34557	34557	771.30	0.000
NumOptn	1	1046077	1046077	1046077	2.3E+04	0.000
ShopLoad	1	8080	8080	8080	180.35	0.000
Schedule	9	289033	289033	32115	716.79	0.000
NumMach*NumOptn	1	3647	3647	3647	81.41	0.000
NumMach*ShopLoad	1	1586	1586	1586	35.40	0.000
NumMach*Schedule	9	1545	1545	172	3.83	0.000
NumOptn*ShopLoad	1	21	21	21	0.47	0.493
NumOptn*Schedule	9	25238	25238	2804	62.59	0.000
ShopLoad*Schedule	9	811	811	90	2.01	0.036
Error	757	33916	33916	45		
Total	799	1444512				

Figure 4.14. Analysis of Variance on Average Flow Time for Single Stage Models with All Factors

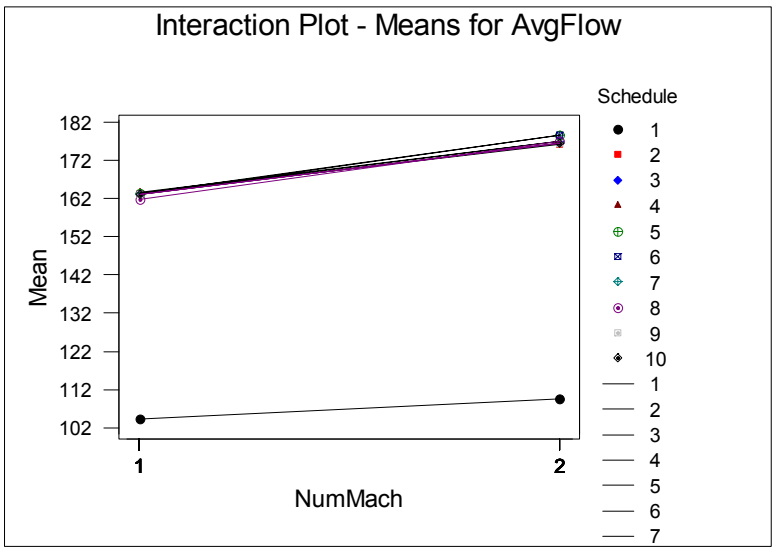


Figure 4.15. Analysis of Variance on Average Flow Time for Single Stage Models with All Factors

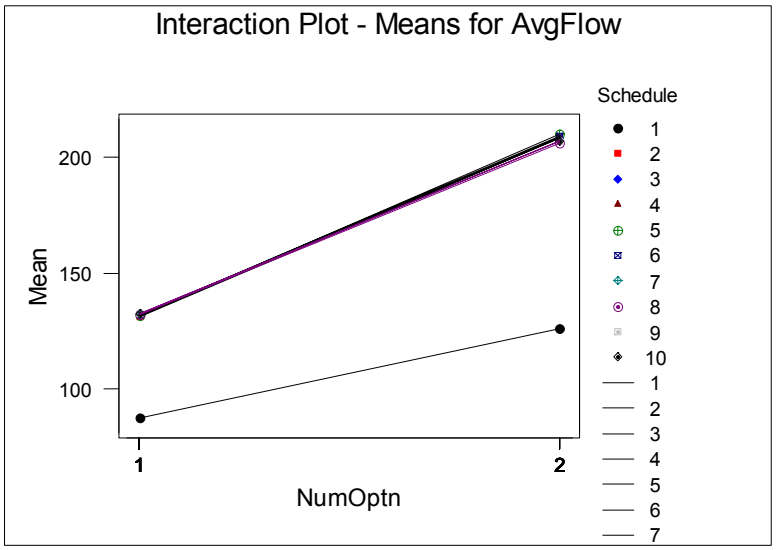


Figure 4.16. Analysis of Variance on Average Flow Time for Single Stage Models with All Factors

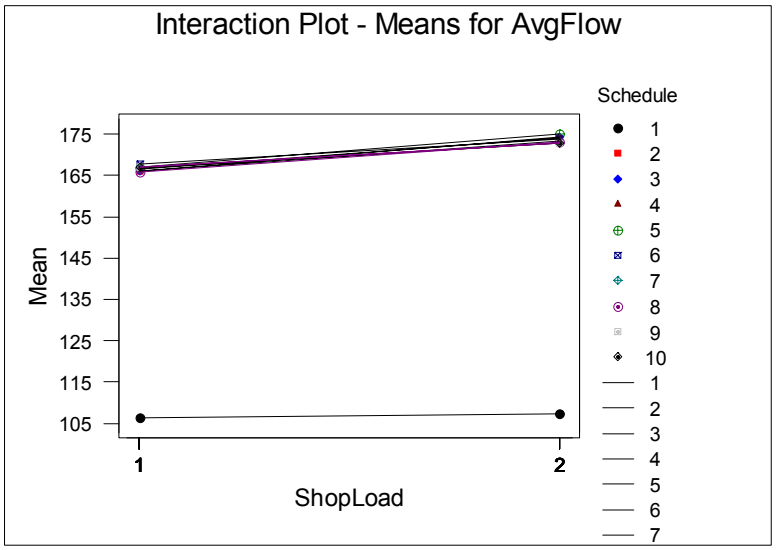


Figure 4.17. Analysis of Variance on Average Flow Time for Single Stage Models with All Factors

4.4.1.4 Horizon Length Effect

In order to investigate the effect of the rolling horizon length on the performance of the proposed scheduling approach, simulation experiments were conducted for each of hypothetical models with seven levels of horizon lengths (in Table 4.17). The shop-load level is assumed at 90% for these experiments. One-way analysis of variance is used for each of hypothetical models to test the significant difference on average flow time for different levels of horizon lengths.

Table 4.17. Levels of Horizon Lengths

Horizon Length	3 min	6 min	9 min	12 min	15 min	100 min	200 min
Levels	1	2	3	4	5	6	7

```
Analysis of Variance on AvgFlow
Source     DF        SS        MS        F        p
Length      6      7342      1224     5.35    0.000
Error      63     14411       229
Total      69     21753
                                    Individual 95% CIs For Mean
                                    Based on Pooled StDev
 Level      N      Mean     StDev  -----+---------+---------+---------+-
     1     10    181.42     19.20            (-----*-----)
     2     10    173.61     18.32      (------*-----)
     3     10    178.82     12.03          (-----*------)
     4     10    168.46      9.31   (-----*------)
     5     10    177.29     12.57         (-----*------)
     6     10    193.76     15.22                    (-----*------)
     7     10    199.60     16.61                        (-----*-----)
                                    -----+---------+---------+---------+-
Pooled StDev =    15.12               165       180       195       210
Tukey's pairwise comparisons
    Family error rate = 0.0500
Individual error rate = 0.00338
Critical value = 4.31
Intervals for (column level mean) - (row level mean)
               1         2         3         4         5         6

     2    -12.81
           28.42
     3    -18.01    -25.82
           23.21     15.41
     4     -7.66    -15.47    -10.26
           33.57     25.76     30.97
     5    -16.49    -24.30    -19.09    -29.44
           24.74     16.93     22.14     11.78
     6    -32.96    -40.77    -35.56    -45.91    -37.08
            8.27      0.46      5.67     -4.69      4.14
     7    -38.80    -46.60    -41.40    -51.75    -42.92    -26.45
            2.43     -5.37     -0.17    -10.52     -1.69     14.78
```

Figure 4.18. One-way Analysis of Variance on Average Flow Time for Hypothetical Job Shop with Six Machines and Ten Operation Steps

In Figure 4.18, F-test with a P-value less than 0.001 indicates that the difference in average flow time among different horizon lengths is significant for the hypothetical job shops with six machines and 10 operation steps. From the main effect plot in Figure 4.19, it can be noticed that the proposed approach is better when horizon length is smaller. However, the differences among first five levels are not significant according to the Tukey's comparison in Figure 4.18.

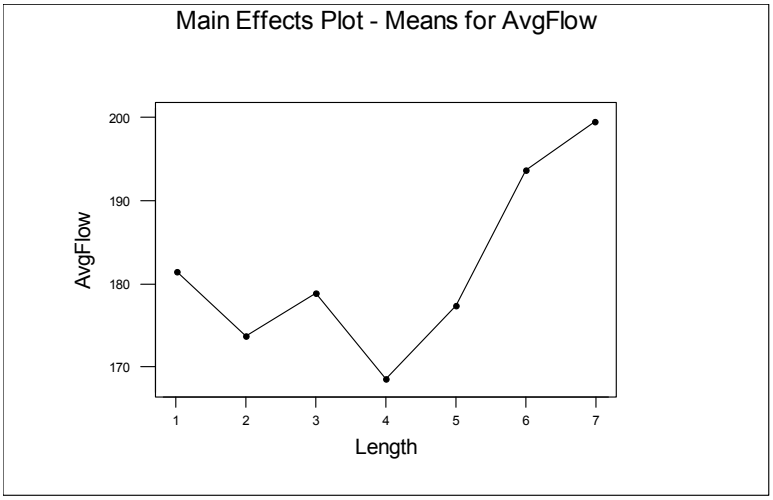


Figure 4.19. The Average Flow Time for Hypothetical Job Shops with Six Machines and Ten Operations Steps

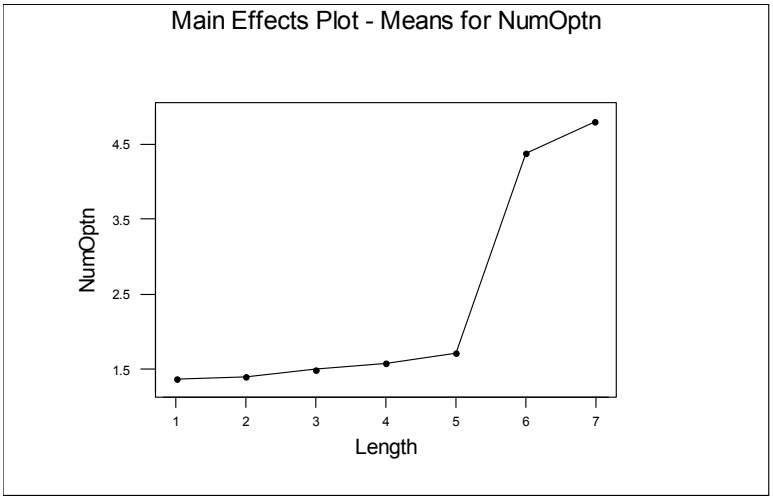


Figure 4.20. The Number of Candidate Jobs in Each Cell-level Sub-problem for Hypothetical Job Shops with Six Machines and Ten Operations Steps

Figures 4.20-4.21 show that the numbers of candidates in the both cell-level and machine- level sub-problems increase when the horizon length increases. It appears that that the more the number of candidate jobs the sub-problems have, the longer the average flow time from the data. Intuitively, one may expect a better system performance when the horizon length increases because the more candidate jobs are involved in the sub-problem decisions, the more global perspectives of the scheduling are taken into account. However, this can be explained by the fact that there are more uncertainties involved in the estimates of job ready time and completion time when the horizon length is longer. Some of the candidate jobs or machines chosen by the sub-problem decisions based on these estimates may become non-optimal over time for their original sub-problems.

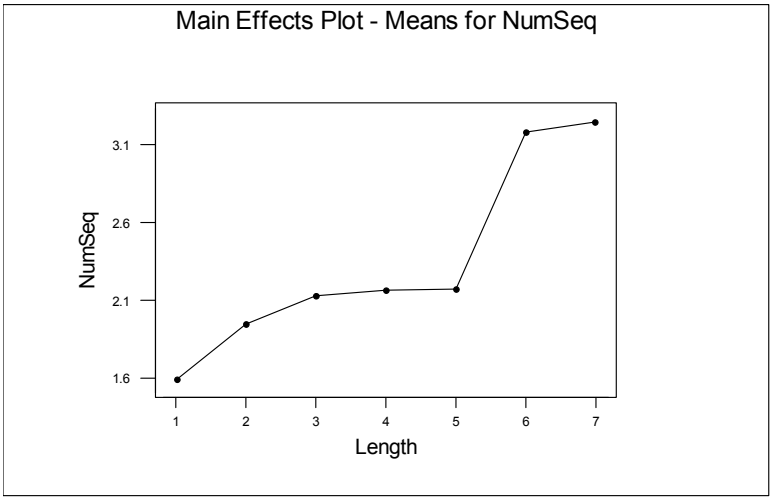


Figure 4.21. The Number of Candidate Operation Sequences in Each Machine-level Sub-problem for Hypothetical Job Shops with Six Machines and Ten Operations Steps

Table 4.18 shows the ANOVA F-test results for all hypothetical models. It shows that the horizon effect is significant on average flow time in most cases, except for single stage models with six machines, and flow shop models with six machines and 15 operation steps. However, for horizon lengths less than to equal to 15 minutes, difference is not significant based on the Tukey's multiple comparison, which implies that the performance of the proposed method is not very sensitive to the horizon length when it is smaller. From Figures 4.20-4.21, it is found that the proposed approach performs relatively better when the average number of candidate jobs is

around 1.5 in the cell level sub-problems and the average number of candidate operations sequences is around 2 in the machine level sub-problems.

Table 4.18. ANOVA F-test for Significance of the Horizon Length on Average Flow Time

	Number of Machines	Number of Operations Steps	P-value in ANOVA F-test
Job shops	6	10	<0.001
	6	15	0.013
	9	10	<0.001
	9	15	<0.002
Flow shops	6	10	<0.001
	6	15	0.063
	9	10	<0.001
	9	15	<0.001
Single Stage	6	10	0.351
	6	15	0.27
	9	10	<0.001
	9	15	<0.001

```
Factor   Levels Values
NumMach       2     1     2
NumOfOpt      2     1     2
Length        7     1     2     3     4     5     6     7

Analysis of Variance for AvgFlow

Source              DF     Seq SS     Adj SS     Adj MS       F      P
NumMach              1     818254     818254     818254  365.84  0.000
NumOfOpt             1    5423385    5423385    5423385 2424.80  0.000
Length               6     304927     304927      50821   22.72  0.000
NumMach*NumOfOpt     1     728580     728580     728580  325.75  0.000
NumMach*Length       6      20505      20505       3417    1.53  0.169
NumOfOpt*Length      6     106607     106607      17768    7.94  0.000
Error              258     577050     577050       2237
Total              279    7979307
```

Figure 4.22. Analysis of Variance on Average Flow Time for Hypothetical Job Shop with Different Horizon Lengths

Figure 4.22 shows the ANOVA F-test for the interaction between the number of operation steps and the horizon length is significant with a P-value less than 0.001. From the interaction plot in Figure 4.23, it can be found that the performance of the proposed approach is more sensitive to the horizon changes when the number of operation steps is higher.

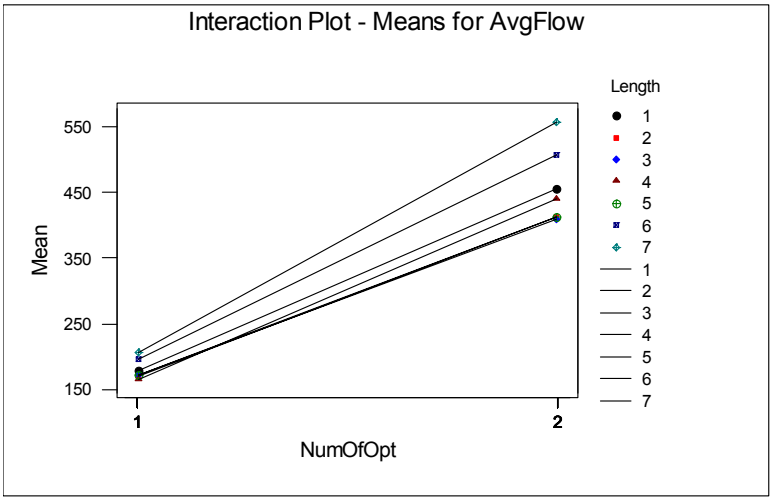


Figure 4.23. Interaction Plot between the Number of Operations and the Horizon Length for Hypothetical Job Shops

Similar analyses are conducted for hypothetical flow shops and single stage models (Figures 4.24-4.27). The results show that there is a significant interaction between the number of operations steps and the horizon length. As in the job shop models, the performance of the proposed approach is more sensitive to the horizon changes when there are more operation steps.

```
Factor    Levels Values
NumMach        2      1      2
NumOfOpt       2      1      2
Length         7      1      2      3      4      5      6      7

Analysis of Variance for AvgFlow

Source              DF      Seq SS      Adj SS      Adj MS        F       P
NumMach              1      455305      455305      455305   146.31   0.000
NumOfOpt             1     3766134     3766134     3766134  1210.20   0.000
Length               6      317694      317694       52949    17.01   0.000
NumMach*NumOfOpt     1      442872      442872      442872   142.31   0.000
NumMach*Length       6       12573       12573        2096     0.67   0.671
NumOfOpt*Length      6       78631       78631       13105     4.21   0.000
Error              258      802893      802893        3112
Total              279     5876103
```

Figure 4.24. Analysis of Variance on Average Flow Time for Hypothetical Flow Shop with Different Horizon Lengths

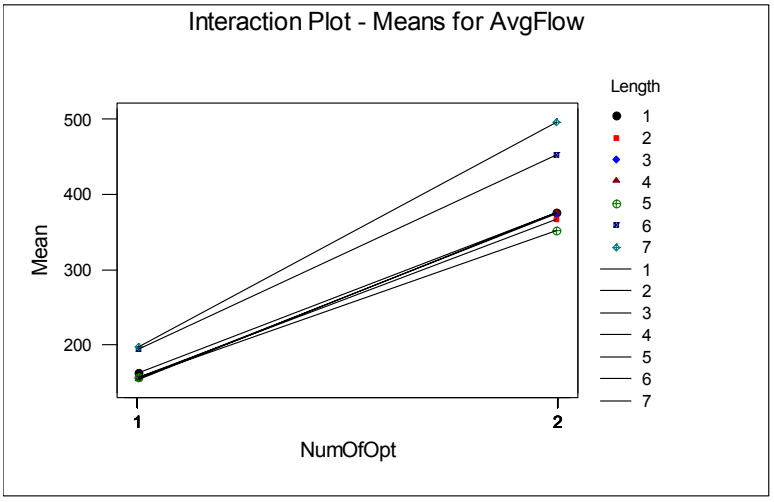


Figure 4.25. Interaction Plot between the Number of Operations and the Horizon Length for Hypothetical Job Shops

```
Factor   Levels Values
NumMach       2      1      2
NumOfOpt      2      1      2
Length        7      1      2      3      4      5      6      7

Analysis of Variance for AvgFlow

Source              DF     Seq SS     Adj SS     Adj MS       F      P
NumMach              1      42695      42695      42695  997.43  0.000
NumOfOpt             1     142659     142659     142659 3332.78  0.000
Length               6      72287      72287      12048  281.46  0.000
NumMach*NumOfOpt     1       1849       1849       1849   43.18  0.000
NumMach*Length       6      68051      68051      11342  264.97  0.000
NumOfOpt*Length      6       6740       6740       1123   26.24  0.000
Error              258      11044      11044         43
Total              279     345325
```

Figure 4.26. Analysis of Variance on Average Flow Time for Hypothetical Single Stage Models with Different Horizon Lengths

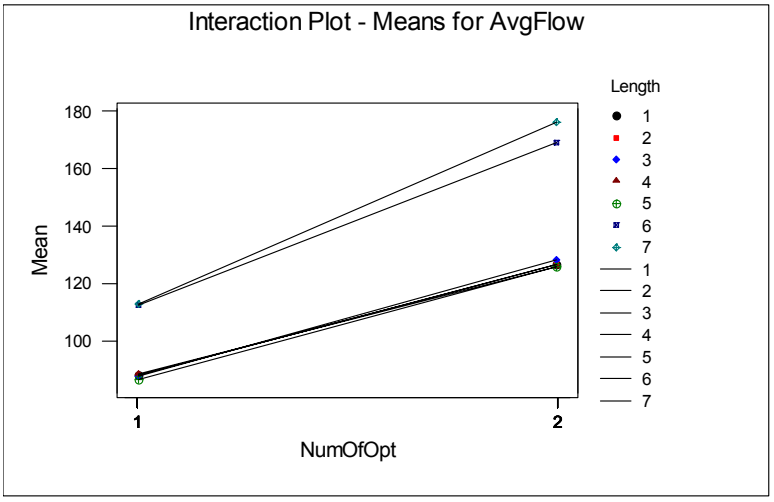


Figure 4.27. Interaction Plot between the Number of Operations and the Horizon Length for Hypothetical Single Stage Models

4.4.2 Industrial Example Model

Three performance measures, mean tardiness, average flow time, and percentage of tardy jobs are used in the simulation experiments. A rolling horizon length of 6 minutes is used for both cell-level and machine-level scheduling. One-way Analyses of Variance on three performance measures are conducted at each level of two factors, the shop-load level and the machine breakdown level. Tukey's multiple comparisons are used to test any significant difference between the proposed scheduling approach and other dispatching heuristics. A multi-way ANOVA of four factors and interaction plots are then employed to test the significance of each factor and the interactions.

Table 4.19. One-way ANOVA and Tukey's Test for Hypothetical Flow Shops

		P value of ANOVA F-test			Tukey's Confidence Interval *		
Shop-Load	Machine Breakdown	Average Flow Time	Tardiness	Percent of Tardy Job	Average Flow Time	Tardiness	Percent of Tardy Job
60% (1)	3% (1)	<0.001	0.079	0.003	(-97.74, -87.03)	(-0.2962, 0.2256)	(-0.0129, 0.008)
60% (1)	6% (2)	<0.001	<0.001	<0.001	(-109.49, -89.48)	(-4.374, 0.107)	(-0.1027, -0.017)
60% (1)	9% (3)	<0.001	<0.001	<0.001	(-162.9, -85.6)	(-48.9, -0.9)	(-0.4125, -0.155)
75% (2)	3% (1)	<0.001	0.074	0.002	(-103.97, -90.1)	(-0.6226, 0.2224)	(-0.0267, 0.004)
75% (2)	6% (2)	<0.001	0.001	<0.001	(-120.0, -88.75)	(-9.506, 1.296)	(-0.1696, -0.02627)
75% (2)	9% (3)	<0.001	<0.001	<0.001	(-206.5, -72.3)	(-94.1, 4.5)	(-0.5748, -0.1928)
90% (3)	3% (1)	<0.001	0.018	<0.001	(-113.82, -87.48)	(-3.096, 0.85)	(-0.09538, -0.00304)
90% (3)	6% (2)	<0.001	0.098	<0.001	(-193.6, -74.4)	(-60.6, 16.9)	(-0.5538, -0.1050)
90% (3)	9% (3)	<0.001	<0.001	<0.001	(-372.0, -93.1)	(-299.5, -36.4)	(-0.8118, -0.512)

* Tukey's confidence interval of the difference between the proposed method and the next best dispatching heuristics. (minutes)

```
Factor   Levels Values
ShopLoad      3     1     2     3
Breakdow      3     1     2     3
Schedule     10     1     2     3     4     5     6     7     8     9    10


Analysis of Variance for Average Flow TimeT

Source              DF     Seq SS     Adj SS     Adj MS       F      P
ShopLoad             2    2193944    2193944    1096972  684.75  0.000
Breakdow             2    4829856    4829856    2414928 1507.44  0.000
ShopLoad*Breakdow    4    1223265    1223265     305816  190.90  0.000
Schedule             9    1501059    1501059     166784  104.11  0.000
ShopLoad*Schedule   18      84178      84178       4677    2.92  0.000
Breakdow*Schedule   18     138394     138394       7689    4.80  0.000
Error              846    1355298    1355298       1602
Total              899   11325994
```

Figure 4.28. Analysis of Variance on Average Flow Time for Industrial Example Models with All Factors

Table 4.19 shows the P-value of the ANOVA F-test and Tukey's confidence interval on Average Flow Time, Tardiness, and Percent of Tardy Jobs for all levels of experimental factors in the industrial example model. It is found that the differences among various scheduling

approaches are significant on Average Flow Time and Percent of Tardy Jobs in all cases. The difference in Tardiness is significant in most cases. From the Tukey's multiple comparisons, the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time in all cases. Percent of Tardy Jobs of the proposed approach is significantly better than that of the next best dispatching heuristics in most cases except for the two cases when the shop-load is 60% and 75%, and the machine breakdown level is 3%.

Figure 4.28 shows the Analysis of Variance on Average Flow time for all factors in the industrial example model. The F-tests for the interactions between the shop load and the scheduling approach; the machine breakdown level and the scheduling approach are all significant. Figures 4.29-4.30 show the interaction plots on Average Flow Time between these three factors. The performance gaps between the proposed scheduling approach and the other heuristics tends to become larger when the shop-load level is higher, and the machine breakdown level is higher. It implies that the proposed scheduling approach is robust when the shop-load level gets higher and there are more disruptions such as machine breakdowns.

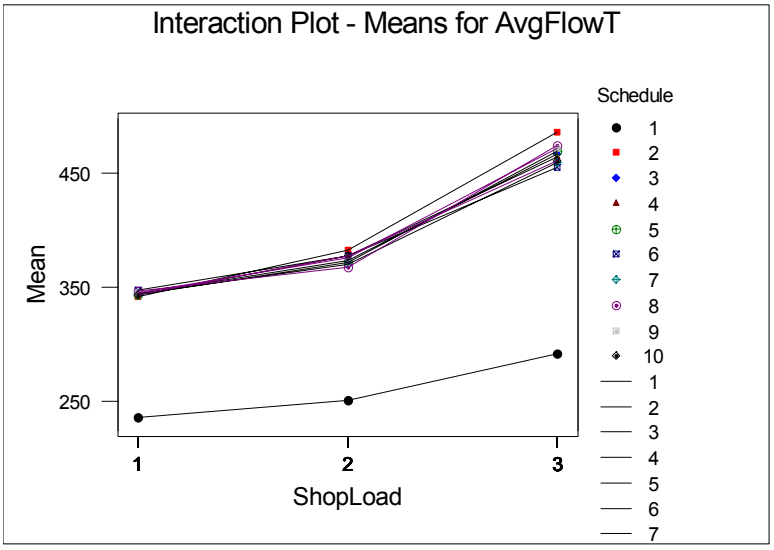


Figure 4.29. Interaction Plot between the Shop-Load and the Scheduling Approach on Average Flow Time for Industrial Example Models

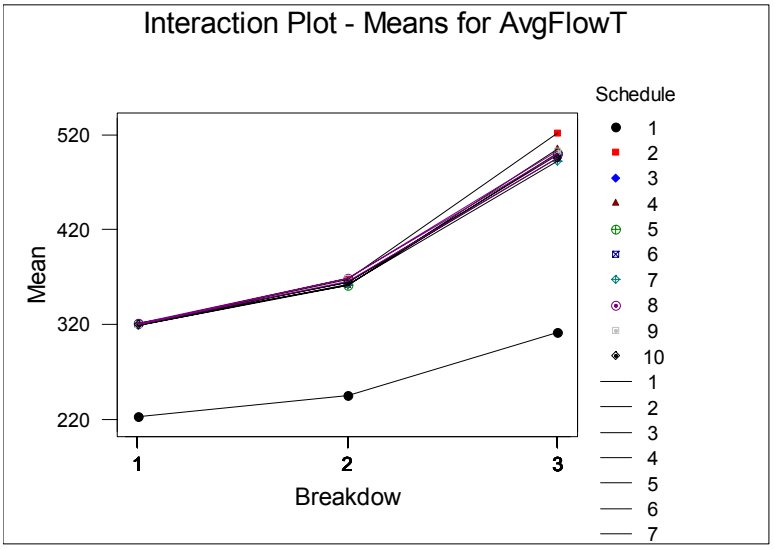


Figure 4.30. Interaction Plot between the Machine Breakdown Level and the Scheduling Approach on Average Flow Time for Industrial Example Models

4.4.3 Tool Loading Models

In order to test the performance of a proposed tool loading method, 16 cases of simulations are conducted as they are described in Section 4.1.3. The proposed rolling horizon approach is used for job-machine assignment and tool flow control at the scheduling stage. A rolling horizon length of 6 minutes is used for both cell-level and machine-level scheduling. The minimum resident tool ratio of 0.5 is used to obtain solutions using the proposed loading heuristic. A multi-way ANOVA of four factors is then employed to test the significance of each factor and their interactions. Paired-t tests at 95% of confidence level are used to compare three performance measures (Average Flow Time, Tardiness, and Percent of Tardy Jobs) of the proposed loading method against those of the random loading for each of 16 cases of simulation experiments.

Table 4.20 shows the simulation outputs and Paired-t tests on three performance measures between the proposed loading method (MRF) and the random loading approach (RAND) for the loading model with nine machines and 30 operation steps. It is found that the proposed loading method performed significantly better than the random loading approach for three measures for this case.

Table 4.20. Paired-t Test for the Loading Model With Nine Machines and Fifteen Operations Steps

	Average Flow Time		**Tardiness**		**Percent Tardy**	
Replication	**RAND**	**MRF**	**RAND**	**MRF**	**RAND**	**MRF**
1	125.283	101.842	6.069	1.191	0.126	0.034
2	135.533	112.275	11.832	1.159	0.197	0.051
3	132.934	103.254	7.427	1.420	0.122	0.051
4	133.721	116.853	19.773	2.727	0.178	0.057
5	133.095	100.809	7.725	2.732	0.122	0.044
6	134.212	94.104	2.404	1.568	0.068	0.049
7	145.415	98.256	4.797	5.305	0.097	0.100
8	124.400	98.749	5.008	0.879	0.101	0.019
9	129.652	105.468	5.539	0.450	0.149	0.026
10	131.423	101.857	5.052	2.175	0.112	0.046
	t-Test: Paired Two Sample for Means					
	RAND	*MRF*	*RAND*	*MRF*	*RAND*	*MRF*
Mean	132.567	103.347	7.563	1.961	0.127	0.048
Variance	34.210	45.560	24.529	1.950	0.001	0.000
Observations	10	10	10	10	10	10
Hypothesized Mean Difference	0		0		0	
df	9		9		9	
t Stat	10.405		3.523		5.485	
P(T<=t) one-tail	0.000		0.003		0.000	
t Critical one-tail	1.833		1.833		1.833	
P(T<=t) two-tail	0.000		0.006		0.000	
t Critical two-tail	2.262		2.262		2.262	

Table 4.21 shows the Paired-t tests on Average Flow Time for all cases of the loading models. It is noticed that the proposed loading method performs significantly better on average flow time except for two cases of the loading models with six machines and 30 operation steps. In Table 4.22, the P-value of Paired-t test on Tardiness indicates that the proposed approach is also significantly better than the random loading in most cases. For the case with six machine and 30 operation steps at 100 m/min tool carrier speed, and those with six machines at 180 m/min tool carrier speed, the Paired-t test on Tardiness is not significant.

Table 4.21. Paired-t test on Average Flow Time for the Loading Models with All Factors

Tool Carrier Speed (m/min)	Number of Machines	Number of Operation	Average Flow Time		P-value of Paired-t test
			MRF	RAND	
100	6	20	222.15	197.56	0.002
100	6	30	156.1	154.1	0.35
100	9	20	266.2	242.8	0.005
100	9	30	241.83	234.31	0.0011
180	6	20	149.99	133.5	0.042
180	6	30	131.9	129.2	0.145
180	9	20	104.6	99.1	0.016
180	9	30	132.56	103.35	<0.001

Table 4.22. Paired-t test on Tardiness for the Loading Models with All Factors

Tool Carrier Speed (m/min)	Number of Machines	Number of Operation	Tardiness		P-value of Paired-t test
			MRF	RAND	
100	6	20	68.124	45.731	0.002
100	6	30	0.286	0.274	0.481
100	9	20	110.373	87.630	0.005
100	9	30	25.630	20.909	0.007
180	6	20	24.167	12.094	0.076
180	6	30	0.097	0.070	0.27
180	9	20	2.220	1.098	0.05
180	9	30	7.563	1.961	0.003

Table 4.23. Paired-t test on Percent of Tardy Jobs for the Loading Models with All Factors

Tool Carrier Speed (m/min)	Number of Machines	Number of Operation	Percent of Tardy Jobs		P-value of Paired-t test
			MRF	RAND	
100	6	20	0.884	0.790	0.024
100	6	30	0.019	0.019	0.497
100	9	20	0.969	0.932	0.125
100	9	30	0.647	0.579	<0.001
180	6	20	0.390	0.324	0.113
180	6	30	0.009	0.006	0.182
180	9	20	0.078	0.054	0.072
180	9	30	0.127	0.048	<0.001

From Table 4.23, it can be found that the Percent of Tardy Jobs of the proposed loading method is significantly better than the random loading in 50% of the cases that have bee tested. Figure 4.31 shows the Analysis of Variance on Average Flow Time based on the General Linear Model with all experimental factors included. The ANOVA F-test indicates the difference between the loading methods is significant with a P-value of 0.001. The two-factor interactions between the loading approach and other three factors are not statistically significant.

Factor	Levels	Values	
TCSpeed	2	1	2
Machine	2	1	2
OptnStep	2	1	2
Load	2	1	2

Analysis of Variance for AvgFlowT

Source	DF	Seq SS	Adj SS	Adj MS	F	P
TCSpeed	1	330605	330605	330605	514.74	0.000
Machine	1	13443	13443	13443	20.93	0.000
OptnStep	1	10716	10716	10716	16.68	0.000
Load	1	7259	7259	7259	11.30	0.001
TCSpeed*Machine	1	82667	82667	82667	128.71	0.000
TCSpeed*OptnStep	1	14754	14754	14754	22.97	0.000
TCSpeed*Load	1	31	31	31	0.05	0.828
Machine*OptnStep	1	10501	10501	10501	16.35	0.000
Machine*Load	1	350	350	350	0.54	0.462
OptnStep*Load	1	575	575	575	0.90	0.346
Error	149	95698	95698	642		
Total	159	566599				

Figure 4.31. Analysis of Variance on Average Flow Time for the Loading Models with All Factors

4.5 Summary of Experiment Results

Three performance measures, mean tardiness, average flow time, and percentage of tardy jobs were used in the simulation experiments. The short-term scheduling window length of 6 minutes is used for both the cell-level and machine-level scheduling. For each of the three shop flow structures (Flow shops, Job shops, and Single Stage Systems), One-way Analysis of Variance on three performance measures are conducted at each level of two factors, the number of machines and the number of operation steps. Tukey's multiple comparisons are used to test any significant difference between the proposed scheduling approach and other dispatching heuristics. A multi-way ANOVA of four factors and interaction plots are then employed to test the significance of each factor and their interactions.

For flexible flow shops, the results indicate that the difference between various scheduling approaches are significant on Average Flow Time and Percent of Tardy Jobs in most cases for hypothetical flow shops. Based on the results of the Tukey's multiple comparison, the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time in most cases. Percent of Tardy Jobs of the proposed approach is significantly better than that of the next best dispatching heuristics when there are nine machines. It is also found that the performance gaps between the proposed scheduling approach and the other heuristics tend to become larger when the number of machines is higher, and the higher the number of operation steps, the better the proposed method performs relative to the other heuristics.

For flexible job shops, it is found that the differences between various scheduling approaches are significant on Average Flow Time in most cases. The proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time 5 out 8 cases. Percent of Tardy Jobs of the proposed approach is significantly smaller than the next best dispatching heuristics when there are nine machines. The performance gaps between the proposed scheduling approach and the other heuristics tend to become larger when the number of machines is higher, the number of operation steps is higher, and the shop-load is higher. This

interaction is also intuitive since the higher number of machines is considered as candidates in the cell-level sub-problems.

For single stage models, it is found that the differences between various scheduling approaches are significant on three performance measures in all cases. From the Tukey's multiple comparison, the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time, Tardiness, and Percent of Tardy Jobs in all cases. The performance gaps between the proposed scheduling approach and the other heuristics also tend to become larger when the number of machines is higher, the number of operation steps is higher, and the shop-load is higher.

In order to investigate the effect of the rolling horizon length, simulation experiments were conducted for each of hypothetical models with seven levels of horizon lengths. It shows that the horizon effect is significant on average flow time in most cases. The more the number of candidate jobs in sub-problems, the longer the average flow time. However, for horizon lengths less than or equal to 15 minutes, the difference is not significant based on the Tukey's multiple comparison, which implies that the performance of the proposed method is not very sensitive to the horizon length changes when the horizon length is relatively small. The proposed approach performs relatively better when the average number of candidate jobs is around 1.5 in the cell level sub-problems and the average number of candidate operations sequences is around 2 in the machine level sub-problems. It is also discovered that the performance of the proposed approach is more sensitive to the horizon length changes when the number of operation steps is higher. The performance of rolling horizon approach deteriorates as the length of the forecast horizon increases.

For industrial example models, one-way Analyses of Variance on three performance measures are conducted at each level of two factors, shop-load level and the machine breakdown level. It is found that the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time in all cases. Percent of Tardy Jobs of the proposed approach is significantly better than the next best dispatching heuristics in most cases. The performance gaps between the proposed scheduling approach and the other heuristics tend to

become larger when the shop-load level is higher, and the machine breakdown level is higher, which implies that the proposed scheduling approach is robust when the shop-load level gets higher and there are more disruptions such as machine breakdowns.

Since the preliminary study results showed that the average flow time of the approach based on Song et al. (1995) is 3 to 4 times of those in the models with the proposed approach and the random loading approach, only the proposed approach and the random loading method are used as in the simulation experimental studies. Paired-t tests indicate that the proposed loading method performs significantly better on average flow time and Tardiness than the random loading in most cases. The Percent of Tardy Jobs of the proposed loading method is significantly better in 50% of the cases that have been tested. The ANOVA F-test also shows that the difference between the loading methods is significant with a P-value of 0.001. The two-factor interactions between the loading approach and other three factors are not statistically significant.

Chapter 5 Conclusion and Further Research

5.1 Summary of the Research

This research has addressed the dynamic scheduling problems in automated manufacturing systems which have jobs with varying part types (each job can have several operations), alternative routing (each operation can be processed at different workstation). Jobs may arrive at different times to be processed on machines for a given amount of time, and shop floor disturbances like machine random failures exist. Both machines and tools are major resource constraints in the system studied. In order to support production of multiple products with quick responsiveness, dynamic tool sharing is used as the underlying tooling strategy which is supported by an automatic tool changing mechanism and a fast tool delivery system to reduce the time for tooling setup, hence allows parts to be processed in small batches. Myopic dispatching rules decompose the part/machine assignment into smaller sub-problems, namely machine routing and part dispatching (prioritization of tool requests). Resources and tasks are assigned sequentially without considering the interaction between sub-problems due to jobs and machines becoming available at different times.

This research presented a dynamic scheduling approach that makes coupled decisions about the part/machine scheduling and the operation/tool assignments on a rolling window basis, while the shop floor disturbances such as machine failures, and rush orders are considered. The proposed dynamic scheduling approach was tested on two types of manufacturing system models. First, hypothetical models, which were generated based on some generic shop flow structures (e.g. flexible flow shops, job shops, and single-stage systems) and configurations, were studied in order to provide more general empirical results about the applicability and the performance of the proposed approach under various production environments. Second, a model based on a real industrial flexible manufacturing system was experimented to see how well it performs when machine types, part routing, tooling, and other parameters are closer to the real metal-cutting operations. The study results show that the proposed scheduling approach significantly outperforms other dispatching heuristics, including Cost Over Time (COVERT),

Apparent Tardiness Cost (ATC), and Bottleneck Dynamics (BD), on due-date related performances under both types of manufacturing systems models.

When tooling resource is shared, parts can be routed to machines that do not have all the required tools, resulting in higher routing flexibility. Research to date in scheduling with shared tooling resources often places more emphasis on the real-time control and manipulation of tools, but pay less attention to the machine loading and initial tool allocation at the planning stage. In this research, a machine-loading model with shared tooling is proposed to maximize routing flexibility while attempting to maintain a minimum ratio of resident tools. The proposed loading heuristic is compared to a random loading method under hypothetically generated single stage system models. The study result indicates that better system performances can be obtained when the resident tools is considered in assigning part types and allocating tools to machines initially at the planning stage.

5.2 Research Contributions and Conclusion

The contributions and conclusion of this research are summarized as follows:

(1) The proposed framework allows two levels of hierarchical, dynamic decision making for resource and task assignment. The job-machine scheduling decision is made at the cell level, while tools are scheduled for each operation sequence at the machine level. Most of the pre-release planning function is not performed, thereby allowing the system to respond quickly to disturbances such as machine breakdowns. This provides industrial software designers a better system design models for the implementation of real-time control software in complex, dynamic manufacturing systems.

(2) The assignment decision at each level is formulated as a minimum-cost flow problem during each short-term window, and solved by the efficient network optimization algorithm. The cost coefficients corresponding to alternative assignment of resources to tasks are defined to optimize weighted completion time during each short-term scheduling window. This

improves the overall due-date related system performance significantly. The sub-problem heuristics can provide researchers a good alternative algorithm to solve dynamic job scheduling problems using a rolling time horizon approach.

(3) The proposed dynamic scheduling approach is tested on multiple manufacturing system models considering variations in generic shop flow structures and configurations. The results show that the proposed approach performs significantly better than all other dispatching heuristics on Average Flow Time. It is also noticed that the performance differences between the proposed scheduling approach and the other heuristics tend to become more significant when there are more machines. It is also noticed that more operation steps, the better the proposed method performs relative to the other heuristics. The system implementers can use this result as a general design guideline to determine when the proposed approach will perform the best.

(4) The performance of the proposed approach is validated by testing a model based on a real industrial flexible manufacturing system. The study results show that the proposed scheduling approach significantly outperforms other dispatching heuristics at various levels of shop load conditions and machine breakdowns. The proposed approach is especially robust when the shop-load level increases and when there are more disruptions such as machine breakdowns.

(5) Through the investigation of the horizon length effect, it is found that the more candidate jobs in sub-problems, the longer the average flow time becomes. The proposed approach performs relatively better when the average number of candidate jobs is approximately 1.5 in the cell level sub-problems and the average number of candidate operations sequences is around 2 in the machine level sub-problems. This provides system designers empirical guidelines in choosing the appropriate horizon length for implementing the proposed scheduling approach.

(6) Research to date in scheduling with shared tools often places emphasis on the real-time control and manipulation of tools, but pay less attention to machine loading and initial tool

allocations at the planning stage. In this research, a machine-loading model is proposed to assign part types and tools to machines while considering minimum ratio of resident tools. It provides a closer coordination of the tool flow control between the tool allocation at the machine loading stage and the real-time assignment of tools at the scheduling stage.

(7) The proposed loading heuristic is tested in single stage systems. The results show that the proposed loading method performs significantly better on average flow time and tardiness than that of the random loading in most cases. It proved the research hypothesis that better system performances can be achieved by considering the ratio of resident tools in assigning part types and allocating tools to machines at the loading stage. This provides production-planning practitioners a new machine loading approach producing better overall system performances when the tools are dynamically shared among machines.

5.3 Recommendations for Future Research

The presented approaches for the dynamic scheduling and machine loading with shared tools are expected to open many new topics for future research. The followings are suggested areas for the future research.

1) The automated manufacturing system models studied in this research are characterized with alternative part-routings and shared cutting tool resources. There are many other production systems that also have these characteristics. For example, in the lithography operation of wafer fabrications, the reticles are often shared among different steppers. Applying the proposed approach to the dynamic scheduling problems in these types systems and comparing its performance to those of other dispatching heuristics will be one of the areas for the future research.

2) In the simulation experiments of this research, it shows that the proposed scheduling approach performs well under some flexible flow shops and job shops when tooling resources are shared between machines. In the future research, the proposed scheduling

approach should be compared to other dispatching heuristics in the generalized job shop and flow shop environments where the tools are not shared among machines.

3) In the simulation experiments of this research, exponential distribution is assumed for the machine breakdown and the repair time. In future research, it would be interesting to investigate how the proposed approach performs when the machine breakdowns follow other distributions such as Weibull and Lognormal.

4) In this research, it is assumed that the tool handling system transports tools one at a time. When a machine failed, the entire set of tools on it can be migrated to other machines in a bulk exchange manner. The proposed loading heuristic can be used to re-allocated tools and re-assign part types to other active machines, which is expected to further improve the overall system performance. Simulation experiments should be conducted to show the effectiveness of the proposed loading methods together with tool migrations in the event of machine breakdowns.

5) It is shown that better system performances can be obtained by taking into account the resident tool ratio at the loading stage in this research. It would be also interesting to investigate the performance of the proposed approach when the resident tool ratio is also considered in the cell-level sub-problem heuristic at the scheduling stage in future research.

References

Agnetis, A., Alfieri, A., Brandimarte, P., and Prinsecchi, P., 1997, "Joint job/tool scheduling in a flexible manufacturing cell with no on-board tool magazine," *Computer-Integrated Manufacturing Systems*, 10 (1), pp 61-68.

Alfieri, A. and Brandimarte, P., 1997, "Job shop scheduling with delay-renewable resources: a comparison of decomposition strategies," *International Journal of Production Research*, 35 (7), pp 1807-1823.

Amoako-Gyampah, Kwasi, 1994, "A Comparative Study of FMS Tool Allocation and Part Type Selection Approaches for a Varying Part Type Mix," *International Journal of Flexible Manufacturing Systems*, 6, pp 179-207.

Amoako-Gyampah, Kwasi and Meredith, Jack R., 1996, "A Simulation Study of FMS Tool Allocation Procedures," *Journal of Manufacturing Systems,* 15 (6), pp 419-431.

Anderson, E.J. and Nyirenda, J.C., 1990, "Two new rules to minimize tardiness in a job shop," *International Journal of Production Research*, 28 (12), pp 2277-2292.

Baker, K.R., 1977, "An experiment study of the effectiveness of rolling schedules in production planning," *Decision science*, 8, pp 19-31.

Baker, K.R. and Bertrand, J.W.M., 1982, "A dynamic priority rule for sequencing against due-dates", *Journal of Operations Management*, 3, pp 37-42.

Baker, K.R. and Petersen, D.W., 1979, "Analytical framework for evaluating rolling schedules," *Management Science*, 25, pp 342-358.

Bazaraa, M.S., Jarvis, J., and Sherali, H.D., 1990, *Linear Programming and Network Flows*, 2nd Edition, Wiley, New York.

Bazaraa, M.S., Sherali, H.D., and Shetty, C.M., 1993, *Nonlinear Programming: theory and algorithms*, 2nd Edition, Wiley, New York.

Blackstone, J.H. Jr., Phillips, D. T. and Hogg, G. L. 1982, "A state-of-art survey of dispatching rules for manufacturing job shop operations," *International Journal of Production Research*, 20 (1), pp 27-45.

Browne, J., Dubois, D., Rathmill, K., Sethi, S.P., and Stecke, K.E., 1984, "Classification of Flexible Manufacturing Systems," *FMS Magazine,* pp 114-117.

Carrie, A.S., and Perera, D.T.S., 1986, "Work scheduling in FMS under tool availability constraints," *International Journal of Production Research*, 24 (6), pp 1299-1308.

Carrie, A.S., and Petsopoulos, A.C., 1985, "Operation sequencing in an FMS," *Robotica,* 3 (4), pp 259-264.

Carter, M.F., 1986, "Designing Flexibility into Automated Manufacturing Systems," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems (Ann Arbor, MI), K.E. Stecke and R. Suri (Eds.), Elsevier, Amsterdam, The Netherlands*, pp. 107-118.

Chandra, Jayanta, 1990, "Optimization-based Opportunistic Part Dispatching in Flexible Manufacturing Systems," Unpublished PhD. Thesis, School of Industrial Engineering, Purdue University, W. Lafayette, IN.

Chandra, J. and Talavage, J., 1991, "Intelligent dispatching for flexible manufacturing," *International Journal of Production Research*, 29 (11), pp 2259-2278.

Chandra, P., Li, S., and Stan, M., 1993, "Jobs and tools sequencing in an automated manufacturing environment," *International Journal of Production Research*, 31 (12), pp 2911-2925.

Chandra, P. and Tombak, M.M., 1992, "Models for the Evaluation of Routing and Machine Flexibility," *European Journal of Operation Research,* 60, pp 156-165.

Chatterjee, A., Cohen, M.A., Maxwell, W.L., and Miller, L.W., 1984, "Manufacturing Flexibility: Models and Measurements," *Proceedings of the First ORSA/TIMS Conference on Flexible Manufacturing Systems (Ann Arbor, MI), K.E. Stecke and R. Suri (Eds.), Elsevier, Amsterdam, The Netherlands.*

Chen, F.F., and Xu, D.D., and Ker, J.I., 1993, "Effects of tool loading and replenishing practices on design and operation of flexible manufacturing centers," *Transactions of NAMRI/SME*, 21, 391-398.

Chung, C.H., 1991, "Planning tool requirements for flexible manufacturing systems," *Journal of Manufacturing Systems*, 10 (6), pp 476-483.

Chung, C.H. and Chen, I.J., 1989, "A Systematic Assessment of the Value of Flexibility from FMS," *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems (Ann Arbor, MI), K.E. Stecke and R. Suri (Eds.), Elsevier, Amsterdam,* The Netherlands, pp 27-34.

Church, L.K., and Uzsoy, R., 1992, "Analysis of periodic and event-driven rescheduling policies in dynamics shops," *International Journal of Computer Integrated Manufacturing*, 5, pp 153-163.

Conway, R.W., Maxwell, W.L. and Miller, L.W., 1967, *Theory of Scheduling*, Addison Wesley Publishing Company, Reading, MA.

Garetti. M., Pozzctti, A., and Bareggi, A., 1990, "On-line and dispatching in flexible manufacturing systems," *International Journal of Production Research*, 28 (7), pp 1271-1292.

Gargeya, V.B. and Deane, R.H., 1999, "Scheduling in the dynamic job shop under auxiliary resource constraints: a simulation study," *International Journal of Production Research*, 37 (12), pp 2817-2834.

Gerwin, D., 1982, "Do's and Don'ts of Computerized Manufacturing," *Harvard Bussiness Review,* 60, pp 107-116.

Ghosh, S., Melnyk, S.A., and Ragatz, G.L., 1992, "Tooling constraints and shop floor scheduling: evaluating the impact of sequence dependency," *International Journal of Production Research*, 30 (6), pp 1237-1253.

Gaalman, G.J., Nawijn, W.M., and Platzer, L.W., 1987, "Tool sharing in an FMS – a flexibility study," *Engineering Cost and Production Economics,* 12, pp 107-115.

Grieco, A., Semeraro, Q., Tolio, T., and Toma, S., 1995, "Simulation of tool and part flow in FMS," *International Journal of Production Research,* 33 (3), pp 643-658.

Grieco, A., Semeraro, Q., and Tolio, T., 2001, "A Review of Different Approaches to the FMS Loading Problem," *International Journal of Flexible Manufacturing Systems,* 13(4), pp. 361-384.

Gray, A.E., Seidmann, A., and Stecke, K.E., 1993, " A Synthesis of Decision Models for Tool Management in Automated Manufacturing," *Management Science*, 39 (5), pp 549-567.

Gunasekaran, A., Martikainen, T., and Yli-Olli, P., 1993, "Flexible Manufacturing Systems: An Investigation for Research and Applications," *European Journal of Operational Research*, 66, pp 1-26.

Hahn, H.S. and Sanders, J.L., 1994, "Performance analysis of a LIM-based high-speed tool delivery system for machining," *International Journal of Production Research,* 32 (1), pp 179-207.

Hammer, H., 1989, “A new game plan for tool control," *Modem Machine Shop*, Jan., pp 52-63.

Han, M.H., Na, Y.K., and Hogg, G.L., 1989, “Real-time control and job dispatching in flexible manufacturing systems”, *International Journal of Production Research,* 27 (8), pp 1257-1267.

Hendry, L.C. and Wong, S.K., 1994, "Alternative order release mechanisms: a comparison by simulation," *International Journal of Production Research*, 32 (12), pp 2827-2842.

Hankins, S. and Rovito, V., 1984, "The impact of tooling on flexible manufacturing systems," *Proceeding of Second Biennial International Machine Tool Technical Conference,* Chicago, Illinois, pp 193-199.

Hinkelmann, K., and Kempthorne, O., 1994, *Design and Analysis of Experiments. Volume I: Introduction to Experimental Design,* John Wiley and Sons.

Ishii, N. and Talavage, J.J., 1991, "A transient-based real-time scheduling algorithms in FMS," *International Journal of Production Research*, 29 (12), pp 2501-2520.

Ishii, N. and Talavage, J.J, 1994, "A mixed dispatching rule approach in FMS scheduling," *International Journal of Flexible Manufacturing Systems*, 6 (1), pp 69-87.

Kanet, J.J., 1982, “On anomalies in dynamic ratio type scheduling rules: a clarifying analysis,” *Management Science,* 28, pp 1337-1341.

Kanet, J.J. and Hayya, J.C., 1982, “Priority dispatching with operation due-dates in a job shop,” *Journal of Operations Management,* 2, pp 155-163.

Kashyap, A.S. and Khator, S.K., 1996, “Analysis of Tool Sharing in an FMS: A Simulation Study,” *Computers and Industrial Engineering,* 30 (1), pp 137-145.

Kelton, W.D., Sadowski, R.P., and Sadowski, D.A., 1998, *Simulation with Arena*, McGraw-Hill.

Khator, S.K. and Leung, L.C., 1994, "Intermediate tool requirement Planning for FMS," *Journal of Manufacturing Systems*, 13 (1), pp 9-19.

Khator, S. and Nof, S.Y., 1985, "A flow control modeling approach for computerized manufacturing systems," *Material Flow*, 2, pp 1-10.

Koo, Pyung-Hoi, 1996, "Flow Planning and Control of Single-Stage Multimachine Systems," Unpublished PhD. Thesis, School of Industrial Engineering, Purdue University, W. Lafayette, IN.

Koo, P-H. and Tanchoco, J.M.A., 1999, “Real time operation in single stage multimachine systems”, *International Journal of Production Research*, 37 (5), pp 1023-1039.

Koo, P.H., Tanchoco, J.M.A. and Talvage, J.J., 1998, “Estimation of tool requirements in single-stage multi-machine systems,” *International Journal of Production Research,* 36 (6), pp 1699-1713.

Koulamas, C.P., 1991, “Total tool requirements in multi-level machining systems,” *International Journal of Production Research*, 29 (2), pp 417-437.

Kumaran, T.K., 1995, *Tool Management in Computer Integrated Manufacturing*, Unpublsihed Ph.D. Thesis, Texas A&M University.

Kutanoglu, E. and Sabuncuoglu, I., 1999, “An analysis of heuristics in a dynamic job shop with weighted tardiness objectives,” *International Journal of Production Research,* 37 (1), pp 165-187.

Lawrence, S.R. and Morton, T.E., 1993, “Resource-constrained multi-project scheduling with tardy costs: comparing myopic, bottleneck, and resource pricing heuristics,” *European Journal of Operational Research*, 64, pp 168-187.

Lee, D. H., and Kim, Y. D., 2000, “Loading algorithms for flexible manufacturing systems with partially grouped machines,” *IIE Transactions,* 32, 33-47.

Lin, G.Y-J., 1993, "A distributed production control for intelligent manufacturing systems," Unpublished Ph.D. Thesis, School of Industrial Engineering, Purdue University, W. Lafayette, IN.

Macchiaroli, R. and Riemma, S., 1996, “Design of a tool management system in a flexible cell,” *International Journal of Production Research,* 34 (3), pp 767-784.

Mason, F., 1986, "Computerized cutting-tool management," *American Machinist and Automated Manufacturing,* May, pp 105-120.

Melnyk, S.A., Carter, P.L., Dilts, D.M., and Lyth, D.M., 1985, *Shop Floor Control*, Dow Jones-Irwin, Homewood, Illinois, 1985.

Melnyk, S.A., Ghosh S., and Ragatz, G.L., 1989, "Tooling constraints and shop floor scheduling: a simulation study," *Journal of Operations Management*, 8 (9), pp 69-89.

Melnyk, S.A., Tan, K.C., Denzler, D.R. and Fredendall, L., 1994, "Evaluating variance control, order review/release and dispatching: a regression analysis," *International Journal of Production Research*, 32 (5), pp 1045-1061.

Montazeri, M. and Van Wassenhove, L.N., 1990, "Analysis of scheduling for an FMS," *International Journal of Production Research*, 28 (4), pp 785-802.

Morton, T.E. and Pentico, D., 1993, *Heuristic Scheduling Systems with Applications to Production and Project Management*, New York, Wiley.

Muhlemann, A.P., Lockett, A.G., and Farn, C.K., 1982, "Job shop scheduling heuristics and frequency of scheduling," *International Journal of Production Research*, 20, pp 227-241.

O'Keefe, R.M and Kasirajan, T., 1992, "Interaction between dispatching and next station selection rules in a dedicated flexible manufacturing system," *International Journal of Production Research*, 30 (8), pp 1753-1772.

Osaki, S., 1992, *Applied Stochastic System Modeling*, Springer-Verlag, New York.

Ovick, I.M. and Uzsoy, R., 1994, "Rolling horizon algorithms for single machine dynamic scheduling problem with sequence dependent setup times," *International Journal of Production Research*, 32, pp 1243-1263.

Ovick, I.M. and Uzsoy, R., 1995, "Rolling horizon procedures for dynamic parallel machine scheduling problem with sequence dependent setup times," *International Journal of Production Research*, 33, pp 3173-3192.

Park, S.C., Raman, N., and Shaw, M., 1997, "Adaptive scheduling in dynamic flexible manufacturing systems: a dynamic rule selection approach," *IEEE Transactions on Robotics and Automation*, 13 (4), pp 486-502.

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

Pinedo, M, 1995, *Scheduling: Theory, Algorithms, and Systems,* Prentice-Hall, New York.

Piramuthu, S., Shaw, M., and Fulkerson, B., 2000, “Information-based dynamic manufacturing system scheduling,” *International Journal of Flexible Manufacturing Systems*, 12, pp 219-234.

Raju, K.R. and Chetty, O.V.K., 1993, "Design and evaluation of automated guided vehicle systems for flexible manufacturing systems: an extended Petri net-based approach," *International Journal of Production Research*, 31 (5), pp 1069-1096.

Ro, I-K. and Kim, J-I., 1990, "Multi-criteria operational control rules in flexible manufacturing system," *International Journal of Production Research*, 28 (1), pp 47-63.

Roh, H. K., and Kim, Y. D., 1997, “Due-date based loading and scheduling methods for a flexible manufacturing system with an automatic tool transporter,” *International Journal of Production Research,* 35(11), 2989-3003.

Roderick, L.M., Phillips, D.T. and Hogg, G. L., 1992, "A comparison of order release strategies in production control systems," *International Journal of Production Research*, 30 (3), pp 611-626.

Savoie, R.M, 1987, “How to simply tool requirement planning: a new approach,” *Proceedings of American Production and Inventory Control Society Conference,* pp 100-103.

Sethi, A.K. and Sethi, S.P., 1990, "Flexibility in manufacturing: a survey," *International Journal of Flexible Manufacturing Systems,* 2, pp 289-328.

Shafaei, R. and Brunn, P., 1999a, “Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach,” *International Journal of Production Research*, 37 (17), pp 3913-3925.

Shafaei, R. and Brunn, P., 1999b, “Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment,” *International Journal of Production Research*, 37 (18), pp 4105-4117.

Shanthikumar, J.G. and Buzacott, J.A., 1981, “Open queuing network models of dynamic job shops,” *International Journal of Production Research,* 19 (3), pp 255-266.

Shaw, M.J., 1989, "FMS scheduling as cooperative problem solving," *Annals of Operations research*, 17, pp 323-346.

Shewchuk, J.P. and Chang, T.C., 1995, “Resource-constrained job scheduling with recyclable resources,” *European Journal of Operational Research,* 81, pp 364-375.

Shewchuk, J.P. and Moodie, C.L., 1998, “Definition and classification of manufacturing flexibility types and measures,” *International Journal of Flexible Manufacturing Systems,* 10, pp 325-349.

Song, C.Y., Hwang, H, and Kim, Y.D., 1995, “Heuristic algorithm for the tool movement policy in flexible manufacturing systems”, *Journal of Manufacturing Systems*, 14(3), 160-168.

Stecke, K.E., 1983, "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems,” *Management Science,* 29(3), pp 273-288.

Stecke, K.E., 1985, “Design, planning, scheduling, and control problems of flexible manufacturing systems,” *Annals of Operations Research*, 3, 3-12.

Stecke, K.E., and Raman, N, 1994, “Production planning decisions in flexible manufacturing systems with random material flows,” *IIE Transactions,* 26(5), 2-17.

Stecke, K.E., and Raman, N., 1995, “FMS planning decisions, operating flexibilities, and System performance,” *IEEE Transactions on Engineering Management,* 42(1), 82-90.

Sun, D. and Lin, L., 1994, “A dynamic job shop scheduling framework: a backward approach,” *International Journal of Production Research*, 32, pp 967-985.

Talavage, J. and Hannam, R.G., 1988, *Flexible Manufacturing Systems in Practice, Applications, Design and Simulation*, Marcel Dekker, Inc., New York.

Tang, C.S., and Denardo, E.V. 1988, “Models arising from a flexible manufacturing machine, part II: minimization of the number switching instants,” *Operations Research*, 36(5), 778-784.

Tempelmeier, H. and Kuhn, H., 1993, *Flexible Manufacturing Systems: Decision Support for Design and Operation*, John Wiley and Sons, Inc, New York.

Upton, D.M., Barash, M.M., and Matheson, A.M., 1991, "Architectures and Auctions in Manufacturing," *International Journal of Computer-Integrated Manufacturing*, 4 (1), pp 23-33.

Vepsalainen, A.P.J., and Morton, T.E., 1987, “Priority rules for job shops with weighted tardiness costs,” *Management Science*, 33, pp 1035-1047.

Wang, J., Yang, J. andGargeya, V.B., 1993, “Tool requirement planning in stochastic job shops: a simulated annealing approach,” *Computers and Industrial Engineering,* 24 (2), pp 249-265.

Wassweiler, W.R., 1982, “Tool requirement planning,” *Proceedings of American Production and Inventory Control Society Conference,* pp 160-162.

Whitt, W., 1983, “The queuing network analyzer,” *The Bell System Technical Journal,* 62 (9), pp 2779-2815.

Xu, Z. and Randhawa, S., 1998, “Evaluation of scheduling strategies for a dynamic job shop in a tool-shared, flexible manufacturing environment,” *Production Planning and Control*, 9 (1), pp 74-86.

Yang, E-H. , 1994, "Investigation of dynamic job allocation methods for computerized manufacturing systems with priority and due-date considerations," Unpublished Ph.D. Thesis, School of Industrial Engineering, Purdue University, W. Lafayette, IN.

Yao, D.D., 1985, “Material and Information Flows in Flexible Manufacturing Systems,” *Material Flow,* 2, pp143-149.

Yao, D.D. and Pei, F.F., 1990, "Flexible parts routing in manufacturing systems," *IIE Transactions*, 22 (1), pp 48-55.

Zeestraten, M.J., 1990, "The look ahead dispatching procedure," *International Journal of Production Research*, 28 (2), pp 369-384.

Appendix A: Additional Simulation Input Data

A.1 Part Types and Routings for Hypothetical Job Shops

PartTypeID	OptnID	OptnNo	StnTypeID	PalletType	NumOfSeq	OptnTime
A1	1	1	VTL	1	10	31.89812922
A1	2	2	VMC	2	10	30.1866512
A1	3	3	SHP	3	10	34.90737633
A2	4	1	VTL	1	10	37.4202704
A2	5	2	SHP	2	10	32.93575854
A2	6	3	VMC	3	10	23.80437636
A3	7	1	VMC	1	10	27.48185675
A3	8	2	VTL	2	10	26.54847865
A3	9	3	SHP	3	10	29.47703482
B1	10	1	VMC	1	10	27.83257546
B1	11	2	SHP	2	10	32.52937407
B1	12	3	VTL	3	10	32.00531022
B2	13	1	SHP	1	10	26.44605853
B2	14	2	VTL	2	10	28.50373852
B2	15	3	VMC	3	10	33.02243111
C1	16	1	SHP	1	10	24.96395764
C1	17	2	VMC	2	10	27.55766473
C1	18	3	VTL	3	10	32.15594958
C2	19	1	VTL	1	10	32.19257179
C2	20	2	VMC	2	10	27.16617329
C2	21	3	SHP	3	10	25.8702353
C3	22	1	VTL	1	10	28.69771416
C3	23	2	SHP	2	10	29.60301523
C3	24	3	VMC	3	10	23.37199011
C4	25	1	VMC	1	10	33.69750053
C4	26	2	VTL	2	10	31.61552782
C4	27	3	SHP	3	10	30.70937224
D1	28	1	VMC	1	10	22.99050874
D1	29	2	SHP	2	10	35.12833033
D1	30	3	VTL	3	10	30.77773369

A.2 Operation Steps for Hypothetical Job Shops

SequenceID	OptnID	SeqNo	ToolTypeID	Seq_Time	SequenceID	OptnID	SeqNo	ToolTypeID	Seq_Time
1	1	1	TT007	4.85888	151	16	1	TT076	2.27812
2	1	2	TT018	1.7405	152	16	2	TT099	2.00308
3	1	3	TT035	3.68905	153	16	3	TT100	1.28248
4	1	4	TT024	3.81796	154	16	4	TT095	2.06375
5	1	5	TT009	4.24021	155	16	5	TT097	3.91379
6	1	6	TT029	2.34941	156	16	6	TT080	3.62471
7	1	7	TT015	2.48564	157	16	7	TT099	1.18348
8	1	8	TT018	2.91009	158	16	8	TT086	3.00836
9	1	9	TT008	4.46385	159	16	9	TT081	2.30155
10	1	10	TT034	1.34254	160	16	10	TT089	3.30464
11	2	1	TT070	3.99973	161	17	1	TT041	1.55055
12	2	2	TT054	4.29221	162	17	2	TT037	4.78002
13	2	3	TT038	2.2692	163	17	3	TT054	2.44462
14	2	4	TT067	4.8977	164	17	4	TT058	2.27335
15	2	5	TT068	1.96475	165	17	5	TT056	1.67617
16	2	6	TT041	2.01334	166	17	6	TT054	4.72826
17	2	7	TT039	3.79501	167	17	7	TT041	4.0079
18	2	8	TT061	3.63533	168	17	8	TT048	2.21122
19	2	9	TT072	1.65761	169	17	9	TT064	1.42457
20	2	10	TT062	1.66176	170	17	10	TT058	2.46098
21	3	1	TT094	2.92608	171	18	1	TT012	4.05698
22	3	2	TT080	4.65111	172	18	2	TT003	4.8667
23	3	3	TT077	3.85018	173	18	3	TT028	1.90103
24	3	4	TT088	3.96542	174	18	4	TT018	3.1551
25	3	5	TT080	3.9808	175	18	5	TT026	4.07675
26	3	6	TT083	4.35716	176	18	6	TT029	3.99374
27	3	7	TT099	3.44295	177	18	7	TT031	3.99948
28	3	8	TT100	3.09333	178	18	8	TT004	3.30354
29	3	9	TT097	2.76031	179	18	9	TT016	1.652
30	3	10	TT098	1.88003	180	18	10	TT007	1.15064
31	4	1	TT019	2.93255	181	19	1	TT025	3.90011
32	4	2	TT031	3.02252	182	19	2	TT009	4.87194
33	4	3	TT017	4.83129	183	19	3	TT021	1.22877
34	4	4	TT018	4.9331	184	19	4	TT028	3.29951
35	4	5	TT003	2.75237	185	19	5	TT030	2.78582
36	4	6	TT009	4.79308	186	19	6	TT015	3.863
37	4	7	TT022	4.55589	187	19	7	TT028	2.1298
38	4	8	TT020	4.47069	188	19	8	TT022	4.51256
39	4	9	TT027	2.5446	189	19	9	TT035	2.78228
40	4	10	TT006	2.58415	190	19	10	TT033	2.81878
41	5	1	TT080	1.68606	191	20	1	TT046	4.50108

42	5	2	TT092	2.12235	192	20	2	TT068	2.73638
43	5	3	TT078	4.81945	193	20	3	TT073	1.2754
44	5	4	TT099	4.99792	194	20	4	TT056	2.31132
45	5	5	TT091	4.79321	195	20	5	TT057	2.07816
46	5	6	TT093	2.99908	196	20	6	TT062	1.65261
47	5	7	TT094	1.8533	197	20	7	TT075	1.81069
48	5	8	TT078	4.1301	198	20	8	TT070	4.17441
49	5	9	TT076	1.01453	199	20	9	TT071	4.12229
50	5	10	TT083	4.51976	200	20	10	TT073	2.50383
51	6	1	TT068	1.11866	201	21	1	TT092	3.26801
52	6	2	TT037	1.68044	202	21	2	TT091	4.62304
53	6	3	TT052	1.07898	203	21	3	TT079	1.29749
54	6	4	TT072	3.91159	204	21	4	TT083	2.46757
55	6	5	TT067	1.49123	205	21	5	TT092	3.13959
56	6	6	TT074	1.03394	206	21	6	TT086	1.28553
57	6	7	TT059	4.01962	207	21	7	TT076	2.2349
58	6	8	TT044	4.84423	208	21	8	TT077	3.60591
59	6	9	TT041	3.15656	209	21	9	TT093	1.69387
60	6	10	TT065	1.46913	210	21	10	TT099	2.25431
61	7	1	TT050	3.64022	211	22	1	TT028	3.51643
62	7	2	TT073	1.77358	212	22	2	TT018	3.4294
63	7	3	TT072	1.1178	213	22	3	TT015	1.09986
64	7	4	TT049	2.96612	214	22	4	TT032	3.76424
65	7	5	TT042	4.43309	215	22	5	TT029	3.03095
66	7	6	TT040	1.03479	216	22	6	TT032	2.59062
67	7	7	TT043	4.11069	217	22	7	TT031	4.41575
68	7	8	TT039	2.57329	218	22	8	TT007	1.67629
69	7	9	TT045	3.11249	219	22	9	TT014	2.94256
70	7	10	TT050	2.71978	220	22	10	TT006	2.2316
71	8	1	TT006	1.57912	221	23	1	TT094	3.92819
72	8	2	TT025	3.83395	222	23	2	TT083	4.35142
73	8	3	TT030	4.16697	223	23	3	TT099	1.02783
74	8	4	TT031	1.54543	224	23	4	TT087	2.40031
75	8	5	TT006	1.26771	225	23	5	TT082	2.35624
76	8	6	TT020	3.55672	226	23	6	TT081	1.43092
77	8	7	TT011	3.23457	227	23	7	TT091	3.89315
78	8	8	TT008	1.8301	228	23	8	TT083	1.90738
79	8	9	TT024	3.36494	229	23	9	TT088	4.28489
80	8	10	TT020	2.16898	230	23	10	TT080	4.02268
81	9	1	TT093	1.50368	231	24	1	TT042	1.5342
82	9	2	TT076	2.35783	232	24	2	TT037	2.61101
83	9	3	TT079	2.53923	233	24	3	TT049	1.25208
84	9	4	TT099	4.08237	234	24	4	TT062	2.00894
85	9	5	TT100	4.23667	235	24	5	TT055	4.81286
86	9	6	TT097	2.28813	236	24	6	TT044	1.83267

87	9	7	TT093	3.99472	237	24	7	TT074	2.51579
88	9	8	TT079	3.07941	238	24	8	TT072	3.12726
89	9	9	TT095	3.85067	239	24	9	TT043	1.64516
90	9	10	TT085	1.54433	240	24	10	TT060	2.03201
91	10	1	TT058	3.88986	241	25	1	TT039	2.52153
92	10	2	TT041	2.41826	242	25	2	TT052	3.92453
93	10	3	TT056	4.04794	243	25	3	TT071	4.19883
94	10	4	TT040	2.45048	244	25	4	TT036	3.36982
95	10	5	TT053	2.56145	245	25	5	TT039	3.54463
96	10	6	TT068	2.69158	246	25	6	TT051	2.94647
97	10	7	TT040	3.65194	247	25	7	TT058	3.99155
98	10	8	TT050	2.49797	248	25	8	TT050	4.01657
99	10	9	TT051	1.44105	249	25	9	TT064	1.72475
100	10	10	TT071	2.18204	250	25	10	TT070	3.45882
101	11	1	TT088	1.15821	251	26	1	TT018	4.33628
102	11	2	TT089	3.57759	252	26	2	TT021	3.30305
103	11	3	TT080	4.68676	253	26	3	TT001	3.31611
104	11	4	TT084	1.15442	254	26	4	TT035	1.0968
105	11	5	TT078	3.67623	255	26	5	TT026	2.61321
106	11	6	TT097	4.40391	256	26	6	TT031	2.53337
107	11	7	TT090	3.13703	257	26	7	TT002	4.23069
108	11	8	TT078	4.80432	258	26	8	TT004	4.16965
109	11	9	TT080	2.35685	259	26	9	TT016	3.0473
110	11	10	TT088	3.57405	260	26	10	TT012	2.96905
111	12	1	TT034	4.9906	261	27	1	TT080	2.69634
112	12	2	TT029	3.42134	262	27	2	TT085	2.50896
113	12	3	TT020	2.08853	263	27	3	TT079	1.72353
114	12	4	TT010	3.28376	264	27	4	TT078	3.14423
115	12	5	TT022	4.87866	265	27	5	TT086	2.62468
116	12	6	TT005	4.23911	266	27	6	TT094	4.14878
117	12	7	TT021	4.66674	267	27	7	TT083	4.10031
118	12	8	TT002	1.02612	268	27	8	TT077	3.41487
119	12	9	TT012	1.51003	269	27	9	TT094	3.74313
120	12	10	TT030	1.90042	270	27	10	TT080	2.60454
121	13	1	TT094	2.4716	271	28	1	TT055	3.69564
122	13	2	TT078	1.91885	272	28	2	TT074	1.07007
123	13	3	TT094	2.57964	273	28	3	TT063	1.69399
124	13	4	TT091	1.43922	274	28	4	TT042	1.28114
125	13	5	TT099	2.78582	275	28	5	TT045	1.03943
126	13	6	TT093	1.61611	276	28	6	TT070	3.51582
127	13	7	TT083	4.57555	277	28	7	TT071	4.90112
128	13	8	TT093	3.16511	278	28	8	TT060	2.09708
129	13	9	TT084	1.53883	279	28	9	TT061	1.06897
130	13	10	TT076	4.35533	280	28	10	TT050	2.62725
131	14	1	TT020	1.09534	281	29	1	TT089	2.29777

132	14	2	TT017	2.38347	282	29	2	TT080	4.301
133	14	3	TT025	2.86822	283	29	3	TT087	4.57042
134	14	4	TT008	1.45851	284	29	4	TT081	3.89975
135	14	5	TT019	4.27866	285	29	5	TT082	2.57573
136	14	6	TT002	1.87075	286	29	6	TT080	3.32759
137	14	7	TT012	3.45894	287	29	7	TT100	4.41795
138	14	8	TT003	4.03647	288	29	8	TT087	4.05246
139	14	9	TT029	2.97186	289	29	9	TT084	3.60213
140	14	10	TT031	4.08151	290	29	10	TT081	2.08353
141	15	1	TT045	4.343	291	30	1	TT018	4.39537
142	15	2	TT069	4.44163	292	30	2	TT002	2.19999
143	15	3	TT063	2.51958	293	30	3	TT018	3.36323
144	15	4	TT049	3.34187	294	30	4	TT022	4.85864
145	15	5	TT062	3.17243	295	30	5	TT022	1.26722
146	15	6	TT071	4.38853	296	30	6	TT004	3.80844
147	15	7	TT065	1.8179	297	30	7	TT022	3.52962
148	15	8	TT055	2.69329	298	30	8	TT017	4.39256
149	15	9	TT075	1.38649	299	30	9	TT027	1.54262
150	15	10	TT070	4.91772	300	30	10	TT012	1.42006

A.3 Tool Life Data for Hypothetical Job Shops

ToolTypeID	ToolLife	ToolTypeID	ToolLife	ToolTypeID	ToolLife
TT001	66.32221	TT037	60.4765	TT073	37.01875
TT002	46.63778	TT038	45.38408	TT074	30.79867
TT003	77.70358	TT039	62.17231	TT075	31.9718
TT004	75.21083	TT040	47.58141	TT076	48.9628
TT005	84.78225	TT041	52.58644	TT077	72.47312
TT006	38.31294	TT042	48.3228	TT078	74.97726
TT007	51.23875	TT043	57.55852	TT079	43.19834
TT008	51.6831	TT044	66.76901	TT080	68.97899
TT009	92.7016	TT045	56.63279	TT081	48.57875
TT010	65.67522	TT046	90.02167	TT082	49.31974
TT011	64.69131	TT048	44.22437	TT083	75.08329
TT012	53.66021	TT049	50.4005	TT084	41.96926
TT014	58.85128	TT050	62.00714	TT085	40.53285
TT015	49.65667	TT051	43.87524	TT086	46.12384
TT016	46.99301	TT052	50.0351	TT087	73.48796
TT017	77.38212	TT053	51.22898	TT088	64.91287
TT018	70.65767	TT054	76.43401	TT089	61.19999
TT019	72.11219	TT055	74.67859	TT090	62.74056
TT020	53.52104	TT056	53.56955	TT091	73.7431
TT021	61.32369	TT057	41.56316	TT092	56.86636
TT022	78.67529	TT058	63.07871	TT093	49.90855
TT024	71.82897	TT059	80.39247	TT094	61.85919
TT025	70.68189	TT060	41.29093	TT095	59.14426
TT026	66.89962	TT061	47.04306	TT097	66.83065
TT027	40.87222	TT062	42.47871	TT098	37.60063
TT028	54.23383	TT063	42.13569	TT099	54.44441
TT029	63.06919	TT064	31.49327	TT100	65.15213
TT030	59.02137	TT065	32.87027		
TT031	65.3269	TT067	63.88928		
TT032	63.54869	TT068	42.55684		
TT033	56.37562	TT069	88.83267		
TT034	63.3314	TT070	80.266		
TT035	50.45422	TT071	79.17124		
TT036	67.39647	TT072	49.07132		

A.4 Part-tool Matrix used for the Loading Model in Single Stage Systems

Part Type/ ToolTypeNo	1	2	3	4	5	6	7	8	9	10
1			1	1		1				
2	1		1		1	1				1
3						1		1		1
4	1									1
5	1			1						
6			1			1	1		1	
7			1				1			1
8	1	1	1		1			1		
9			1			1				
10			1						1	1
11		1					1	1		
12			1	1		1			1	
13	1			1				1	1	
14	1	1		1		1	1	1		
15									1	1
16		1		1						1
17			1			1				1
18		1			1	1			1	
19			1		1			1	1	
20				1						1
21		1		1	1					
22					1		1		1	
23		1				1	1		1	
24		1	1						1	1
25							1	1	1	1
26	1			1			1		1	
27		1	1	1	1		1			
28		1			1			1		
29			1			1			1	1
30	1			1			1	1	1	
31						1	1			
32					1		1	1		
33	1					1			1	
34			1		1	1		1		1
35				1	1		1			1
36					1			1		1
37		1		1						
38				1			1			1
39	1	1		1						
40			1		1					1

41	1			1						
42					1	1				
43					1	1		1		
44	1		1							
45						1	1			1
46			1	1						1
47						1		1		1
48			1	1		1		1	1	
49						1			1	1
50	1							1		
51						1			1	
52	1								1	1
53				1		1				
54		1		1						
55	1				1				1	
56			1				1			
57		1	1		1		1	1		
58						1	1		1	
59	1						1		1	
60	1				1			1		1
61	1					1				1
62	1	1	1					1		1
63		1	1	1	1			1		1
64	1	1					1			1
65							1		1	
66		1			1			1		
67			1	1	1	1			1	
68	1			1			1			
69				1			1	1		
70	1	1					1			
71			1	1				1		
72	1	1	1	1	1					
73		1					1		1	
74				1	1	1		1	1	
75	1	1						1		
76	1						1		1	
77	1	1			1	1		1	1	1
78				1	1		1			
79	1	1			1			1	1	1
80		1					1	1		
81	1		1		1	1				
82		1	1	1				1	1	
83			1		1	1	1			
84		1			1					1
85			1				1	1		

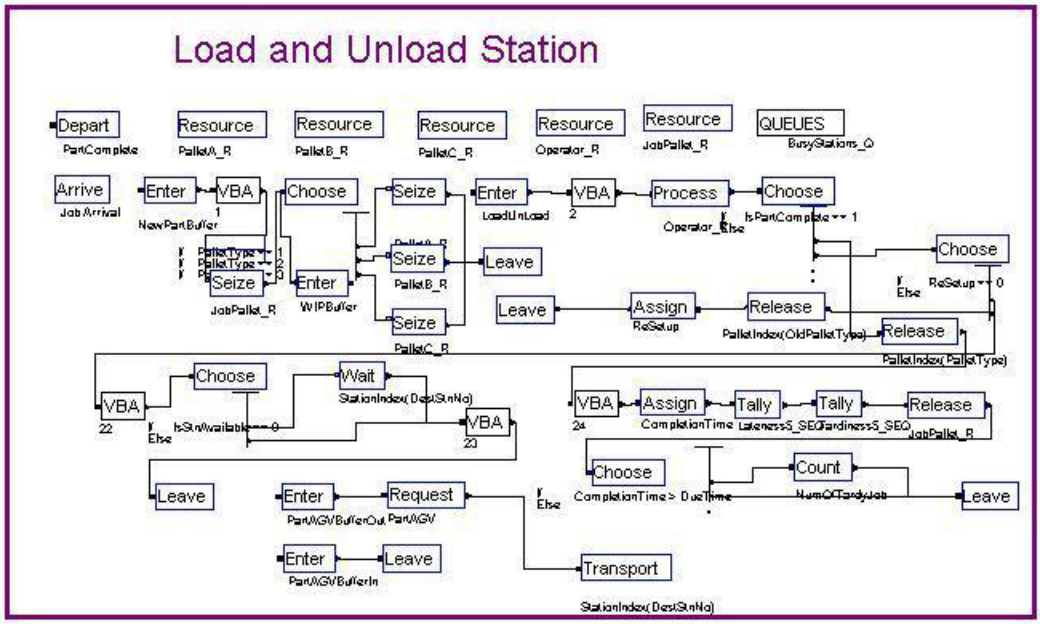
86	1	1	1							
87	1	1			1					
88		1		1		1				

A.5 Machine Downtime Data used for Hypothetical Models

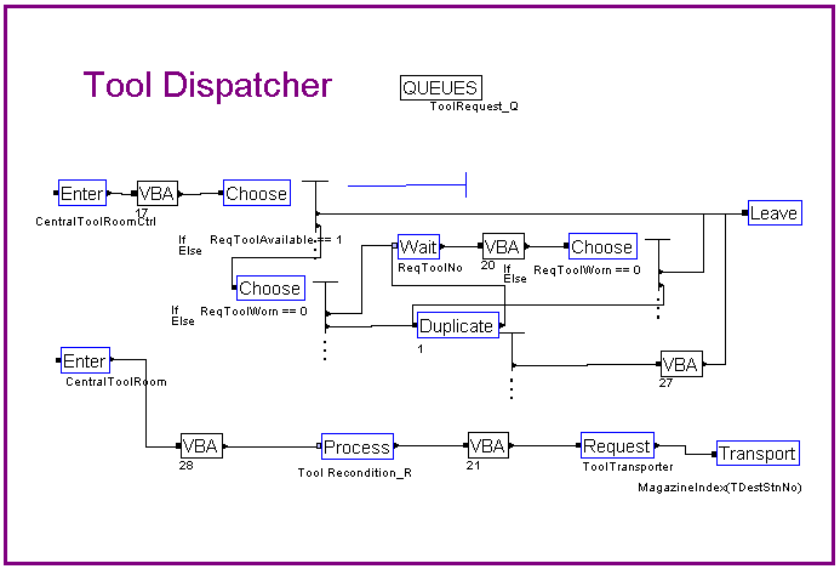
StationNo	StnNo	StnTypeID	MTBF	MTTR
1	661	VTL	829.20011	92.13335
2	662	VTL	660.408338	73.3787
3	663	VTL	957.890561	106.4323
4	664	VMC	1139.46348	126.6071
5	665	VMC	1130.76571	125.6406
6	666	VMC	1175.07859	130.5643
7	667	SHP	608.697775	67.63309
8	668	SHP	844.453261	93.82814
9	669	SHP	1117.94794	124.2164

Appendix B: Arena Simulation Logic

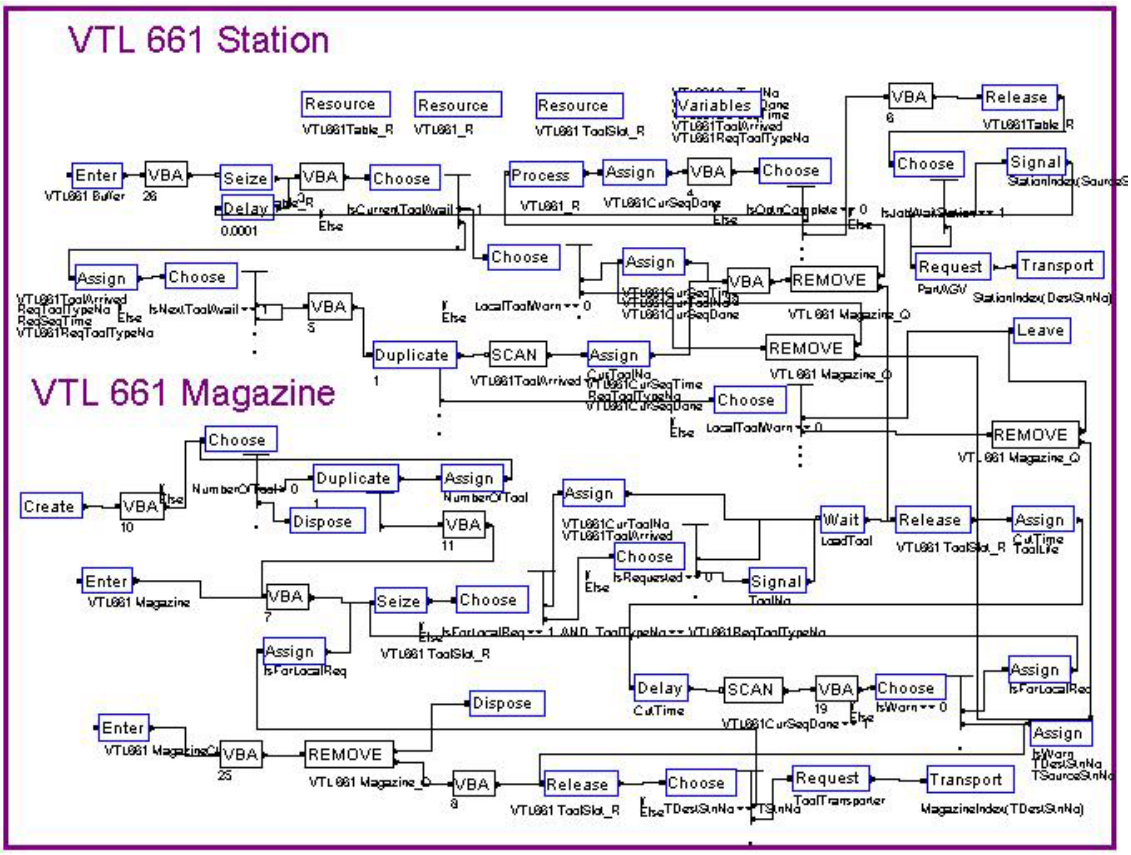
B.1 Load and Unload Station Logic



B.2 Central Tool Room and Tool Dispatcher Logic



B.3 Workstation and Tool Magazine Logic



Appendix C: Source Code for Dynamic Scheduler

C.1 Visual Basic Code for Building MCF Model for Sub-problems

```
Public Nodes As New Collection
Public NumOfFlow As Integer
Public MPL As OptiMax
Public MCFModel As MPLLib.Model
Private ModelString As String
Dim PathVar As VariableVector

Public Sub ParseAndSolve()
   Dim result As Integer
   Set MPL = New OptiMax
   Set MCFModel = New MPLLib.Model
   MPL.Solvers.Add ("CPLEX")
   MPL.StatusWindow.Automatic = False
   MPL.StatusWindow.visible = False
   Set MCFModel = MPL.Models.Add("MCFModel")
   Call SetModelString
   result = MCFModel.ParseModel(ModelString)
   If (result > 0) Then
      MsgBox MCFModel.ErrorMessage, vbOKOnly, "Error found in MCF model"
      Exit Sub
   End If
   result = MCFModel.Solve
   If (result > 0) Then
      MsgBox MCFModel.ErrorMessage, vbOKOnly, "Can not solve the MCF model"
      Exit Sub
   End If
   Call ReadNetworkSolution

End Sub

Private Sub ReadNetworkSolution()
   Dim newNode As New Node
   Dim newArc As New PArc
   Dim i As Integer
   Set PathVar = MCFModel.VariableVectors("Path")
   PathVar.MoveFirstPos
   Do While PathVar.PosValid
      Set newNode = Nodes.Item(PathVar.Subscripts(1).value)
      If newNode.Arcs.Count > 0 Then
         For i = 1 To newNode.Arcs.Count
            Set newArc = newNode.Arcs.Item(i)
            If newArc.ToIndexNo = PathVar.Subscripts(2).value Then
               newArc.Activity = PathVar.Variable.Activity
               Exit For
            End If
```

```
            Next i
        End If
        PathVar.MoveNextPos
    Loop
End Sub

Private Sub SetModelString()
    Dim newNode As New Node
    Dim newArc As New PArc
    Dim NumOfNode As Integer
    Dim NumOfArc As Integer
    Dim i As Integer
    Dim j As Integer
    NumOfNode = Nodes.Count
    If NumOfNode = 0 Or NumOfFlow = 0 Then
        MsgBox "Nodes and arcs should be build before parsing model!"
        Exit Sub
    End If
    ModelString = ""
    ModelString = ModelString and "TITLE" and vbCrLf
    ModelString = ModelString and "  MCFModel;" and vbCrLf
    ModelString = ModelString and "INDEX" and vbCrLf
    ModelString = ModelString and "  node := 1.." and CStr(NumOfNode) and ";" and vbCrLf
    ModelString = ModelString and "  FromNode := node;" and vbCrLf
    ModelString = ModelString and "  ToNode   := node;" and vbCrLf
    ModelString = ModelString and "SPARSE DATA" and vbCrLf
    ModelString = ModelString and "  Distance[FromNode, ToNode] := [" and vbCrLf
    For i = 1 To NumOfNode
        Set newNode = Nodes.Item(i)
        NumOfArc = newNode.Arcs.Count
        If NumOfArc > 0 Then
            For j = 1 To NumOfArc
                Set newArc = newNode.Arcs.Item(j)
                If i = NumOfNode - 1 And j = NumOfArc Then
                    ModelString = ModelString and "    " and CStr(newNode.IndexNo) _
                      and ", " and CStr(newArc.ToIndexNo) and ", " _
                      and Format(newArc.Distance, "#####.#####") and " ];" and vbCrLf
                Else
                    ModelString = ModelString and "    " and CStr(newNode.IndexNo) _
                      and ", " and CStr(newArc.ToIndexNo) and ", " _
                      and Format(newArc.Distance, "######.#####") and "," and vbCrLf
                End If
            Next j
        End If
    Next i
    ModelString = ModelString and "  StartCity := 1;" and vbCrLf
    ModelString = ModelString and "  EndCity := " and CStr(NumOfNode) and ";" and vbCrLf
    ModelString = ModelString and "  Flow := " and CStr(NumOfFlow) and ";" and vbCrLf
    ModelString = ModelString and "VARIABLES" and vbCrLf
    ModelString = ModelString and "  Path[FromNode, ToNode] WHERE (Distance > 0);" and vbCrLf
    ModelString = ModelString and "MODEL" and vbCrLf
    ModelString = ModelString and "  MIN TotalDistance" and vbCrLf
    ModelString = ModelString and "      = SUM(FromNode, ToNode: Distance * Path);" and vbCrLf
```

```
        ModelString = ModelString and "  FlowBalance[node] :" and vbCrLf

        ModelString = ModelString and "SUBJECT TO" and vbCrLf

        ModelString = ModelString and "    Flow IF (node=StartCity) - Flow IF (node=EndCity) +" and vbCrLf


        ModelString = ModelString and "     SUM(FromNode: Path[FromNode, ToNode:=node])" and vbCrLf
        ModelString = ModelString and "    =" and vbCrLf
        ModelString = ModelString and "     SUM(ToNode: Path[FromNode:=node, ToNode]);" and vbCrLf
        ModelString = ModelString and "  FlowCapacity[FromNode, ToNode] WHERE (Distance > 0):" and vbCrLf
        ModelString = ModelString and "    Path[FromNode, ToNode] <= 1; " and vbCrLf
        ModelString = ModelString and "END" and vbCrLf

    End Sub
```

C.2 Visual Basic Code for the Cell-level Sub-problem Heuristics

```
Private Sub OptimizeOptnSchedule(OptnSlacktime As Double, OptnHorizonlength As Double)
    Dim i As Integer
    Dim j As Integer
    Dim NumOfOperation As Integer
    Dim NumOfStation As Integer
    Dim newStnTypes As New Collection
    If Jobs.Count = 0 Then Exit Sub
    Dim OptnMCFModel As New MCFProblem
    Set OptnMCFModel = New MCFProblem
    NumOfOperation = 0
    NumOfStation = 0
    g_SIMAN.VariableArrayValue(g_OptnSchedStartTimeidx) = Timer

    Dim newStartNode As New Node
    Set newStartNode = New Node
    newStartNode.IndexNo = 1
    newStartNode.NodeType = STARTCITYNODE
    OptnMCFModel.Nodes.Add newStartNode

    Call CheckStationStatus

    For i = 1 To Jobs.Count
        Dim newJob As New Job
        Set newJob = New Job
        Set newJob = Jobs.Item(i)
        Dim newparttype As New Part_Type
        Set newparttype = PartTypes.Item(newJob.PartTypeNo)
        If newJob.CurrentStatus <> PARTIDLEBFLOAD And _
            newJob.CurrentStatus <> PARTISCOMPLETE Then
            If newJob.CurrentOptnStatus = PARTOPTNPENDING _
                Or newJob.CurrentOptnStatus = PARTOPTNDECIDED Then
                NumOfOperation = NumOfOperation + 1
                Dim newoptn As New Operation
                Set newoptn = New Operation
                Set newoptn = newparttype.Operations.Item(newJob.CurrentOptnNo)
                Dim newNode As New Node
                Set newNode = New Node
                newNode.JobNo = newJob.JobNo
                newNode.OptnNo = newJob.CurrentOptnNo
                newNode.NodeType = OPERATIONNODE
                newNode.StnTypeID = newoptn.StnTypeID
                newNode.ProcTime = newoptn.OptnTime
                newNode.TotalProcTime = newJob.TotalProcTime
                newNode.RemainProcTime = newJob.RemainProcTime
                newNode.StnNo = newJob.CurrentStnNo
                newNode.IndexNo = 1 + NumOfOperation
                newNode.Resetup = newJob.CurPalletSetUp
                OptnMCFModel.Nodes.Add newNode
                newJob.CurrentOptnProcTime = newoptn.OptnTime
```

```
            Dim newStnType As New StationType
            Set newStnType = New StationType
            Dim IsStnTypeIn As Boolean
            IsStnTypeIn = False
            If newStnTypes.Count > 0 Then
               For j = 1 To newStnTypes.Count
                  Set newStnType = newStnTypes.Item(j)
                  If newStnType.StationTypeID = newNode.StnTypeID Then IsStnTypeIn = True
               Next j
            End If
            If IsStnTypeIn = False Then
               Dim StnType As New StationType
               Set StnType = New StationType
               StnType.StationTypeID = newNode.StnTypeID
               newStnTypes.Add StnType
            End If

         Else
            If newJob.CurrentOptnStatus = PARTOPTNCOMMIT _
               And (newJob.NextOptnStatus = PARTOPTNPENDING Or newJob.NextOptnStatus =
PARTOPTNDECIDED) _
               And newJob.NextIdleTime < g_SIMAN.RunCurrentTime + OptnHorizonlength Then

               NumOfOperation = NumOfOperation + 1
               Dim newOptn1 As New Operation
               Set newOptn1 = New Operation
               Set newOptn1 = newparttype.Operations.Item(newJob.NextOptnNo)
               Dim newNode1 As New Node
               Set newNode1 = New Node
               newNode1.JobNo = newJob.JobNo
               newNode1.OptnNo = newJob.NextOptnNo
               newNode1.NodeType = OPERATIONNODE
               newNode1.StnTypeID = newOptn1.StnTypeID
               newNode1.ProcTime = newOptn1.OptnTime
               newNode1.TotalProcTime = newJob.TotalProcTime
               newNode1.RemainProcTime = newJob.RemainProcTime
               newNode1.StnNo = newJob.CurrentStnNo
               newNode1.IndexNo = 1 + NumOfOperation
               newNode1.Resetup = newJob.NextPalletSetup
               OptnMCFModel.Nodes.Add newNode1
               newJob.NextOptnProcTime = newOptn1.OptnTime

               Dim newStnType1 As New StationType
               Set newStnType1 = New StationType
               Dim IsStnTypeIn1 As Boolean
               IsStnTypeIn1 = False
               If newStnTypes.Count > 0 Then
                  For j = 1 To newStnTypes.Count
                     Set newStnType1 = newStnTypes.Item(j)
                     If newStnType1.StationTypeID = newNode1.StnTypeID Then IsStnTypeIn1 = True
                  Next j
               End If
```

```
                Dim StnType1 As New StationType

            If IsStnTypeIn1 = False Then

                Set StnType1 = New StationType



                StnType1.StationTypeID = newNode1.StnTypeID
                newStnTypes.Add StnType1
            End If

        End If
    End If
  End If
Next i
If NumOfOperation = 0 Then
  g_SIMAN.VariableArrayValue(g_OptnSchedEndTimeidx) = Timer
  Exit Sub
End If
For i = 1 To newStnTypes.Count
  Dim newStn As New Station
  Dim StnType2 As New StationType
  Set StnType2 = New StationType
  Set StnType2 = newStnTypes.Item(i)
  For j = 1 To Stations.Count
    Set newStn = New Station
    Set newStn = Stations.Item(j)
    If newStn.StnTypeID = StnType2.StationTypeID And IncludeStation(i) = True Then
      Dim newNode2 As New Node
      Set newNode2 = New Node
      NumOfStation = NumOfStation + 1
      newNode2.NodeType = STATIONNODE
      newNode2.StnTypeID = newStn.StnTypeID
      newNode2.StnNo = newStn.StnNo
      newNode2.IndexNo = 1 + NumOfOperation + NumOfStation
      OptnMCFModel.Nodes.Add newNode2
    End If
  Next j
Next i

Dim newEndNode As New Node
Set newEndNode = New Node
newEndNode.IndexNo = NumOfOperation + NumOfStation + 2
newEndNode.NodeType = ENDCITYNODE
OptnMCFModel.Nodes.Add newEndNode

If NumOfStation = 0 Then
  g_SIMAN.VariableArrayValue(g_OptnSchedEndTimeidx) = Timer
  Exit Sub
End If
Dim newFromnode As New Node
Set newFromnode = OptnMCFModel.Nodes(1)
For i = 2 To NumOfOperation + 1
  Dim newToNode As New Node
```

```
        Set newToNode = OptnMCFModel.Nodes.Item(i)
        If newToNode.NodeType <> OPERATIONNODE Then MsgBox "Error in Operation node data"
        Dim newStartArc As New PArc
        Set newStartArc = New PArc
        newStartArc.ToIndexNo = i
        newStartArc.Distance = 0.0001
```

```
        newFromnode.Arcs.Add newStartArc
```

```
        For j = 1 To NumOfStation
          Dim nextToNode As New Node
          Set nextToNode = New Node
          Set nextToNode = OptnMCFModel.Nodes.Item(1 + NumOfOperation + j)
          If nextToNode.NodeType <> STATIONNODE Then MsgBox "Error in Station node data"
          If newToNode.StnTypeID = nextToNode.StnTypeID Then
            Dim newVarArc As New PArc
            Set newVarArc = New PArc
            newVarArc.ToIndexNo = 1 + NumOfOperation + j
            Dim newSchedJob As New Job
            Dim newSchedStn As New Station
            Set newSchedJob = Jobs.Item(FindJobIndex(newToNode.JobNo))
            Set newSchedStn = Stations.Item(nextToNode.StnNo)

            newVarArc.DueTime = newSchedJob.DueTime
            If newToNode.Resetup = 0 Then
              newVarArc.StartTime = Max(Max(g_SIMAN.RunCurrentTime, newSchedJob.NextIdleTime) _
                + PartAGVTime(HMC668, newToNode.StnNo) + PartAGVTime(newToNode.StnNo, _
                PARTAGVBUFOUT) + 2 + PartAGVTime(HMC668, PARTAGVBUFOUT) +
PartAGVTime(PARTAGVBUFOUT, nextToNode.StnNo), newSchedStn.NextIdleTime)
            Else
              newVarArc.StartTime = Max(Max(g_SIMAN.RunCurrentTime, newSchedJob.NextIdleTime) _
                + PartAGVTime(HMC668, newToNode.StnNo) + PartAGVTime(newToNode.StnNo, _
                nextToNode.StnNo), newSchedStn.NextIdleTime)
            End If

            newVarArc.FinishTime = GetCompleteTime(newVarArc.StartTime, newToNode.ProcTime,
nextToNode.StnNo)

            newVarArc.Distance = Max(1, newVarArc.DueTime - g_SIMAN.RunCurrentTime -
newSchedJob.RemainProcTime)
            newVarArc.Distance = Max(0.0001, newVarArc.Distance / newSchedJob.RemainProcTime)

            newVarArc.Distance = Max(0.0001, newVarArc.Distance * newVarArc.StartTime)
            newToNode.Arcs.Add newVarArc

          End If

        Next j

      Next i
```

```
Set endNode = OptnMCFModel.Nodes.Item(NumOfOperation + NumOfStation + 2)
For i = NumOfOperation + 2 To NumOfOperation + NumOfStation + 1
   Dim StnEndNode As New Node
   Set StnEndNode = OptnMCFModel.Nodes.Item(i)
   If StnEndNode.NodeType <> STATIONNODE Then MsgBox "Error in Station Node data"
   Dim newEndArc As New PArc
   Set newEndArc = New PArc
   newEndArc.ToIndexNo = endNode.IndexNo
   newEndArc.Distance = 0.0001
   StnEndNode.Arcs.Add newEndArc
Next i
```

```
OptnMCFModel.NumOfFlow = Min(CDbl(NumOfOperation), CDbl(NumOfStation))
OptnMCFModel.ParseAndSolve

Dim newReadFromnode As New Node
Set newReadFromnode = OptnMCFModel.Nodes(1) ' The start city node
For i = 1 To NumOfOperation ' for each Operation node
   Dim newReadToNode As New Node
   Set newReadToNode = New Node
   Set newReadToNode = OptnMCFModel.Nodes.Item(i + 1)
   If newReadToNode.NodeType <> OPERATIONNODE Then MsgBox "Error in Operation node data"
   Dim newReadStartArc As New PArc
   Set newReadStartArc = New PArc
   Set newReadStartArc = newReadFromnode.Arcs.Item(i)
   If newReadStartArc.Activity = 1 Then
      ' if the operation is selected, then inspect its arc variables
      For j = 1 To newReadToNode.Arcs.Count
         Dim newReadVarArc As New PArc
         Set newReadVarArc = New PArc
         Set newReadVarArc = newReadToNode.Arcs.Item(j)
         If newReadVarArc.Activity = 1 Then
            Dim nextReadToNode As New Node ' for each station node
            Set nextReadToNode = New Node
            Set nextReadToNode = OptnMCFModel.Nodes.Item(newReadVarArc.ToIndexNo)
            If nextReadToNode.NodeType <> STATIONNODE Then MsgBox "Error in Station node data"
            Dim newReadJob As New Job
            Dim newReadStation As New Station
            Set newReadJob = New Job
            Set newReadStation = New Station
            Set newReadJob = Jobs.Item(FindJobIndex(newReadToNode.JobNo))
            Set newReadStation = Stations.Item(nextReadToNode.StnNo)
            If newReadToNode.OptnNo = newReadJob.CurrentOptnNo Then
               newReadJob.CurrentOptnStnNo = nextReadToNode.StnNo
               newReadJob.CurrentOptnStartTime = newReadVarArc.StartTime
               newReadJob.CurrentoptnFinishTime = newReadVarArc.FinishTime
               If newReadJob.CurrentOptnStartTime <= g_SIMAN.RunCurrentTime + OptnSlacktime Then
                  newReadJob.CurrentOptnStatus = PARTOPTNCOMMIT
                  newReadJob.NextIdleTime = newReadJob.CurrentoptnFinishTime
```

```
                    Set newReadSchedOptn = New SchedOptn
                    Set newReadSchedOptn = New SchedOptn
                    newReadSchedOptn.JobNo = newReadJob.JobNo
                    newReadSchedOptn.OptnNo = newReadJob.CurrentOptnNo
                    newReadSchedOptn.OptnProcTime = newReadJob.CurrentOptnProcTime
                    newReadSchedOptn.OptnStartTime = newReadJob.CurrentOptnStartTime
                    newReadSchedOptn.OptnFinishTime = newReadJob.CurrentoptnFinishTime
                    If newReadSchedOptn.OptnNo = 0 Then MsgBox "Error in OptnNo = 0 for Scheduled Operation"
                    If newReadSchedOptn.OptnNo > newReadJob.NumOfOptn Then MsgBox "Error in OptnNo >
numofOptn for Scheduled Operation"

                    newReadStation.SchedOptns.Add newReadSchedOptn
                    newReadStation.NextIdleTime = newReadSchedOptn.OptnFinishTime

                    Dim newparttype1 As New Part_Type
                    Dim newReadCurOptn As New Operation
```

```
                    Set newparttype1 = New Part_Type
```

```
                    Set newReadCurOptn = New Operation
                    Set newparttype1 = PartTypes.Item(newReadJob.PartTypeNo)
                    Set newReadCurOptn = newparttype1.Operations.Item(newReadJob.CurrentOptnNo)
                    newReadJob.NextSeqIdleTime = newReadJob.CurrentOptnStartTime
                    newReadJob.CurSeqStatus = PARTSEQPENDING
                    newReadJob.CurSeqNo = 1
                    If newReadCurOptn.NumOfSeq > 1 Then
                        newReadJob.NextSeqNo = 2
                        newReadJob.NextSeqStatus = PARTSEQPENDING
                    Else
                        newReadJob.NextSeqStatus = PARTSEQINACTIVE
                    End If
                  Else
                    newReadJob.CurrentOptnStatus = PARTOPTNDECIDED
                  End If
                Else
                  If newReadToNode.OptnNo <> newReadJob.NextOptnNo Then MsgBox "Error in read OptnNo for
update schedule"
                  newReadJob.NextOptnStnNo = nextReadToNode.StnNo
                  newReadJob.NextOptnStartTime = newReadVarArc.StartTime
                  newReadJob.NextOptnFinishTime = newReadVarArc.FinishTime
                  If newReadJob.NextOptnStartTime <= g_SIMAN.RunCurrentTime + OptnSlacktime Then
                    newReadJob.NextOptnStatus = PARTOPTNCOMMIT
                    newReadJob.NextIdleTime = newReadJob.NextOptnFinishTime
                    Dim newReadSchedOptn1 As New SchedOptn
                    Set newReadSchedOptn1 = New SchedOptn
                    Set newReadSchedOptn1 = New SchedOptn
                    newReadSchedOptn1.JobNo = newReadJob.JobNo
                    newReadSchedOptn1.OptnNo = newReadJob.NextOptnNo
                    newReadSchedOptn1.OptnStartTime = newReadJob.NextOptnStartTime
                    newReadSchedOptn1.OptnProcTime = newReadJob.NextOptnProcTime
                    newReadSchedOptn1.OptnFinishTime = newReadJob.NextOptnFinishTime
```

```
                    If newReadSchedOptn1.OptnNo > newReadJob.NumOfOptn Then MsgBox "Error in OptnNo >
numofOptn for Scheduled Operation"
                    newReadStation.SchedOptns.Add newReadSchedOptn1
                    newReadStation.NextIdleTime = newReadSchedOptn1.OptnFinishTime

                  Else
                    newReadJob.NextOptnStatus = PARTOPTNDECIDED
                  End If

                End If

              End If

            Next j

          End If

        Next i
        g_SIMAN.VariableArrayValue(g_OptnSchedEndTimeidx) = Timer


      End Sub
```

c.3 Visual Basic code for calculation of estimated finish time

```
Public IsMachineDown As Boolean
Public ZeroTime As Double
Public StartTime As Double
Public CompleteTime As Double
Public ProcTime As Double
Public MTBF As Double
Public MTTR As Double
Public Sub Estimate()
    CompleteTime = FindGSCompleteTimeUp(StartTime, ProcTime, MTBF, MTTR)
End Sub

Private Function FindGSCompleteTimeUp(Xa As Double, _
  ProcTime As Double, MTBF As Double, MTTR As Double) As Double
  ' find the solution of completion time by Gold Section algorithm
  ' given start time Xa, proctime, and p(0,0), mtbf, mttr.
  Dim i, k As Integer
  Dim CompletionTime As Double
  Dim Alpha As Double ' Golden section parameter
  Dim Xb As Double
  Dim GSStart As Double
  Dim GSEnd As Double
  Dim GSTime As Double  ' Time used for finding completion time
  GSStart = Timer
  For i = 1 To 1000 ' find the upper bound of search scope
    Xb = Xa + i * MTTR
    If ThetaValueUp(Xa) * ThetaValueUp(Xb) < 0 Then
      Exit For
    End If
  Next i
  If ThetaValueUp(Xa) * ThetaValueUp(Xb) >= 0 Then
    MsgBox "Error in finding upper bound for golden section search"
  End If
  Alpha = 0.618
  Dim A(300) As Double
  Dim B(300) As Double
  Dim L(300) As Double
  Dim M(300) As Double
  Dim thetaL(300) As Double
  Dim thetaM(300) As Double
  A(1) = Xa
  B(1) = Xb
  L(1) = Xa + (1 - Alpha) * (Xb - Xa)
  M(1) = Xa + Alpha * (Xb - Xa)
  For k = 1 To 299
    ' Step 1 if bk-ak < 0.1 then stop
    If B(k) - A(k) <= 0.1 Then
      CompletionTime = (A(k) + B(k)) / 2
      Exit For
    End If
```

```
        thetaL(k) = Abs(ThetaValueUp(L(k)))
        thetaM(k) = Abs(ThetaValueUp(M(k)))
        If thetaL(k) > thetaM(k) Then
            ' Step 2 if thetal(k) > thetam(k)
            A(k + 1) = L(k)
            B(k + 1) = B(k)
            L(k + 1) = M(k)
            M(k + 1) = A(k + 1) + Alpha * (B(k + 1) - A(k + 1))
        Else
            ' Step 2 if thetal(k) <= thetam(k)
            A(k + 1) = A(k)
            B(k + 1) = M(k)
            L(k + 1) = A(k + 1) + (1 - Alpha) * (B(k + 1) - A(k + 1))
            M(k + 1) = L(k)
        End If
    Next k
    If B(k) - A(k) > 0.1 Then
        CompletionTime = (A(k) + B(k)) / 2
        MsgBox "Golden section search does not converge to 0.1"
    End If
    GSEnd = Timer
    GSTime = GSEnd - GSStart
    'Debug.Print "Found a completion time in " and CStr(GSTime) and " seconds"
    FindGSCompleteTimeUp = CompletionTime

End Function

Private Function ThetaValueUp(X As Double) As Double
    ' Find the Theta(X) function value of Finish time X, given Start Xa
    Dim lamda As Double
    Dim Mu As Double
    Dim Proc As Double
    Dim Xa As Double
    Dim TValue As Double
    lamda = 1 / MTBF
    Mu = 1 / MTTR
    Proc = ProcTime
    Xa = StartTime
    TValue = (Mu / (lamda + Mu)) * (X - Xa) - (lamda / ((lamda + Mu) ^ 2)) * (Exp(-1 * (lamda + Mu) * X) - Exp(-1
* (lamda + Mu) * Xa)) - Proc
    ThetaValueUp = TValue
End Function
```

C.4 Visual Basic code for Heuristic Machine Loading Procedure

```
Dim Operations As Collection
Dim OptnsAvailable As Collection ' P0 list of parts unassigned
Dim OptnsAssigned As Collection ' P1 list of parts assigned
Dim ToolTypes As Collection
Dim ToolsAvailable As Collection ' T0 list of tools for Part i at machine j
Dim ToolsLoaded As Collection 'T1 list of tools loaded
Dim ToolsConsidered As Collection  'T3 list of tools considered
Dim Stations As Collection
Dim StnsAvailable As Collection ' M0 list
Dim StnsAssigned As Collection ' M1 list
Dim StnsConsidered As Collection ' M2 list
Dim StnsPreselected As Collection 'M3 List

Const iCount As Integer = 10
Const jCount As Integer = 5
Const tCount As Integer = 50
Const MRTR As Double = 0.7
Const LMRTR As Double = 0.5
Const MaxOptn As Integer = 2
Const MagC As Integer = 20 ' Magazine capacity for loading
Dim CurMRTR As Double ' Current MRTR ratio
'parameters and variables
Dim Atj(tCount, jCount) As Integer ' 1 if tool t is already on machine j
Dim Bti(tCount, iCount) As Integer ' 1 if part i requires a tool type t
Dim Xij(iCount, jCount)  As Integer ' 1 if part i is assigned to machine j
Dim BETAij(iCount, jCount) As Double ' resident tool ratio of part i at machine j
Dim Ytj(tCount, jCount) As Integer ' 1 if tool type t is assigned to machine j
Dim PRi(iCount) As Double ' part mix ratio of part type i
Dim NSEQi(iCount) As Integer ' Number of tools required by part type i
Dim Mt(tCount) As Integer ' number of avaialble tools of type t

Private Sub Cmd_Load_Click()
   ' Loading heuristic to maximize the routing flexibility MRFD
   Dim OptnIdx As Integer
   Dim CurOptn As LOperation
   Dim CurStn As LStation
   Set Operations = New Collection
   Set OptnsAvailable = New Collection
   Set OptnsAssigned = New Collection
   Set ToolTypes = New Collection
   Set ToolsAvailable = New Collection
   Set ToolsLoaded = New Collection
   Set ToolsConsidered = New Collection
   Set Stations = New Collection
   Set StnsAvailable = New Collection
   Set StnsAssigned = New Collection
```

```
Set StnsPreselected = New Collection
Set StnsExtra = New Collection
Set StnsConsidered = New Collection
```

```
Call ReadOptnTool 'Read part tool matrix
Dim i, j, t As Integer
'step 0-a) generate {P0} a list of unassigned parts
'and {P1} a list of assigned part
Call MakeP0List

For i = 1 To iCount
   If i = 1 Then
      'step 0-b) Select a part(operation)
      OptnIdx = FindOptnLNT()
   Else
      'step 1-a) Select a part i from {P0} with largest #of common tools
      OptnIdx = FindOptnLNCT()
   End If
   Set CurOptn = New LOperation
   Set CurOptn = OptnsAvailable.Item(OptnIdx)
   Dim CurPartNo As Integer

   CurMRTR = MRTR
   CurPartNo = CurOptn.OptnNo

   Do While StnsAssigned.Count + StnsPreselected.Count = 0 And CurMRTR >= LMRTR
      'step 1-b) generate a list of unassigned machines {M0}
      ' and a list of assigned machines {M1}
      Call MakeM0List(CurPartNo)

      Dim NumAvailStn As Integer
      NumAvailStn = StnsAvailable.Count
      If NumAvailStn > 0 Then
      For j = 1 To NumAvailStn
         'step 2-b) select a machine from {M0} with largest number of tools
         ' required by part i already loaded on machine j.
         Set CurStn = New LStation
         Dim MachIdx As Integer
         MachIdx = FindMachidxSNOptn(CurOptn.OptnNo)
         'MachIdx = FindMachidxSNTOptnReq(CurOptn.OptnNo)
         Set CurStn = StnsAvailable.Item(MachIdx)
         Dim CurStnNo As Integer
         CurStnNo = CurStn.StationNo

         'generate a list of unassigned tools {T0}
         ' and a list of assigned tools {T1)
         Call MakeT0List(CurPartNo, CurStnNo)

         Dim tNum As Integer ' number of tools in {T0}
         Dim LoadCurStn As Integer ' flag to indicate if load current station
```

```
numReq = GetNumToolReq(CurPartNo)
NumLoadedOptn = GetNumToolLoadedOptn(CurStnNo, CurPartNo)

LoadCurStn = 0
Dim CurToolTypeNo As Integer
Dim CurTool As New LTool
tNum = ToolsAvailable.Count
```

```
If tNum > 0 Then
```

```
    For t = 1 To tNum
        Dim CurToolidx As Integer
        'step 3-b) select a tool from {T0} required most
        ' by parts in {P0}
        CurToolidx = FindToolidxLNOReq(CurStnNo)
        Set CurTool = New LTool
        Set CurTool = ToolsAvailable.Item(CurToolidx)
        CurToolTypeNo = CurTool.ToolTypeNo
        ToolsAvailable.Remove (CurToolidx)
        ToolsLoaded.Add CurTool
        ' Check the new RTR (resident tool ratio )
        ' step 3-c) if RTR >= MRTR
        If NumLoadedOptn + ToolsLoaded.Count >= CurMRTR * numReq Then
            LoadCurStn = 1
            Exit For
        End If

    Next t
Else
    If NumLoadedOptn + ToolsLoaded.Count >= CurMRTR * numReq Then
        LoadCurStn = 1
    End If

End If

'step 3-a) all tool considered
' If MRTR is maintained, commit all the loading
' i.e.  Xij = 1, Atj = 1 and Mt = Mt -1
If LoadCurStn = 1 Then
    StnsAssigned.Add CurStn
    Xij(CurPartNo, CurStnNo) = 1
    Dim NTLoaded, tL As Integer
    Dim ToolLoaded As LTool
    NTLoaded = ToolsLoaded.Count
    If NTLoaded > 0 Then
        For tL = 1 To NTLoaded
            Set ToolLoaded = New LTool
            Set ToolLoaded = ToolsLoaded.Item(tL)
            Atj(ToolLoaded.ToolTypeNo, CurStnNo) = 1
            Mt(ToolLoaded.ToolTypeNo) = Mt(ToolLoaded.ToolTypeNo) - 1
        Next tL
```

```
            Else
                StnsConsidered.Add CurStn
                'Xij(CurPartNo, CurStnNo) = 0
            End If

            StnsAvailable.Remove (MachIdx)

        Next j
        End If ' end if StnsAvailable.count > 0
        If StnsAssigned.Count = 0 Then
            CurMRTR = Max(0, CurMRTR - 0.1)
        End If
```

```
        'If part is still not assigned to any station, reduce the current MRTR ratio
```

```
        Loop
        OptnsAvailable.Remove (OptnIdx)

        'clear Station list M0, M1, M2
        Do While StnsAvailable.Count > 0 ' clear M0 list
            StnsAvailable.Remove (1)
        Loop
        Do While StnsAssigned.Count > 0  ' clear M1 list
            StnsAssigned.Remove (1)
        Loop
        Do While StnsConsidered.Count > 0  ' clear M2 list
            StnsConsidered.Remove (1)
        Loop
        Do While StnsPreselected.Count > 0  ' clear M3 list
            StnsPreselected.Remove (1)
        Loop


    Next i

    'When all tools are assigned, but there is still some tools unloaded
    Call AssignExtraTool
    ' Update the RTR ratio of every  part-station assignment
    Call UpdateRTR
    ' Write the results to excel worksheets
    Call UpdateToolLoad


End Sub

Private Sub AssignExtraTool()
    Dim t1, t1Num As Integer
    Dim t As Integer
    Dim j As Integer
    Dim tStn As LStation
```

```
    Dim sntStnidx As Integer
    For t = 1 To tCount
        Dim stnsExtras As New Collection
        'Clear M3 list
        Do While stnsExtras.Count > 0
            stnsExtras.Remove (1)
        Loop
        If Mt(t) > 0 Then
            'create M3 list of stations
            For j = 1 To jCount
                If Atj(t, j) = 0 Then
                    Set tStn = New LStation
                    tStn.StationNo = j
                    stnsExtras.Add tStn
                End If
            Next j
            If Mt(t) > stnsExtras.Count Then
```

```
                MsgBox "it is not possible"
```

```
            End If
            t1Num = Mt(t)
            If t1Num = 0 Then MsgBox "No place to load extra tools"
            For t1 = 1 To t1Num
                'Assign the tool to station with smallest #of tools loaded
                Mt(t) = Mt(t) - 1
                sntStnidx = FindMachSNTLoaded(stnsExtras, t)
                Set sntStn = New LStation
                Set sntStn = stnsExtras.Item(sntStnidx)
                If Atj(t, sntStn.StationNo) = 1 Then
                    MsgBox "Atj index wrong"
                End If
                Atj(t, sntStn.StationNo) = 1
                stnsExtras.Remove (sntStnidx)
            Next t1

        End If
    Next t

End Sub


Private Sub UpdateRTR()
    'Update the Resident tool ratio RTR of operation-machine assignment
    Dim i As Integer
    Dim j As Integer
    Dim myRTR As Double
    Dim numLoaded, numReq As Double
    For i = 1 To iCount
        For j = 1 To jCount
            numLoaded = GetNumToolLoadedOptn(j, i)
```

```
            BETAij(i, j) = Round(numLoaded / numReq, 3)
            If BETAij(i, j) >= LMRTR Then Xij(i, j) = 1
        Next j
    Next i
    For i = 1 To iCount ' If there is any part unassigned
        If GetNumStnLoaded(i) = 0 Then
            Xij(i, FindLBStnNo(i)) = 1
        End If
    Next i
    ' Write loading plan
    For i = 1 To iCount
        For j = 1 To jCount
            Sheet8.Cells((i - 1) * jCount + j + 1, 1) = i
            Sheet8.Cells((i - 1) * jCount + j + 1, 2) = j
            Sheet8.Cells((i - 1) * jCount + j + 1, 3) = Xij(i, j)
            Sheet8.Cells((i - 1) * jCount + j + 1, 4) = BETAij(i, j)
        Next j
    Next i

End Sub

Private Function FindLBStnNo(OptnNo As Integer) As Integer
```

```
    Dim LBeta As Double
```

```
    Dim j As Integer
    LBeta = 0
    For j = 1 To jCount
        If BETAij(OptnNo, j) > LBeta Then
            FindLBStnNo = j
            LBeta = BETAij(OptnNo, j)
        End If
    Next j

End Function

Private Function GetNumStnLoaded(OptnNo As Integer) As Integer
    Dim j As Integer
    GetNumStnLoaded = 0
    For j = 1 To jCount
        GetNumStnLoaded = GetNumStnLoaded + Xij(OptnNo, j)
    Next j

End Function


Private Sub UpdateToolLoad()

    Dim t, j As Integer
    Dim ToolID As Integer
    Dim Mt As Integer
```

```
    ToolID = 0
    For t = 1 To 50 ' Loop through all tool types
        For j = 1 To jCount
            Sheet9.Cells((t - 1) * jCount + j + 1, 1) = t
            Sheet9.Cells((t - 1) * jCount + j + 1, 2) = j
            Sheet9.Cells((t - 1) * jCount + j + 1, 3) = Atj(t, j)
            If Atj(t, j) = 1 Then
                ToolID = ToolID + 1
                'Populate the Tools table
                Sheet6.Cells(1 + ToolID, 3).Value = j
            End If
        Next j
    Next t


End Sub

Private Sub ReadOptnTool()
    'Read part-tool matrix Bti(tCount, iCount)
    Dim t, i As Integer
    For t = 1 To tCount
        Mt(t) = Sheet4.Range("ToolCopy").Cells(t, 1)
        For i = 1 To iCount
            Bti(t, i) = Sheet4.Range("PartTool").Cells(t, i)
        Next i
    Next t
    For i = 1 To iCount
```

```
        PRi(i) = Sheet1.Range("MixRatio").Cells(i, 1)  ' read part mix ratio
```

```
        NSEQi(i) = Sheet2.Range("NumOfSeq").Cells(i, 1)  ' read number of tools required by part i
    Next i

End Sub

Private Sub MakeP0List()
    Dim i As Integer
    Dim lOptn As LOperation
    For i = 1 To iCount
        Set lOptn = New LOperation
        lOptn.OptnNo = i
        lOptn.MixRatio = PRi(i) ' part mix ratio
        lOptn.NumOfSeq = NSEQi(i) ' number of tools required by part i
        Operations.Add lOptn
        OptnsAvailable.Add lOptn
    Next i
    ' M1 (OptnsAssigned) list is empty by default

End Sub

Private Function FindOptnLNT() As Integer
```

```
    Dim OptnNoLNT As Integer
    Dim LNumTool As Integer
    Dim MaxRatio As Double
    Dim Optn As LOperation
    Dim i, numOptn As Integer
    OptnNoLNT = 0
    LNumTool = 0
    MaxRatio = 0
    numOptn = OptnsAvailable.Count
    For i = 1 To numOptn
        Set Optn = New LOperation
        Set Optn = OptnsAvailable.Item(i)
        If Optn.MixRatio > MaxRatio Then
            OptnNoLNT = i
            LNumTool = Optn.NumOfSeq
            MaxRatio = Optn.MixRatio
        Else
            If Optn.MixRatio = MaxRatio And Optn.NumOfSeq > LNumTool Then
                OptnNoLNT = i
                LNumTool = Optn.NumOfSeq
                MaxRatio = Optn.MixRatio
            End If
        End If

    Next i
    FindOptnLNT = OptnNoLNT

End Function

Private Function FindOptnLNCT() As Integer
' Find the operation index in P0 with largest number of common tools
' with parts in list P1 or tools already loaded
```

```
    Dim OptnNoLNCT As Integer
```

```
    Dim LNumComTool As Integer
    Dim MaxRatio As Double
    Dim Optn As LOperation
    Dim i, j, t, numOptn As Integer
    OptnNoLNT = 0
    LNumComTool = 0
    MaxRatio = 0
    ' define the z(t) array =1 if tool t is already loaded on some machine
    Dim z(tCount) As Integer
    For t = 1 To tCount
        z(t) = 0
        For j = 1 To jCount
            If Atj(t, j) = 1 Then
                z(t) = 1
                Exit For
            End If
```

```
    Next t

    numOptn = OptnsAvailable.Count
    For i = 1 To numOptn
        Set Optn = New LOperation
        Set Optn = OptnsAvailable.Item(i)
        Dim NumComTool As Integer
        NumComTool = 0
        For t = 1 To tCount ' Calculate the common tools required by part i optnNo
            NumComTool = NumComTool + z(t) * Bti(t, Optn.OptnNo)
        Next t

        If Optn.MixRatio > MaxRatio Then
            OptnNoLNCT = i
            LNumComTool = NumComTool
            MaxRatio = Optn.MixRatio
        Else
            If Optn.MixRatio = MaxRatio And NumComTool > LNumComTool Then
                OptnNoLNT = i
                LNumTool = NumComTool
                MaxRatio = Optn.MixRatio
            End If
        End If
    Next i
    FindOptnLNCT = OptnNoLNCT

End Function

Private Function FindMachLNTLoaded() As Integer
    'find the machine index with largest number of tools in M0
    Dim t, j As Integer
    Dim Stnidx As Integer
    Dim LNT, NT As Integer
    Dim Stn As New LStation
    LNT = 0
    Stnidx = 0
    For j = 1 To StnsAvailable.Count
```

```
        Set Stn = New LStation
```

```
        Set Stn = StnsAvailable.Item(j)
        NT = 0
        For t = 1 To tCount
            NT = NT + Atj(t, Stn.StationNo)
        Next t
        If NT > LNT Then
            Stnidx = j
            LNT = NT
        End If
    Next j
    FindMachLNTLoaded = Stnidx
```

```
Private Function FindMachSNTLoaded(extra As Collection, tNo As Integer) As Integer
    'find the machine index with smallest number of tools in M0
    Dim t As Integer
    Dim j As Integer
    Dim i As Integer
    Dim Stnidx As Integer
    Dim SNT, NT As Integer
    Dim numStn As Integer
    Dim tStn As New LStation
    Dim LowBeta As Double
    Dim LBStnidx As Integer
    Dim numLoaded, numReq As Double
    numStn = extra.Count
    If numStn = 0 Then MsgBox "M3 list is empty"
    'If Xij = 1 and Bti =1 with lowest BETAij

    SNT = jCount * MagC ' set small least #of tool to a large number
    LowBeta = 1
    LBStnidx = 0
    Stnidx = 0
    For j = 1 To numStn
        Set tStn = New LStation
        Set tStn = extra.Item(j)

        ' check to see if any part i assigned to stn j
        ' requires tool type tNo and with Lowest Betaij
        For i = 1 To iCount
            If Xij(i, j) = 1 And Bti(tNo, i) Then
                numLoaded = GetNumToolLoadedOptn(j, i)
                numReq = GetNumToolReq(i)
                BETAij(i, j) = Round(numLoaded / numReq, 3)
                If BETAij(i, j) < LowBeta Then
                    LBStnidx = j
                    LowBeta = BETAij(i, j)
                End If
            End If
        Next i

        NT = 0 ' Count # of tools loaded on tStn.Station
        For t = 1 To tCount
            NT = NT + Atj(t, tStn.StationNo)
```

```
        Next t
```

```
        If NT < SNT Then
            Stnidx = j
            SNT = NT
        End If
    Next j
    If LBStnidx > 0 Then
```

```
    Else
        FindMachSNTLoaded = Stnidx
    End If

End Function


Private Sub MakeM0List(OptnNo As Integer)
    ' create a list of unassinged machines and assigned machines.

    Dim j As Integer
    Dim lStn As LStation
    Do While StnsAvailable.Count > 0 ' clear M0 list
        StnsAvailable.Remove (1)
    Loop
    Do While StnsAssigned.Count > 0  ' clear M1 list
        StnsAssigned.Remove (1)
    Loop
    Do While StnsConsidered.Count > 0  ' clear M2 list
        StnsConsidered.Remove (1)
    Loop

    Do While StnsPreselected.Count > 0  ' clear M3 list
        StnsPreselected.Remove (1)
    Loop

    For j = 1 To jCount
        Set lStn = New LStation
        lStn.StationNo = j
        Dim numReq, numLoaded, NumLoadedOptn, NumNeed As Integer
        numReq = GetNumToolReq(OptnNo)
        numLoaded = GetNumToolLoaded(j)
        NumLoadedOptn = GetNumToolLoadedOptn(j, OptnNo)
        NumNeed = GetNumToolNeedOptn(j, OptnNo) ' #of candidate tools
        'Step 2-c) check if machine j already has the required
        ' tools for part i with minimum resident tool ratio MRTR

        'If NumLoadedOptn >= MRTR * numReq Then
        If NumLoadedOptn >= CurMRTR * numReq Then
            ' if this station already have enough tools
            StnsPreselected.Add lStn
            'StnsConsidered.Add lStn
            'StnsAssigned.Add lStn
            Xij(OptnNo, lStn.StationNo) = 1
        Else
            'step 2-d) check if machine j has enough tool slots
            ' to maintain the MRTR for part i


            'step 2-e) check if enough tools available to keep MRTR


            ' if part i assigned to machine j
```

```
            Or Round(CurMRTR * numReq - NumLoadedOptn, 0) > NumNeed _
            Or FindNumOptnAtMach(j) >= MaxOptn Then
            StnsConsidered.Add lStn
            'Xij(OptnNo, lStn.StationNo) = 0
         Else
            StnsAvailable.Add lStn
         End If
      End If
   Next j

End Sub

Private Sub MakeT0List(OptnNo As Integer, StnNo As Integer)
   'create a list of unassigned tools for part i at machine j
   Dim t As Integer
   Dim LTool As LTool
   Do While ToolsAvailable.Count > 0 ' clear T0 list
      ToolsAvailable.Remove (1)
   Loop
   Do While ToolsLoaded.Count > 0  ' clear T1 list
      ToolsLoaded.Remove (1)
   Loop
   Do While ToolsConsidered.Count > 0   ' clear T2 list
      ToolsConsidered.Remove (1)
   Loop

   For t = 1 To tCount
      Set LTool = New LTool
      If Bti(t, OptnNo) = 1 Then
         If Atj(t, StnNo) = 1 Then
            LTool.ToolTypeNo = t
            ToolsConsidered.Add LTool
         Else
            If Mt(t) > 0 Then
               LTool.ToolTypeNo = t
               ToolsAvailable.Add LTool
            End If

         End If

      End If

   Next t

End Sub


Private Function GetNumToolLoaded(StnNo As Integer) As Integer
   ' find the number of tools already loaded on machine j (Stnno)
   Dim t As Integer
   Dim numTool As Integer
   numTool = 0
```

```
    For t = 1 To tCount
        numTool = numTool + Atj(t, StnNo)
    Next t
    GetNumToolLoaded = numTool
End Function

Private Function GetNumToolLoadedOptn(StnNo As Integer, OptnNo As Integer) As Integer
    ' find the number of tools already loaded for optn i on machine j (Stnno)
    Dim t As Integer
    Dim numTool As Integer
    numTool = 0
    For t = 1 To tCount
        numTool = numTool + Atj(t, StnNo) * Bti(t, OptnNo)
    Next t
    GetNumToolLoadedOptn = numTool
End Function

Private Function GetNumToolNeedOptn(StnNo As Integer, OptnNo As Integer) As Integer
    ' find the number of candidate tools that could be loaded for optn i on machine j (Stnno)
    Dim t As Integer
    Dim numTool As Integer
    numTool = 0
    For t = 1 To tCount
        numTool = numTool + (1 - Atj(t, StnNo)) * Bti(t, OptnNo) * IsToolAvailable(t)
    Next t
    GetNumToolNeedOptn = numTool
End Function

Private Function IsToolAvailable(t As Integer) As Integer
    Dim IsTool As Integer
    IsTool = 0
    If Mt(t) > 0 Then IsTool = 1
    IsToolAvailable = IsTool
End Function


Private Function GetNumToolReq(OptnNo As Integer) As Integer
    ' find the number of tools already loaded for optn i on machine j (Stnno)
    Dim t As Integer
    Dim numTool As Integer
    numTool = 0
    For t = 1 To tCount
        numTool = numTool + Bti(t, OptnNo)
    Next t
    GetNumToolReq = numTool
End Function

Private Function FindMachidxLNTOptnReq(OptnNo As Integer) As Integer
    ' Find the machine j index (stnno) from M0 with the largest number of tools
    ' required by part type i (OptnNo)
    Dim j As Integer
    Dim MjLNTOptnReq, LNTOptnReq, LNTLoaded As Integer
    Dim NTOptnReq, NTLoaded As Integer
```

```
    If StnsAvailable.Count = 0 Then MsgBox "M0 list is empty!"

    Dim lStn As LStation

    MjLNTOptnReq = 1 ' Let j = 1 first,


    LNTOptnReq = 0
    LNTLoaded = 0
    NTOptnReq = 0
    NTLoaded = 0
    For j = 1 To StnsAvailable.Count
        Set lStn = New LStation
        Set lStn = StnsAvailable.Item(j)
        NTOptnReq = GetNumToolLoadedOptn(lStn.StationNo, OptnNo)
        NTLoaded = GetNumToolLoaded(lStn.StationNo)
        If NTOptnReq > LNTOptnReq Then
            LNTOptnReq = NTOptnReq
            LNTLoaded = NTLoaded
            MjLNTOptnReq = j
        Else
            If NTOptnReq = LNTOptnReq And NTLoaded > LNTLoaded Then
                LNTOptnReq = NTOptnReq
                LNTLoaded = NTLoaded
                MjLNTOptnReq = j
            End If
        End If

    Next j

    FindMachidxLNTOptnReq = MjLNTOptnReq
End Function


Private Function FindMachidxSNTOptnReq(OptnNo As Integer) As Integer
    ' Find the machine j index (stnno) from M0 with the smallest number of tools
    ' required by part type i (OptnNo)
    Dim j As Integer
    Dim MjLNTOptnReq, LNTOptnReq, LNTLoaded As Integer
    Dim NTOptnReq, NTLoaded As Integer
    Dim lStn As LStation
    If StnsAvailable.Count = 0 Then MsgBox "M0 list is empty!"
    MjLNTOptnReq = 1 ' Let j = 1 first,
    LNTOptnReq = jCount * tCount
    LNTLoaded = jCount * tCount
    NTOptnReq = 0
    NTLoaded = 0
    For j = 1 To StnsAvailable.Count
        Set lStn = New LStation
        Set lStn = StnsAvailable.Item(j)
        NTOptnReq = GetNumToolLoadedOptn(lStn.StationNo, OptnNo)
        NTLoaded = GetNumToolLoaded(lStn.StationNo)
        If NTOptnReq < LNTOptnReq Then
            LNTOptnReq = NTOptnReq
```

```
            MjLNTOptnReq = j
        Else
            If NTOptnReq = LNTOptnReq And NTLoaded < LNTLoaded Then
                LNTOptnReq = NTOptnReq
                LNTLoaded = NTLoaded
                MjLNTOptnReq = j
```

```
            End If
```

```
        End If

    Next j

    FindMachidxSNTOptnReq = MjLNTOptnReq
End Function

Private Function FindNumOptnAtMach(StnNo As Integer) As Integer
    Dim i As Integer
    FindNumOptnAtMach = 0
    For i = 1 To iCount
        FindNumOptnAtMach = FindNumOptnAtMach + Xij(i, StnNo)
    Next i
End Function



Private Function FindMachidxSNOptn(OptnNo As Integer) As Integer
    ' Find the machine j index (stnno) from M0 with smallest # of optn assigned
    ' break tie with the largest number of tools required by part type i (OptnNo)
    Dim j As Integer
    Dim MjLNTOptnReq, LNTOptnReq, SNOLoaded As Integer
    Dim NTOptnReq, NOLoaded As Integer
    Dim lStn As LStation
    If StnsAvailable.Count = 0 Then MsgBox "M0 list is empty!"
    MjLNTOptnReq = 1 ' Let j = 1 first,
    LNTOptnReq = 0
    SNOLoaded = iCount
    NTOptnReq = 0
    NOLoaded = iCount
    For j = 1 To StnsAvailable.Count
        Set lStn = New LStation
        Set lStn = StnsAvailable.Item(j)
        NTOptnReq = GetNumToolLoadedOptn(lStn.StationNo, OptnNo)
        NOLoaded = FindNumOptnAtMach(lStn.StationNo)
        If NOLoaded < SNOLoaded Then
            LNTOptnReq = NTOptnReq
            SNOLoaded = NOLoaded
            MjLNTOptnReq = j
        Else
            If NTOptnReq > LNTOptnReq And NOLoaded = SNOLoaded Then
                LNTOptnReq = NTOptnReq
```

```
            MjLNTOptnReq = j
        End If
    End If

    Next j
    FindMachidxSNOptn = MjLNTOptnReq
End Function
```

```
Private Function FindToolidxLNOReq(StnNo As Integer) As Integer
```

```
    'Find the tool that requested most by operations in P0 list
    ' break tie by selecting the one requested most by optns in P1 list
    Dim Toolidx As Integer
    Dim aTool As LTool
    Dim t, tNum As Integer
    Dim NTP0OReq, NTP1OReq, LNTP0OReq, LNTP1OReq As Integer
    NTP0OReq = 0
    NTP1OReq = 0
    LNTP0OReq = 0
    LNTP1OReq = 0
    If ToolsAvailable.Count = 0 Then MsgBox "T0 list is empty!"
    tNum = ToolsAvailable.Count
    Toolidx = 1 ' Let t = 1 first every time
    For t = 1 To tNum
        Set aTool = New LTool
        Set aTool = ToolsAvailable.Item(t)
        NTP0OReq = GetNumP0OptnReq(aTool.ToolTypeNo)
        NTP1OReq = GetNumP1OptnReq(aTool.ToolTypeNo, StnNo)
        If NTP0OReq > LNTP0OReq Then
            Toolidx = t
            LNTP0OReq = NTP0OReq
            LNTP1OReq = NTP1OReq

        Else
            If NTP0OReq = LNTP0OReq And NTP1OReq > LNTP1OReq Then
                Toolidx = t
                LNTP0OReq = NTP0OReq
                LNTP1OReq = NTP1OReq

            End If

        End If

    Next t

    FindToolidxLNOReq = Toolidx
End Function
```

```
Private Function GetNumP0OptnReq(tNo As Integer) As Integer
    ' find the number operations in P0 list that require tool type tNo
    Dim NumP0OptnReq As Integer
    Dim numPart As Integer
    Dim i As Integer
    Dim lOptn As LOperation
    NumP0OptnReq = 0
    numPart = OptnsAvailable.Count
    If numPart = 0 Then MsgBox "P0 list is empty!"
    For i = 1 To numPart
        Set lOptn = New LOperation
        Set lOptn = OptnsAvailable.Item(i)
        If Bti(tNo, lOptn.OptnNo) = 1 Then NumP0OptnReq = NumP0OptnReq + 1

    Next i


    GetNumP0OptnReq = NumP0OptnReq


End Function

Private Function GetNumP1OptnReq(tNo As Integer, jNo As Integer) As Integer
    ' find the number operations in P1 list that require tool type tNo
    Dim NumP1OptnReq As Integer
    Dim numPart As Integer
    Dim i As Integer
    Dim lOptn As LOperation
    NumP1OptnReq = 0
    numPart = OptnsAssigned.Count
    'If numPart = 0 Then MsgBox "P1 list is empty!"
    For i = 1 To numPart
        Set lOptn = New LOperation
        Set lOptn = OptnsAssigned.Item(i)
        If Bti(tNo, lOptn.OptnNo) = 1 Then NumP1OptnReq = NumP1OptnReq + 1

    Next i
    GetNumP1OptnReq = NumP1OptnReq
End Function


Private Function Max(Xa As Double, Xb As Double) As Double
    If Xa > Xb Then
        Max = Xa
    Else
        Max = Xb
    End If
End Function

Private Function Min(Xa As Double, Xb As Double) As Double
    If Xa > Xb Then
        Min = Xb
```

```
        Min = Xa
    End If
End Function
```

VITA

Jie Chen received his Master of Science degree in Industrial Engineering from the University of Toledo. He worked as an Industrial Engineer for three years at ABB Flexible Automation in China before he started his graduate studies. He began his PhD studies at Virginia Polytechnic Institute and State University (Virginia Tech) in 1999. He worked as an Operations Research Intern at Enterprise Systems Lab in General Motors' R&D center in summer 2000. He received Burrows Fellowship award from the Grado Department of Industrial and Systems Engineering of Virginia Tech in 2001-2002. He also taught an undergraduate course in summer 2001 at Virginia Tech. He published his research papers in the International Journal of Production Research and International Journal of Advance Manufacturing Technology.