DEVELOPMENT OF A BIDDING ALGORITHM USED IN AN AGENT-BASED SHOP-FLOOR CONTROL SYSTEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUHTAR URAL ULUER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JANUARY 2007

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Kemal İDER
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Prof. Dr. Sadık Engin KILIÇ
Supervisor

Examining Committee Members

Prof. Dr. Y. Samim ÜNLÜSOY	(METU,ME)	_____
Prof. Dr. S. Engin KILIÇ	(METU,ME)	_____
Prof. Dr. Mustafa İlhan GÖKLER	(METU,ME)	_____
Prof. Dr. Ömer ANLAĞAN	(TÜBİTAK)	_____
Assoc. Prof. Dr. Tayyar D. ŞEN	(METU,IE)	_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Muhtar Ural ULUER

ABSTRACT

DEVELOPMENT OF A BIDDING ALGORITHM USED IN AN AGENT-BASED SHOP-FLOOR CONTROL SYSTEM

Uluer, Muhtar Ural

M. Sc., Department of Mechanical Engineering

Supervisor: Prof. Dr. S. Engin KILIÇ

January 2007, 144 pages

In this study a time based bidding framework is developed which is used for dispatching jobs to manufacturing resources in a virtual shop-floor environment. Agent-based shop-floor control approach is implemented with machine and part agents. The Contract-net communication protocol is utilized as the negotiation scheme between these agents. Single step product reservation (SSPR) technique is adopted throughout the study. Primary objective is determined as meeting the due dates and if the lateness is inevitable, avoiding the parts of high priority from being late. A balanced machine utilization rate is set as the secondary objective.

During bid construction step, the SSPR technique is augmented with W(SPT+CR) sequencing rule in order to obtain weighted tardiness results. Bids containing Earliest Finishing Time (EFT) and machine loading values of the corresponding machine are evaluated with considering the priority of the part. An elimination algorithm which discards the highly deviated bids having obvious differences is implemented at the initial stage of the bid evaluation step. A basic algorithm to control the maximum tardiness value is applied, as well.

A simulation test bed is developed in order to implement the time concept into the presented bidding framework. The test bed is mainly based on the Computer Integrated Manufacturing Laboratory (CIMLAB) located in Middle East Technical University, Department of Mechanical Engineering.

The developed bidding algorithm is tested under several cases. Results revealed that the proposed bidding framework was quite successful in meeting the objectives. The study is concluded with some specific future work, outlined in the light of the results obtained.

Keywords: Bidding, Auction-Based Distributed Scheduling, Simulation of Flexible Manufacturing Systems, Agent-Based Systems, Shop-Floor Control

ÖZ

AJAN TEMELLİ ATÖLYE DENETİM SİSTEMİNDE KULLANILACAK BİR TEKLİF ALGORİTMASI GELİŞTİRİLMESİ

Uluer, Muhtar Ural
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. S. Engin KILIÇ

Ocak 2007, 144 sayfa

Bu çalışmada, sanal atöyle ortamında üretim tezgahlarına iş gönderecek zamana dayalı bir teklif yapısı oluşturulmuştur. Makina ve parça ajanları kullanılarak ajan temelli atölye denetim yaklaşımı uygulanmış ve ajanların görüşmelerinde Contract-net haberleşme protokolü kullanılmıştır. Tüm çalışma boyunca tek aşamalı parça rezervasyonu tekniğinden faydalanılmıştır. Birincil hedef belirlenen bitiş tarihlerine uyulması eğer mümkün olmuyorsa, yüksek önem taşıyan parçaların geç kalmaması olarak belirlenmiştir. Atölyede dengeli bir yük dağalımı ise ikinci derece hedef olarak koyulmuştur.

Teklif oluşturma aşamasında ağırlıklı geç kalma değerleri elde etmek için tek aşamalı parça rezervasyon tekniği, W(SPT+CR) sıralama kuralı ile bütünleşik olarak kullanılmıştır. En erken bitirme zamanlarını ve karşılık gelen makinanın yük değerini içeren teklifler, parçanın önceliği de göz önünde bulundurularak değerlendirilmiştir. Teklif değerlendirme aşamasının ilk basamağında yüksek sapma değerine sahip olan teklifleri çıkaran bir eleme algoritması uygulanmıştır. En çok geç kalma değerini denetleyen basit bir algortma da eklenmiştir.

Önerilen teklif yapısına zaman kavramını tanıtabilmek için bir simülasyon ortamı geliştirilmiştir. Bu ortam, Orta Doğu Teknik Üniversitesi, Makina Mühendisliği Bölümünde bulunan Bilgisayar Tümleşik Üretim laboratuarındaki sistem temel alınarak yapılmıştır.

Geliştirilen teklif algoritması değişik durumlar için test edilmiş ve elde edilen sonuçlar belirlenen hedeflere başarılı bir şekilde ulaşıldığını göstermiştir. Çalışma, elde edilen sonuçların ışığında, ileride yapılabilecek diğer araştırmalar belirtilerek sonlandırılmıştır.

Anahtar Kelimeler: Teklif, Artırma Temelli Dağatık Çizelgeleme, Esnek İmalat Sistemlerinin Simülasyonu, Ajan Temelli Sistemler, Atölye Denetimi

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Dr. S. Engin Kılıç, for his continuous guidance, encouragement and both academic and personal support throughout my thesis study.

I would like to express my appreciation to my colleagues, Yusuf Başıbüyük, Dr. Burak Sarı, Deniz Yücel and Burak Ay in Integrated Manufacturing Technologies Research Group (IMTRG) for numerous fruitful discussions and wise comments during the development of my thesis.

I also thank my friends, especially my colleague Murat Kandaz for making the thesis period enjoyable and colorful and Cihan Selçuk for his invaluable support during the programming stage.

Finally, for and beyond this thesis study, I am indebted to my parents for their never ending love, belief and patience throughout my academic life. Without their encouragement, completion of this thesis would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Manufacturing industry has been evolving to meet the changing nature of customer demands. Demand versatility has lead to product variations which cannot be dealt with the fixed transfer lines and fixed automation of mass production systems and low production rate of stand alone NC machines of job-shop production. As a result the new concept of Flexible Manufacturing Systems (FMS) having a moderate level of production volume and product variety is introduced.

Flexible manufacturing systems have different numerically controlled machines and resources linked together through a communication network. As the number of parts and machines increase, the control of the information using the communication network starts to become a problem. The control architectures bring solutions to those problems by defining the interactions between the manufacturing components and identifying the decision making responsibilities of each system component.

Traditional centralized control relies on a single control unit and hierarchical control architectures have increased number of intermediate level communication links allowing only top-down or bottom up information flow. In addition, using these kinds of control architectures makes it difficult to modify or extend the system which is conflicting with the flexible nature of the flexible manufacturing systems. As a result, number of flexible manufacturing cells implementing heterarchical control has proliferated in recent years. In heterarchical control architecture the decision making responsibilities are distributed to each component of the system so that each component has sufficient knowledge to accomplish its own task autonomously.

Agent-based technology can be implemented into the heterarchical control architecture because the physical system components can be easily represented by agent structure. Agents can be defined as autonomous software objects having the capability to respond the changes in the environment and communicate with other agents to achieve their goal.

Agent-based technology has been widely recognized as a promising paradigm for developing software applications able to support complex tasks. However there can be several tasks that a single agent is unable to finish alone. Such tasks require the cooperation of a group of agents. Communication is a means of establishing such cooperation between those autonomous agents. A popular scheme to achieve cooperation among autonomous agents is through the negotiation-based contract-net protocol (Smith, 1980). The contract-net protocol provides the advantage of real-time information exchange, making it suitable for shop-floor control and scheduling.

A shop-floor scheduling problem can be investigated under two sub-categories: job routing and job sequencing. The job is assigned and dispatched to the machines. Allocation of jobs to machine centers is referred as job routing. The sequence of the incoming jobs to the machines is determined through job sequencing. These two activities constitute the scheduling of the part. There are number of scheduling objectives under two main groups: customer oriented objectives and shop-floor efficiency based objectives. Meeting due dates and minimizing the average flow time are the most common customer oriented objectives and minimizing work in process (WIP) inventory and maximizing the machine utilization are the shop-floor efficiency based objectives. However, it is not possible to optimize all objectives simultaneously since some of these objectives conflict. The common practice for multi-objective studies is to select a point on the trade-off curve of the conflicting objectives.

In the bidding based scheduling problems, parts arriving in the system find suitable machines to themselves by simply negotiating with all available machines. As the name bidding implies each machine compete for the job and try to give the best offer in an auction mechanism. In this mechanism, parts receive bids which include either the time to start or finish the part or the total cost to manufacture the part. After evaluating all of the received bids, part awards itself to the most suitable machine. The cooperation between machines and parts are regulated by a negotiation scheme which dictates the steps in reaching a compromise.

1.1 Motivation and Scope

In this study a time based bidding framework is proposed to dispatch the incoming jobs to the machines on the shop-floor according to a specific decision making procedure. The shop-floor is modeled by utilizing agent based approach and the Contract-net protocol is implemented as the negotiation scheme between agents.

Two types of agents are developed for the system: Part agents and machine agents. Attributes for each part such as process plans, arrival time, processing time, due date, priority and events such as bid requesting, bid evaluation and awarding are embedded in the part agent structure. The machine agent structure contains attributes such as waiting queue and the history queue status, machine loading condition and machine type as well as the events like sequencing the coming part in the machine reservation queue and bid construction upon request.

Each part arriving at the shop-floor requires at least one and at most two operations. Although the process plans of the incoming parts are fixed, there exist alternative machines for an operation. The availability of multiple machines capable of processing a specific operation results in the routing flexibility problem. The problem is dealt with proposed bidding framework which plays a role during the selection of the best route among the alternative machines.

Single step product reservation technique is utilized for the scheduling mechanism. In this technique each machine has its own reservation list and each job is allocated on a machine when the work order arrives at the manufacturing system. Single step refers to the reservation scenario where the part does not complete allocation of all operations to the machines upon arrival, but does its reservations one operation at a time. The incoming parts and the parts which complete their first operations are held in the AGV and conveyor respectively. Product reservation technique allows the negotiation procedure to continue at the same time with the processing of the parts which are reserved beforehand. This technique eliminates the time loss due to negotiation messages during bidding before the system actually starts processing the part. It is mostly effective when the reservation list populates and the rate of the new coming parts to the system increases.

Research in this area generally aims at optimizing a single system objective such as makespan, average flow time or number of tardy parts and neglects the combined objectives and job priorities. This study integrates the part and machine objectives considering the part priorities as well. The primary objective taken into account when developing the bidding framework is to meet the due dates of the incoming parts and if lateness is unavoidable, parts of low priority should be late instead of the ones having high priorities. Secondary objective is to obtain a balanced machine loading rates throughout the shop-floor. While trying to achieve the first and the secondary objectives, keeping the maximum tardiness (positive lateness) value under control is set to be the last objective.

In the developed framework, during the bid construction step, W(SPT+CR) sequencing rule is augmented with the single step product reservation technique. The aim of using W(SPT+CR) rule is obtaining weighted tardiness results which will fulfill the primary objective considering the part priorities and the due dates. Constructed bids involve the Earliest Finishing Time (EFT) of the corresponding machine for the part that has requested a bid. The bids reaching to the part agents are evaluated based on the EFT values, loading factors of the corresponding

machines and the part priorities. An elimination algorithm plays a role at the initial stage of the bid evaluation step. This algorithm basically discards the highly deviated bids having obvious differences from the rest of the received bids. The secondary objective, work load balancing of the machines, is realized during bid evaluation step.

A virtual simulation test-bed is developed in order to implement the time concept into the developed bidding framework. The simulation structure simply checks the status of the system and all the parts in the system at every time instant. The performance of the bidding framework of flexible manufacturing system is tested under different conditions by running the simulation with different parameters. The results are interpreted by the help of the statistic collectors which are built in the simulation structure.

1.2 Outline

Chapter 2 consists of a literature survey on different type of control architectures, agent-based systems which are utilized in shop-floor control and the common negotiation scheme: Contract-net protocol. Finally various bidding based scheduling and job routing approaches related to the study are presented.

In Chapter 3 the modeled flexible manufacturing cell and its constant and variable physical attributes are described. The modeled bidding framework is defined in detail with the bid construction and bid evaluation mechanisms.

Chapter 4 is dedicated to the structure of the system. Simulation efforts are presented with the parameters utilized. The structure of the part and machine agents which take part in the contract-net protocol are explained with their attributes and events.

Chapter 5 involves the results of the various case studies on the developed system. The function of each algorithm is verified and cross comparisons are made with the results of previously developed bidding schemes.

Finally, concluding remarks and recommendations for possible future work which can be based on the developed system are presented in Chapter 6.

CHAPTER 2

LITERATURE SURVEY

2.1 Control Architectures

Flexible manufacturing cells have different automated components linked together through a communication network. As the capacity of the cell increases, the control of the information using the communication network starts to become a problem. Control architectures bring solutions to those problems by defining the interactions between the manufacturing components and identifying the decision making responsibilities of each system component.

Dilts et al. (1991) classified the control architectures into four main groups: Centralized Form, Proper Hierarchical Form, Modified Hierarchical Form and Heterarchical Form. Figure 2.1 shows the four different categories. Control components are represented by the boxes and the manufacturing resources are represented by the circles. The connecting lines show the control interrelationships.

2.1.1 Centralized Form

The earliest control architecture is the centralized form (Figure 2.1.a). It is characterized by a large computer performing all the information processing and maintaining global databases to record all the activities of the system. The simple machine controllers are distributed in the manufacturing environment and they execute the commands that are coming from the centralized control unit.

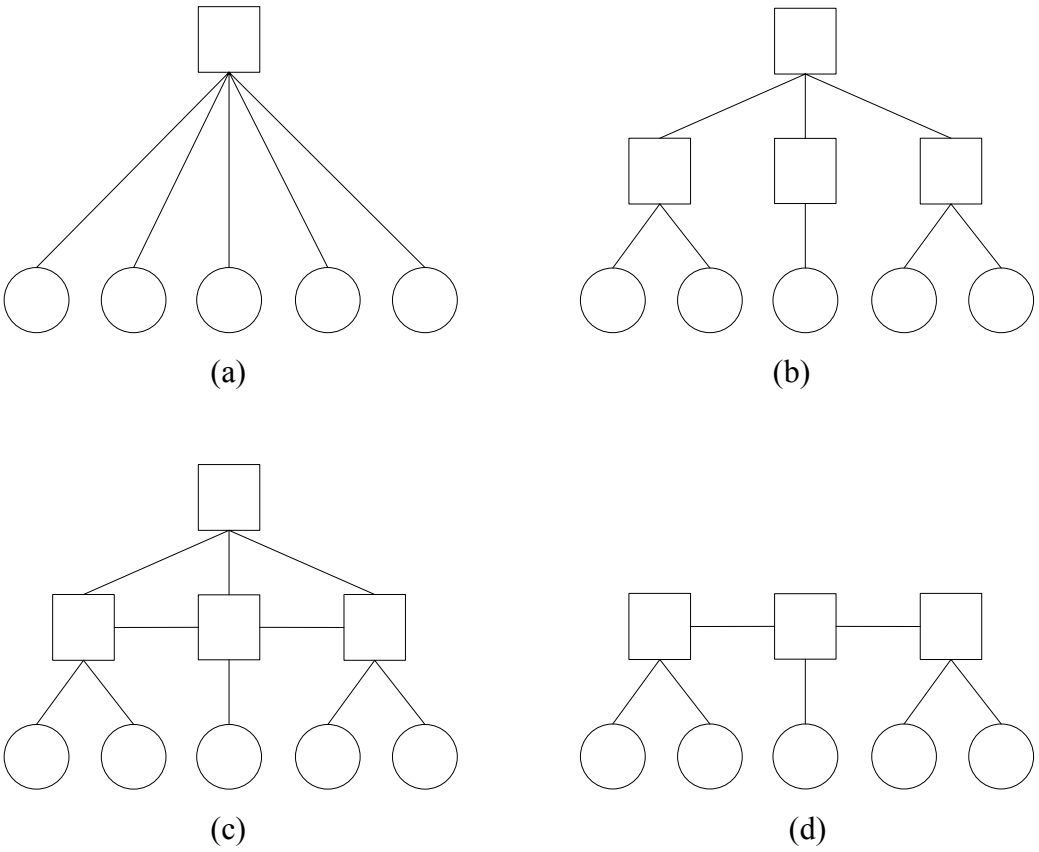

Figure 2.1 The Evolution of Control Architectures by Dilts et al. (1991)
(a) Centralized Form, (b)Proper Hierarchical Form, (c)Modified Hierarchical Form
(d) Heterarchical Form

One of the advantages of the centralized control architecture is that it simplifies global optimization since it holds all global information in a single control unit which receives monitoring information from shop-floor and machine controllers to use in making its global control decisions. Another advantage of a centralized control architecture is the overall system status that can be obtained by accessing the central control unit.

The centralized form has also some disadvantages. As the manufacturing system gets larger and becomes more complicated, the speed of response decreases because of the limited capacity of the central control unit. Also the response may become inconsistent due to the numerous tasks that are carried out by the control unit.

Another disadvantage of the centralized form is its dependence on the central control unit. This deteriorates the fault-tolerance of the entire system since the survival of the system is directly dependent on the central control unit. Finally, the flexibility of the central form is low because it is difficult to modify the control software.

According to Dilts et al. (1991) the centralized control architecture is not commonly used in the entire manufacturing facility, however, it is common to find this type of architecture applied to the control of a single manufacturing cell.

2.1.2 Proper Hierarchical Form

The research trying to eliminate the deficiencies of the centralized form resulted in the development of the proper hierarchical form (Figure 2.1.b). This form is also referred as "hierarchical form" by Crowe and Stahlman (1995) and Duffie et al. (1986, 1987, 1988, 1994). The idea behind the proper hierarchical form is distributing the load on the central computer by introducing the philosophy of "levels" (Duffie et al. 1988). Proper hierarchical systems contain a number of control modules arranged in a pyramidal structure. Rigid master/slave relationships are created between levels. Each component in the hierarchy is only able to communicate with the components that are one layer above or below. Command information flows top-down, and feedback information flows bottom-up through the hierarchy. Commands input at the highest level are decomposed into more detailed commands and passed on to the next lower level in the hierarchy. The upper-level layers have more authority and responsibility for decision-making than the lower-level layers. Modules at each level make decisions based on commands received from the level above, and feedback received from the level below.

When the control of an entire manufacturing facility is considered, the proper hierarchical form has been widely implemented by industries. One popular model is

the National Institute of Standard and Technology, Advanced Manufacturing Research Facility (NIST/AMRF). It is composed of five level control hierarchy in which the controller at each level communicates only with those above and below it but not with its peers. These levels of hierarchy are: facility, shop, cell, workstation and equipment. *Facility level* performs the planning, production management, information management, and other business functions. *Shop level* is responsible for coordination of resources and jobs on the shop-floor. The *cell level* control is concerned with the planning and sequencing of jobs in the cell. The cell also manages the resources such as tooling and part programs in the cell, and controls the material handling between, and processing on, the work stations. The *workstation level* typically consists of a machine tool serviced by a robot, a material storage buffer, and a control computer. The job of the computer is to sequence the processing tasks. Finally, equipment level is responsible for monitoring the execution of the production tasks (Jones and McLean 1986).

There are several advantages associated with the proper hierarchical form as stated by Dilts et al.(1991). Since the control system is not concentrated in a single central unit, according to the control requirements vertical control units may be added. Some level of fault-tolerance can be introduced by having other computers in the hierarchy taking over the tasks of the failed computers. Besides, achieving global optimization may be still possible. Finally, the rigid structure of purely hierarchical systems and the tight master/slave coupling between modules usually result in fast response times.

The proper hierarchical form has some disadvantages as well. Duffie et al. (1988) states some disadvantages as:

> The organization and structure of these systems become fixed in the early stages of their design. Extensions must be foreseen in advance, making subsequent unforeseen modifications difficult. A module at a given level in the hierarchy requires substantial knowledge of the module above it in the hierarchy as well as the modules below it, particularly when fault tolerance must be

incorporated. This tends to make large hierarchical systems difficult and expensive to design, maintain, and modify. Experience has shown that fault tolerance is obtained in hierarchical systems with considerable expense and complexity.

According to Dilts et al (1991), the inefficient information exchange between the local computers and the higher level controllers results in poorly responding to the occurrence of real time events.

2.1.3 Modified Hierarchical Form

The third category of the control architectures is the modified hierarchical form (Figure 2.1.c). This form is also referred as the quasi-heterarchical by Crowe and Stahlman (1995) and hybrid hierarchical/heterarchical by Ou-Yang and Lin (1998). It shares many of the characteristics of the proper hierarchical form such as the concept of levels of control with established supervisor/subordinate relationships. However the modified form allows significant autonomy to the subordinates, and also peer to peer communication between the entities. The autonomy given to the subordinates results in the looseness of the master/slave relationships between the levels of hierarchy. This causes the subordinates act as an intelligent assistant to the supervisor and not as a slave. Advances in LAN technology and the availability of inexpensive computing power have made modified hierarchical form possible. (Dilts et al. 1991)

Modified hierarchical control architecture has all the advantages that the proper hierarchical form bears. Because of the subordinate autonomy, the loading level of the supervisor decreases. This results in a decreased response time of the supervisor to the subordinate requests. Subordinate autonomy also increases the fault-tolerance of the system, since the subordinates can operate independently for a certain time during failure at higher levels.

Other than those of the proper form, the modified hierarchy architecture has some characteristic disadvantages of its own. The peer-to-peer communication between the subordinates may cause connectivity problems. This, in turn, complicates the control of the system, resulting in a decreased system extensibility and modification.

2.1.4 Heterarchical Form

In order to overcome the disadvantages of the centralized and hierarchical control structures, a heterarchical (decentralized) approach is proposed (Figure 2.1.d). In a heterarchical system there is no organizational hierarchy and master/slave relationship. Each component of the system has autonomy and can communicate between each other in peer-to-peer fashion. Each component possesses sufficient local knowledge to accomplish its own task. A task that a single component is unable to finish alone may require the cooperation of a cluster of components. Communication is the key for achieving cooperation between the autonomous components. Decision making occurs at the point of information gathering rather than at a central controller. Duffie and Prabhu (1994) presented several design principles for a heterarchical system:

- The system should be decomposed into a set of quasi-independent entities with relatively weak interactions.
- Master-slave relationships should not exist between entities.
- The physical system configuration should be transparent to entities in the system, and entities should not need to know where other entities reside.
- Time-critical responses should be contained within entities and should not be dependent on time-critical responses from other entities.
- Entities should cooperate with other entities whenever possible, but should not assume that other entities will cooperate with them.

- Entities should delay establishing relationships with other entities for as long as possible and should terminate these relationships as soon as possible.

Many advantages of the heterarchical control architecture are because of the full local autonomy property. Elimination of the master/slave relationships results in reduced coupling between the modules. This causes a natural increase in the fault-tolerance level of the system without any need of external intervention such as introducing redundant resources or programming. The major characteristic of cooperative decision making will continue even if one or more components fail. Moreover, heterarchical systems require relatively less complex software. According to the study of Duffie and Piper (1987) the line of the source codes of centralized controller, hierarchical controller and heterarchical controller are 680, 2450 and 259 respectively. This also verifies the reduced software complexity of the heterarchically controlled systems. Finally heterarchical form enhances the reconfigurability and adaptability of the system. As long as the capacity of the network allows, a machine can be physically added or removed from the manufacturing system. This causes the control component of the machine to connect or disconnect from the system as well.

The disadvantages of the heterarchical form are mainly derived from the technological reasons. Resolving this defect requires a robust mechanism to support cooperation between the autonomous components having the same type of peer-to-peer control component with a well developed operating software. Another disadvantage is the poor global optimization. This problem arises because of the full local autonomy of the individual components that do not possess a global perspective.

Okubo et al. (2000) compared the distributed and centralized production control systems by response time, planning scope, and progressive accuracy. Progressive accuracy is defined by Okubo et al. as: “the difference between a plan and the result of production progress”. Their simulation results showed that a distributed control

system enables a shorter response time, narrower planning scope, and higher progressive accuracy than a centralized control system. However, when the system is in heavy work load condition, the lead times of centralized architecture turn out to be smaller when compared to the distributed architecture. This is because of the wide planning scope of the centralized form so that it can control the work in process (WIP) level with a more global perspective.

Because of the implicit fault tolerance, ease of reconfigurability and adaptability of the heterarchical control architectures, they become attractive alternatives for manufacturing systems. Having fully autonomous components, this form is frequently accompanied with agent based technology, where each resource in the shop-floor is represented by corresponding agents. Next section consists of the agent based shop-floor control concepts and literature review.

2.2 Agent Based Shop Floor Control

Due to the structural rigidity and reliance on single control unit of classical centralized architectures, the heterarchical structure has been more appealing for control of manufacturing systems. (Duffie and Piper, 1986, 1987; Duffie et al., 1988; Crowe and Stahlman, 1995; Dilts et al., 1991; Duffie and Prabhu, 1994). One of the major properties of the heterarchical structure is that the decision-making responsibilities are fully distributed to each component of the system. Each component is autonomous and possesses local knowledge that is sufficient to accomplish its own task. Implementing a distributed control architecture, the requirements of the next generation of manufacturing systems, such as good fault-tolerance, ease of reconfigurability and adaptability, and agility, can be achieved (Shaw and Norrie, 1999).

Table 2.1 Summary of Control Architectures by Dilts et al.(1991)

	Features	Advantages	Disadvantages
Centralized	• single mainframe computer • all control decisions made at a single location • global database records all system activities	• access to global information • global optimization possible • single source for system status information	• slow and inconsistent response speed • reliance on single control unit • difficult to modify control software
Proper Hierarchical	• multiple, variety of computers • rigid master/slave relationships between decision making levels • supervisor coordinates all activities of subordinates • aggregated databases at each level	• gradual implementation, redundancy, reduced software development problems • incremental addition of control possible • allowance of differing time scales • fast response times	• computational limitations of local controllers • increased number of inter-level communication links • difficulties with dealing with dynamic adaptive control • difficulty of making future unforeseen modifications
Modified Hierarchical	• multiple, variety of computers • loose master/slave relationships between decision making levels • supervisor initiates sequence of activities in subordinates • subordinates cooperate to complete sequence	• all the advantages of proper hierarchical control • ability of local systems to have local autonomy • ability to off-load some linkage tasks to local controllers	• most of the disadvantages of the proper hierarchical form • connectivity problems • limitations of low-level controllers • increased difficulty of control system design
Heterarchical	• multiple, but less variety computers • no master/slave relationships • full local autonomy • distributed decision making for activity coordination • local databases only	• full local autonomy • reduced software complexity • implicit fault-tolerance • ease of reconfigurability and adaptability • faster diffusion of information	• primarily due to technical limits of controllers • no communication standards • high likelihood of only local optimization • requires high network capacity

Agent-based technology fits naturally into the heterarchical control structure because the autonomous component can easily be represented by an agent that is defined as an autonomous, pro-active element with the capability to communicate with other agents. From the perspective of a software application, an agent can be viewed as a computational module that is able to act autonomously to achieve its goal.

According to Arazy et al. (2002) agents should have the following properties:

- **Autonomy**: An agent can operate without the direct intervention of external entities, and has some kind of control over their behavior
- **Cooperation:** The agents interact with other agents, in order to achieve a common goal.
- **Reactivity:** The agents perceive their environment and response quickly to changes that occur on it.
- **Proactivity:** The agents do not simply act in response to their environment, but are able to taking the initiative, controlling its behavior.
- **Adaptation and Decentralization:** The agents can be organized in a decentralized structure, and easily be reorganized into different organizational structures.

The design of heterarchical systems of autonomous agents, so-called Multi-agent systems (MASs) for use in manufacturing gained much attention in the robotics and automation research community. Fan and Wong (2003) state that multi-agent technology has been applied to various concepts such as manufacturing enterprise integration, supply chain management, manufacturing planning, scheduling and control (Shen, 2001; Shen and Norrie, 2002), materials handling, and holonic manufacturing systems (Kadar et al. 1998) since each agent can be used to represent physical shop-floor components such as parts, machines, tools, and resources. Under the application of multi-agent systems, agents are in charge of information collection, data storage, and decision-making for the corresponding shop-floor

component. Due to their distributed nature, MASs promise, at least theoretically, some advantages that make them attractive structures for control and execution of manufacturing processes. Their main advantages are modularity, robustness, fault tolerance, maintainability, and extendibility. These features of MASs hold the potential of building manufacturing systems with greater flexibility then the currently used monolithic ones.

MASs are described from two different points of view. First, from the viewpoint of a single agent and second, from the viewpoint of the system as a collection of interacting agents. Obviously, the coordination and communication of the participating agents play an important role. A single agent is modeled as a skilled subsystem, which performs actions that are related to its locally defined goal, whereas a combination of agents with different skills and goals form a system (Friedrich et al., 1998)

The essential feature of MASs is that they are distributed and autonomous in their intelligence such that they would have capabilities for scheduling in relative isolation while resolving conflicts among themselves to maintain consistent local schedules. The type of schedule carried out by an entity is the application of rules to dispatch the next job for execution, rather than any maintenance of a local schedule.

According to Alataş (2003), using multi-agent technology has some more advantages. These are:

- **Platform independency**: The use of Object Oriented programming language and distributed communications platforms, such as CORBA, to develop control applications, allows the use of the same application in different operating systems environments (such as Windows, Linux and Unix), being platform independent.
- **Application development**: Using the agent-based approach, the software necessary to develop the application is shorter and simpler to write, to debug and to maintain.

- **Code re-usability**: The multi-agent technology concept allows an easy and modular development of control applications. Additionally, some components of the developed control application can be re-used for other applications.
- **Distribution and Autonomy**: Each agent has autonomy, has control about its behavior and has local and community knowledge. By this way, it is possible to build distinct and independent agents that can be placed transparently in a distributed environment.
- **Plugging Intelligence:** The addition of intelligence to an agent, for example to take decisions, manage disturbances or learning, is a transparent process for the agent and can be viewed as a plug-in of an intelligence module, which takes easier the development of control applications.

Agent-based technology has been widely recognized as a promising paradigm for developing software applications able to support complex tasks. However there can be several tasks that a single agent is unable to finish alone. Such tasks require the cooperation of a group of agents. Communication is a means of establishing such cooperation between those autonomous agents. A popular scheme to achieve cooperation among autonomous agents is through the negotiation-based contract-net protocol (Smith, 1980). The contract-net protocol provides the advantage of real-time information exchange, making it suitable for shop-floor scheduling and control.

2.2.1 The Contract Net Protocol

The contract net protocol has been developed to arrange cooperation between distributed agents in a manufacturing system. It was first proposed by Smith (1980) and demonstrated on a distributed sensor system.

The contract net model consists of a set of nodes representing a set of decision makers that accomplish a negotiation procedure between each other. Each node in the net takes one of the two roles related to the execution of an individual task: manager or contractor. A *manager* is responsible for monitoring the execution of a task and processing the results of its execution, and a *contractor* is responsible for the actual execution of the task. A *contract* is the agreement between the two nodes. Parunak (1987) changed the definition of the nodes by introducing the *bidder* node (contractor candidate) by defining it as a node that offers to perform a task. As a result the contractor node is defined as the successful bidder whose bid is accepted by the manager node.

The basic idea behind the negotiation is that available contractor candidates evaluate task announcements made by several managers and submit bids on those for which they are suited. The manager nodes evaluate the bids and award the nodes they determine to be the most suitable for the task. The negotiation process may then recur. A contractor may further partition a task and award contracts to other nodes. It is then the manager for those contracts. This leads to the hierarchical control structure that is typical of task sharing. Control is distributed because processing and communication are not focused at particular nodes, but rather every node is capable of accepting and assigning tasks. (Smith, 1980)

Nodes communicate between each other by means of messages and after receiving each message a procedure is triggered within the node. The negotiation procedure is normally initiated by the node that generates a task to be done by declaring the existence of the particular task to the other nodes through broadcasting a **task announcement** message. The node that announces the availability of a task becomes the manager node. A task can be announced to all the nodes, to a limited number of nodes or to a single node in the net. Announcing the task to a limited number of nodes provides reduced message traffic and processing time since the non-addressed nodes are allowed to ignore the task announcements by only

examining the *addressee* slot of the corresponding message. A task announcement message has four main slots (Figure 2.2).

To: * indicates a broadcast message
From: 25
Type: TASK ANNOUNCEMENT
Contract: 22-3-1
Task Abstraction:
TASK TYPE SIGNAL
POSITION LAT 47*N* LONG 17*E*
Eligibility Specification:
MUST-HAVE SENSOR
MUST-HAVE POSITION AREA *A*
Bid Specification:
POSITION LAT LONG
EVERY SENSOR NAME TYPE
Expiration Time:
28 1730Z Feb 2005

Figure 2.2 Task Announcement Message by Smith (1980)

In the *eligibility specification* slot, the criteria necessary for a node to submit a bid is present. This reduces the message traffic by eliminating the nodes which would submit unacceptable bids. Difference of eligibility specification slot from the addressee slot is that, it is used to eliminate the nodes when the manager is not certain about the specific names of the nodes but can describe the properties of a node that it is willing to cooperate. *Task abstraction* is used to describe the task briefly. According to this slot, nodes rank the tasks relatively. The *bid specification* slot is a description of the expected form of a bid. Using this slot, manager specifies the required information from the contractor candidate nodes for constructing their bids. By this way, during the bid construction, contractor node only includes information about its capabilities relevant to the task rather than a complete

description. This simplifies the bid evaluation procedure of the manager node and reduces the message traffic. *Expiration time* is the final time for the nodes to submit bids. These deadlines dictated by the manager node are not crucial during the negotiation since the nodes can send bids after the expiration time and still have a chance to make a contract with the manager. An example task announcement message is given in Figure 2.2.

Each node maintains an ordered list of announcements that have been received. During task announcement processing a node initially checks the eligibility specifications of the announced task. If the node is eligible to submit a bid, it ranks the new announcement relative to the others under consideration. If any other criterion is not defined, most recently received task announcement obtains the highest rank. This procedure is called as task announcement processing by Smith (1980).

After ranking the tasks, the contractor candidate is enabled to submit bid with a **bid message** to the announced tasks. It checks the list of announced tasks and selects the task to submit a bid. If there is only a single task in the list then the bid is submitted on the particular task. However, if there are several tasks announced, then the contractor node must select one of them. By default, the Contract Net Protocol defined by Smith (1980) selects the task that is most recently received. Other selection criteria defined by users can also be implemented. An idle node can submit a bid when either the node receives a new task announcement or the expiration time is reached for any received task announcements. At each specific instance, the node makes a decision whether to send a bid message of to wait for the new coming task announcements. The task bid message proposed by Smith (1980) is shown in Figure 2.3. The *node abstraction* slot contains a short specification of the capabilities of the node that are relevant to the announced task. It is generated according to the bid specification of the task announcement message dictated by the manager node.

To: 25
From: 42
Type: BID
Contract: 22-3-1
Node Abstraction:
POSITION LAT 62*N* LONG 9*E*
SENSOR NAME *S1* TYPE *S*
SENSOR NAME *S2* TYPE *S*
SENSOR NAME *T1* TYPE T

Figure 2.3 Task Bid Message by Smith (1980)

In the manager node, the contractors are queued locally until they are awarded. The manager node also keeps a rank ordered list of the submitted bids by the contractor nodes for the announced task. Upon the arrival of a new bid, the manager ranks the bid relative to the others that are under consideration. If any bid is found to be satisfactory for the specific task then the corresponding contractor node is awarded immediately. Since the manager node is allowed not to wait until the expiration time to award a contractor, the average negotiation time is kept low. If there is not any satisfactory bid at that moment, the manager waits for further bids. When the expiration time is reached and the task is not awarded to a contractor node, there are several actions that can be taken. First, task can be awarded to the most acceptable bid. Second alternative is to transmit another task announcement message for the same task or to wait for a period before transmitting another task announcement.

The winning bidders are informed that they are the contractors to the specific task by an **award message** generated by the manager node (Figure 2.4). The *task specification* slot contains the information that is needed to start the execution of the task and any additional data that was requested by the bidder.

After receiving the award message, the contractor still has a chance to accept or reject the awarded task by an **acknowledgement message**. The manager can also

interrupt the execution of a contract with a **termination message**. Upon receiving such a contract, the contractor terminates its performance of the contract and all of its subcontracts if there are any.

To: 42
From: 25
Type: AWARD
Contract: 22-3-1
Node Abstraction:
SENSOR NAME *S1*
SENSOR NAME *S2*

Figure 2.4 Task Award Message by Smith (1980)

Figure 2.5 shows the decision making process of the contract net protocol schematically.

Although the contract net approach is quite simple and can be efficient, when the number of nodes increases, the number of messages on the network increases as well. This, in turn, results in a situation where agents spend more time processing messages than doing the actual work, or worse the system stops through being flooded by messages. Thus, various improvements to the basic contract net approach have been proposed (Shen and Norrie, 2001), such as:

- sending offers to a limited number of nodes, instead of broadcasting them;
- anticipating offers, i.e., contractors send bids in advance;
- varying the time when commitment is decided;
- allowing de-commitment (breaking commitments);
- allowing several agents to answer as a group (coalition formation);
- introducing priorities for solving tasks.

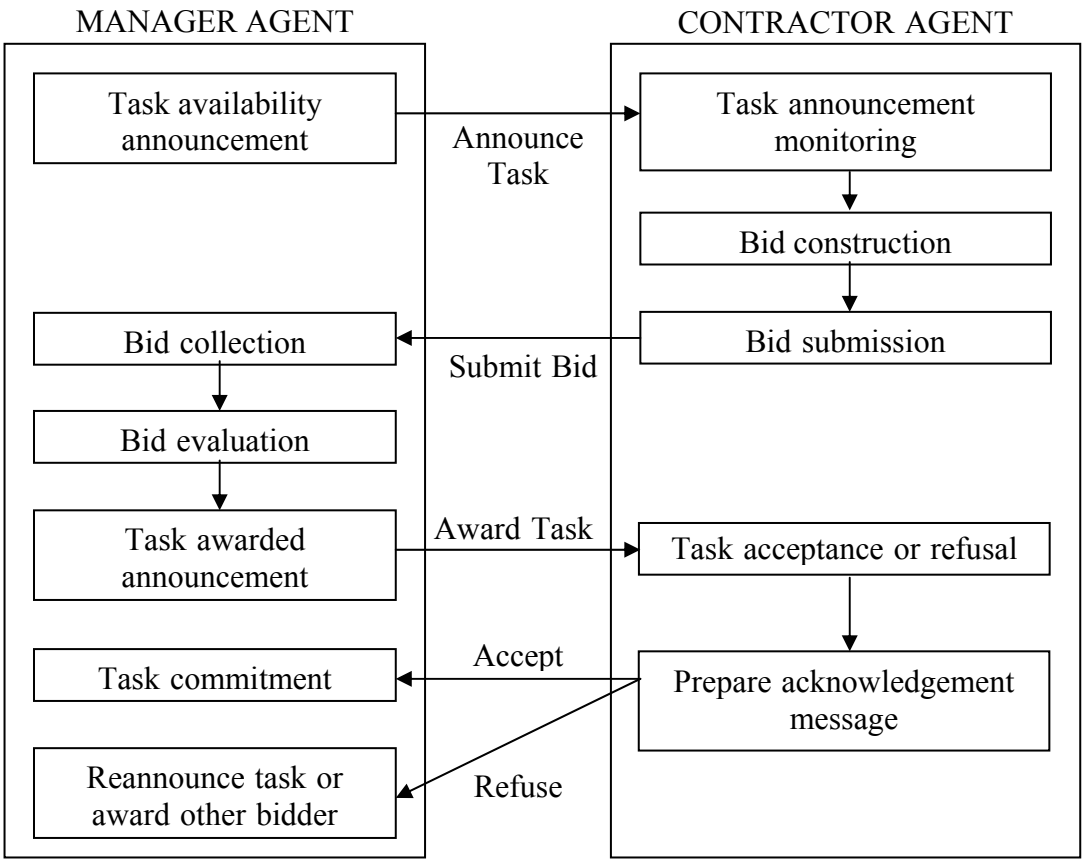

Figure 2.5 The Contract Net Protocol

In the standard Contract Net Protocol, bids are compared corresponding to a particular offer from the manager agent. An example for the improved case allowing de-commitment can be making the protocol similar to a market approach by introducing penalties.

Even though the variation of contract net protocol may exist in different implementations, two main procedures are usually present which are bid construction and bid evaluation. Bid construction refers to the process of calculating the bid value and providing other required information along the bid submission. Bid evaluation refers to the process of comparing different submitted bids and deciding which bid will be awarded as the winner. Various criteria and algorithms

may be employed in this stage, ranging from simple minimum cost or time to mathematical function.

2.3 Bidding Based Scheduling Approaches

In a general multi-resource system, an activity or set of activities may have several choices of resources to accomplish their completion. As an example, a workpiece can be drilled in a CNC milling machine or it can be turned on a manual lathe to obtain the same sized hole. Different choices for an activity are called different *routes* and finding the best route is called the *job routing problem*.

Lin and Solberg (1991) identified four types of routing flexibility based on the availability of alternative machines for an operation, alternative operations for a feature and alternative operations sequences for a job. For the case of *no routing flexibility*, a job is completed using a fixed sequence of operations and each operation must be processed on a specific machine. There are no alternative machines capable of performing the same operation. For the *fixed sequencing* type, the operations of a job must be performed in a fixed sequence, but there can be more than one machine capable of processing any given operation. This case is extended in third type, *flexible sequencing*, where alternative sequences of the operations are permitted. The last type is *flexibly processing* where alternative sequences are permitted whereby alternative operations may be available for machining each feature and alternative machines employed to perform the selected operation. These four types of routing flexibility are summarized in Table 2.2.

Chan (2001) used Taguchi experimental design techniques to study the effects of different levels of routing flexibility on the performance of a FMS. In the study, routing flexibility is defined as a measure of the average number of choices of a machine that an individual part can choose. He found that increasing routing flexibility does not guarantee an improvement in system performance. Chan

concluded routing flexibility with a measure of 2 (meaning that on average, each job has two options of which machine to use for its next operation) provided the best system performance when makespan and flow time are considered.

Table 2.2 Types of Routing Flexibility

	No Flexibility	Fixed Sequencing	Flexible Sequencing	Flexible Processing
Alternative M/C for an operation	No	Yes	Yes	Yes
Alternative operation for a feature	No	No	No	Yes
Operation sequence of a job	Fixed	Fixed	Flexible	Flexible

2.3.1 Agent Based Approaches

The job routing problem is solved through the negotiation messages that are sent or received by the agents representing the manufacturing system components. As mentioned in Section 2.2.1 a popular scheme to achieve cooperation among autonomous agents is through the negotiation-based contract-net protocol. The cooperation can be achieved with two different bidding schemes: Part initiated and Resource initiated negotiation schemes.

In **Part Initiated Negotiation Scheme**, upon its arrival, the part agent makes a task announcement that involves the type of operation and additional information such as processing time, due date and the priority of the part if applicable. The interested or all of the resources construct bids and compete with each other to obtain the part for processing. In the next step, the part agent collects the offers including additional information about the resource status. The part awards itself to the machine offering the most convenient bid. In this type of negotiation scheme, the

machine *pulls* the parts through the manufacturing system. The part initiated scheme is show in Figure 2.6.

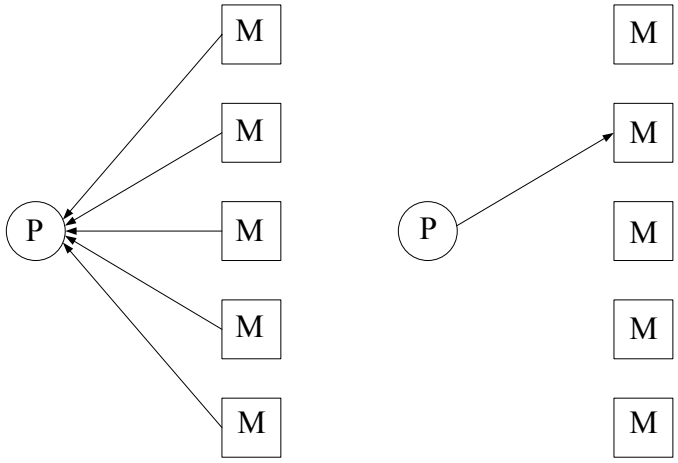

Figure 2.6 Part Initiated Negotiation Scheme by Saad et al. (1997).

In **Resource Initiated Negotiation Scheme**, upon finishing its current task, the resource agent announces its availability to all part agents. The interested part agents make proposals addressed to the available resource according to its process plan. The target of the bidding is the services or operations that are offered by the machines. The resource agent determines the winning part agent according to a certain objective. In this type of negotiation scheme, parts *push* themselves through the manufacturing system. The resource initiated scheme is show in Figure 2.7.

Shaw (1987, 1988) employed the contract-net method for dynamic scheduling in cellular manufacturing systems. In his approach, when an operation of a job at a cell is finished, the control unit of a cell will make the decision regarding which cell the job should visit next. The cell's control unit broadcasts the task announcements to the other cell control units. When the cell control unit receives a task announcement message it checks whether the required operation is within its capability and

submits its estimation on the earliest finishing time (EFT) or shortest processing time (SPT). Route of each job is determined through the negotiation between the cells. Shaw's experimental results indicated that the bidding scheme with EFT (earliest finishing time) outperformed the bidding scheme with SPT. The difference between two schemes is because of the additional information such as estimated waiting time or estimated transporting time which is taken into account within bidding scheme with EFT.

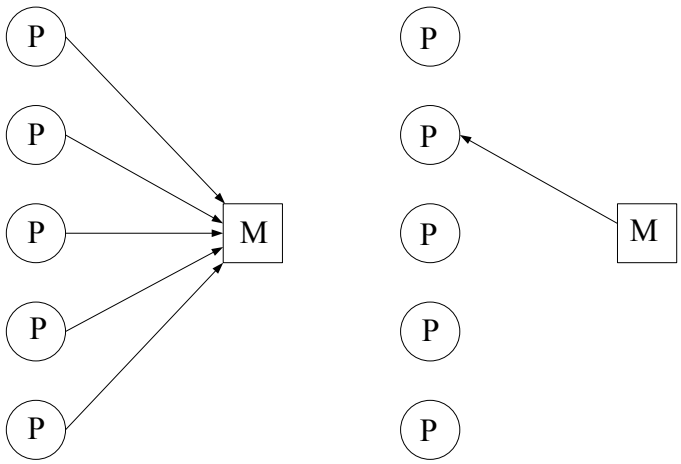

Figure 2.7 Resource Initiated Negotiation Scheme by Saad et al. (1997)

Saad et al. (1997) proposed a contract-net-based heterarchical scheduling approach for flexible manufacturing systems. In their study, two scheduling mechanisms were tested. The first is the Product reservation (PR) method where all the operations of a job are scheduled completely at the time when it arrives to the system. The other method, referred as Single Step Product reservation (SSPR), schedules one operation at a time with the job agent delaying negotiation of its next operation until the current operation is finished. In the contract-net protocol, a job agent selects the machine that can finish processing the required operation first. If at least two alternatives are tied for this criterion, the job agent will choose the machine with

fewer jobs in its reservation list. The PR and SSPR approaches are compared with some traditional dispatching rules. Results revealed that PR outperformed the traditional dispatching rules, while SSPR only outperformed PR on average tardiness. However, unexpected events such as machine breakdowns or emergent jobs were not considered in their experiments. Otherwise, SSPR will be more advantageous under the effect of those uncertainties.

Xue et al. (2001) developed an intelligent optimal scheduling mechanism that uses a constraint-based search mechanism to identify the best sequence to accomplish the required tasks, as well as timing parameter values (the earliest and the latest task finish times). Given the timing parameter values, the agent-based collaborative mechanism was used to generate a production schedule. The agent-based collaborative mechanism consists of a bidding mechanism and a mediator mechanism. The bidding mechanism is implemented based on the contract-net protocol and the mediation mechanism is used to coordinate the activities of the relevant agents to improve the scheduling efficiency. In the study, the manufacturing resources, including facilities and persons are modeled as agents. Two mediators, facility mediator and personnel mediator, are used to coordinate the activities of the resource agents.

Sousa and Ramos (1999) proposed a contract-net based negotiation protocol for scheduling in manufacturing systems. The bid submitted from the resource agent consists of the information concerning the time windows that the resources are free. Selecting bid was based on the resources being able to finish the part before the due date and with more free time intervals. The authors also mentioned about renegotiation phase when a machine malfunctions. However, no further explanation is given on how to deal with the scheduled operations that are affected by this malfunction.

Brennan and O (2000, 2001) used the contract-net based bidding scheme to compare FCFS rule (AUC_BID) and SPT rule (AUC+JSEQ) to sequence the

incoming parts in the machine queues. The part agents request the Earliest Start Time from the machine agents and select the machine agent that can start processing the part first. In the study, the effect of violating the previous commitment for the sake of renegotiation is also investigated (COMT+AUC+JSEQ). It is found that the scheme including the renegotiation scheme (COMT+AUC+JSEQ) is superior to the other scheme in all cases. However, the bidding scheme using AUC+JSEQ is close to the COMT+AUC+JSEQ case. The control strategy using the (AUC_BID) is always outperformed by the other two rules.

Another agent-based negotiation approach called **market-like approach** is very similar to the contract-net protocol except currency is used for bid evaluation. Each job agent carries some amount of fictitious currency and tries to achieve the processing requirement to reach its weighted objectives by bargaining with the resource agents. The objective of the part represents the need of a customer. In every bidding process, the job agent who is able to offer the highest bid takes priority of being processed. The agent negotiation strategies in the studies presented below employ a market-based approach.

Lin and Solberg (1992) presented an agent-based shop-floor scheduling and control framework based on a market-like model that combined the objective and price mechanisms. In the proposed system, each job agent with its unique set of weighted objectives enters the system with some currency and alternative process plans. To achieve the objectives, job agents will try to fulfill the processing requirements by bargaining with resource agents. Each resource agent sets its charging price based on its status. The part agent tries to minimize the price paid, but the resource agent's goal is to maximize the price charged. Each deal is completed once the part agent and resource agent are mutually committed. One important feature of this market-like mechanism is that the negotiation among agents is invisibly guided by an adjustable price to improve the system performance. Lin and Solbergs' results essentially showed that their system was able to handle unexpected resource failures

and part objective changes. Lin and Solberg (1994) later presented a manufacturing simulation system based on the dynamic price mechanism for agent negotiation. The proposed agent-based framework simplifies implementation of different negotiation strategies in manufacturing systems.

Gu et al. (1997) demonstrated the use of bidding scheme in process planning and scheduling in agile manufacturing systems with hierarchical control structure. A shop manager agent is a coordinating agent and keeps track of the system state. The shop manager agent assigns the part to the machine agent that can process the part with the lowest total cost. The total cost consists of machine schedule, machining time, setting up time, tool change time, cost of tooling and the penalty for part lateness.

Ou-Yang et al. (1998) developed a hybrid hierarchical/heterarchical shop-floor control system using bidding method in routing selection. Machines submit a bid with price calculated from processing cost, inventory cost, and shortage cost. The selection criterion is based on the production price and the utilization rate of each cell.

Dewan and Joshi (2000, 2001) developed an auction-based scheduling mechanism for a job shop environment. They also used currency as a means for agent negotiation. Their market-like approach differed from Lin and Solbergs' (1992) in using Lagrangian relaxation to decompose the problem formulation. The study is based on the resource initiated bidding scheme. Whenever a machine agent is available, it announces an auction for time slots from the current time to the end of the time horizon. Each job agent will bid for the time slots with the cost that they are willing to pay. The job agent's goal is to minimize cost, while the machine agent uses the submitted bids for price adjustment. If more than one job demands the same time slot, the price for that slot will increase. The price adjustment and bid calculations continue iteratively until the price converges. The machine agent determines the best bid for the earliest time slot as the next operation. After

processing is finished for that operation, the above auction procedure is executed again. Dewan and Joshi (2000) further used the above mechanism to schedule the jobs with different objectives.

Ottaway and Burns (2000) proposed an agent-based negotiation involving a currency scheme. In their model, the amount of currency that a job agent carries is based on the job's objective function, a weighted linear combination of time, cost, and quality. The resources determine the amount of currency to be charged for their production services based on their capabilities and the demand for their services. It is noted that there is a factor for preventing a job from being stuck in the system due to a lack of currency. This factor is used to increase the budgeted funds for the jobs that kept failing in the bidding process. Ottaway and Burns also addressed the importance of using supervisor agents to balance the production load and maximize overall throughput. The supervisor agents essentially played a key role for dynamically switching the system structure between a hierarchy and a heterarchy.

Siwamogsatham and Saygin (2004) developed an auction based model based on the study done by MacChiaroli and Riemma (2002). In this type of negotiation scheme after completing a previous task the resource agent, announces its availability to all the part agents. The interested part agents make a service purchase proposal addressed to the available resource and to all other eligible resources according to the process plan. The resources receiving any proposal then construct an offer taking into account the proposals received and the service quality available in terms of expected completion time. Each part agent selects the offer with certain criteria. If more part agents accept the offer of a resource, the resource will then choose which part to execute. If there exists a spread between part proposals and resource offers, an iterative re-negotiation process is initiated which aims at reaching convergence. If parts and resources do not reach an agreement at the first step, a re-negotiation occurs. Proposals are increased and offers are reduced up to a predetermined limit with a gradient proportional to the current spread. After a predetermined number of iterations, both agents recognize that an agreement cannot

be found and the whole process is reset, so as to include other production resources that were not included in the previous negotiation. In order to benchmark the performance of the proposed auction based scheme 3 different job routing rules and 8 sequencing rules used. The results revealed that the proposed auction based approach outperformed the priority rules on most of the performance measures.

2.3.2 Other Approaches

Ro and Kim (1990) proposed three machine selection heuristics namely Alternative Routings Directed Dynamically (ARD), Alternative Routings Planned (ARP) and Alternative Routings Planned and Directed Dynamically (ARPD). The ARD rule is a rule to select the machine that has the shortest time composed of a sum of travel time, queuing time, and processing time. Use of the ARP rule requires that routes be determined by a linear programming (LP) model whose objective is to minimize makespan. Implementation of the ARP rule requires that the LP model to be solved whenever a new job arrives or a machine breaks down. The ARPD rule is a combination of ARD and ARP. Initially, the routes are determined by solving the LP model, but if the primary machine (from LP solution) is busy, a machine is selected based on the ARD rule. Ro and Kim compared their three heuristics with two other heuristics namely No Alternative Routings (NAR)and Work in Queue (WINQ). The NAR is a rule to select the route with the minimum total processing time (no alternative routes are permitted). From the simulation results, ARD gave the best results in four performance measures (makespan, mean flow time, mean tardiness, and maximum tardiness) except for system utilization. It also found that ARD, APRD, and WINQ were significantly better than ARP and NAR in every performance measure.

Chandra and Talavage (1991) developed a heuristic dispatching system for FMS. In their system, a part after completing an operation is not routed to a specific machine, but is sent to a global buffer. The routing decisions are not made by the

parts, but by the machines. Their dispatching mechanism categorizes and selects the jobs based on a predefined algorithm. The mechanism was also able to deal with a scheduling problem with multiple objectives. The authors compared their system to the four traditional dispatching rules (SPT, EDD, LSPO, LRS). The developed dispatching system consistently outperformed those dispatching rules under various circumstances. They concluded that making decisions with simple commonsense reasoning combining some empirically proven dispatching rules could achieve a significant improvement.

Subramaniam et al. (2000-1) proposed three route selection rules: LAC, LAP, and LACP. LAC selects the machine with the lowest average cost of processing every operation in the machine queue. For LAP machine selection is based on the lowest average processing time of every operation in the machine queue. LCAP awards the highest priority to the machine that has the minimum aggregate cost and processing time. The results revealed that LAC and LAP rules perform well for the mean cost and mean tardiness performance measures, respectively, while the LACP rule exhibits performance that is between the LAC and LAP rules.

Subramaniam et al. (2000-2) proposed an approach of dynamic dispatching rule selection based on the analytic hierarchical process (AHP), which considers the shop conditions existing at every decision point. In fact, AHP is an approach to help the decision makers to make better decisions in problems involving multiple objectives. The AHP provides a framework that ranks the alternatives based on the decision maker's knowledge and preferences. The results in the article showed that the AHP method is not guaranteed to generate the optimal schedule, but it is superior to the method using single dispatching rule for the measure of makespan.

CHAPTER 3

SYSTEM MODELING

3.1 Part Input Model

The bidding framework is mainly responsible for dispatching the parts to the resources on the shop-floor. So the parts entering to the system are crucial components of the framework and should have suitable and well selected properties to be used for reflecting the system performance. Generally, an incoming part has some time independent properties most of which are either dictated by the customer or decided upon engineering estimates. Those properties will be referred as *fixed attributes* throughout the study. This section is dedicated to the fixed part attributes and the models behind those attributes.

Part Number: Every part coming to the shop-floor has a unique part identification number which is given in the order of introduction to the shop-floor.

Part Type: The type indicates the process plan of the corresponding part. The modeled system is mainly based on the Computer Integrated Manufacturing Laboratory (CIMLAB) layout located in Middle East Technical University, Department of Mechanical Engineering. In the CIMLAB layout there are CNC Turning and Milling centers and therefore the parts in the modeled system can only undergo turning and milling processes. This results in a part having four distinct process plans. However, there is an assumption that a part can only have single pass in a particular type of machine and can not return to the same type for a second pass of the same process. Part types and the corresponding process plans are given in Table 3.1.

Table 3.1 Part types and process sequences

PART TYPE	FIRST PROCESS	SECOND PROCESS
1	Milling	Turning
2	Milling	-
3	Turning	-
4	Turning	Milling

Different types are assigned to different parts to obtain the variability in part attributes and results. It is assumed that the occurrence probability of part types is equal for a specific part. The assumption is simplifying, yet sensible, since the types are not superior to each other. Therefore, uniform distribution is utilized to generate the part types for a group of parts. Further information about uniform distribution is provided in Appendix B.1.

Priority: As the name implies, priority shows the precedence of a part over other parts. The priority concept is first introduced to CIMLAB layout by Alataş (2003). These values are assumed to be given by the manufacturing engineers once the new part is introduced into the system. The priorities can be treated as the urgency of a part as well as a representation of the importance of the customer or the budget of the job undertaken. They are mainly used for sequencing the jobs in the reservation lists of the machines.

In the modeled system a scale from 1 to 10 is used, 10 being the highest priority. It is assumed that for a specific part, taking any priority value is equally probable. Therefore, uniform distribution is utilized for assigning priority values to the incoming parts. Further information about uniform distribution is provided in Appendix B.1.

Time of Arrival: Time of arrival value indicates the time at which a part is accepted as a job to the system. There are two cases used for modeling the part arrivals: static and dynamic. If all of the jobs arrive simultaneously at the beginning of a time interval, this case is called as *static*. In *dynamic* case, jobs arrive in the system continuously and one at a time or in small batches. It is obvious that the static case is an artificial situation and dynamic case is a close match to reality where the parts arrive randomly over time. The dynamic modeling is also used in this study.

The arrival process is modeled based on the time passing between the arrivals of two successive parts which is called the interarrival time. The parts are assumed to come one at a time and completely at random. This case allows the utilization of exponential distribution for interarrival times. Further information about exponential distribution is provided in Appendix B.2.

Time of arrival is an important set of data for the simulation of the modeled system. By manipulating time of arrival data, the system loading level can be altered. To increase the loading of a system, the mean interarrival time should be decreased and vice versa. However, the arrival data should be so arranged that, the system does not end up with an excessively congested condition or with a few parts far from representing the real system characteristics.

In the modeled system, the initial part is assumed to have zero arrival time and the values obtained from the exponential distribution is added on each other to find the time of arrivals for the successive parts. The time of arrival data are in minutes. Different mean interarrival time values are used for different scenarios which will be mentioned in Chapter 5.

Processing Time: Processing time is the time required for a part to complete its operation on the corresponding machine. A processing time is not known until the part finishes its operation and it is an engineering estimate. There are two types of

processing times in the modeled system: *Turning time* and *milling time*. A part can have one or two of these process times according to its type. *Machining time* is simply defined as the sum of the individual processing times of a part. Of course, the machining time of a part requiring only one operation will be the processing time of its single operation.

The machines of similar type are assumed to be identical and therefore processing time for each type of operation for a specific part is taken to be the same on all similar type machines. The machine setup times are assumed to be included in the processing time for each operation.

Processing times are treated as deterministic without random input. However, they are modeled by triangular distribution. Triangular distribution is used for data whose exact form of distribution is not known. The distribution utilizes the estimates of minimum, maximum and most likely values for generating processing times. Further information about triangular distribution and pseudo code used for generating the processing times are provided in Appendix B.3.

According to the length of the processing times the bottleneck characteristics of a machine is determined. If long time values are used for the parameters of the triangular distribution, the possibility of obtaining a bottleneck resource increases for the corresponding type of machine. Processing time generation is important in this sense. In this study unit for the processing time is minutes. Results with different processing time lengths will be presented in Chapter 5.

Due Date: Due dates are the promised times by which all of the processes of a part should be finished. There are two cases for due dates: *Common* and *distinct* due dates. Common due dates may be used for situations where several parts constitute an order of a single customer or for assembly components where all parts should be ready at the same time. Distinct due dates are used for situations where all the parts

are independent of each other. Since all parts represent different customers, this study utilizes distinct due date modeling.

Due dates are generally modeled by adding an extra time on the total processing time of a part to create a proper slack. This value is then added on the time of arrival to obtain the due date. The following equation is used for generating due dates for the incoming parts:

Due Date = Time of Arrival + Machining Time * Random[a..b]

In the above equation Random[a..b] generates a random number from the (a,b) range. Therefore *a* and *b* are the parameters that will determine the slack. In order to avoid a part being late at the instant of completing all of its operations without any delay, the parameters *a* and *b* should be greater than one.

Due date is an important parameter for the shop-floor simulation study since it may reveal the system performance under loose and tight due dates. Higher the parameters *a* and *b*, looser the due dates get. Due dates always take integer values and like the other input parameters, the unit of due date is minutes in this study.

3.2 Shop-Floor Model

3.2.1 Present System – CIMLAB Test-bed

The current heterarchical agent-based flexible manufacturing system has been implemented by Integrated Manufacturing Technologies Research Group (IMTRG) in Computer Integrated Manufacturing Laboratory (CIMLAB) located in Department of Mechanical Engineering, METU. The control model has been developed using the three-tiered model of Windows DNA. User, Business, and Data Services of the "Agent" has been mostly written under Visual Basic 6.0. For the

communication and event driven messaging of agents, Microsoft Message Queue Server (MSMQ) has been used, stateless objects for database search and update has been deployed in Microsoft Transaction Server (MTS). The common database of the "Agent" has been constructed using SQL Server 7.0. Internet Information Server (IIS) has been used to grant access to the web sites as ASP and HTML pages, which are designed in Visual InterDev 6.0, a product of Microsoft Visual Studio.

The system consists of a single manufacturing cell as show in Figure 3.1. The main material handling system utilized in the system is composed of a closed loop buffer (conveyor) and a 6 axis robot. The closed loop buffer is used as an intermediate storage and for movements between Computer Numerical Controlled (CNC) Turning and Milling Machines and the static buffer. The static buffer (AGV) is used for loading and unloading parts to the system. It has distinct places for accepted and rejected parts. The movement of the robot between the CNC Turning and CNC Milling Machine is accomplished by a Pneumatic Linear Robot Drive (PLRD). PLRD lets the robot move linearly between part load-unload and CNC Turning and CNC Milling Stations. CNC Turning and Milling Machines are loaded and unloaded using the robot.

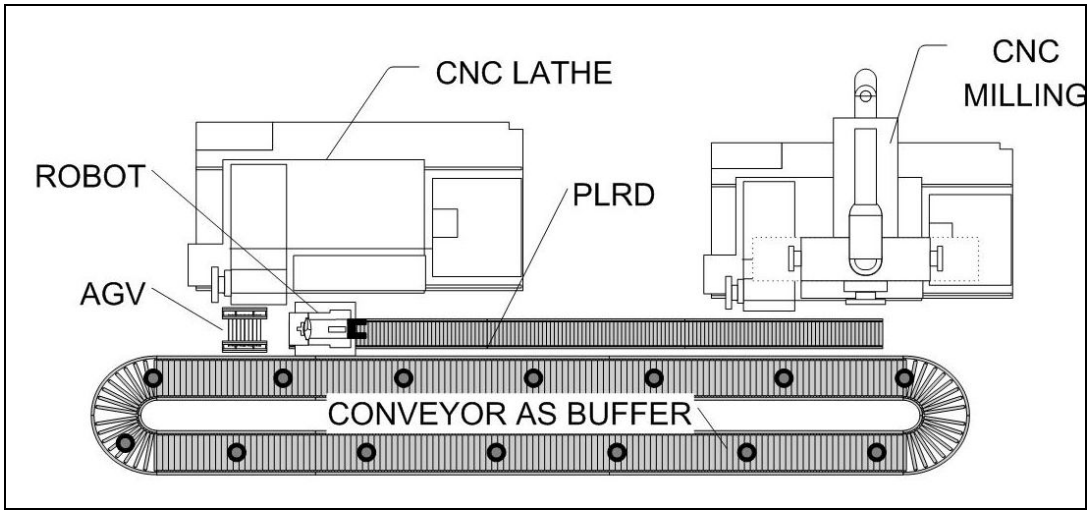

Figure 3.1 Layout of CIMLAB

The functionality, property and capability details of the manufacturing and transport hardware are stated below:

1. *CNC Turning Machine*: The model of the machine is Mirac/Denford/UK. It is a medium duty, PC based lathe having 2 simultaneously controlled axes. It is equipped with a turret having 8 stations. The door and the chuck of the machine are pneumatically powered. It can handle typically bars up to 50 mm in diameter and 150 mm in length and has a maximum spindle speed of 2500 rpm. A built-in user-friendly interface is utilized to visualize and debug part programs. The control is via standard RS 232 serial communication port and I/O card at a single sensor channel.

2. *CNC Milling Machine*: The model of the machine is Triac/Denford/UK. It is a medium duty, PC based milling machine having 3 simultaneously controlled axes. It is equipped with an automatic tool magazine with 6 stations. The door, chuck and tool magazine are pneumatically powered. It can handle parts up to 200 mm in width and 500 mm in length and has a maximum spindle speed of 2500 rpm. A built-in user-friendly interface is utilized to visualize and debug part programs. The control is via standard RS 232 serial communication port and I/O card at a single sensor channel.

3. *Closed Loop Buffer*: The model of the conveyor is SKF/UK. It is a unidirectional, constant speed, closed loop buffer having 14 cups. Typically, it can handle cylindrical parts up to 50 mm in diameter. It is driven by a motor with gearbox. The conveyor has a speed of 87 mm/sec and a total length of approximately 7100 mm resulting in a full rotation time about 82 seconds. The control is via 48 channel I/O card. The conveyor has one operate channel and one counter channel. To start the conveyor rotation, the operate channel is switched to ON mode and to stop the rotation it is switched OFF. The counter channel is used to count the number of the cups passed.

4. *Robot*: The model of the robot is Movemaster EX/Mitsubishi/Japan. The robot is capable of handling bars of 50 mm in diameter and has a weight of approximately 3 kg. The control of the robot is achieved by storing positions taught by the user in its EPROM and these programmed positions can be executed by external triggering of program commands through RS232 connection from the computer. A DSR (data set ready) signal from the serial port indicates that there is no active program running or the task is finished. The robot is used for loading and unloading operations between AGV and conveyor or conveyor and machines. Each operation carried out by the robot lasts approximately 30 seconds.

5. *Pneumatic Linear Robot Drive (PLRD)*: The model of the PLRD is FESTO/Germany. It is a pneumatically powered linear drive for the robot having a movement range of 2 meters. The stop positions of the PLRD are at both ends only. In CIMLAB configuration it is used to move the robot from CNC turning machine to CNC milling machine neighborhood. The control is via 48 channel I/O card. The PLRD has two operate - and two sensor channels. When the first operate channel is triggered and immediately released it moves to right and vice versa for the second. Sensor channels on the left and right positions indicate ON when the robot is at left and right ends of its range respectively. The traversing speed of the robot is not constant during the 2 meter movement due to its acceleration during starting and stopping.

6. *Static Buffer (AGV)*: The stationary buffer is used to import parts to the cell and export the finished parts out of the cell. It has 3 input and 3 output stations which can handle bars of 70-90-100 millimeters. Although the buffer is not physically connected or driven by a computer and it has no control or moving capabilities, it is modeled as an automated guided vehicle (AGV) in the system.

Overall view of the entire system in CIMLAB and the individual components are given in the Appendix A.

3.2.2 Modifications and Assumptions

The flexible manufacturing system model of this study is based on the Computer Integrated Manufacturing Laboratory (CIMLAB) in Department of Mechanical Engineering, METU. However, the present flexible manufacturing system is not modeled according to its present state. Several modifications are done to make the current flexible system suitable to implement the developed bidding framework. The basic difference of the modeled system from the flexible manufacturing cell in CIMLAB is the number of CNC turning and milling machines utilized. Other modifications are the auxiliary ones accompanying the increase in the number of CNC machines. Detailed descriptions of the modifications and assumptions on the manufacturing and transport hardware are as follows:

- *CNC Turning Machines* having the same specifications as the one already present in the CIMLAB can be added to the modeled system. By this way a competitive environment can be created between each of the alternative CNC Turning Machines and therefore the job routing problem is introduced to the shop-floor for turning operations. The machines do not have any input buffer. The part that will be processed is called from the AGV which is used as a raw material buffer. The CNC Turning machines are assumed to finish their process cycles for a given group of parts without requiring any maintenance. Also, no machine breakdown occurs during production.

- *CNC Milling Machines* having the same specifications as the one already present in the CIMLAB can be added to the modeled system. By this way a competitive environment can be created between each of the alternative CNC Milling Machines and therefore the job routing problem is introduced to the shop-floor for milling operations. The machines do not have any input buffer. The part that will be processed is called from the AGV which is used as a raw material buffer. The CNC Milling machines are assumed to finish their process

cycles for a given group of parts without requiring any maintenance. Also, no machine breakdown occurs during production.

- The length of the *closed loop buffer (conveyor)* is increased to accompany the new number of CNC machines. The distance for the conveyor to travel between loading/unloading stations of two successive machines or the AGV and a machine is assumed to be constant and 2 meters. Since the linear speed of the conveyor is 87 mm/sec, the travel time between two successive loading/unloading stations approximately takes 20 seconds. The number of the cups on the conveyor is also increased to meet the increased production rate. In order to be consistent with the previous configuration having 14 cups for 7100 mm length, number of cups on the conveyor is approximated by the formula below. The formula assumes that there are 3 available cups in the vicinity of each CNC machine and the AGV.

  Number of Cups on Conveyor = (Number of Total CNC Machines + 1) x 3 +3

  In the developed system, the conveyor is assumed to be bidirectional, resulting in a rotation in both directions. This allows the conveyor to reach the destination resource in a shorter time, selecting the shorter path by minimizing the distance traveled. Besides, the conveyor is assumed to be stationary during loading/unloading action of the robot and there is an empty cup present at the loading/unloading location. This eliminates the need for indexing one of the empty cups on the conveyor to the loading/unloading station of the resources. The parts that have finished their first operations are unloaded to the conveyor and the conveyor is used as a buffer for those parts that are waiting their second operation.

- The movement range of the *Pneumatic Linear Robot Drive* (PLRD) is increased so that the robot is able to serve all the added CNC machines. Since the main purpose of the PLRD is to move the robot between the CNC Turning and CNC

Milling Machines in the original system the 2 stop positions at both ends were sufficient. However, in the modeled system apart from the stop positions at both ends, intermediate stop positions are included to the PLRD because of the increased number of CNC machines. Since the traversing speed of the robot is not constant between the two successive stop positions because of the acceleration and deceleration during the starting and stopping, an average constant travel time of 5 seconds assumption of Yücel (2005) is also used in this model. In fact, the travel time of the conveyor (20 seconds) between two stations will be the limiting time when compared to the travel time of the robot (5 seconds).

- The loading and unloading time of the *robot* remained at a constant value of 30 seconds. However, robot is assumed to be available for loading/unloading action at all times. This assumption eliminated the robot job queue. Also the robot reach is enough to reach both CNC Turning and CNC Milling Machines at the stop positions of the robot.

- *AGV* is modeled as a static buffer having unlimited buffer size. Therefore, parts coming to the system or the outgoing parts which have completed all of their operations can be accepted and held in the AGV no matter how many parts have already been waiting in the buffer.

- The layout of the shop-floor is also altered to accommodate for the increase in the number of CNC Machines. Each group of CNCs having Turning or Milling Machines is located in a linear double row layout along opposite sides of the conveyor. The central space of the conveyor is increased so that the PLRD and the robot are located into the space. The change in the location of the robot made it possible to serve both sides. It is assumed that the robot reach is sufficient to serve 2 opposite CNC machines of different types. The static buffer (AGV) is located to the left of the closed loop buffer (conveyor) so that it has 2 meters of conveyor distance to the nearest CNC turning and CNC milling machine.

The final state of the modeled flexible manufacturing cell is demonstrated in Figure 3.2. In the figure 2 CNC Turning and 2 CNC Milling Machines are utilized in the shop-floor. The 2 by 2 arrangement of the CNC machines is given as an example and other arrangements can also be handled by the bidding framework. Results with different number of machines will be presented in Chapter 5.

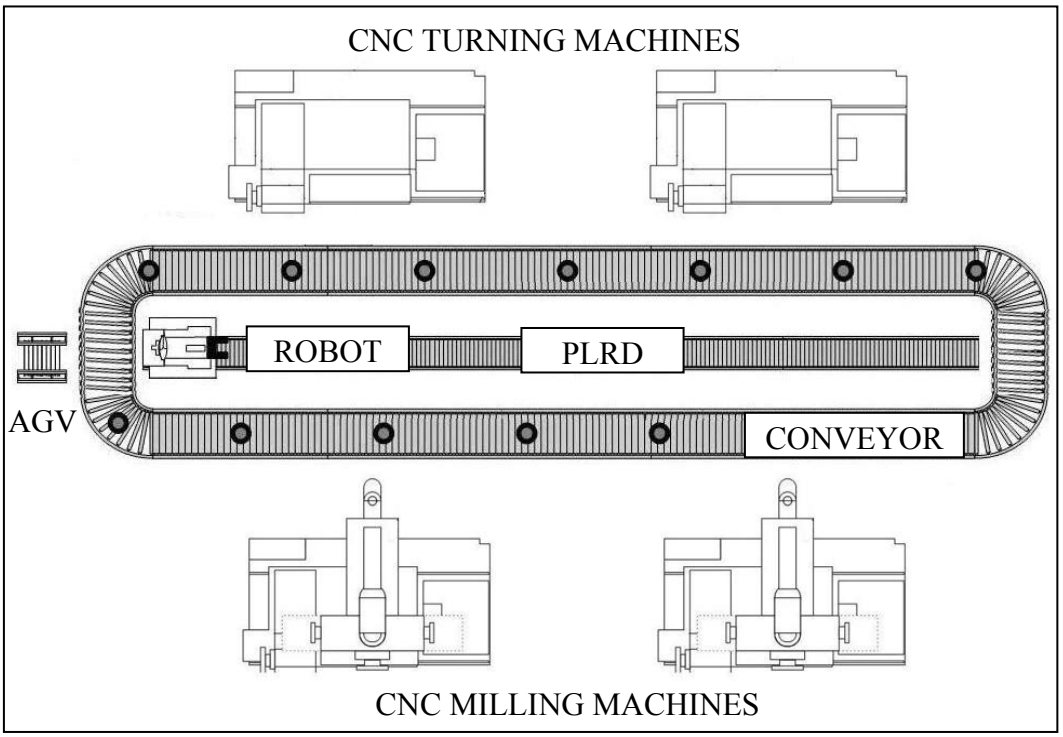

Figure 3.2 Modified Layout with 2 CNC Turning and 2 CNC Milling Machines

3.3 Shop Floor Objectives

The shop-floor objectives in this study are determined as:

1) Due dates dictated by the customers should be met.
2) If all the due dates are not met, the tardy parts should be of low priority.
3) Utilization of the machines in the shop-floor should be balanced.
4) Maximum tardiness value should be under control.

The complete study is customer oriented therefore the primary objective is meeting the due dates. However according to the due date tightness and the shop-floor loading conditions it is not possible to meet all of the due dates and tardiness is unavoidable. In such cases, weighted tardiness is used as the primary objective. Weighted tardiness can be stated as the parts having high priorities should have low tardiness value. As the priority of a part decreases, the tardiness is assumed to be more tolerable. The weighted tardiness objective is dealt by W(SPT+CR) sequencing rule which is introduced in Section 3.4.3.3.

A machine with a high machine loading rate requires a frequent maintenance schedule which increases the frequency of the machine idleness. Also the breakdown probability of such machines is high. Besides, machines with low machine loading imply that the machine stays idle most of the time. Of course, in a shop-floor system, presence of these two extreme conditions simultaneously is unacceptable. Therefore the machine loading among the same type of machines should be balanced as a secondary objective. A way of checking the utilization balance of the machines in the shop-floor environment is achieved by looking the difference between the machine loading percents of the most heavily loaded machine and the idlest machine of the same type. A percent difference within 10% implies a balanced utilization. Balancing the machine load objective is reached by the weight algorithm described in Section 3.4.4.3.

An auxiliary maximum tardiness control objective is important in the shop-floor. Some customer may tolerate small tardiness values. However, if the value exceeds specified tolerance limits, it will cause a decrease in the reputation of the shop-floor and the customers whose parts are delivered with high tardiness values will not work with the shop-floor again. This problem is dealt with the maximum tardiness algorithm described in the bid construction algorithm which will be explained in Section 3.4.3.3. The algorithm is implemented in the sequencing rule and is not a stand alone algorithm.

3.4 Bidding Based Scheduling Model

A shop-floor scheduling problem can be investigated under two sub-categories: job routing and job sequencing. Job routing involves assigning operations of a job to specific machines. The sequence of the incoming jobs to the machines is determined through job sequencing. These two activities constitute the scheduling of the part. In bidding based scheduling, job sequencing and job routing are not independent from each other. At the end of the negotiation between the part and the machines, one of the factors affecting the routing decision is the sequence of the part on the corresponding machine.

Bidding based scheduling can be explained as a mechanism that defines a set of rules for the allocation of the machines for certain parts based on the evaluation of the bids submitted by the parts participated in the bidding process. Therefore, the developed bidding framework is responsible for generating production schedules for incoming jobs on alternative machines according to the processing specifications. The framework is based on a heterarchical architecture having distributed components acting autonomously. These components are the physical shop-floor elements which were explained in Section 3.2. Among those elements the machines and the parts are designed as agents. Overall scheduling of the system is done by using Product Reservation technique. The framework is developed as part initiated (Figure 2.6) and the Contract-net structure (Figure 2.5) is utilized for the negotiation between resources and parts.

In the part initiated contract-net based bidding approach, bidding starts by a part requesting bids from the corresponding machines. This occurs when a new part enters to the system or when a part completes its first operation and needs a machine to process its second operation. Machines check their reservation list and find a suitable place for the incoming part after a sequencing operation. Then every machine participating in the bidding process calculates the time by which it will finish the processing of the part. Time calculation constitutes the bid construction

step. The constructed bid which also includes the machine loading factor is then sent to the part. After collecting all of the bids from the machines, part selects the most suitable bid according to the system objectives. Once the bids are evaluated, the machine having the winning bid is informed and it adds the part to the proposed place in the reservation list. If the bidding process is done for the first operation of an incoming part to the system, the part waits on the AGV for its turn to come. If the bidding process is done for the second process of the part, the part waits its turn on the conveyor.

In this section the modeled bidding framework will be described in details. The necessary scheduling concepts including the primary performance measures of a scheduling system will be introduced in Section 3.4.1. The product reservation scheduling technique and the alternative single step product reservation technique will be described in Section 3.4.2. The bid construction steps, starting from the bid request of the parts and including the sequencing of the incoming part according to W(SPT+CR) rule with the proposed algorithm that controls the maximum tardiness value is explained in Section 3.4.3. In Section 3.4.4 a set of rules including the elimination algorithm and the weight algorithm that are used to evaluate the bids coming to the parts and award the winning machine are given.

3.4.1 Basic Scheduling Terminology

This section describes the primary output measures of a scheduling system. Most of the shop-floor objectives are the derivatives of the stated output measures. Possible inputs used in the system have bee discussed in Section 3.1.

Flow Time: It is the time that a part spends in the system. Average flow time and maximum flow time are commonly used as measures of system performance. Flow time starts with the part introduced into the system on AGV and finishes with the part leaving the system again on AGV.

Completion Time: It is the time at which processing of a specific task is finished and ready to leave the system. Defined as:
Completion Time = Arrival Time + Flow Time

Makespan: It is the completion time of the part which is finished last. It can be also defined as the maximum completion time in a group of parts. Minimizing the makespan is a common objective in sequencing problems.

Lateness: It is the amount of time by which the completion time of a part exceeds its due date. It can be either positive or negative. Lateness can not be expressed for individual operations of a part. Defined as:
Lateness = Completion Time – Due Date

Tardiness: It is the lateness of a job if it fails to meet its due date, zero otherwise. Number of tardy parts, maximum tardiness and average tardiness are generally used as performance measures in the system. Defined as:
Tardiness = MAX(0, Lateness)

Earliness: It is the amount of time that a part is finished before it is due date, zero otherwise. Minimized as an objective generally in Just in Time (JIT) systems when the finished part inventory has storage cost. Defined as
Earliness = MAX(0, -Lateness)

3.4.2 Product Reservation Scheduling

Product reservation is a scheduling technique in which all the operations of a part are allocated in the machines on the shop-floor once the part arrives in the system. However, for the parts with two operations, scheduling both of the operations at the instant of arrival limits the dynamic and reactive nature of the system. An alternate product reservation technique called Single Step Product Reservation (SSPR) solves

the stated shortcoming. Single step refers to the reservation scenario where the part does not complete allocation of all operations to the machines upon arrival, but does its reservations one operation at a time.

In the conventional scheduling systems using the bidding approach a machine should be available in order to participate in a bidding process. When a part arrives in a system it requests bids from all the available machines and then bids are evaluated and the winning bidder is announced. The time passing during the negotiation messages between the part and the available machines is lost and all the available machines stay idle during this time. However, product reservation scheduling technique eliminates this loss. In this technique to submit a bid, it is not necessary for the machine to be available meaning that a machine can receive bid request and submit bids during processing. By this way, the negotiation time is overlapped with the processing time. The only loss because of the negotiation messages occur at the initial condition of the system where all or most of the machines are idle. However as the time passes, the reservation lists of the machines populate and the incoming parts to the system increases. Therefore production reservation approach is mostly efficient when the system comes to a steady state so that the bidding process occurs simultaneously with the processing of the parts that are allocated on the machines beforehand.

Another advantage of product reservation is the ability to reach the objectives more easily. The objectives for the system are described in Section 3.3. The bidding process is not limited to only available machines at the instant when the part requests for bids. Instead, all the machines are included in the negotiation process whether they are occupied or available. This distributes the decision mechanism to all of the machines in the system thus resulting in decisions which are much more objective oriented compared to the decisions of limited number of machines.

A main characteristic of product reservation is that the machines do not have any physical input buffer where the parts are held before processing. Each machine has

a reservation list which does not have a physical correspondence. A machine agent keeps its own reservation list and knows its production sequence by referring to this list. A part which is not associated with any of the machines on the shop-floor for its operation negotiates with the machines to find a suitable route. After the bidding process finishes and the part is informed about the destination machine which will process itself, the part is not dispatched to that specific machine immediately. Instead, it is kept in its current position and added to the reservation list of the machine having the winning bid. Current position of a part changes according to the process plan. The position of a new part that is introduced to the system can only be AGV. The bidding process of the first operation for the new part is done on the AGV and it is also kept on the AGV till its turn comes in the reservation list of the corresponding machine. For a part having two operations, the bidding for the second process occurs at the last second of its processing time in the first machine. After the negotiation concludes, the part is unloaded from the first machine to the conveyor where it will wait its turn for the new destination machine for the second operation.

Unless combined with any other sequencing rule, production reservation technique utilizes the First Come First Served (FCFS) sequencing rule by default to construct bids containing Earliest Finishing Time (EFT) for a part. This means that no changes are made in the order of the reservation list to take the part priorities into account. In this study a different sequencing rule W(SPT+CR) is utilized to sequence the incoming jobs on the machines. The details of the implemented sequencing rule are explained in the following section.

An important shortcoming of the implemented single step product reservation system is, once the route of a part for its single operation is determined and the commitment is made to the destination machine, it is not possible to break the commitment and route the part to a different machine for the same operation. This obviously deteriorates the system performance since not being able to break commitments conflicts the flexibility property of the system.

3.4.3 Bid Construction

3.4.3.1 Bid Request

Constructed bids are the main information that a part uses to make routing decisions. Bid construction step is triggered by the bid request done by the part. Parts can request bids at different times according to their processing plans. An incoming part is introduced into the system by the AGV. The first routing decision of a new part should be done on the AGV before it is dispatched to the shop-floor. Therefore an incoming part requests bids immediately. There is a second instance when a part can request bids from the machines. If a part has finished one of the two operations according to the process plan, it should request bid to find a machine to complete the second operation. The routing decision via bidding process is made at the last second of part processing on the first machine. Of course, depending on the process, bid request is made to machines of corresponding type. CNC Milling machines are not involved in a bid submission process for a part requesting a turning operation and vice versa.

When a machine receives a bid request, it follows certain steps to form a proper bid. A constructed bid mainly includes the earliest finishing time (EFT) of the particular operation of the part along with the machine loading rate (ML) of the machine that is constructing the bid. Therefore calculation of the EFT value and the ML value is the most crucial part of bid construction.

3.4.3.2 Sequencing Rules

In the modeled system, most of the time, the number of parts allocated to the machines exceeds the machine number. This results in a machine queue consisting of the allocated parts. It was stated in Section 3.4.2 that there is no physical machine queue and it is handled as a reservation list. In order to sequence the parts in the

reservation list, a precedence index should be defined for each part. Precedence index determines which part is more important when the shop-floor objectives are considered. This problem is solved by the sequencing rules. A sequencing rule is defined as the criteria by which a machine selects the next part from its reservation list. More than 100 such rules are listed in the literature (Panwalkar and Iskander 1977 and Blackstone et. al. 1982). Panwalker and Iskander (1977) grouped such rules under three main groups:

- Simple Sequencing Rules
- Combination of Simple Sequencing Rules
- Weighted Sequencing Rules

Precedence index in this study is found by W(SPT+CR) sequencing rule. It is a weighted sequencing rule making use of simple and combination of simple sequencing rules. Those rules are:

First-Come First-Served (FCFS): This rule sequences the parts according to the time that they enter the system. This is the most basic sequencing rule and does not take the important information such as part priorities, due dates or processing times into account.

Shortest Processing Time (SPT): This rule sequences the parts in increasing order of their processing times. A part with shorter processing time will have a high precedence index. It is grouped under Simple Sequencing Rules according to Panwalker and Iskander (1977).

Critical Ratio (CR): This rule sequences the parts according to the ratio of remaining time until due date to the remaining processing time of the part. It is grouped under Combination of Simple Sequencing Rules according to Panwalker and Iskander (1977).

3.4.3.3 Bid Construction Algorithm

Upon receiving a bid request from the part, the machine checks its reservation list and sequences the part. The W(SPT+CR) sequencing rule is used which is described as:

$$W(SPT+CR)_{ij} = \frac{w_i}{p_{ij} \times \max\left(1 \ , \ \frac{d_i - t_{now}}{\sum_{q=j}^{m_i} p_{iq}}\right)}$$

where

m_i = Number of operations of part i
w_i = Priority of part i
d_i = Due date of part i
p_{ij} = Processing time of operation j of part i
t_{now} = Current time

Since part sequencing is done only when the part requests a bid, the $\sum_{q=j}^{m_i} p_{iq}$ term in the above equation becomes p_{i1} for a part having a singe operation, $p_{i1} + p_{i2}$ for the first operation of a part requiring two operations and p_{i2} for the second operation for a part requiring two operations.

Kutanoğlu and Sabuncuoğlu (1999) compared various sequencing rules under different conditions, including the recently developed W(SPT+CR) sequencing rule. The statistical tests conducted on the overall results show that the W(SPT+CR) rule is significantly better than the others especially when the weighted tardiness results are compared. This is mainly because of the nature of the sequencing rule which

combines the advantage of CR and SPT rule into one. Also, W(SPT+CR) rule does not need much shop-floor status data which simplifies its implementation. As weighted tardiness is one of the objectives of the shop-floor stated in Section 3.3, utilization of W(SPT+CR) sequencing rule during the bid construction step makes perfect sense.

During the bid construction step, the part requesting a bid is hypothetically inserted to the reservation list and a proposed sequence number is obtained. The proposed sequence implies the place in the reservation list of the machine if the part awards itself to the specific machine and all the calculations during the bid construction step are done according to this proposed queue number. The part can only be inserted permanently on the reservation list of the machine which has given the winning bid after the bid evaluation step.

Once the proposed queue number of the part is found in the reservation list, the machine determines the **Earliest Finishing Time (EFT)** of the given part. Earliest Finishing Time is calculated using the formula below:

EFT = N + Q + T + P

where

N = Remaining time of the current process of the machine
Q = Total processing and transportation time of the parts in the queue
T = Transportation time of the part from its current position to the machine
P = Processing time of the corresponding operation of the part

If the proposed sequence of the part is (n+1) in the reservation list, then the total processing and transportation time of the parts in the queue will be:

$$Q = \sum_{i=1}^{n} (T_i + P_i)$$

It is obvious that the Q term will be zero if the reservation list is empty and the part requested bid is the only part in the reservation list.

When the processing of a part finishes in a machine, the machine calls the next part in its reservation list. The part is not removed from the list until it completes its transportation and loaded on the destination machine. Therefore, if the machine is waiting a part that is on a conveyor, N is equal to zero and Q is calculated by:

$$Q = T_{1_{remaining}} + P_1 + \sum_{i=2}^{n} (T_i + P_i)$$

EFT data are correct for a part at the instant of bid construction. As time passes, there is a possibility that another part is introduced into the system and requests bid. As a result of the bidding, the new part may have a higher precedence index calculated by W(SPT+CR) rule and may enter between other parts that are scheduled on the machine before. This results in a shift in the earliest finishing times of the parts that stay behind of the new coming part in the reservation list. For example, a low priority part can find itself at a sequence number bigger that it was first agreed. This case is shown in Figure 3.3. Figure shows the reservation list of a given machine including same operation of different parts. The production sequence of the parts is in 1 – 2 – 3 – 4. However, the part with identification number 5 which is newly introduced to the system finds a place between the parts 2 and 3 as a result of its bidding process. This leads to the production sequence become 1 – 2 – 5 – 3 – 4 and a shift in finishing times for the parts 3 and 4. For example, during the flow time estimation of part 3 the EFT value which was found before part 5 is introduced will be used. However, the flow time of the part 3 will be higher than the estimated one because of the shift. This causes a deviation between the proposed flow time and the actual flow time of a part when W(SPT+CR) rule is utilized. This deviation

in the proposed flow time is mainly because the Q term in EFT equation is subject to change when W(SPT+CR) rule is utilized and EFT values are only calculated at the instant when a part request bids. Furthermore, the EFT value associated with the part is the one that belongs to the machine that gives the winning bid.

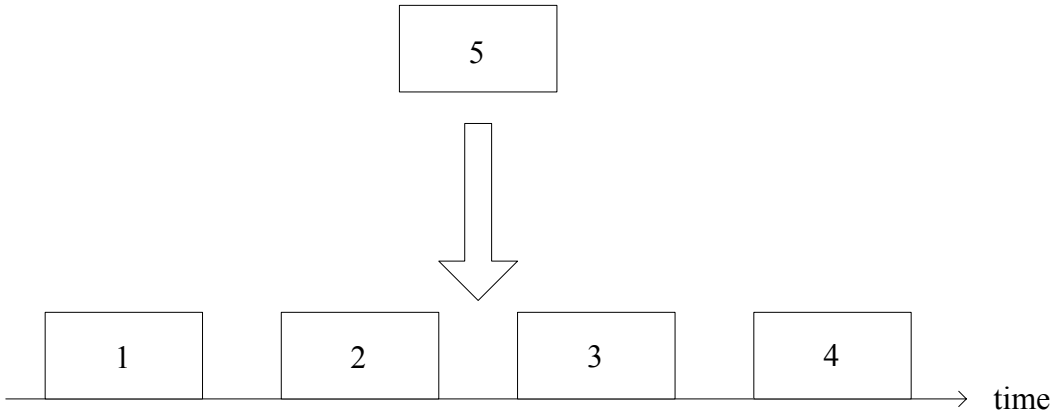

Figure 3.3 Sequencing causing a shift in the schedule

Along with the EFT value, the **machine loading** value is also sent to the part with the constructed bid. The machine loading data will be used in bid evaluation step simply for realizing the shop-floor utilization balance. Machine loading rate (ML) is calculated at every time instant according to the formula below:

$$ML = \frac{\text{total processing time}}{\text{total time}}$$

Generally parts with low priority values tend to have low precedence indexes and mostly sent to the end of the reservation list as the time passes. This results in very high queue time and therefore high tardiness values. In order to prevent this shortcoming **maximum tardiness control algorithm** is implemented in the bid construction algorithm. A maximum tardiness control threshold is defined in the system with unit in minutes. When the tardiness value of a part becomes equal to

the tardiness value stated by the control threshold, maximum tardiness control algorithm is initiated.

The algorithm simply changes the sequencing rule when activated. In a reservation list, if a part having a tardiness value higher than the control threshold exists, no part introduced to the system can overtake the part with high tardiness value even if the new coming part has a higher precedence index. The sequencing rule beyond the part with high tardiness value is again done by finding precedence indexes by W(SPT+CR) rule and sequencing them in the decreasing order.

An important point about the maximum tardiness control algorithm is that the control threshold does not promise to keep the maximum tardiness level under the given value. It only activates the algorithm. This is because the algorithm just avoids new parts overtaking the part with high tardiness value. However, the tardy part still has to wait the processing of the parts that are scheduled in front of it.

3.4.4 Bid Evaluation

3.4.4.1 Bid Collection

After the bid construction is concluded, each machine successively submits the bid to the part. In the structure of the part agent, a list is allocated for the incoming EFT values, another list is for the machine identification numbers and a different one for the machine loading rates. The EFT values are collected in the order of machines. The bids are kept with the machine identification numbers and the machine loading rates of the corresponding machines. The second step in the bid collection is sorting the recorded bids in the order of increasing EFT. Since the smaller EFT values are more desirable, the first machine in the list is the most appealing one to the part. As a result of this sorting, if any evaluation algorithm does not take place the machine which has submitted the smallest EFT value will be awarded by the part.

3.4.4.2 Elimination Algorithm

As the part number to be produced increases, the bidding evaluation step is executed more frequently. Especially to accommodate for the increased part number, the number of machines may also be increased in the shop-floor environment. This causes the number of bids collected to increase. The purpose of the elimination algorithm is to decrease the load on the central processing unit (CPU) of the system hosting the part agent. This is achieved by eliminating some of the submitted bids or by directly awarding a bid with the lowest EFT without considering the other submitted bids.

The main function of the elimination algorithm is checking the obvious difference between the submitted EFT values and if found any, truncating and ignoring the EFT values in the list, from the beginning of the obvious difference to the end of the list. Elimination process results in a simplified and a short EFT list to evaluate.

This algorithm is generally more effective when the importance given to the machine loading is increased. When the decisions are based on the machine loadings, there occurs the possibility of awarding one of the high EFT values just for the sake of balancing the machine loading. For the high EFT values which are truncated, the possibility of being selected by the low values of machine loading is eliminated. In fact, this algorithm favors the primary objective of minimizing tardy parts when compared to the secondary objective of shop-floor utilization balancing.

The elimination algorithm is mainly based on the percent calculations. First the percent difference between the maximum and minimum EFT values is found and if this value is greater than a threshold value, a percent list is generated in which the percent differences between the successive EFT values are kept. It can be understood that, the percent list length is less than the EFT list length by one. The percent list is checked starting from its first element and if a value greater than a certain elimination limit is found, corresponding EFT value causing the high

percent difference and the EFT values which are greater than that value are eliminated. A simplified flow chart of the elimination is shown in Figure 3.4. The parameters used in the elimination algorithm are:

Elimination Threshold (ET): This value determines the possible need for the elimination algorithm for a given set of EFT values. The value represents the threshold after which the elimination algorithm is activated. It is defined as the maximum allowable value of the percent difference between the maximum EFT value and the minimum EFT. The EFT data are eligible for elimination if:

$$\text{Elimination Treshold} \leq \frac{EFT_{max} - EFT_{min}}{EFT_{min}} \times 100$$

The elimination of EFT data is not guaranteed even if the percent difference is greater than the elimination threshold.

Elimination Limit (EL): This value determines whether the high percent difference between the maximum and minimum EFT values is distributed evenly or localized between any two successive EFT values. If latter is the case, the EFT values after the localized value are eliminated and do not taken into account for the further bid evaluation calculations. If after the elimination done, only one EFT value remains, then the corresponding machine is directly awarded without making any further evaluations.

3.4.4.3 Weight Algorithm

The purpose of the weight algorithm is to involve the machine loading factors in the bid evaluation step. The algorithm consists of factors to adjust the weight of either EFT or ML during the evaluation. Using these weights a modified EFT (mEFT) is obtained and the part is awarded to the machine having the minimum mEFT value.

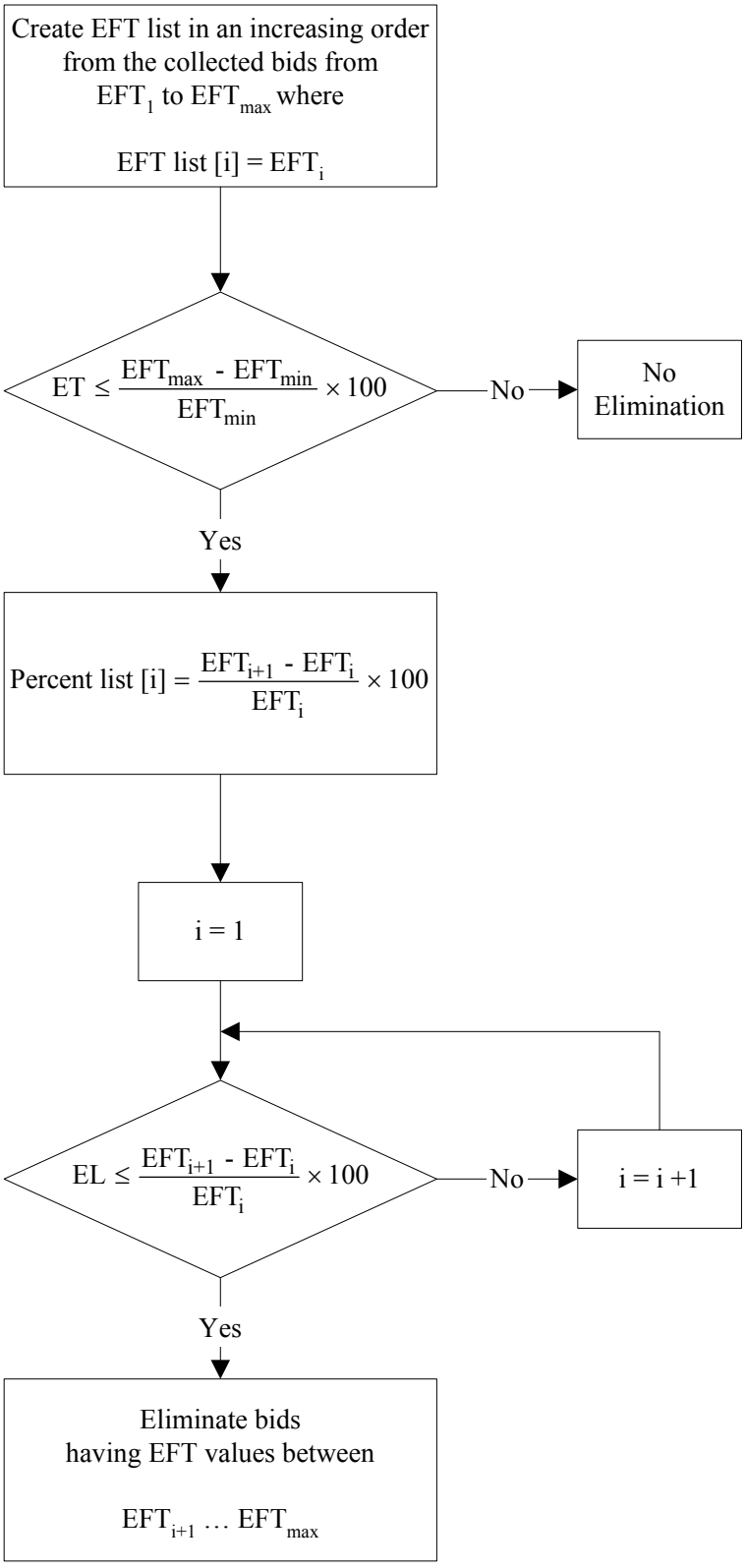

Figure 3.4 Flow chart of elimination algorithm

Modified EFT value of the i^{th} machine can be calculated using the following formula:

$$mEFT_i = \frac{w_{ML} \cdot ML_i + (100 - w_{ML})}{100} \times \frac{w_{EFT} \cdot EFT_i + (100 - w_{EFT})}{100}$$

where

ML_i = Machine loading rate of machine i

EFT_i = Proposed earliest finishing time value of machine i

w_{ML} = Weight of machine loading rate and $0 \leq w_{ML} \leq 100$

w_{EFT} = Weight of the earliest finishing time value and $0 \leq w_{EFT} \leq 100$

In the above formula weights of machine loading and earliest finishing time are independent. However, these two weights can be made dependent on each other so that by increasing the weight of machine loading, the weight of earliest finishing time is reduced. The complementary effects of the weights can be achieved by simply using the formula below:

$$w_{ML} + w_{EFT} = 100$$

Investigating the extreme cases, if w_{ML} is equal to 100, then $mEFT_i$ equation will be reduced to ML_i term only, meaning that the decisions will be purely made according to the machine loading data of the corresponding machine. If w_{ML} is equal to 0, then mEFT equation will only include EFT_i term. This implies that the decisions will be made according to the proposed earliest finishing time values of the machines.

The main modified EFT formula is independent of part priorities. However in this study, balancing of shop-floor utilization is generally tried to be achieved by parts of low priority. So for those parts, smaller weights of earliest finishing time (w_{EFT})

and higher weights of machine loading rate (w_{ML}) are used. Of course, the opposite is true for the parts with higher priorities. It is a useless and tiring effort to specify different machine loading and earliest finishing time weights for each 10 type of priority. So only the weights of the lowest priority 1 and the highest priority 10 are defined in the system. Weight of the parts having priorities in between are linearly interpolated The priority dependence is simply achieved by defining $mEFT_{i,j}$ meaning the modified earliest finishing time of machine i which is submitted to a part of priority j. Therefore:

$$mEFT_{i,1} = mEFT(w_{ML,1}, w_{EFT,1})$$

$$mEFT_{i,10} = mEFT(w_{ML,10}, w_{EFT,10})$$

and the modified earliest finishing time value is calculated as:

$$mEFT_i = \frac{(mEFT_{i,10} - mEFT_{i,1}) \times (\text{part priority} - 1)}{9} + mEFT_{i,1}$$

Following the elimination step, the rest of the EFT values of the corresponding machines are modified. After all mEFT values are obtained for a part in a mEFT list that is stored again in the machine agent, the values are ordered in the increasing order. The machine which has the lowest mEFT value is declared to be the winner with the best bid.

3.4.4.4 Awarding

Bid evaluation step is concluded with the awarding of the winning machine. As a result of the weight algorithm, a winning machine is determined; however the machine should also be informed about the result. The information about the part that the machine won to process is sent to the machine. The machine accepts the

part by sequencing it in the reservation list as proposed by the W(SPT+CR) sequencing rule explained in Section 3.4.3.3.

3.4.4.5 Task Commitment

Task commitment is related to the part agent. The resulting machine as a result of the bid evaluation should also be associated with the part agent so that when its turn comes in the machine it will know its destination resource. Also the winning earliest value is stored in the part agent in order to calculate the proposed flow time in the system. However, if W(SPT+CR) rule is used, it was stated in Section 3.4.3.3 that there will be a deviation between the proposed and real flow time of the part.

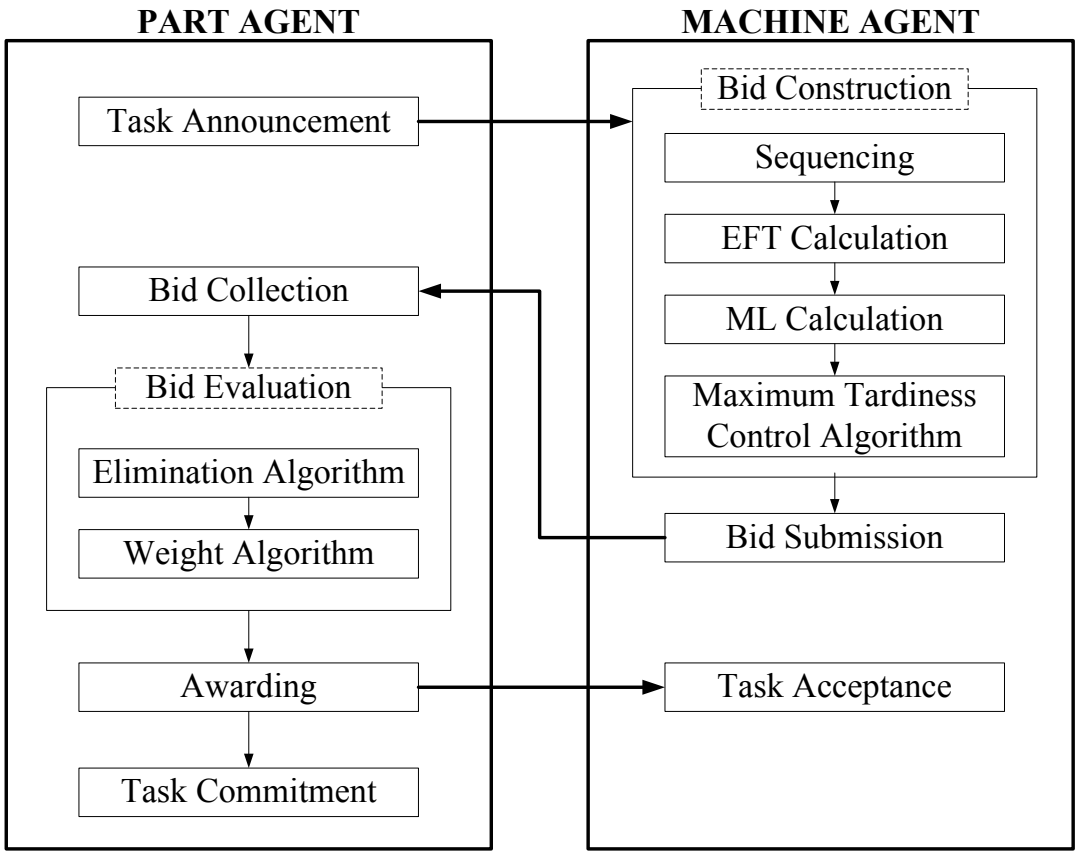

Figure 3.5 Steps in the contract-net based scheduling model

CHAPTER 4

SYSTEM STRUCTURE

In this chapter, structure of the virtual system which is constructed in C# programming language to test the developed bidding framework is explained. Simulation structure section contains the simulation components along with a brief terminology. It also involves the efforts for simulation modeling. Second section of the chapter describes the agent structure used in the system. Attributes and events of the part and machine agents used in the system are presented in this section.

4.1 Simulation Structure

4.1.1 Simulation Structure Components

All of the simulation structures have some basic components no matter how advanced or complex the simulated system is. The structural components of a simulation include entities, resources, attributes, global variables, events and statistics (Kelton et. al. 2004 and Yücel 2005). These components and their correspondences in the developed system are given in the following sections.

4.1.1.1 Entities

Entities are the dynamic objects in a simulation that move around, change the status of the system and affect the output performance measures. In a simulation structure entities are created, move in the simulation to fulfill their objectives and are

disposed when they leave. Most of the entities in a simulation have real equivalents in the system. There can also be different kinds of same type of entities.

Entities used in this simulation structure are the parts entering to the system to be processed. They are created upon arrival, queued in the machine reservation lists according to their process plans, dispatched to the machines to complete their processes and terminated when they are finished and need to be delivered to the customer. There exist several kinds of part entities having different processing plans and different attributes. These were introduced in Section 3.1. Since each part entity will make its own dispatching decisions according to the bid evaluation results they need to be autonomous and have communication capability with machines in the system. Therefore, in the simulation structure, the part entities are modeled as autonomous agents. Description of the part agents with their attributes and events are explained in Section 4.2.1.

4.1.1.2 Resources

Resources mainly provide different types of services to entities in the simulation structure and the entities generally compete with each other for the given service. An entity seizes a resource for a definite or indefinite time during the simulation and releases it when the required operations are finished. Resources in the simulation structure have real equivalents in the system such as personnel, machine or a storage area of limited capacity.

The main resources in the developed system are the CNC turning and milling machines. Several machine resources can be introduced to the simulation structure in order to create the competitive environment for bidding process in the shop-floor. They can only allocate one entity at a specific time instant and they are assumed to work without any breakdowns. After participating in the communication protocol during the bidding process, winning CNC machine resource is seized by the part

entity for a specific processing time period. During the bidding process, each CNC machine calculates the necessary components and submits bid to a part upon receiving a bid request from that part agent. Each machine will have different attributes and therefore will submit different bids. Also each machine needs some events to properly communicate with part agents. Therefore in order to implement autonomy in the CNC machine resources, they are modeled as agents in the simulation structure. Description of the CNC machine agents with their attributes and events are explained in Section 4.2.2.

Other auxiliary resources in the system are the robot, conveyor and the automated guided vehicle (AGV) which is modeled as a static buffer. Although modeled as static, the AGV introduces the incoming parts to the system and delivers the finished parts out of the system. Conveyor is mainly responsible for transportation of the parts in the simulation structure. A part entity seizes a place (a cup) on the conveyor during transportation and releases the place when it reaches its final destination. Part entities need to be transferred to the conveyor upon arrival in the AGV or upon finishing its process in a machine. Part entities also need to be transferred from the conveyor to AGV when all of its processes are finished or to a machine when the machine will start the processing of the part. Robot in the structure simply simulates these loading and unloading actions.

Detailed information about all the resources is provided in Section 3.2 with their technical specifications, modifications and assumptions done for the simulation structure.

4.1.1.3 Attributes

An attribute is a common characteristic of the entities and resources whose value can differ from one entity or resource to another. Attributes are used to individualize entities or resources to create different kinds. Distinct attributes values

are tied to specific entities and resources and most of the time, the same attribute will have different value for different entities or resources. These values can be assigned at the beginning of a simulation run or they can be created during the run. It is a modeling effort to determine the needed attributes, naming them, assigning values to them, changing those values and recalling them during the simulation run.

The entities and the resources created in the simulation structure have various types of attributes. These attributes may stay fixed during the run or may change as the simulation run progresses. Attributes such as due dates and time of arrivals of parts do not change all through the run and they will be referred as *fixed attributes*. However, the location or the queue time of a part changes as the simulation advances. These types of attributes are referred as *variable attributes*. Attributes which are unknown at the beginning of the run but found at the end of the simulation run such as completion time of a part are also considered as variable attributes. So, type or number of a machine is a fixed attribute whereas, the loading rate of the same machine will be a variable attribute. The complete list of attributes of the part and machine agents taking part in the simulation structure will be given in Section 4.2.1.1 and Section 4.2.2.1.

As stated in Section 3.2, AGV which is modeled as static buffer does not have a limited buffer size so there is not an attribute associated with AGV. The only attribute for the robot is the loading and unloading times. These are taken to be same and constant all throughout the simulation run. The loading and unloading value is 30s by default but can be changed at the initialization of the simulation run to create a constant delay. Conveyor has two attributes, namely the time of travel between two successive stations and cup number. Time of travel between successive stations is taken to be 20s by default and can be changed at the initialization of the simulation run. However, cup number of the conveyor is fixed and calculated according to the formula given in Section 3.2. The cup number of the conveyor is so arranged that it increases with increasing machine number in the shop-floor.

As stated, the attributes of the conveyor and robot can be changed during initialization of the run but are fixed all through the run. A different kind of robot and a different conveyor can be obtained by changing the attributes during initialization of simulation run. If the system contains more than one robot and one conveyor, the robot loading/unloading time and travel time of conveyor between two successive stations will definitely be attributes. However, since there are only one robot and one conveyor in the system with fixed attributes, their attributes can also be treated as global variables.

4.1.1.4 Global Variables

A global variable is a piece of information which reflects some characteristic of the simulated system. Global variables are unique values which are independent of the number or kind of entities or resources. The main difference of the global variables from the attributes is variables are values related to the whole system whereas attributes are specifically tied to the entities or resources. The global variables are accessible by entities and resources and some of them can be changed by entities.

Global variables can be used to represent different information. They can be set to an appropriate value at the initialization of the simulation run so that they can be recalled whenever necessary. These kinds of variables stay unchanged throughout the run. If those variables are needed to be changed, this should be done before the simulation run is executed. Number of CNC machines in the shop-floor is such a variable that can not be changed during the simulation run. Global variables can also be used to represent information that changes throughout the simulation run such as the current time of the simulation.

Table 4.1 shows the global variables used in the developed simulation structure with brief descriptions. These variables are important ones that affect the system outputs or that are indispensable components of the simulation structure.

Table 4.1 Global variables and their descriptions

Variable Name	Description
Smallest_id	The part with the smallest part number in the system at a particular instant. It is generally used as the lower limit in looping structures checking the situations of all the parts in the system.
p	The part with the biggest part number in the system at a particular instant. It can also be defined as the part that has arrived last. It is generally used for as the upper limit in looping structures checking the situations of all parts in the system.
rem_parts	Number of remaining parts that are not finished and delivered. It is mainly used to stop the simulation run when its value equals to zero.
mach_count	Number of CNC machines in the system. The first half of the number represents the CNC turning machines and the other half represents the CNC milling machines. Used for determining the type of the machine and as a limit for the looping structures checking the situations of all machines in the system.
t	Current time. It is the main element of the simulation structure. Incremented by 1 after checking all of the structure components and executing necessary events.
elim_thres	Elimination threshold.
elim_limit	Elimination limit.
control_thres	Maximum tardiness control threshold.
w_1ml	Weight of machine loading rate for priority 1.
w_1eft	Weight of the earliest finishing time for priority 1.
w_10ml	Weight of machine loading rate for priority 1.
w_10eft	Weight of the earliest finishing time for priority 10.

4.1.1.5 Events

An event is something that happens at a particular instant of time affecting the attributes of entities and resources, global variables or statistical accumulators and causing a change in the state of the system.

Events in the simulated system are mainly triggered because of the dispatching decisions made cooperatively by machine and part agents. find_EFT event for the machine agent where the EFT value for the bid is calculated or collect_bids event for the part agent where the bids of the machine agents are collected and sorted in increasing order are examples of such events. Events related to the agents are given in Section 4.2.1.2 and Section 4.2.2.2 for part and machine agents respectively. Events can also be triggered as a result of the simulation activities such as adding a part to the finished list of a machine (add_to_finished_queue) when the processing time of the part is completed. Such events are explained in Section 4.1.2.

4.1.1.6 Statistics

Performance of the simulated system can be evaluated by collecting statistical data. Statistical values do not involve in the decision making process of the simulation structure. They only show the values or changes in different system parameters. There are three different groups of statistics: tally, time-persistent and counter

Tally statistics are created from taking the average, maximum or minimum of a list of numbers. Average tardiness value for a simulation run is a tally statistic since it involves the average of tardiness values of all tardy parts. *Time-persistent statistics* are created by taking the average, minimum or maximum of a plot of a parameter in the simulation structure. Time-persistent averages involve the accumulated area under the plotted curve where the x-axis is continuous time. Average number of parts on the conveyor is a time-persistent statistic since the number of parts on the

conveyor at every instant of time is averaged over the total time. *Counters*, as the name implies, are the summations of the occurrence of an event during the simulation run. Number of times a CNC milling machine is used by the parts is a simple counter example.

The statistical data collected in the developed simulation structure is classified as Part, Resource, Objective-Based and Algorithm Statistics. These statistics are given in Table 4.2, Table 4.3, Table 4.4 and Table 4.5 respectively with their brief definitions. Tally statistics are represented with T, time-persistent statistics are represented by P and counters are represented by C.

Table 4.2 Part Statistics

Name	Description	
type_no(1..4)	Number of parts of each type	C
type_tardy_no(1..4)	Number of tardy parts of each type	C
type_early_no(1..4)	Number of early parts of each type	C
type_avetur(1..4)	Average turning time of each type	T
type_avemil(1..4)	Average milling time of each type	T
type_avemach(1..4)	Average machining time of each type	T
pri_no(1..10)	Number of parts of each priority	C
pri_tardy_no(1..10)	Number of tardy parts of each priority	C
pri_early_no(1..10)	Number of early parts of each priority	C
pri_avetardy(1..10)	Average tardiness value of each priority	T
pri_avelate(1..10)	Average lateness value of each priority	T
pri_aveflow(1.10)	Average flow time of each priority	T
wip	Number of parts in the system at every time instant	P

Table 4.3 Resource Statistics

Name	Description	
conv_no	Number of parts on the conveyor at every time instant	P
conv_aveno	Average number of parts on the conveyor	P
conv_maxno	Maximum number of parts on the conveyor	P
conv_util	Instantaneous utilization of the conveyor	P
conv_aveutil	Average utilization of the conveyor	P
AGV_no	Number of parts on the AGV at every time instant	P
AGV_aveno	Average number of parts on the AGV	P
AGV_maxno	Maximum number of parts on the AGV	P
turning_totalno	Scheduled times of turning machines	C
milling_totalno	Scheduled times of milling machines	C
turning_no(1..n)	Scheduled times of each turning machine	C
milling_no(1..n)	Scheduled times of each milling machine	C
activemach_no	Number of occupied machines at every time instant	P

Table 4.4 Objective-based Statistics

Name	Description	
aveflow	Average flow time of parts	T
maxflow	Maximum flow time of parts	T
avecomp	Average completion time of parts	T
maxcomp	Maximum completion time of parts	T
avelate	Average lateness of parts	T
maxlate	Maximum lateness of parts	T

Table 4.4 (continued) Objective-based Statistics

avetardy	Average tardiness of parts	T
maxtardy	Maximum tardiness of parts	T
tardy_no	Number of tardy parts	C
aveearly	Average earliness of parts	T
maxearly	Maximum earliness of parts	T

Table 4.5 Algorithm Statistics

Name	Description	
elim_elig_tur	Number of set of bids eligible for elimination for turning operations	C
elim_elig_mil	Number of set of bids eligible for elimination for milling operations	C
elim_done_tur	Number of eliminations done for turning operations	C
elim_elig_mil	Number of eliminations done for milling operations	C
elim_award_tur	Number of bids directly awarded after elimination for turning operations	C
elim_award_mil	Number of bids directly awarded after elimination for milling operations	C
w_pri_turmin(1..10)	Number of winning bids having minimum EFT values for turning operations	C
w_pri_milmin(1..10)	Number of winning bids having minimum EFT values for milling operations	C
w_pri_totmin(1..10)	Number of winning bids having minimum EFT values	C
pri_tur_no(1..10)	Number of turning operations of each priority	C
pri_mil_no(1..10)	Number of milling operations of each priority	C
pri_total_no(1..10)	Total number of operations of each priority	C

4.1.2 Simulation Modeling

Simulation structure involves different kinds of entities, various resources offering service, attributes of these entities and resources, many distinct events and statistics that reflect the performance of the simulation. Simulation modeling section mainly deals with determining the relations between the simulation structure components by integrating the time concept.

The changes in the system are caused by events and most of the time these events are triggered by certain values of some particular attributes. These particular attributes are the *status* and the *remainingtime* of a part agent. Status indicates the place of a part in the system and remaining time is a finite or indefinite delay introduced to the part as a result of waiting in the queue or utilizing a service provided by a particular resource. The idea of the simulation model is checking the status and the remaining time of each part at every time instant and realizing some events at proper times. In fact, these events in turn, alter the status and the remaining time along with other attributes and sometimes trigger other events causing a change in the state of the system. By this way, the continuity of the simulation is maintained until there are no parts coming to the system in order to be processed and there are no parts left uncompleted in the system.

The following sections include the simulation modeling efforts for parts and machines. The main frame of the simulation will be explained without mentioning how and where the statistical data are collected.

4.1.2.1 Part Model

As mentioned above, the main reason of the events in the simulation structure is the status and the remaining time attributes of the parts. The simulation model for the parts simply checks the remaining time of the parts that are in the system at every

instant. Remaining time values of a part can be indefinite or finite. Remaining time values are indefinite when a part is waiting in a queue in the reservation list of a machine. This is due to the fact that any other part having higher precedence can overtake the part changing the estimated waiting time. In such a case, the part can be on AGV waiting for its first operation to be done or on the conveyor waiting its second operation to be done. Remaining time values of parts take finite values during transportation between two specific stations or when being processed by a machine.

A part is introduced to the system when the time of arrival of a part becomes equal to the current time of the simulation. Upon arrival, all the parts are located on the AGV. The first step for the incoming part is determining the destination machine via the bidding process. The winning machine of the bidding process is assigned to the attribute *dest_mach_no* of the part. The model of the bidding process is explained in Section 3.4 and the bidding events executed by the machine and part agents are given in Section 4.2.1.2 and Section 4.2.2.2. At the end of the bidding process, the remaining time of the part is set to be indefinite (*remainingtime* = -1) since the part will wait on the AGV until it is called by the destination machine when turn of the part comes.

According to the remaining time values of parts at a particular instant three actions can be done. For the parts with *finite remaining time* values other than zero, no event is done and the remaining time values are decremented by 1. Parts having *indefinite remaining time* values are handled by the machine model which is described in Section 4.1.2.2. Since the parts with indefinite remaining times imply waiting in the reservation lists, the machines simply check their situations and if they are empty and there are parts in their reservation lists, first part in the list is called to the machine. Finite remaining time values are assigned to such parts which are the transportation times calculated from the current position of the part to the machine. Parts with *zero remaining time* values constitute the most important part of the part model. Remaining time being equal to zero implies that the part has

come to the end of its certain task and some action should be taken according to the status of the part. The status of a part can be *on machine*, *on AGV*, *on conveyor* or *finished*. Since the robot loading and unloading time is taken as constant value, it is simply added to the transportation time of the part from one machine to another and there is not a status as on robot.

If a part is on machine when its remaining time is equal to zero, it means that the part has finished its processing on that particular machine. First of all, the part is added to the history list of the machine containing the identification numbers of the parts that the machine has processed before. This is achieved by event *add_to_finished_queue*. The machine is declared to be empty by setting the value of *isidle* attribute to true. As stated, the zero remaining value of a part on machine implies that the processing of the part has finished. Therefore the part is set to be milled or turned according to the current machine of the part. This is done by assigning true value to the part attributes *ismilled* or *isturned*. The status of the part should not be *on machine* any more so the new status value should be assigned to the part. Of course, the next status of the part will be *on conveyor* which means that the part will start to be unloaded to the conveyor. Being on the conveyor, the destination machine of the part should be determined so that the remaining time can also be set. If all the operations of the part is completed after the recent processing (if *isdone* = true), the part should be sent to the AGV (*dest_mach_no* = -1) in order to be delivered to the customer. Therefore the remaining time of the part is set as the transportation time from the recently utilized machine to the AGV. If the part needs a second operation after its first operation then the bidding process is executed in order to determine the new destination machine of the part. The found machine is assigned to the attribute *dest_mach_no* of the part. At the end of the bidding process, the remaining time of the part is set to be indefinite (*remainingtime* = -1) since the part will wait on the conveyor until it is called by the destination machine when turn of the part comes. This is the case even if the part is not going to wait in the reservation list of an idle machine and will directly be called by the destination machine.

If a part is on conveyor when its remaining time is equal to zero, it means that the part has reached its destination resource. This resource can be either one of the machines or AGV. If a part reaches AGV after transportation, it means that the part is completed and ready to be sent out of the system. Therefore the new status of the part will be AGV where it will leave the system. Remaining time of the part is set to be zero again, so that in the next time instant, it will be on the AGV waiting to be disposed. If the part reaches to one of the CNC machines, it means that the new status of the part will be *on machine.* The new remaining time of the part will be either its turning time or milling time according to the machine type. The remaining time is obtained by the part event *get_process_time*. The part is removed from the reservation list of the machine since it is now on the machine by the event *remove_from_reslist*. The machine is declared to be occupied by assigning false to the attribute *isidle*. Since the part is arrived, *iswaiting* attribute is set to be false meaning that there is not a part coming on the conveyor to the machine.

If a part is on AGV when its remaining time is equal to zero, it means that the part has completed all of its operations and ready to be delivered to the customer. In this case the part is declared to be finished by setting the status as *finished*. There is no valid remaining time for the part since it will no longer be in the system. The remaining parts (*rem_parts*) are decremented by 1.

After checking all of the parts and machine situations, the current time value is incremented by 1 so that the new time instant is simulated and the same procedure is repeated.

4.1.2.2 Machine Model

The main role of the machine model in the simulation structure is calling the parts having indefinite remaining time values because of waiting in the reservation list. The part that is first in the reservation list at the instant when the machine becomes

idle is called. Evidently that will be the part having the highest precedence in the reservation list belonging to that specific machine.

The procedure is realized in a looping structure for all CNC machines on the shop-floor at any time instant. If any machine is available at the particular time instant (*isidle* = true), not waiting a part to be transferred by the conveyor (*iswaiting* = false) and the reservation list of the machine is not empty (*islistempty* = false) it retrieves the first part in its reservation list. This is done by the event *readpop.* Since that part will be processed by the machine, it should be transferred to the machine by the conveyor. Since the machine is now allocated to the specific machine the attribute *isidle* is set to be false. The machine is waiting for the part to come on the conveyor so the *iswaiting* attribute is set to be true. Now the part should be assigned a remaining time. In the machine model, parts with indefinite remaining times are called. Therefore the part that will be transported on the conveyor can be on AGV waiting for its first process to be done or on conveyor waiting for its second process. If it is *on AGV*, the status of the part is set to be *on conveyor* and the remaining time will be assigned as the transportation time of the part from the AGV to the current machine. If the status of the part is *on conveyor*, then the remaining time is assigned as the transportation time of the current position of the part on the conveyor to the current machine.

On the other hand, during the looping structure, if the machine turns out to be occupied (*isidle* = false) and being occupied is not because of a part that is on the way to the machine (*iswaiting* = false) then it means that there is a part that is currently being processed on the machine. In such a case no action is taken, only the machine busy time (*total_working_time*) which is used to calculate the machine loading rate is incremented by 1.

After checking all of the parts and machine situations, the current time value is incremented by 1 so that the new time instant is simulated and the same procedure is repeated.

4.2 Agent Structure

4.2.1 Part Agent

Part agents are the entities of the simulation structure. They are also the main agents taking part in the negotiation protocol of the bidding process. In this section, fixed and variable attributes and the events utilized to model this agent type will be described.

4.2.1.1 Attributes

Attributes of the part agents are used to individualize each part agent and to state the condition of a part entity at any instant during the simulation run. Part agents have different attributes which are classified as fixed and variable. Definitions of fixed and variable attributes are given in Section 4.1.1.3.

no: A fixed attribute which is used to represent the part number described in Section 3.1. It can have values starting from 0 for the first part and the upper limit is less than the identification number of the last part by one dictated by the input file.

type: A fixed attribute which is used to represent the process plan of corresponding part. It can have values between 1 and 4 each standing for different process plans described in Section 3.1.

priority: A fixed attribute which is used to represent the importance of a part over others. In Section 3.1 the priority of a part is modeled to have values between 1 and 10, where 10 corresponds to the most important part.

time_of_arrival: A fixed attribute which is used to represent the time when a part arrives in the system. Model used to create these values is described in Section 3.1.

due_date: A fixed attribute which is used to represent the time by which the part should be finished. Values of *due_date* are calculated using the machining time and the time of arrival value of the part. Details are given in section 3.1.

turning_time: A fixed attribute which is used to represent the estimated time for the turning operation of the part. Turning time of a specific part is fixed for all CNC turning machines since the machines are assumed to be similar. Model used to create turning time values is described in Section 3.1.

milling_time: A fixed attribute which is used to represent the estimated time for the milling operation of the part. Milling time of a specific part is fixed for all CNC milling machines since the machines are assumed to be similar. Model used to create milling time values is described in Section 3.1.

machining_time: A fixed attribute which is used to represent the total processing time of a part in the CNC machines in the shop-floor. It is calculated by simply adding values of *turning_time* and *milling_time* attributes.

isdone: A variable attribute which is used to represent whether all of the machining operations of a part is finished or not. It is a Boolean type attribute and should have a value either true or false. *isdone* attribute for the parts requiring only single operation becomes true when the single operation is finished. However, for parts requiring two operations, the value is true when both of the operations are finished.

isturned: A variable attribute which is used to represent whether the turning operation of a part is finished or not. It is a Boolean type attribute and should have a value either true or false.

ismilled: A variable attribute which is used to represent whether the milling operation of a part is finished or not. It is a Boolean type attribute and should have a value either true or false.

dest_mach_no: A variable attribute which is used to represent the target resource of the part. It can have integer values between -1 and *machcount*.

dest_mach_no = -1 $\Rightarrow$ AGV

$0 \leq$ *dest_mach_no* < *machcount* /2 $\Rightarrow$ CNC Turning Machines

machcount /2 $\leq$ *dest_mach no* < *machcount* $\Rightarrow$ CNC Milling Machines

Except for the value -1 meaning that the part should go to AGV, all other values are determined through the bidding process.

status: A variable attribute which is used to represent the location or the state of a part. It may have four different values: *on_machine*, *on_conveyor*, *on_AGV* or *finished*.

remainingtime: A variable attribute which is used to represent either a finite delay because of a part using the service provided by a particular resource or an indefinite delay which is due to waiting of a part in the reservation list of a machine. In fact, the nature of the *remainingtime* value is dependent on the *status* attribute. A part having a finite delay is either being processed by a machine (*on_machine*) or being transported by the conveyor (*on_conveyor*). On the other hand a part having an indefinite delay value is waiting in the reservation list of a machine either *on_conveyor* or *on_AGV*.

time_queue: A variable attribute which is used to represent the total waiting time of a part in queues. The value starts with zero and incremented by 1 for each second of a part waiting in a queue.

time_transport: A variable attribute which is used to represent the total transportation time of a part in queues. It may include components because of the following transportation actions of the conveyor:

Transportation of a part from AGV to a CNC machine

Transportation of a part from one type of CNC machine to another

Transportation of a part from a CNC machine to AGV

Since the robot is assumed to be always available and has a constant service time, robot loading and unloading times are also added as constants to the transportation time values above.

time_flow: A variable attribute which is used to represent the time that a specific part spends in the system. The value is set to zero when a part arrives and incremented by 1 for each second that the part spends in the system. The value of the flow time can also be verified by adding *time_queue*, *time_transport* and *machining_time* value of a part.

time_completion: A variable attribute which is used to represent the time that a specific part is completed and ready to be delivered to the customer. This attribute has a value of zero all through the life of a specific part and the correct time value is assigned to *time_completion* attribute when the part arrives back at the AGV. The value can also be verified by adding *time_flow* value to the *time_of_arrival* value of the part.

earliness: A fixed attribute which is used to represent the earliness value of a part. The earliness value of a part is obtained by the following equation at the end of the simulation run for each part.
max[0, due_date - *time_completion*]

tardiness: A fixed attribute which is used to represent the tardiness value of a part. The tardiness value is obtained by the following equation at the end of the simulation run for each part.
max[0, *time_completion – due_date*]

eft_tur_win: A fixed attribute which is used to represent the earliest finishing time (EFT) value of the CNC turning machine having the winning bid. The attribute is created and the winning EFT value is assigned at the end of the bidding process for a part.

eft_mil_win: A fixed attribute which is used to represent the earliest finishing time (EFT) value of the CNC milling machine having the winning bid. The attribute is created and the winning EFT value is assigned at the end of the bidding process for a part.

time_proposed_flow: A fixed attribute which is used to represent the proposed flow time. Proposed flow time is the estimated flow time for a part which is equal to the *time_flow* if no part overtakes the current part before it is produced. For part types 1 and 4 time_propoesd_flow is calculated by adding *eft_tur_win*, *eft_mil_win* and the transportation time between the last *dest_mach_no* (before AGV) and AGV. For part type 2 it is calculated by adding *eft_mil_win* and the transportation time between the last *dest_mach_no* (before AGV) and AGV. For part type 3 it is calculated by adding *eft_tur_win* and the transportation time between the last *dest_mach_no* (before AGV) and AGV.

deviation: A fixed attribute which is used to represent the deviation of the flow time from the proposed flow time. Detailed explanation about the deviation of the proposed flow time is given in Section 3.4.3.3.

4.2.1.2 Events

Most of the events of part agents stem from the cooperation with the machine agent in the contract-net protocol. As a result of the part events, the attributes of part agent and the machine agent can be changed or other events can be triggered.

get_process_time: An event which is used to retrieve the processing time of the part according to the *dest_mach_no*. It is used for assigning remaining time value to the part when the part arrives in the machine by means of the conveyor. Main reason of defining such an event is that it simplifies the coding efforts eliminating the need for recursive conditional statements.

bid_request: An event which is used to trigger the bid construction step of the bidding process. The *bid_request* event occurs at the instant when a new part arrives on AGV or at the final second (when remainingtime=0 and status = *on manchine*) of the first operation of a part which is requiring a second operation.

collect_bids: An event which is triggered by the *find_EFT* event of the machine agents constructing the bids. The bids are retrieved form the machine agents and they are sorted in the increasing order of EFTs in a dummy list in the part agent. The event triggers either of the *evaluate_milling_bids* or *evaluate_milling_bids* events according to the part type.

evaluate_turning_bids: An event which is triggered by the *collect_bids* event of a part agent seeking a proper CNC turning machine to be dispatched. It corresponds to the bid evaluation step of the bidding process. The EFT values in the dummy list that is created in the *collect_bids* event are modified by considering the weights coming from the weight algorithm. Details of the weight algorithm are given in Section 3.4.4.3. The part is awarded to the machine having the lowest modified EFT value. A dummy variable having an integer value representing the winning machine number (*no*) is the output of the event.

evaluate_milling_bids: An event which is triggered by the *collect_bids* event of a part agent seeking a proper CNC milling machine to be dispatched. It corresponds to the bid evaluation step of the bidding process. The EFT values in the dummy list that is created in the *collect_bids* event are modified by considering the weights coming from the weight algorithm. Details of the weight algorithm are given in Section 3.4.4.3. The part is awarded to the machine having the lowest modified EFT value. A dummy variable having an integer value representing the winning machine number (*no*) is the output of the event.

task_commitment: An event which is triggered by the bid evaluation step of the bidding process. The dummy integer which is found by either of the

evaluate_turning_bids or *evaluate_milling_bids* events is assigned to the *dest_mach_no* attribute of the part agent requesting the bid. This event concludes the bidding process for the part agent.

4.2.2 Machine Agent

Machine agents are the most important resources of the simulation structure directly affecting the system outputs. They are also the main agents taking part in the negotiation protocol of the bidding process. In this section, fixed and variable attributes and the events utilized to model this agent type will be described.

4.2.2.1 Attributes

Attributes of the machine agents are used to individualize each machine agent and to state the condition of a machine at any instant during the simulation run. Machine agents have different attributes which are classified as fixed and variable. Definitions of fixed and variable attributes are given in Section 4.1.1.3.

no: A fixed attribute which is used to represent the CNC machine identification number. Each CNC machine has a unique *no* value regardless of its type. It can have integer values starting from 0 and the upper limit is less than the *machcount* value of by one.

machine_type: A fixed attribute which is used to represent the type of the CNC machine. It has a value of 0 for CNC turning machines and 1 for CNC milling machines. Therefore:

$0 \leq no < machcount/2 \quad \Rightarrow \quad machine\_type = 0$

$machcount/2 \leq no < machcount \quad \Rightarrow \quad machine\_type = 1$

isidle: A variable attribute which is used to represent whether a machine is currently allocated to a part or not. It is a Boolean type attribute and should have a value either true or false. An important point for the attribute is, if a machine is waiting (*iswaiting*=true) a part to come by means of the conveyor then the *isidle*=false for the machine.

iswaiting: A variable attribute which is used to represent whether a machine is waiting a part to come by means of the conveyor. It is a Boolean type attribute and should have a value either true or false. When combined with *isidle*, *iswaiting* results in different situations:

iswaiting = false and *isidle* = false

⇒ Machine is currently processing a part.

iswaiting = false and *isidle* = true

⇒ Machine is empty and not allocated to any part.

iswaiting = true and *isidle* = false

⇒ Machine is empty and waiting for a part on the conveyor.

iswaiting = true and *isidle* = true

⇒ Not a valid case.

islistempty: A variable attribute which is used to represent whether there are parts in the reservation list of a machine or not. It is a Boolean type attribute and should have a value either true or false. This attribute is utilized for calling a part to a machine if *islistempty* = false

total_working_time: A variable attribute which is used to represent the busy time of a machine with processing a part. The value starts with zero and incremented by 1 for each second of a part being processed in the particular machine.

loading_factor: A variable attribute which is used to represent the machine loading rate of a particular machine. It is calculated at every instant by:

loading_factor = total_working_time / t

finished_queue: A variable attribute which is used to represent the parts that a particular machine has processed before. It is a dynamic array whose dimension is incremented by 1 when a part finishes its processing on the particular machine. It keeps the track of the *no* parts.

reservation_list: A variable attribute which is used to represent the parts in the reservation list. It is a dynamic array whose dimension is incremented when a new part is awarded to the machine. It keeps the track of the *no* parts.

4.2.2.2 Events

Most of the events of machine agents stem from the cooperation with the part agent in the contract-net protocol. As a result of the machine events, the attributes of machine agent and the part agent can be changed or other events can be triggered.

add_to_finished_queue: An event which is used to add the identification number of the part (*no*) in the *finished_queue* array of the machine that the part has been processed. The event is triggered when the remaining time of the part on a machine becomes zero.

add_to_reslist: An event which is used to represent the task acceptance step of the bidding process. It is triggered by the bid evaluation step of the part agent and the event occurs only in the machine agent that is declared to be the winner. Winner machine is the one having the identification number equal to the output dummy integer of either *evaluate_turning_bids* or *evaluate_milling_bids*. The identification number (*no*) of the part agent requesting the bid is added to the reservation list of the winner machine at the exact place which the machine agent proposed by *hyp_sequence* event. This event concludes the bidding process for the machine agent.

remove_from_reslist: An event which removes a part from the reservation list of a particular machine. The event is triggered when a part reaches the machine by means of the conveyor so that it can start being processed.

readpop: An event which is used to retrieve the part number of the first element in the *reservation_list* of the corresponding machine. The event is triggered by the machine agent when it finishes processing of its current part (*iswaiting* = false and *isidle* = true) and needs to call an unprocessed part from its reservation list.

hyp_sequence: An event which is triggered by the *find_EFT* event in order to find the place of a part in the reservation list of a machine according to the precedence values. Precedence values are found considering the other parts in the reservation list of the particular machine agent. They are calculated in this event considering the W(SPT+CR) sequencing rule and the maximum tardiness control algorithm. Details are given in Section 3.4.3.3. The result of the event is an integer value and it is assigned to a dummy variable which is used in the *find_EFT* event for EFT calculation of a part.

find_EFT: An event which is triggered by the *bid_request* event of the part agent. It constitutes the main part of the bid construction by calculating the EFT value for a part. The event uses the place of the part in the reservation list which is found by *hyp_sequence* event to calculate the EFT value. Each machine of the corresponding type executes this event for a particular part requesting bid. It triggers the *collect_bids* event of the part agent.

CHAPTER 5

TEST RUNS

This chapter is dedicated to the results of the simulation studies done using the developed bidding framework. In objective verification part, the algorithms used to reach the shop-floor objectives stated in Section 3.3 are verified. The results with and without the corresponding algorithms are compared. In cross comparison part, results of the developed bidding framework are compared with the bidding framework developed by Cangar (2000) which is currently implemented in CIMLAB. Comparisons are mainly done by investigating the tardy part numbers with changing loading rate of the shop-floor, due date tightness, number of parts and number of machines. The unit used all throughout this chapter is minutes.

5.1 Objective Verification

This section involves the results for the sequencing algorithm, weight algorithm and maximum tardiness control algorithm. The individual effect of each algorithm is investigated by comparing the system results. Combined effects of those algorithms are presented here.

The input for the simulation runs is held constant. 3 CNC turning and 3 CNC milling machines are used for an input of 200 parts Time of arrival values are modeled using random interarrival times having a mean value of 2.2 minutes between successive parts. Due dates are obtained by the following formula:

Due Date = Time of Arrival + Machining Time * Random[2..4]

Processing time of different part types are modeled by using triangular distribution. The maximum, minimum and the most probable values used are given in Table 5.1.

Table 5.1 Process durations used in objective verification

PART TYPE	FIRST PROCESS	SECOND PROCESS
1	Milling / TRIA(5.8, 5.9, 6)	Turning / TRIA(4.9, 5, 5.2)
2	Milling / TRIA(5.3, 5.5, 5.7)	-
3	Turning / TRIA(2.5, 2.8,3.1)	-
4	Turning / TRIA(7, 7.3,7.4)	Milling / TRIA(12.2, 12.3, 12.5)

5.1.1 First-Come First-Served Results

This section is dedicated to the results of the developed system where none of the algorithms are present. The bidding process continues without the W(SPT+CR) sequencing rule, weight algorithm to balance the loads and the maximum tardiness control algorithm. In fact, this system is similar to the one currently implemented in CIMLAB.

The simulation run ended with a makespan of 486.57 minutes. Total number of tardy parts turned out to be 162 which is an excessive number compared to the total number of parts. Maximum tardiness is 90.87 minutes and the average tardiness of all the parts is 28.53 minutes. The tardiness values seem to be acceptable, considering high system loading conditions (mean arrival between parts is 2.2). However, the tardy part number is very high. Besides, Table 5.2 and Figure 5.1 imply that the tardiness results are priority independent which is not desired according to the system objectives. Therefore having such high number of tardy parts and even the uniformly distributed tardiness values reveal that the primary objectives defined in Section 3.3 are not met at all.

Table 5.2 Tardy part percent vs. Priority for FCFS rule

Priority	Quantity	Tardy	Early	Tardy Percent
1	19	14	5	**73.68**
2	19	16	3	**84.21**
3	19	15	4	**78.95**
4	20	18	2	**90.00**
5	25	20	5	**80.00**
6	29	22	7	**75.86**
7	16	14	2	**87.50**
8	19	17	2	**89.47**
9	16	10	6	**62.50**
10	18	16	2	**88.89**

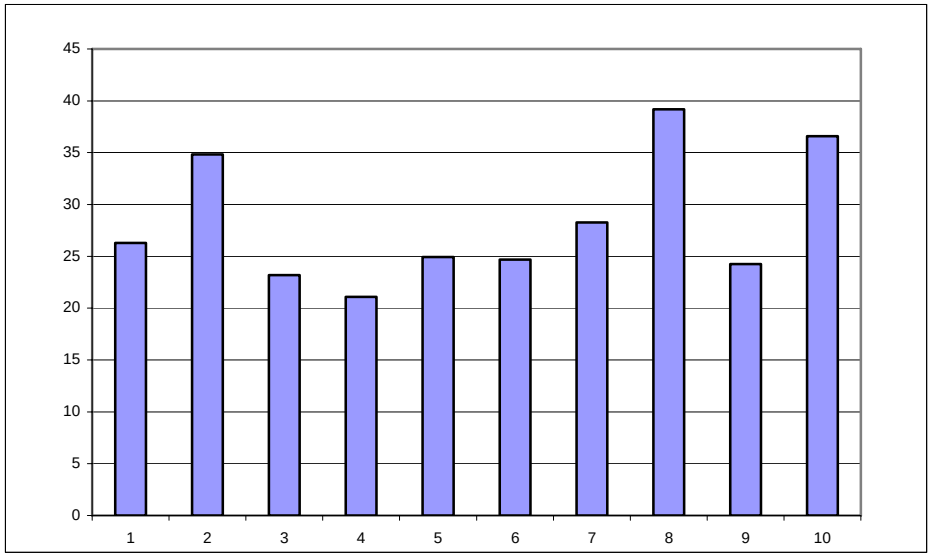

Figure 5.1 Tardiness vs. Priority for FCFS rule

Flow times of the parts are continuously estimated during the simulation. The estimation process is mainly based on the Earliest Finishing Time values given by the winner machines. EFT values only include the transportation time of the part to the machine where it will be processed. However, proposed flow time should also

include the transport time that will pass for unloading of the part from machine, carrying it to its final station, AGV and unloading from the conveyor. So the flow time values are estimated by:

For parts requiring only one operation (i.e. part types 2 and 3):
Proposed Flow Time = $EFT_{winner\ M/C} + t_{trans}(M/C,AGV)$

For parts requiring two operations (i.e. part types 1 and 4):
Proposed Flow Time = $EFT_{winner\ turning\ M/C} + EFT_{winner\ milling\ M/C} + t_{trans}(M/C,AGV)$

The deviation time of a part is defined as:
Deviation = Real Flow Time – Proposed Flow Time

Flow times according to the priority of the parts are shown in Figure 5.2. It is important to note that a considerable amount of the flow is spent by waiting in the queue. This is because of the low mean value (2.2 minutes) of arrival that is used to model the part arrivals causing crowded reservation lists to build up and resulting in long queues for a particular machine.

First-Come First-Served (FCFS) rule does not take into account of precedence according to the priority or the processing time of a part. So, once a part is allocated in the reservation list, another part coming after the reserved part can not overtake the part and begin processing in the same machine. Therefore the proposed flow time for a part will be exactly the same with the real flow time and there will not be any deviations in the parts produced according to the FCFS scenario.

The same reason applies for the low values of maximum tardiness. Since the system does not consider priorities and precedence, no part can overtake another one which avoids any part getting stuck in the system for a long time. The objective of keeping the maximum tardiness value under control is naturally obtained as a result of the nature of FCFS.

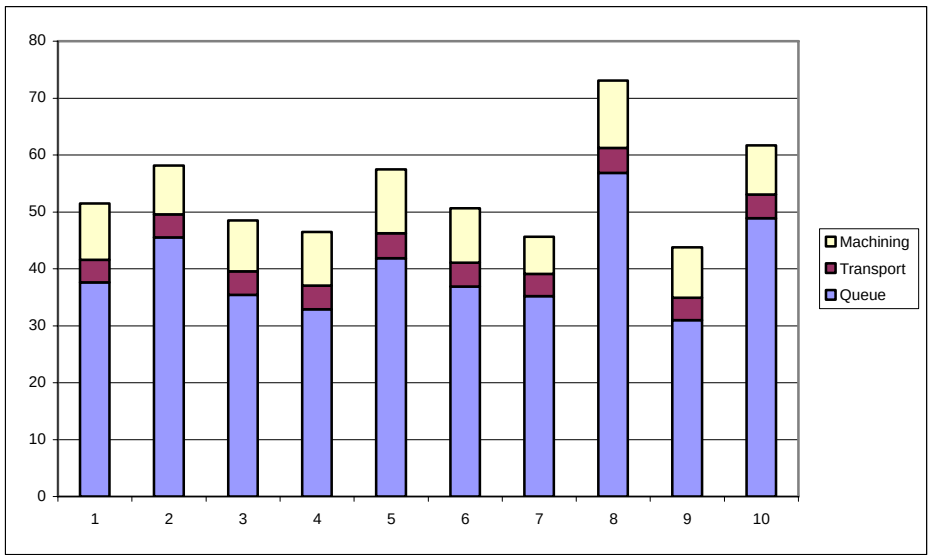

Figure 5.2 Flow time for FCFS rule

5.1.2 Sequencing Algorithm – W(SPT+CR) Results

One of the important objectives of the study is obtaining weighted tardiness results. This means, if tardiness of parts is unavoidable, the values should be so distributed that high priority parts should have low tardiness values. For this reason, during the sequencing step of bid construction W(SPT+CR) algorithm is used.

The simulation run ended with an increased makespan of 522.78 minutes. Total number of tardy parts turned out to be 89 which is a good improvement when compared to the 162 tardy parts of FCFS. Maximum tardiness is 365.37 minutes and the average tardiness of all parts is 27.57 minutes. Although an excessive increase in the maximum tardiness value is obtained, the average tardiness value remained the same (In fact, decreased by 1 minute). Also there occurred deviations between the real and proposed flow times. Use of W(SPT+CR) sequencing rule revealed results having weighted tardiness as shown in Figure 5.3 and Table 5.3. So the primary objective of weighted tardy values is reached at the expense of increased maximum tardiness.

Table 5.3 Tardy part percent vs. Priority for W(SPT+CR) rule

Priority	Quantity	Tardy	Early	Tardy Percent
1	19	13	6	**68.42**
2	19	17	2	**89.47**
3	19	15	4	**78.95**
4	20	8	12	**40.00**
5	25	8	17	**32.00**
6	29	11	18	**37.93**
7	16	6	10	**37.50**
8	19	7	12	**36.84**
9	16	4	12	**25.00**
10	18	0	18	**0.00**

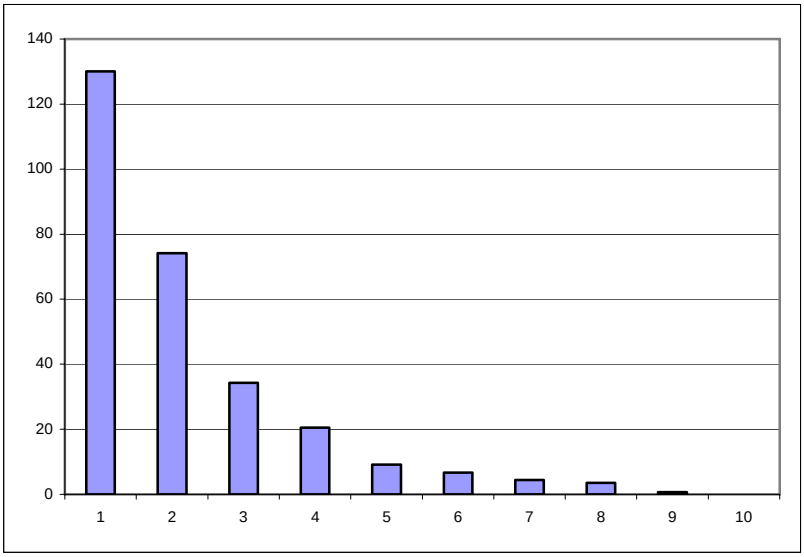

Figure 5.3 Tardiness vs. Priority for W(SPT+CR) rule

It can be seen from the Table 5.3 and Figure 5.3 that there is a decreasing trend in the number of tardy part percentage and the average tardiness value with increasing priority. This is due to the W(SPT+CR) rule where parts with higher priorities and shorter processing times have high precedence index causing them to be processed

earlier. On the other hand, parts having low priorities and high processing times result in low precedence values. Other parts introduced to the system can easily overtake the parts of low precedence. This causes very high queue times and therefore those parts can stuck in the system for very long times. As a result, the maximum tardiness value increases also leading to an increase in the average tardiness values for the parts of low priority.

Investigating the flow times according to the part priorities, it can be concluded that for high priority parts, machining, transport and queue times are relatively balanced. However as the priority of a part decreases, the queue time fraction increases and when priority becomes 1, most of the time is spent in the queue rather than processing and transporting the part. Comparing with the FCFS case, the queue time of high priority parts decreased and low priority parts increased as expected.

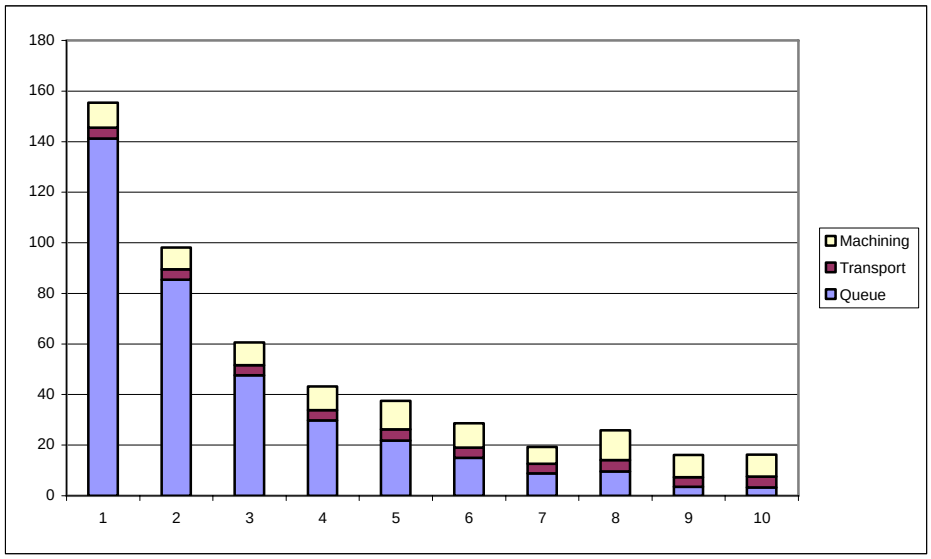

Figure 5.4 Flow time for W(SPT+CR) rule

The flow time deviations also occur when W(SPT+CR) rule is used. The deviations are due to the change of the reservation list orders of the machines. When a part

requests bids from the machines, machine checks its reservation list, sequence the part and proposes an EFT value based on the present sequence. The winning EFT value is used for proposed flow time calculation. After a part is allocated on a machine, another part having a higher precedence index may overtake the part causing a delay in the earliest finishing time. The calculated EFT value and therefore the proposed flow time of that part remains constant but in reality the flow time of the part increases due to the increased queue time causing the deviation. Of course, the deviations are higher for low priority parts where the possibility of another part overtaking the part is high. Figure 5.5 shows the deviation distribution when W(SPT+CR) rule is utilized.

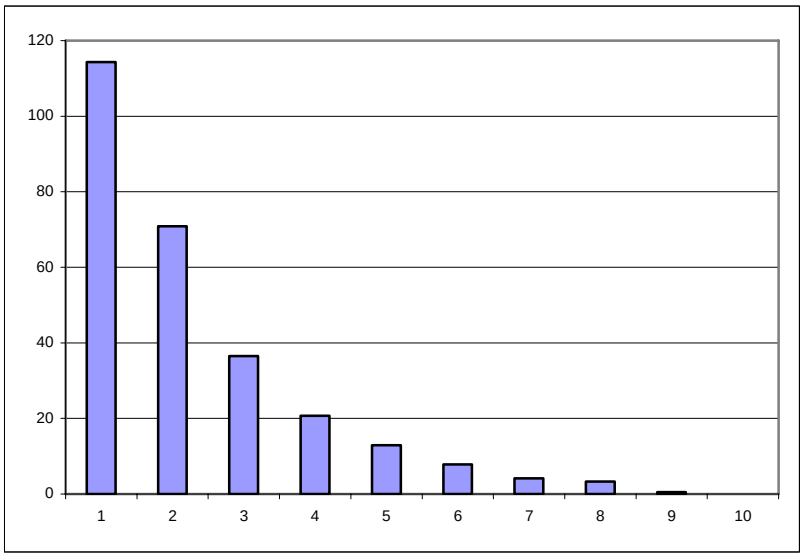

Figure 5.5 Flow time deviation for W(SPT+CR)

5.1.3 Weight Algorithm Results

The secondary objective of the system is obtaining a shop-floor environment where the utilization of each machine type is balanced. For this reason the weight algorithm is implemented into the system having machine loading and EFT weights

that can be adjusted according to the need. The results will be presented with constant weights.

A simulation study is made augmenting the W(SPT+CR) rule with the weight algorithm. The weights used are:

$w_{ML,1} = 99$, $w_{EFT,1} = 100$, $w_{ML,10} = 20$, $w_{EFT,10} = 100$

The simulation run ended with a decreased makespan of 495.9 minutes. Total number of tardy parts decreased further to 80. Maximum tardiness is 330.42 minutes and the average tardiness including all of the parts is 28.38 minutes. When compared to the previous case a 35 minute decrease in maximum tardiness value is achieved and the average tardiness value increased approximately by 1 minute. The weighted distribution of tardiness values and the deviation between the flow time values are still present since W(SPT+CR) algorithm is being used. The average tardiness values are compared in Figure 5.6. It is seen that using weight algorithm with W(SPT+CR) does not significantly change the average tardiness values.

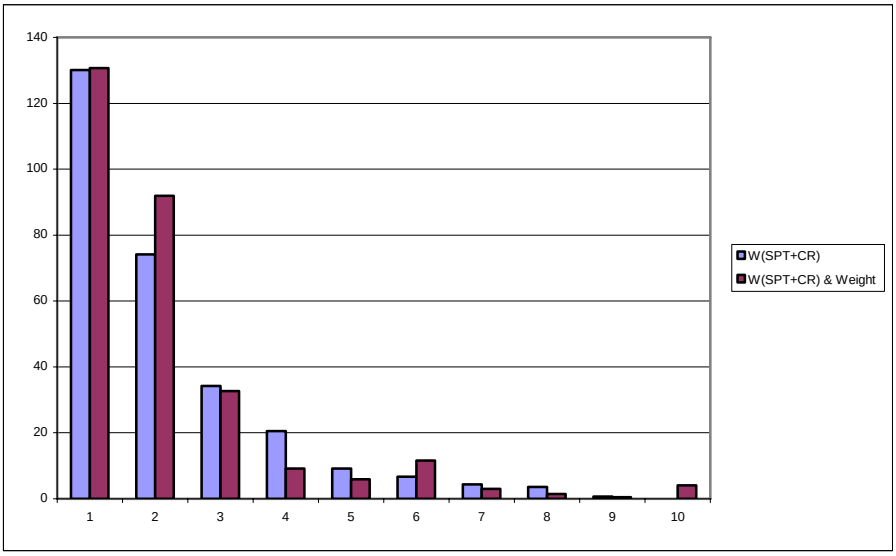

Figure 5.6 Tardiness vs. Priority for W(SPT+CR) and weight algorithm

The results differ when the machine utilizations are investigated. The difference between the maximum and minimum machine loads are used to represent the level of balance in the shop-floor for the corresponding machine type. Comparing Tables 5.4 and 5.5 it can be seen that the balance of the turning machines improved from 3.91 to 3.64 percent. The improvement of the milling machines is more significant. The value dropped to 5.42 from 13.35 percent. Since for the simulated case average milling times are higher compared to the average turning times, the occurrence of bottlenecks in the milling machine reservation lists is quite possible. Therefore weight balancing algorithm becomes more efficient for milling machines in this case.

Table 5.4 Utilization rates for W(SPT+CR) with 6 Machines

CNC Turning M/C ID	Scheduled Times	Percent Utilization	CNC Milling M/C ID	Scheduled Times	Percent Utilization
Turning 1	51	50.37	Milling 1	48	68.62
Turning 2	48	46.46	Milling 2	49	80.53
Turning 3	47	48.73	Milling 3	49	67.18
Max. Difference		**3.91**	**Max. Difference**		**13.35**

Table 5.5 Utilization rates for weight algorithm with 6 Machines

CNC Turning M/C ID	Scheduled Times	Percent Utilization	CNC Milling M/C ID	Scheduled Times	Percent Utilization
Turning 1	51	53.25	Milling 1	48	75.17
Turning 2	50	49.61	Milling 2	50	79.16
Turning 3	45	50.59	Milling 3	48	73.74
Max. Difference		**3.64**	**Max. Difference**		**5.42**

Merit of the machine load balancing weight algorithm can be shown more clearly by using higher number of resources. In this case, W(SPT+CR) algorithm is not used and only the effect of the weight algorithm is tested. The utilization rates are

compared for a shop-floor having 9 CNC turning and milling machines. For production of the given input with 200 parts, using 9 CNC turning and milling machines is unnecessary and under normal shop-floor conditions where the weight algorithm is not used, some machines will be used very rarely and will stay idle most of the time. When the weight algorithm is implemented, the parts will be forced to be dispatched to those machines having very low utilization rates for the sake of balancing the shop-floor utilization.

According to the results in Table 5.6 and 5.7, the differences between the maximum and minimum values of percent utilizations of the corresponding machines decreased significantly. The initial values of 52.84% for turning and 64.8% for milling machines obviously indicate that the utilization of the shop-floor is not balanced. However when the weight algorithm is applied with weights $w_{ML,1}$ = 95, $w_{EFT,1}$ = 5, $w_{ML,10}$ = 75 and $w_{EFT,10}$ = 25 the deviations reduced to 10.7% for turning and 7.2% for milling machines at the expense of increased tardy part number from 10 to 12. The reason of this substantial decrease is 9 CNC turning and milling machines is excessive for the system and the shop-floor can be balanced with the disadvantage of 2 more parts being tardy. However, when the shop-floor loading rate is higher and when there are few machines, increasing the machine loading weights can not give such smooth results. When the weights are investigated it can be seen that the weights for the machine loading is very high. For a shop-floor with a higher loading rate, increasing the weights of the machine loadings might result in a remarkable increase in the tardy parts.

5.1.4 Maximum Tardiness Control Algorithm Results

When using the W(SPT+CR) sequencing rule, the parts having low priority can wait in a queue and stuck in the system for a long time causing excessive tardiness values. Therefore an auxiliary maximum tardiness algorithm is implemented in the W(SPT+CR) sequencing algorithm. This algorithm works within the sequencing

algorithm and omitted if W(SPT+CR) rule is not used. As explained in Section 3.4.3.3 when the tardiness value of a part exceeds a specified threshold the algorithm is activated and tardiness is tried to be kept under control as the auxiliary objective. The threshold does not mean a guaranteed maximum tardiness value.

Table 5.6 Utilization rates for FCFS with 18 Machines

CNC Turning M/C ID	Scheduled Times	Percent Utilization	CNC Milling M/C ID	Scheduled Times	Percent Utilization
Turning 1	42	56.27	Milling 1	35	72.11
Turning 2	35	45.10	Milling 2	27	59.12
Turning 3	25	33.05	Milling 3	28	45.46
Turning 4	15	19.87	Milling 4	19	38.83
Turning 5	6	7.49	Milling 5	14	22.82
Turning 6	3	5.33	Milling 6	9	15.95
Turning 7	2	3.43	Milling 7	4	7.31
Turning 8	6	8.69	Milling 8	3	9.63
Turning 9	12	18.33	Milling 9	7	22.4
Max. Difference		**52.84**	**Max. Difference**		**64.8**

Table 5.7 Utilization rates for weight algorithm with 18 Machines

CNC Turning M/C ID	Scheduled Times	Percent Utilization	CNC Milling M/C ID	Scheduled Times	Percent Utilization
Turning 1	27	27.74	Milling 1	27	36.85
Turning 2	22	26.27	Milling 2	22	34.03
Turning 3	18	25.38	Milling 3	20	30.89
Turning 4	16	22	Milling 4	15	31.49
Turning 5	15	19.83	Milling 5	15	29.65
Turning 6	13	20.58	Milling 6	12	30.82
Turning 7	12	19.84	Milling 7	11	31.46
Turning 8	11	17.04	Milling 8	12	34.64
Turning 9	12	18.22	Milling 9	12	32.81
Max. Difference		**10.7**	**Max. Difference**		**7.2**

A simulation run with a threshold value of 70 minutes is made. The simulation run ended with a decreased makespan of 485.8 minutes. However, the total number of tardy parts increased to 97. Maximum tardiness value is decreased to 186.43 and the average tardiness including all of the parts is 25.57 minutes. Since the weight algorithm is still in use, utilization of the machines is still balanced with 1.31% for turning machines and 2.83% for milling machines. All of the results are presented in Table 5.8, Figures 5.7, 5.8 and 5.9.

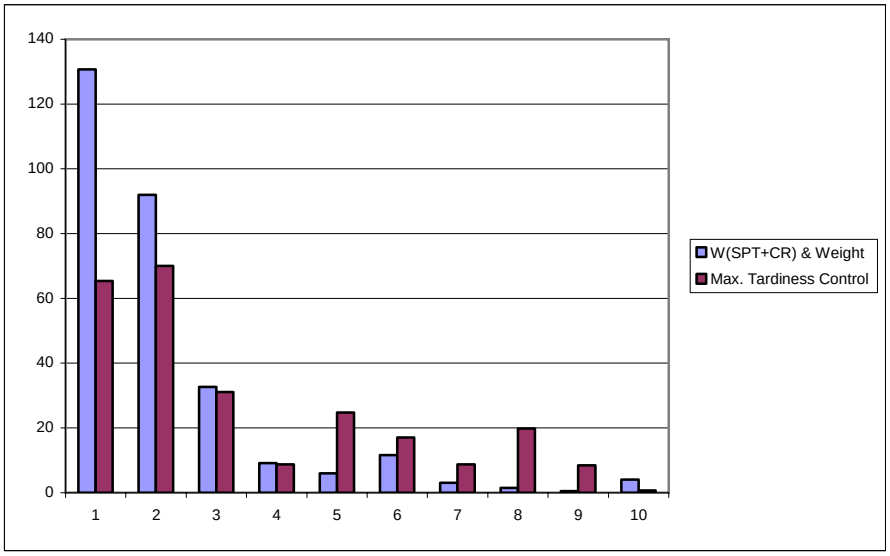

Figure 5.7 Tardiness vs. Priority for maximum tardiness control

It can be seen from Figure 5.7 that when the maximum tardiness control algorithm is used, the average tardiness values are decreased for lower priorities. In fact, this is due to the extreme maximum tardiness values mostly occur for the parts of low priority since they may wait in the queue for very long time. When maximum tardiness control algorithm is active, a part with a high priority does not always have a higher precedence over a part with low priority. If the low priority part is in the system for a long time and the tardiness value of it has exceed the control threshold, any other part introduced to the system can not be allocated in front of

that specific part in the reservation list. By this way further increase in tardiness value of the low priority part is avoided. The decrease in the maximum tardiness values also results in the decrease in the average tardiness values of the low priority parts. Of course, when the algorithm is in use, higher priority parts are not guaranteed to have a higher precedence value and therefore they may not be processed immediately. This causes increased average tardiness values of high priority parts. If Figure 5.7 is investigated for both high priorities and low priorities, the decrease in average tardiness values for the lower priorities and the increase for the high priorities can be observed.

Table 5.8 Tardy part percent vs. Priority for maximum tardiness control

Priority	Quantity	Tardy	Early	Tardy Percent
1	19	13	6	**68.42**
2	19	14	5	**73.68**
3	19	12	7	**63.16**
4	20	6	14	**30.00**
5	25	13	12	**52.00**
6	29	14	15	**48.28**
7	16	7	9	**43.75**
8	19	10	9	**52.63**
9	16	6	10	**37.50**
10	18	2	16	**11.11**

Same reasoning can be used to explain the increased number of tardy parts when maximum tardiness control algorithm is used. Controlling maximum tardiness necessitates decreasing the excessive tardiness value of individual parts. Those parts will most probably be tardy again with a smaller tardiness values. Besides, the higher priority parts may not overtake the lower priority parts having high tardiness values, causing higher priority parts to be late as well. This results in the increase in the overall number of tardy parts. This means maximum tardiness control is

achieved at the expense of an increase in the average tardiness values of the parts with high priorities and therefore an overall increase in the number of tardy parts in the system.

It is important to note that the tardiness values are still weighted because of the W(SPT+CR) algorithm. However, the steep descend in the average tardiness values is replaced by a smoother descend when maximum tardiness algorithm is used since the values are more uniformly distributed. Table 5.8 and Figure 5.7 show the weighted tardiness trend.

Flow time values are given in Figure 5.8. The flow time of low priority parts decreased and high priority parts increased. Since the average machining time is constant and the changes in the transport time can be neglected, the reason of the change is the variations in queuing times. Parts with high priority may wait the processing of the low priority parts with high tardiness values causing an increase in the queue time and low priority parts may have higher precedence because of their high tardiness value decreasing their queue time.

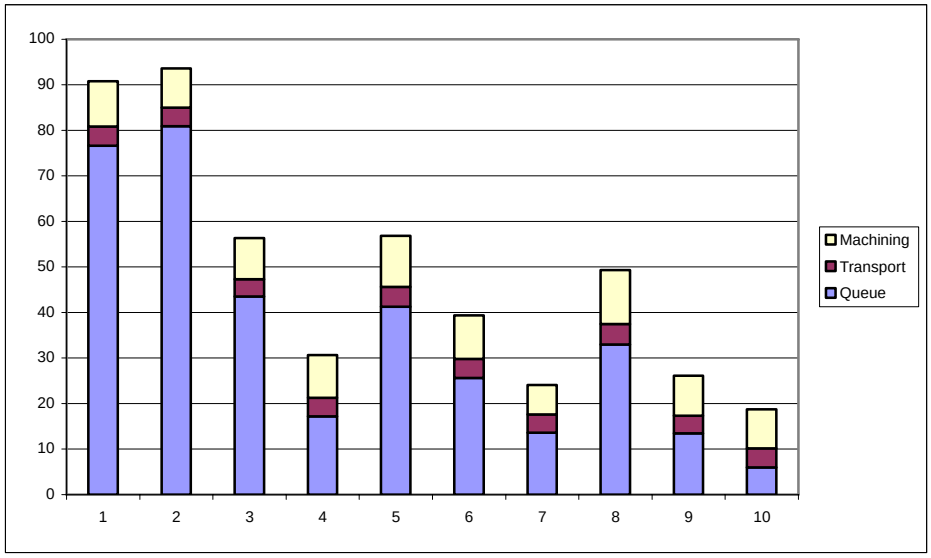

Figure 5.8 Flow time for maximum tardiness control

Flow time deviations are comparatively given in Figure 5.9. When maximum tardiness algorithm is used along with the weight and W(SPT+CR) sequencing algorithm, the deviations become more uniformly distributed. Since flow time deviations originate mainly from the variations in the queue time, it is an expected result to obtain values close to each other.

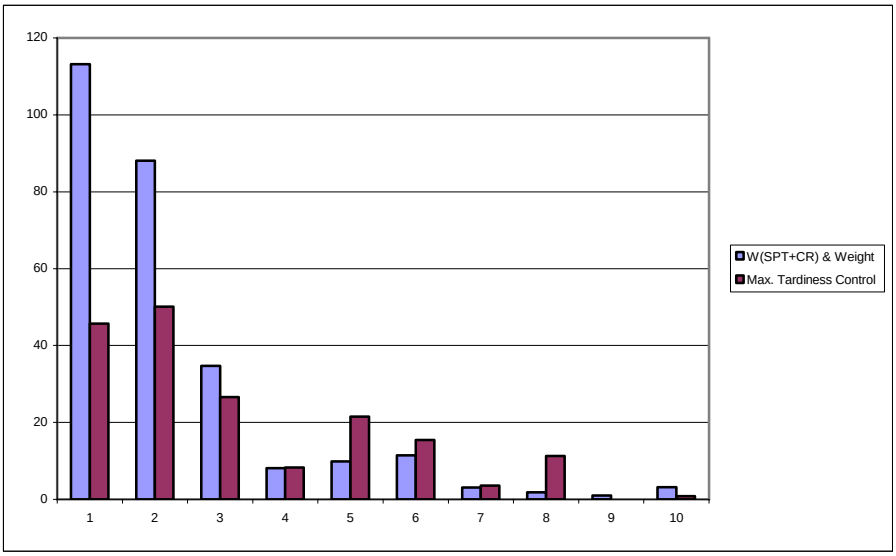

Figure 5.9 Flow time deviation for maximum tardiness control

5.1.5 Overall Results

The results of the investigated four cases are summarized and compared in this section. The results will be referred by using the case numbers for convenience. Numbers and the corresponding cases are given in Table 5.9. In each case an algorithm is added to observe the differences and improvements in the performance of the system. Overall results and the trends in terms of important performance measures are presented in Table 5.10 for number of tardy parts, average tardiness, maximum tardiness, machine utilizations and makespan of the corresponding cases. The values according to the part priorities are not taken into account.

Table 5.9 Conventions used for different case studies

Case 1	First-Come First-Served
Case 2	W(SPT+CR) Algorithm
Case 3	W(SPT+CR) and Weight Algorithms
Case 4	W(SPT+CR), Weight and Maximum Tardiness Control Algorithms

Table 5.10 Performance measures for used algorithms

	Case 1	**Case 2**	**Case 3**	**Case 4**
Number of Tardy Parts	162	89	80	97
Average Tardiness	28.53	27.57	28.38	25.57
Maximum Tardiness	90.87	365.37	330.42	186.43
Turning M/C Utilization	12	3.91	3.64	1.31
Milling M/C Utilization	4.84	13.35	5.42	2.83
Makespan	486.57	522.78	495.9	485.8

Figure 5.10 shows the trend in the number of tardy parts as each algorithm is activated. For Case 1 since only FCFS rule is used 162 tardy parts are obtained. However when W(SPT+CR) algorithm is activated, the processing time of parts are taken into account, causing parts having short processing times to be processed first. This results in a decreased flow time and lateness and therefore several parts with small tardiness values to be early. When the weight algorithm is activated, the number of tardy parts decrease, but not in a great amount compared to Case 2. However as the maximum tardiness control algorithm is activated as well, number of tardy parts increase to 97. This is mainly because the algorithm changes the sequencing rule, giving higher precedence values to parts with extreme tardiness.

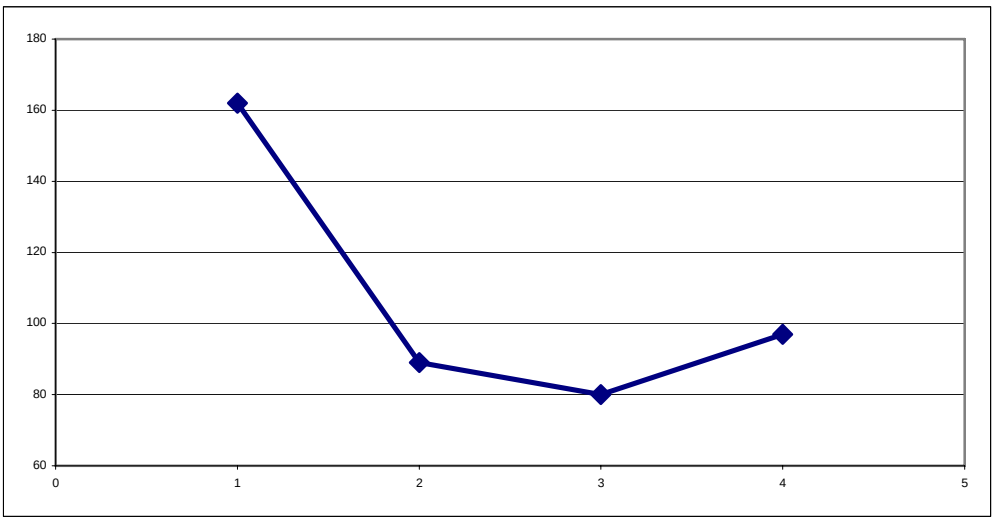

Figure 5.10 Number of tardy parts vs. case numbers

Figure 5.11 shows the average tardiness values for each case. It was expected to obtain the highest value for FCFS case since it does not consider any system input during sequencing step of bid construction. When W(SPT+CR) rule is implemented, flow times are reduced which in turn reduces the tardiness values. In fact, applying the W(SPT+CR) sequencing algorithm resulted in very high extreme tardiness values of some parts. However, the average value is still smaller compared to the FCFS case. When weight algorithm is activated, the parts are forced to be dispatched to the machines having low utilizations. Those machines may or may not be the ones that have submitted the bid with smallest EFT. Therefore when weight algorithm is used, parts may be dispatched to the machines with low utilization values for sake of balancing the shop-floor. This causes an increase in the average flow time and in turn an increase in average tardiness value as shown in Figure 5.11. The lowest average tardiness value is obtained when all the algorithms are in use including the maximum tardiness control. The main reason for high average tardiness values is the individual extreme tardiness values which belong generally to the parts of low priority. This shortcoming is controlled in the Case 4 and those high tardiness values are decreased. Decreasing the individual extreme tardiness values also decreased the average tardiness for Case 4.

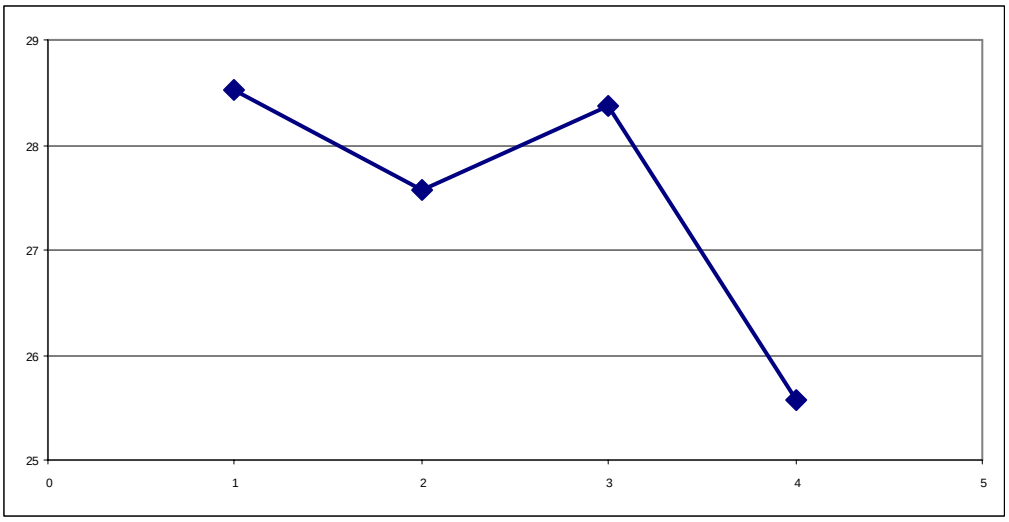

Figure 5.11 Average tardiness values vs. case numbers

Figure 5.12 shows the trend of the maximum tardiness value for all of the cases. It should be noted that maximum tardiness value is not a value showing the trend of all the parts but it belongs to a single part. The maximum tardiness value is in its minimum when FCFS sequencing rule is used. In FCFS sequencing, any part that is introduced to the system can not have a higher precedence over the parts that are introduced beforehand. Therefore there is not a possibility that an extreme tardiness value is obtained. When the system switches to W(SPT+CR) sequencing algorithm there occurs a steep ascend in the value to 365.37 minutes. This is because the parts of low priority and high processing times are sent to the end of the reservation list and wait in the list for very long times. Utilization of the weight algorithm in Case 3 does not decrease the maximum tardiness value remarkably since weight algorithm aims at fulfilling the secondary objective of shop-floor balancing. When the maximum tardiness control algorithm is used with a control threshold of 70 minutes, parts having tardiness values greater than 70 minutes will have the greatest precedence and any other part can not be processed before it. Using the algorithm, maximum tardiness value can be decreased to 186.43 minutes from 330.42 minutes. It should be noted that the obtained result is still higher than 90.87 minutes of FCFS rule and tardy parts are increased from 80 to 97 in Case 4.

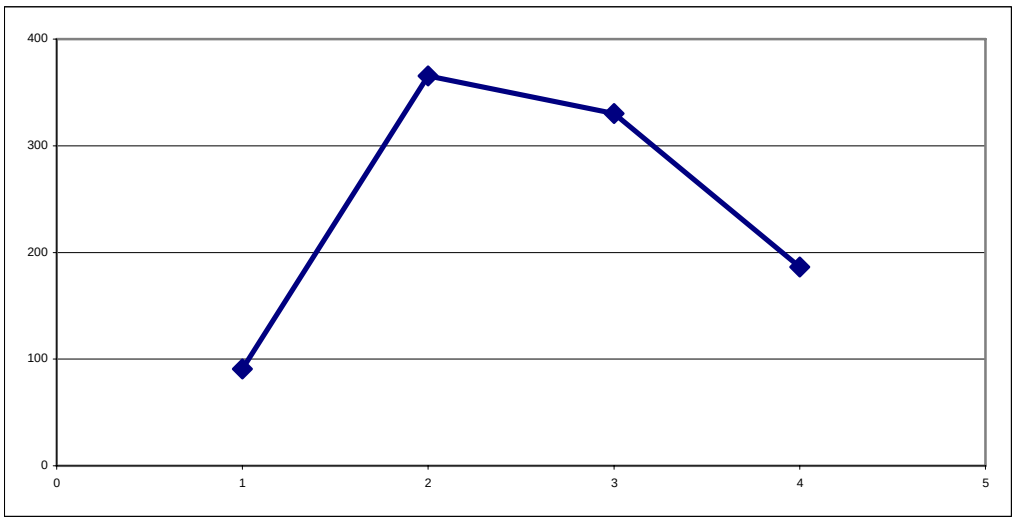

Figure 5.12 Maximum tardiness values vs. case numbers

Shop-floor utilization values are presented in Figure 5.13. Before weight algorithm is introduced, the percent difference can go up to 12 and 13.35. However once the weight algorithm is implemented those values drop to 5% approximately. Generally utilization balance is deteriorated by the parts that are stuck in the system. Since those parts are avoided in Case 4, percent difference can drop to 2 approximately.

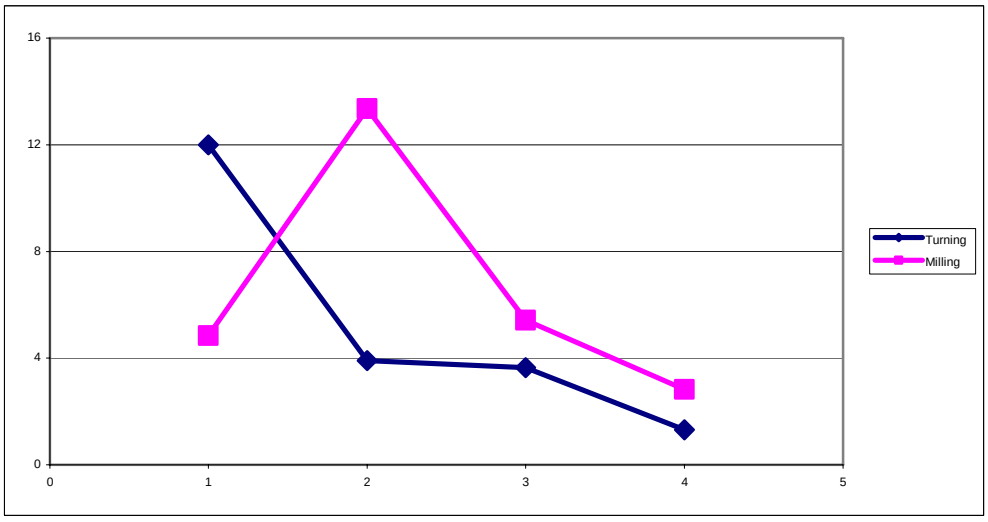

Figure 5.13 Shop-floor utilization vs. case numbers

Table 5.11 demonstrates the makespan for each case. Also the parts corresponding to the makespan values and their attributes are given.

As seen from the table the last finished product (No:198) for FCFS case is close to the final product (No:200) that is introduced to the system. This is because the 198th part is of type 4, requiring both milling and turning operations. However 199th and 200th parts are of type 3 requiring only turning operations. Although the parts 199 and 200 are introduced later to the system for Case 1, their machining (2.9 minutes) and queue time (11.5 minutes) are smaller compared to the 198th part. So the makespan value, which is the completion time of the last part, is determined by part 198. The priority value of the latest finished job is at random since FCFS rule does not take into account of the part priorities. For Case 1 Part 198 has a priority value of 5.

When switched to W(SPT+CR) sequencing rule, makespan increases to 522.78 minutes. As seen from the table, the last finished part is now 106 which is introduced before the last part (No:200). This is due to the increased queue time of the parts having low priorities. Parts of low priority and high processing time stuck in the system and wait for such excessive time that they may be finished even after the part that is introduced to the system last. Increased queue time results in increased flow time values which in turn causes high completion time. It should be noted that the part having the makespan value for Case 2 has a priority 1 and has a machining time of 19.47 which is a high value when Table 5.1 is considered. According to that table a part may have the maximum machining value of 19.9 minutes (12.5+7.4).

In Case 3 the last finished part is again of priority 1 and has a high machining time (19.62 minutes). This is because the W(SPT+CR) rule is still active and low priority parts with high processing time are sent to the end of the reservation lists of the machines. The activated weight algorithm deals with the balanced distribution of the parts to the machines. This avoids a high priority part with low processing time

to be allocated in front of the reservation list of a machine just because it guarantees the lowest finishing time. The loading of the machine is also checked during the bid evaluation step and part is then allocated in a reservation list. By this way, bottleneck machines resulting in a shift in the completion time of parts of low priority is eliminated resulting in shorter queue time and decreased makespan.

For the last case where maximum tardiness control algorithm is introduced the last finished part (No:185) determining the makespan is closer to the final part introduced in the system (No:200). Priority value 1 and high machining time is again because of the W(SPT+CR) algorithm. The maximum tardiness control algorithm prevents high priority parts overtaking parts with extreme tardiness value which is the dominant value of high makespan values. One of the natural results of maximum tardiness control is the reduced queue time for low priority parts leading to reduced flow time. This causes another part to have the longest completion time which is smaller than that of Case 3. Therefore the makespan value is reduced in Case 4.

Table 5.11 Corresponding parts for makespan values of each case

Case	No	Priority	Time of Arrival	Completion Time	Machining Time	Queue Time	Transport Time	Flow Time
1	198	5	357.42	486.57	19.7	104.12	5.33	129.15
2	106	1	198.1	522.78	19.47	299.88	5.33	324.68
3	123	1	217.62	495.9	19.62	253.33	5.33	278.28
4	185	1	340.83	485.8	19.48	120.15	5.33	144.97

5.2 Cross Comparison

This section mainly includes the cross comparisons of the developed bidding framework with the framework currently implemented in CIMLAB developed by Cangar (2000). Different scenarios are used to test both frameworks to reveal the performance of the developed framework.

In order to simulate the current system in CIMLAB, the results are taken without any algorithm activated. In fact, the obtained FCFS system without any algorithm will still have some discrepancies with the current system. The most important one is related to the conveyor and robot. The conveyor and the robot are not agents in the developed system whereas in the current system both conveyor and robot are modeled as agents. So, the developed system assumes the means of transportation and loading/unloading is always available. Omitting robot and conveyor agent decreases the flow time in the developed system since there will not be any queue for the robot service. The results without any algorithm are still efficient in reflecting the behavior of the current system and comparing it to the developed framework since the assumption will be active during testing of the current system.

The two frameworks will be compared by considering different scenarios. Those scenarios will be different shop-floor loading conditions, different due date tightness level, different total number of parts to be processed and different machine numbers in the shop-floor. The effect of maximum tardiness control algorithm is also compared to the current system results with different control threshold values.

Considering all the fixed attributes, only the processing time distributions is constant for modeling different scenarios. Processing time of different part types are modeled by using triangular distribution. The maximum, minimum and the most probable values used are given in Table 5.12. These values are constant even for the different number of parts.

Table 5.12 Process durations used in cross comparison

PART TYPE	FIRST PROCESS	SECOND PROCESS
1	Milling / TRIA(4.6, 5, 5.3)	Turning / TRIA(3.2, 3.4, 3.8)
2	Milling / TRIA(10,10.3,10.5)	-
3	Turning / TRIA(4.3, 4.5, 4.7)	-
4	Turning / TRIA(6, 6.2, 6.5)	Milling / TRIA(7.1, 7.3, 7.6)

5.2.1 Effect of Shop-Floor Loading

The simulation run in order to observe the shop-floor loading effect is executed using 2 CNC turning and 2 CNC milling machines with an input of 100 parts. Due dates of the parts are obtained by the following formula:

Due Date = Time of Arrival + Machining Time * 3

A constant arrival slack of twice the machining time is given to each part. The aim is to eliminate the random effect of due date when the equation introduced in Section 3.1 is used so that the effect of shop-floor loading is more clearly observed. Weights $w_{ML,1} = 70$, $w_{EFT,1} = 30$, $w_{ML,10} = 10$, $w_{EFT,10} = 90$ are used for the weight algorithm in the developed system. The maximum tardiness control algorithm is not activated. In order to model different shop-floor loading conditions random interarrival times having different mean values between successive parts are used. Mean values used are 1, 2, 3, 4 and 5 minutes. As the value decreases, frequency of the parts coming to the system increases resulting in a higher shop-floor loading.

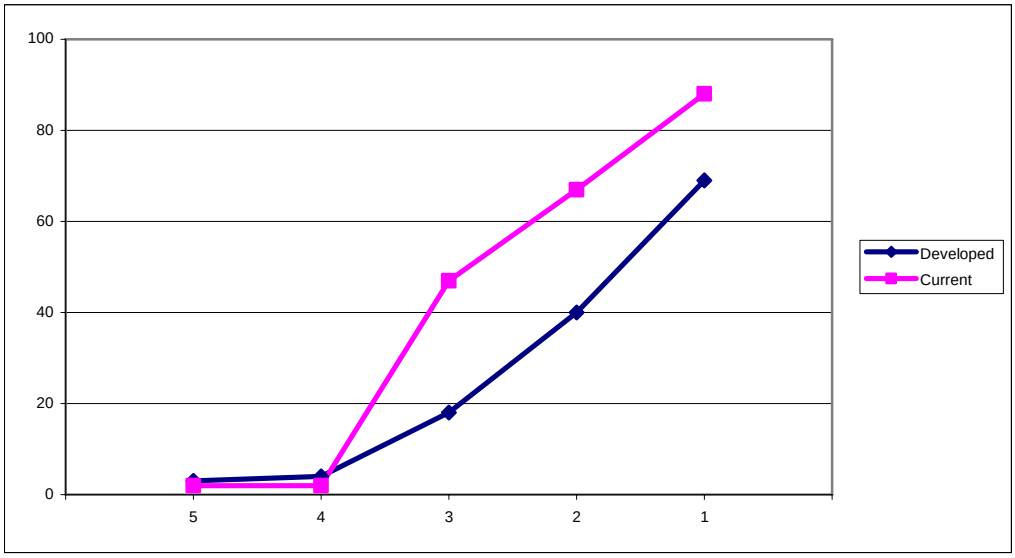

Figure 5.14 Number of tardy parts vs. mean interarrival time

Figure 5.14 shows the obtained number of tardy parts with the developed and the current system for different mean interarrival time values. It can be seen from the figure that when the shop-floor loading level is low, the two systems give tardy part numbers close to each other. In fact, the current system gives smaller number of tardy parts for low loading levels. This is mainly because of the sequencing rule used in the developed system. Since most of the machines will not have populated reservation lists due to the low loading rate, sequencing the parts according to their priorities and processing times creates an unnecessary precedence for parts generating longer reservation lists for particular machines as the arriving parts overtake the parts that are reserved beforehand. Parts having relatively high processing times and low priorities are sent back in the reservation list causing redundant queuing times for those parts. This in turn results in tardy parts which would not have been tardy with the current system. However, the strength of the developed system dominates when the shop-floor loading is increased. FCFS rule implemented in the current system is blind in terms of the system input parameters. Therefore when W(SPT+CR) is introduced for high loading levels, the parts with shorter processing times are given higher precedence and the flow time and therefore the lateness values are reduced. The effect of the W(SPT+CR) algorithm is important when the system loading is high since queue times of the parts are reduced more if the reservation list becomes populated. This results in decreased completion time and number of tardy parts.

5.2.2 Effect of Due Date Tightness

The simulation run in order to observe the due date tightness effect is executed using 2 CNC turning and 2 CNC milling machines with an input of 100 parts. Weights $w_{ML,1} = 70$, $w_{EFT,1} = 30$, $w_{ML,10} = 10$, $w_{EFT,10} = 90$ are used for the weight algorithm in the developed system. The maximum tardiness control algorithm is not activated. In order to investigate the due date effect with two different shop-floor loading conditions random interarrival times having mean values of 2 and 3 minutes

between successive parts are used. Due dates of the parts are obtained by the following formula:

Due Date = Time of Arrival + Machining Time * Random[a..b]

Parameters a and b determine the tightness of the due dates. As the parameters increase, the arrival slack of the part increases allowing a longer time for the part to be processed before its due date. On the other hand, decreasing the parameters results in tight due dates.

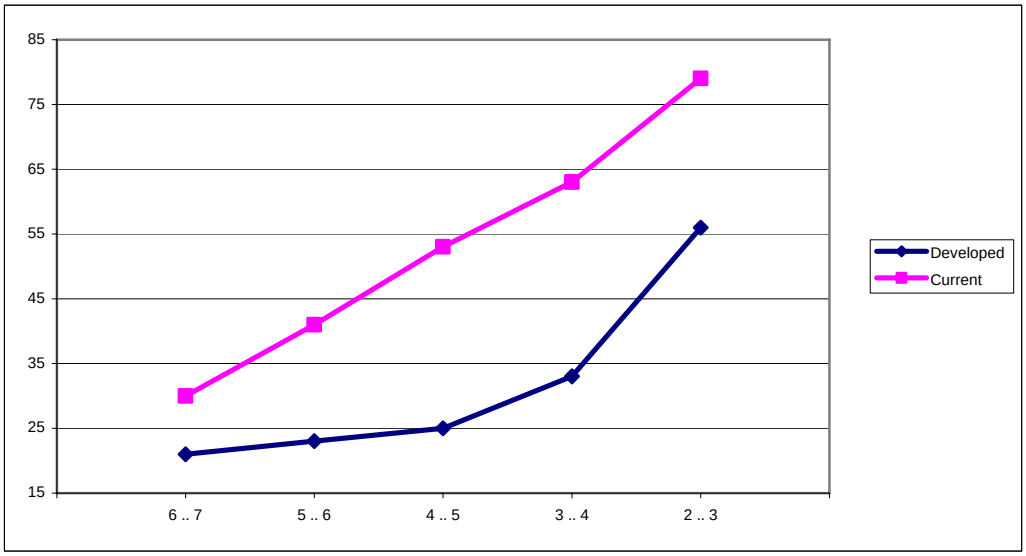

Figure 5.15 Number of tardy parts vs. due date parameters with mean interarrival time of 2 minutes

It is observed from Figures 5.15 and 5.16 that the number of tardy parts decrease linearly as the due date tightness is decreased for the current system. Since the parts are processed in the order of their arrival for the current system, there is no precedence according to the part processing times or priorities. The only reason for the parts being late is the waiting time for the parts in the reservation list which is

the queuing time. The queue time is shown to be uniformly distributed among the part priorities for the current system (utilizing FCFS) in Figure 5.2 of Section 5.1.1. Therefore there can not be any part with extreme tardiness value because of waiting in the queue for a very long time. As the tightness of the due dates are decreased by increasing the parameters a and b, the number of tardy parts decrease gradually creating a linear decrease in the number of tardy pats.

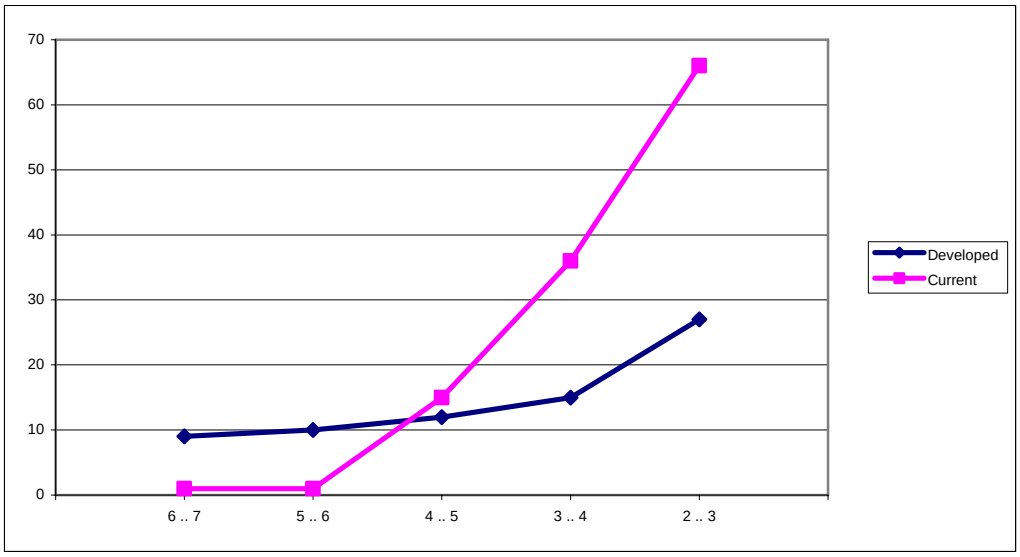

Figure 5.16 Number of tardy parts vs. due date parameters with mean interarrival time of 3 minutes

When the developed system is considered there are precedence for parts because of the W(SPT+CR) sequencing rule. These precedence values cause parts of low priority and high processing times to be stuck in the reservation list resulting in very high queuing times and in turn extreme tardiness values. The parts having those extreme tardiness values can not be made early easily no matter how loose the due dates are. Therefore when the due dates are increased, the developed system can easily finish all the parts before their due dates. However a same increase in the due dates does not enable the developed system to finish all of the parts with extreme

tardiness values before their due dates. This is the reason why the current system outperforms the developed system for loose due dates as in Figure 5.16.

The trend in the number of tardy parts is generally same for different shop-floor loading conditions. For loose due dates the current system outperforms the developed system but then at some point as the due dates become tighter the developed system generates lower tardy parts than the current system. This case is shown in Figure 5.16 in the vicinity of parameters a=4 and b=5. The difference in the number of tardy parts increase up to a point at which the developed system can not deal with the high tightness level of the due dates and the difference start to decrease. This case starts to appear at different due date tightness levels for different shop-floor loading levels. It starts with the parameters a=4 and b=5 in Figure 5.15 and a=3 and b=4 in Figure 5.16. This is reasonable since relatively low shop-floor loading levels can tolerate tighter due dates. At the extreme case of very tight due dates both of the current and the developed system results in every part being tardy.

Comparing Figures 5.15 and 5.16 reveals that for a given set of parameters a and b, the number of tardy parts are less for the case with 3 minutes of mean interarrival time between successive parts. The result can be explained by the decreased shop-floor loading level and can be based on the same reasoning given in the Section 5.2.1.

5.2.3 Effect of Total Number of Parts

The simulation run in order to observe the effect of the number of parts is executed using 2 CNC turning and 2 CNC milling machines with an input of 50, 100 and 200 parts. Due dates of the parts are obtained by the following formula:

Due Date = Time of Arrival + Machining Time * Random[4.7..4.8]

In order to minimize the random effect of the due date the parameters are determined close to each other so that the effect of the number of parts is more clearly observed. Weights $w_{ML,1} = 70$, $w_{EFT,1} = 30$, $w_{ML,10} = 10$, $w_{EFT,10} = 90$ are used for the weight algorithm in the developed system. The maximum tardiness control algorithm is not activated. Random interarrival times having mean value of 2 minutes between successive parts is used to model the shop-floor loading condition.

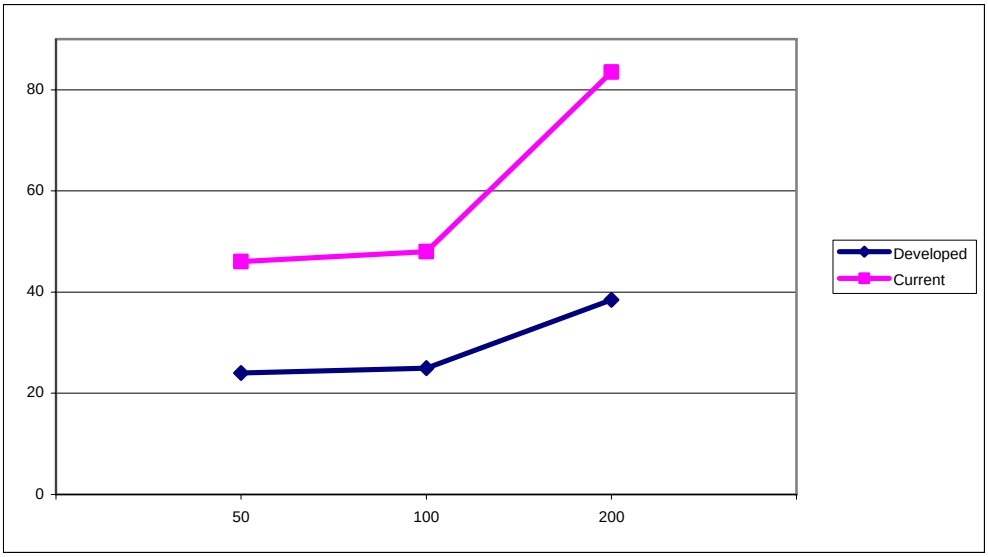

Figure 5.17 Tardy part percent vs. number of parts

The tardiness results are compared considering the tardy part percent since part numbers are different for each case. Results revealed that the developed system outperformed the current system for different number of parts. The results are valid for high shop-floor loading (mean interarrival time of 2 minutes). For high number of parts the strength of the developed system becomes more obvious as the difference between the tardy part percent increases. Investigating the performance measures given in Table 5.13 reveals that as the number of the parts increases the average number of the parts on conveyor and AGV increase as well. Also the

number of maximum WIP is the greatest for 200 parts at a particular instant. Those results imply that although the shop-floor loading level is same, the congestion in the shop-floor increases with increasing part number. As a result populated reservation lists are generated and by the sequencing rule of the developed system the order of the parts in the reservation lists are so arranged that the flow time and therefore the tardiness values are decreased giving low number of tardy parts. Figure 5.18 reveals that the makespan increases with increasing number of parts, as expected. Comparing the developed and the current system, the makespan values are nearly the same with the developed one being a little higher.

Table 5.13 Selected performance measures for the developed system

	50 parts	**100 parts**	**200 parts**
Maximum WIP Number	22	31	65
Average Part Number. on AGV	6.63	8.41	23.55
Average Part Number. on Conveyor	2.24	2.85	7.03

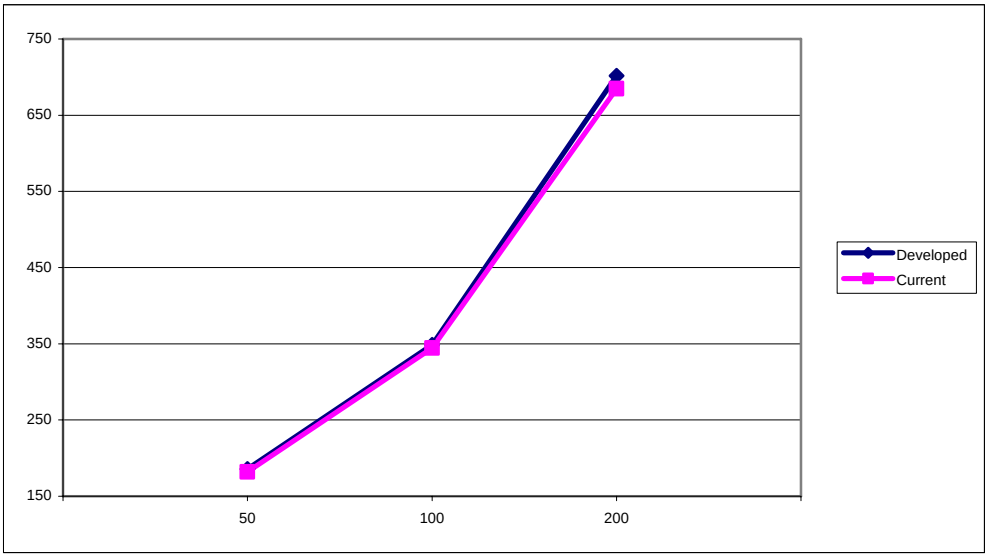

Figure 5.18 Makespan vs. number of parts

5.2.4 Effect of Machine Number

The simulation run in order to observe the effect of the number of machines is executed using with an input 200 parts. Due dates of the parts are obtained by the following formula:

Due Date = Time of Arrival + Machining Time * Random[2..4]

Weights $w_{ML,1} = 70$, $w_{EFT,1} = 30$, $w_{ML,10} = 10$, $w_{EFT,10} = 90$ are used for the weight algorithm in the developed system. The maximum tardiness control algorithm is not activated. Random interarrival times having mean value of 2 minutes between successive parts is used to model the shop-floor loading condition.

The number of machines is increased starting from 2 CNC turning and 2 CNC milling machines up to 10 CNC turning and 10 CNC milling machines. Increasing the number of machines generates an equivalent effect of increasing the interarrival time between two successive parts which decreases the shop-floor loading level.

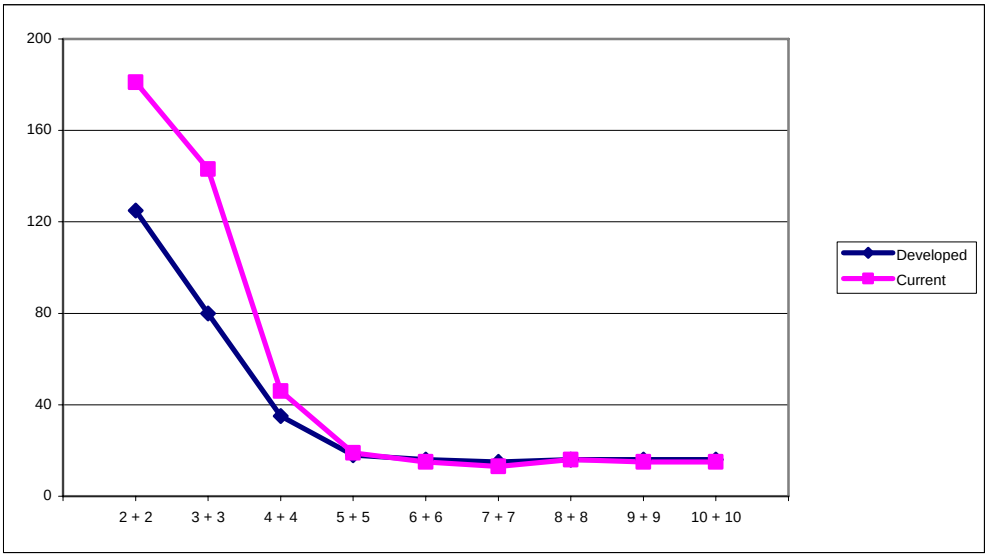

Figure 5.19 Number of tardy parts vs. number of machines

Figure 5.19 shows that as the machine number in the shop-floor decreases reservation lists of the machines become more populated and the strength of the developed system can be observed more clearly. For the given part input and system input parameters the developed system outperforms the current system for 5 CNC turning and 5 CNC milling machines and less.

Increasing the both the CNC turning and CNC milling machines above 5 does not decrease the number of tardy parts for both systems. Above 5 CNC turning and 5 CNC milling machines the limiting factor ceases to be the waiting times in the reservation lists. This time tardiness of parts is generally due to the increased transportation times accompanied with the increased number of machines. In fact, observing Figure 5.19, it can be seen that the current system gives smaller number of tardy parts above 5 CNC turning and 5 CNC milling machines. This is because of the nature of the developed system where the shop floor balancing is determined as a secondary objective. In the developed system, parts can forced to be dispatched to a further machine than the machine it would be dispatched in the current system for the sake of balancing the shop-floor utilization. Therefore the transportation time is further increased for the developed system causing couple of more tardy parts than the current system.

Figure 5.20 and Figure 5.21 show the maximum percent difference in the machine loading rates for turning and milling machines respectively. Increase in the difference implies a trend toward an unbalanced utilization in the shop-floor. As the number of machines increases, the shop-floor becomes more unbalanced when turning and milling machines are considered. Some machines may become idle or may have very low utilization with increasing machine number because of being far from AGV or from other machines.

However, when the developed and the current systems are compared, it can be seen that the developed system generally generates a less unbalanced system. The weight algorithm forces the parts to be dispatched to machines having low utilizations. By

this way, the decisions are not limited only to the earliest finishing times submitted by the machines as in the case of the current system and better utilization values are obtained in the developed system.

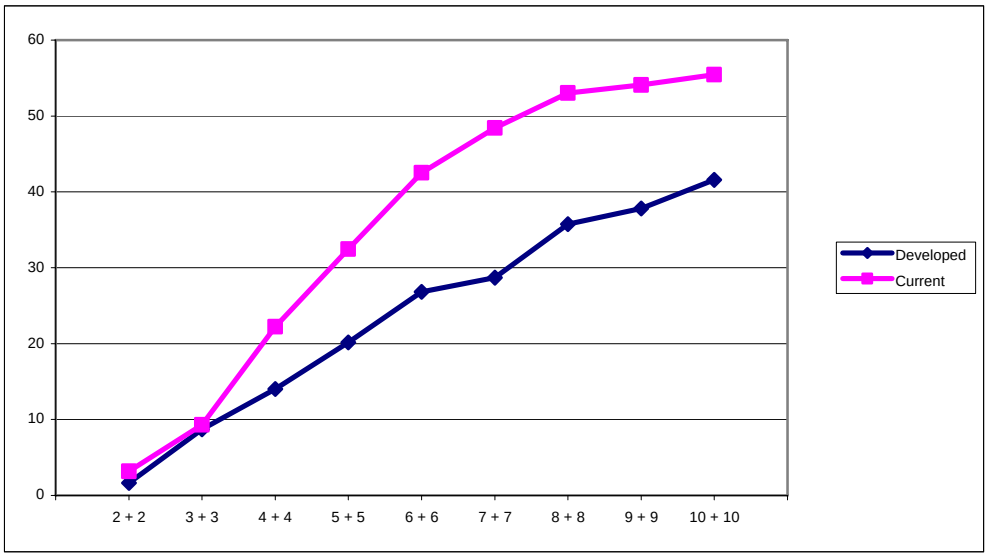

Figure 5.20 Maximum utilization differences for turning machines

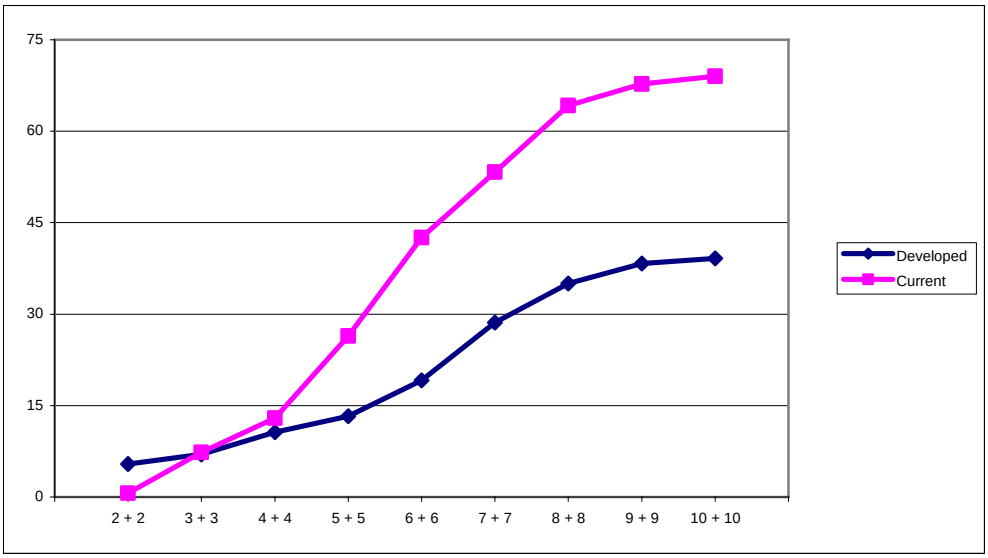

Figure 5.21 Maximum utilization differences for milling machines

5.2.5 Effect of Control Threshold in Maximum Tardiness Control

The simulation run in order to observe the shop effect of control threshold in maximum tardiness control is executed using 2 CNC turning and 2 CNC milling machines with an input of 100 parts. Due dates of the parts are obtained by the following formula:

Due Date = Time of Arrival + Machining Time * 3

A constant arrival slack of twice the machining time is given to each part. The aim is to eliminate the random effect of due date when the equation introduced in Section 3.1 is used. Weights $w_{ML,1} = 70$, $w_{EFT,1} = 30$, $w_{ML,10} = 10$, $w_{EFT,10} = 90$ are used for the weight algorithm in the developed system. Random interarrival times having mean value of 3 minutes between successive parts is used to model the shop-floor loading condition.

In the current system there is no precedence for the parts which means any part can not get stuck in the reservation list for a long time. Therefore for the current system utilizing FCFS the maximum tardiness value is automatically kept low. The maximum tardiness control algorithm in the developed system tries to reduce the extreme tardiness values by a control threshold, details of which are given in Section 3.4.3.3.

Figure 5.22 shows the effect of the control threshold on the developed system. As the threshold is decreased, the maximum tardiness value is gradually decreased. However, the value can not be decreased till it becomes equal to the current system. This is because the maximum tardiness control algorithm is initiated when the tardiness value of a part reaches the control threshold and before this threshold is reached some parts overtake the part with an undesired tardiness value due to the W(SPT+CR) sequencing rule. When the part reaches the control threshold it does not let any other part to overtake it starting from that instant. However, it has to wait

for the parts in front of the reservation list to be processed. This also explains why the maximum tardiness values are always greater than the control threshold. It should also be noted that the primary objective is still reached for the developed system since the tardiness results are still weighted because of the W(SPT+CR) sequencing rule.

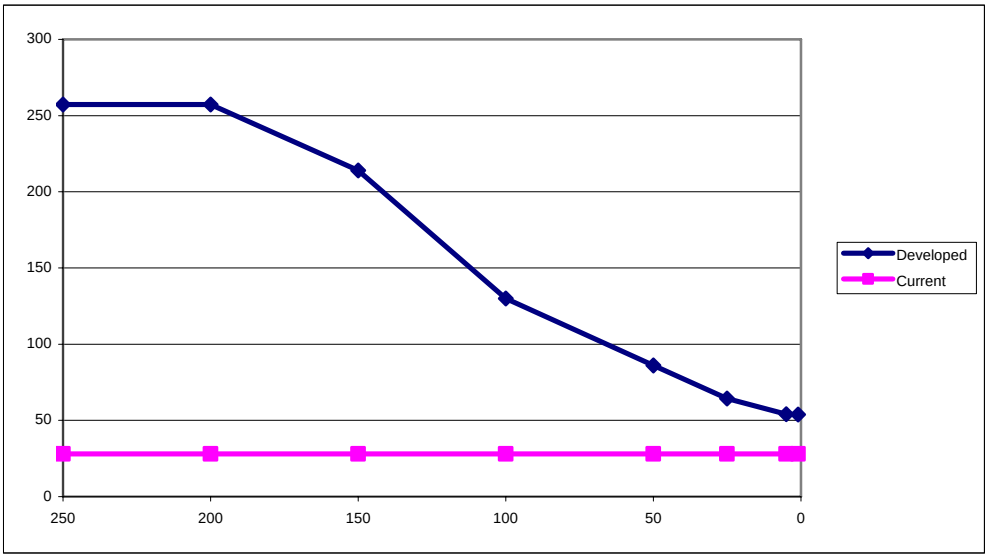

Figure 5.22 Maximum tardiness vs. control threshold

Figure 5.23 shows the number of tardy parts as the control threshold is decreased. The current system gives the maximum number of tardy parts since it utilizes FCFS sequencing rule which does not take into account any system parameters. Of course, for the developed system, the minimum tardy parts are obtained when the system works without maximum tardiness control algorithm. This is because the algorithm changes the precedence values giving higher importance to tardy parts rather than the priorities and processing times. As the control threshold is increased it becomes easy to switch to maximum tardiness control algorithm and the developed system starts to generate higher number of tardy parts. Of course, the effect of W(SPT+CR) continues and the tardy part number never reaches to that of the current system.

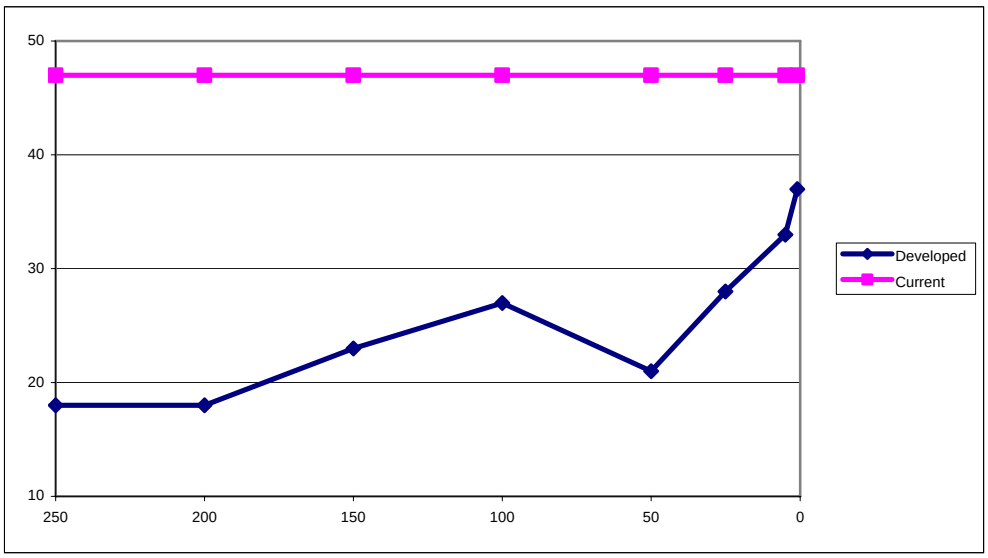

Figure 5.23 Number of tardy parts vs. control threshold

CHAPTER 6

CONCLUSION AND FUTURE WORKS

In this research, a multi-objective time based bidding framework which can be used for auction based scheduling within a distributed decision-making environment is developed. An agent based virtual simulation test bed is created in order to implement the bidding framework. Bid construction and bid evaluation stages are designed so that the multi-objective nature of the bidding framework could be achieved.

Specific outcomes of the study can be given as follows:

- A fundamental agent-based scheduling system including the sequencing and routing mechanisms, combined with the virtual simulation structure is generated.

- The distributed agent system is generated and tested by the developed simulation structure. The negotiation scheme Contract-net is implemented for the agent communication. Only two types of agents are modeled for the system: Part agents and machine agents

- Unlike the commonly used scheduling systems having two distinct steps of job routing and job sequencing, a bidding framework dealing with the two steps simultaneously is developed. Job sequencing step of the scheduling system is integrated into the bid construction step using W(SPT+CR) sequencing rule and job routing is integrated into the bid evaluation step.

- Using the product reservation technique, the idle time during the negotiation messages between the part and the available machines is eliminated. The negotiation process can be carried out even if a machine is busy with processing a part.

- The traditional bidding systems are generally initiated with the machine availability announcement and the part can request bids only from available machines. In the developed framework using product reservation technique, all of the machines are involved in the bidding process. By this way decision making is distributed to all of the machines in the shop-floor allowing the system reach the global objectives easily.

- Weighted tardiness results are obtained using the W(SPT+CR) sequencing rule in the bid construction step. Shop-floor utilization is balanced using the weight algorithm in the bid evaluation step. Maximum tardiness value is kept under control using the maximum tardiness control algorithm integrated in the W(SPT+CR) sequencing rule. All of these shop-floor objectives are achieved at the same time.

- Results outperforming the current system in CIMLAB are obtained. The results are especially superior when the shop-floor congestion level increases. This is due to the current system in CIMLAB utilizing FCFS rule is blind in terms of the system parameters. The developed system organizes the parts when the system becomes busy by taking into account of the precedence values of each part. For low level of congestion, the developed system can generate redundant precedence values which cause some parts to get stuck in the queue unnecessarily. This in turn causes the current system in the CIMLAB generating smaller number of tardy parts compared to the developed system. However, this difference in the performances of the current system and the developed system for low level of congestion is negligible if the strength of the developed system is considered for busy shop-floor conditions.

The developed framework revealed some limitations and drawbacks which should be corrected for better system performance. Besides, the framework gave an insight into the other research topics mainly complementing the current system. The main improvements and future research topics which can be based on this study are given below:

- In the current system, once a part is reserved in a machine it has to be processed on the same machine. However, when W(SPT+CR) sequencing rule is used, there will be parts that will be inserted in the reservation lists of the machines causing a shift in the finishing time of some other parts. Those cases will conflict with the initial commitment of the affected parts. So those affected parts should be given a chance to renegotiate for its committed operation. If no other machines can give bids better than the current machine then the part will chose to stay in the original machine. Otherwise, it will be removed from the reservation list of the original machine and added in the reservation list of the machine with the new winning bid. By this way a more flexible system can be obtained.

- Conveyor and the robot should be designed as agents in order to model the system more realistic. In the developed system conveyor and the robot are represented by limited parameters. The places of the parts that are being carried on the conveyor are not traced and both the robot and the conveyor are assumed to be always available in need.

- In the developed system all of the machines of the same type (turning or milling) are similar. Machine parameters should be implemented in order to complicate the system and model it closer to reality.

- Using the reliability and risk analysis, machine breakdowns should be modeled. A machine breakdown makes the system more realistic. Besides, the flexibility of the negotiation scheme can also be efficiently used.

- System is blind in terms of the future load of the machines. Machine loading rates that are used during the bid evaluation step are calculated based on the past workload of the machine and the reservation lists are not considered. Therefore a bottleneck analysis can be implemented checking the length of the reservation list and avoiding the parts to be routed to the bottleneck machines.

- Sequencing according to the W(SPT+CR) rule is only done when a bid request reaches and when a bid is evaluated in a machine. The sequencing rule can be used more frequently at specific time intervals.

- The bidding structure can be modified by introducing penalties for tardy products. This can be done by setting due dates proportional to the processing times of parts with two operations. Also penalties can be assigned to the early parts to apply the Just in Time (JIT) manufacturing concept.

- The Earliest Reservation First (ERF) technique proposed by Saad et. al. (1997) can be implemented and can be combined with the current tardiness control algorithm to obtain more efficient maximum tardiness results.

- It is not possible to observe the merits of the elimination algorithm in a virtual simulation environment. Therefore the elimination algorithm should be verified in a real distributed manufacturing environment.

REFERENCES

Akalp, M.K., (2002), "Development of a Job-Shop Scheduling System in CIM Environment", Masters Thesis, Graduate School of Natural and Applied Sciences, Department of Mechanical Engineering, Middle East Technical University

Alatas, B., (2003), "Development of a web-based dynamic scheduling methodology for a flexible manufacturing cell using agent based distributed internet applications", Masters Thesis, Graduate School of Natural and Applied Sciences, Department of Mechanical Engineering, Middle East Technical University

Arazy, O., Woo, C.C., (2002), "Analysis and design of agent-oriented information Systems", The Knowledge Engineering Review, Vol. 17, No.3, pp. 215–260

Baker, A.D., (1988), "A Survey of Factory Control Algorithms that can be implemented in a Multi-Agent Heterarchy: Dispatching, Scheduling and Pull", Journal of Manufacturing Systems, Vol. 17, No. 4, pp. 297-320

Baker, K.R., (1995), "Elements of Sequencing and Scheduling", Amos Tuck School, Dartmouth College, Hanover, N.H.

Barber, K.S., Martin, C.E., (2001), "Flexible problem-solving roles for autonomous agents", Integrated Computer-Aided Engineering, Vol. 8, pp. 1-15

Blackstone, J.H., Phillips, D.T., Hogg, G.L., (1982), "A state-of-the-art survey of dispatching rules for manufacturing job shop operations", International Journal of Production Research, Vol. 20, No. 1, pp. 27-45

Brennan, R., O, W., (2000), "A simulation test-bed to evaluate multi-agent control of manufacturing systems", Proceedings of the 2000 Winter ,simulation Conference, pp. 1747-1756

Brennan, R., O, W., Walker, S.S., Norrie, D.H., (2001), "Job Sequencing and Dispatching in Multi-Agent Heterarchical Control Systems", Third International NAISO Congress on World Manufacturing Congress

Burkolter, D., Henoch, J., Pratsini, E., (2001), "A multi-agent bidding system for task assignment", Proceedings of the Third International World Manufacturing Congress, Rochester, NY, 2001

Cangar, T., 2000, "Development of an agent based flexible manufacturing cell controller using distributed internet applications", Masters Thesis, Graduate School of Natural and Applied Sciences, Department of Mechanical Engineering, Middle East Technical University

Chan, F.T.S., (2001), "The effects of routing flexibility on a flexible manufacturing system", International Journal of Computer Integrated Manufacturing, Vol. 14, No. 5, pp. 431-445

Chandra, J., Talavage, J., (1991), "Intelligent Dispatching for Flexible Manufacturing", International Journal of Production Research, Vol. 29, No. 11, pp. 2259-2278

Crowe, T.J., Stahlman, E.J., (1995), "A Proposed Structure for Distributed Shopfloor Control", Integrated Manufacturing Systems, Vol. 6, No. 6, pp. 31-36.

Dewan, P., Joshi, S., (2000), "Dynamic Single Machine Scheduling Under Distributed Decision Making", International Journal of Production Research, Vol. 38, No. 16, pp. 3759-3777

Dewan, P., Joshi, S., (2001), "Implementation of an Auction-Based Distributed Scheduling Model for a Dynamic Job Shop Environment", International Journal of Computer Integrated Manufacturing, Vol. 14, No. 5, pp. 446-456

Dewan, P., Joshi, S., (2002), "Auction-Based Distributed Scheduling a Dynamic Job Shop Environment", International Journal of Production Research, Vol. 40, No. 5, pp. 1173-1191

Dilts, D.M., Boyd, N.P., Whorms, H.H., (1991), "The Evolution of Control Architectures for Automated Manufacturing Systems", Journal of Manufacturing Systems, Vol. 10, No. 1, pp. 79-93

Duffie, N.A., Piper, R.S., (1986), "Non-Hierarchical Control of Manufacturing Systems", Journal of Manufacturing Systems, Vol. 5, No. 2, pp. 137-139

Duffie, N.A., Piper, R.S., (1987), "Non-Hierarchical Control of a Flexible Manufacturing Cell", Robotics and Computer Integrated Manufacturing, Vol. 3, No. 2, pp. 175-179

Duffie, N.A., Chitturi, R., Mou, J., (1988), "Fault-tolerant Heterarchical Control of Heterogeneous Manufacturing System Entities", Journal of Manufacturing Systems, Vol. 7, No. 4, pp. 315-328

Duffie, N.A., Prabhu, V.V., (1994), "Real-Time Distributed Scheduling of Heterarchical Manufacturing Systems", Journal of Manufacturing Systems, Vol. 13, No. 2, pp. 94-107

Fan, C.K., Wong T.N., (2003), "Agent-based architecture for manufacturing system control", Integrated Manufacturing Systems, Vol. 14, No. 7, pp. 599-609

French, S., (1982), "Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop", New York: Wiley

Friedrich, H., Rogalla, O., Dillmann, R., (1998), "Integrating Skills into multi-agent systems", Journal of Intelligent Manufacturing, Vol. 9, No. 2, pp. 119-127

Goel A., (2004), "A Multi-Agent System and Auction Mechanism for Production Planning over Multiple Facilities in an Advanced Planning and Scheduling System", Masters Thesis, Faculty of the Virginia Polytechnic Institute and State University, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University

Gu, P., Balasubramanian, S., Norrie, D.H., (1997), "Bidding-Based Process Planning and Scheduling in a Multi-Agent System", Computers and Industrial Engineering, Vol.32, No. 2, pp. 477-496

Jones, A.T., McLean, C.R., (1986), "A Proposed Hierarchical Control Model for Automated Manufacturing Systems," Journal of Manufacturing Systems, Vol. 5, No. 1, pp. 15-25

Kadar, B., Monostori, L., Szelke, E., (1998), "An object-oriented framework for developing distributed manufacturing architectures", Journal of Intelligent Manufacturing, Vol. 9, No. 2 pp. 173-179

Kelton, W.D., Sadowski, R.P., Sturrock, D.T., (2004), "Simulation with Arena", 3rd Edition, New York : McGraw-Hill

Kim, K.H., Song, J.Y., Wang, K.H., (1997), "A Negotiation Based Scheduling for Items with Flexible Process Plans", Computers and Industrial Engineering, Vol.33, No. 3-4, pp. 785-788

Kouiss, K., Pierreval, H., Mebarki, N., (1997), "Using multi-agent architechture in FMS for dynamic scheduling", Journal of Intelligent Manufacturing, Vol. 8, pp. 41-47

Krothapalli, N.K.C., Deshmukh, A.V., (1997), "Effects of Negotiation Mechanisms on Performance of Agent Based Manufacturing Systems", Proceedings of the Seventh International Conference on Flexible Automation and Intelligent Manufacturing, pp. 704-717

Kutanoğlu, E., Sabuncuoğlu, İ., (1999), "An analysis of heuristics in a dynamic job shop with weighted tardiness objectives", International Journal of Production Research, Vol. 42, No. 3, pp. 547-572

Lim, M.K., Zhang, D.Z., (2001), "An Iterative algorithm for multi-agent based integrated process planning and scheduling", Proc. of the 29th International Conference on Computers and Industrial Engineering, Montreal, Canada, 1-3 November 2001, pp. 26-33

Lim, M.K., Zhang, D.Z., (2003), "A multi-agent based manufacturing control strategy for responsive manufacturing", Journal of Materials Processing Technology, Vol. 139, No. 1-3, pp. 379-384.

Lim, M.K., Zhang, D.Z., (2004), "An integrated agent-based approach to the responsive control of manufacturing resources", Computers and Industrial Engineering, Vol. 46, No. 2, pp: 221-232

Lin, G.Y., Solberg, J.J., (1991), "Effectiveness of Flexible Routing Control", The International Journal of Flexible Manufacturing Systems, Vol. 3, No. 3-4, pp. 189-211

Lin, G.Y., Solberg, J.J., (1992), “Integrated Shop Floor Control Using Autonomous Agents”, IEE Transactions, Vol. 24, No. 3, pp. 57-71

Lin, G.Y., Solberg, J.J., (1994), “An Agent-based flexible routing manufacturing control simulation system”, Proceedings of the 1994 Winter Simulation Conference, pp. 970-977

Macchiaroli, R., Riemma, S., (2002), “A negotiation scheme for autonomous agents in job shop scheduling”, International Journal of Integrated Manufacturing, Vol. 15, No. 3, pp. 222-232

McDonnell, P., Smith, G., Joshi, S., Kumara, S.T., (1999), “A Cascading Auction Protocol as a Framework for Integrating Process Planning and Heterarchical Shop Floor Control”, The International Journal of Flexible Manufacturing Systems, Vol.11, pp. 37–62

Morton, T.E., Pentico, D.W., (1993), “Heuristic Scheduling Systems: With Applications to the Production Systems and Production Management”, New York : Wiley

Okubo, H., Jiahua, W., Onari, H. (2000), “Characteristics of Distributed Autonomous Production Control”, International Journal of Production Research, Vol. 38, No. 17, pp. 4205-4215

Ottaway, T.A., Burns, J.R., (2000), “An Adaptive Production Control System Utilizing Agent Technology”, International Journal of Production Research, Vol. 38, No. 4, pp. 721-737

Ou-Yang, C., Lin, J.S., (1998), “The development of a hybrid hierarchical / heterarchical shop floor control system applying bidding method in job dispatching”, Robotics and Computer-Integrated Manufacturing, Vol. 14, pp. 199-217

Panwalkar, S.S., Iskander, W., (1977), “A Survey of Scheduling Rules”, Operations Research, Vol. 25, No. 1, pp. 45-62

Parunak, H., (1987), “Manufacturing Experience with the Contract Net”, Distributed Artificial Intelligence, Morgan Kaufman, pp. 285-310

Pierreval, H., Mebarki, N., (1997), "Dynamic Selection of Dispatching Rules for Manufacturing System Scheduling", International journal of Production Research, Vol. 35, No.6, pp. 1575-1591

Ro, I.-K., Kim, J.-I., (1990), "Multi-Criteria Operational Control Rules in Flexible Manufacturing Systems (FMSs)", International Journal of Production Research, Vol. 28, No. 1, pp. 47-63.

Saad, A., Kawamura, K., Biswas, G., (1997), "Performance Evaluation of Contract Net-Based Heterarchical Scheduling for Flexible Manufacturing Systems" International Journal of Automation and Soft Computing: Special Issue on Intelligent Manufacturing Planning, Vol. 3, No. 3, pp. 229-248

Shen, W., Norrie, D.H., (1999), "Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey", Knowledge and Information Systems: an International Journal, Vol. 1, No. 2, pp. 129-156

Shen, W., Norrie, D.H., (2001), "Dynamic manufacturing scheduling using both functional and resource related agents", Integrated Computer-Aided Engineering, Vol. 8, pp. 17-30

Shen, W., (2002), "Genetic algorithms in agent-based manufacturing scheduling systems", Integrated Computer-Aided Engineering, Vol. 9, pp. 207-217

Shukla, C.S., Chen, F.F., (2001), "An Intelligent Decision support system for Part Launching in a Flexible Manufacturing System", International Journal of Advanced Manufacturing Technology, Vol. 18, No. 6, pp. 422-433

Siwamogsatham, T., Saygın C., (2004), "Auction-Based Distributed Scheduling and Control Scheme for Flexible Manufacturing Systems", International Journal of Production Research, Vol. 42, No. 3, pp. 547-572

Smith, R G., (1980), "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", IEEE Transactions on Computers, Vol. 29, No. 12, pp. 1104-1113

Shaw, M.J., (1987), "A Distributed Scheduling Method for Computer Integrated Manufacturing: The Use of Local Area Networks in Cellular Systems", International Journal of Production Research, Vol. 25, No. 9, pp. 1285-1303

Shaw, M.J., (1988), "Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Network Decision Making", Journal of Manufacturing Systems, Vol. 7, No. 2, pp. 83-94

Sodhi, M.S., Askin, R.G., Sen, S., (1994) "A hierarchical model for control of flexible manufacturing systems", Journal of the Operational Research Society, Vol. 45, No. 10, pp. 1185-1196

Sousa, P., Ramos, C., (1999), "A Distributed Architecture and Negotiation Protocol for Scheduling in Manufacturing Systems", Computers in Industry, Vol. 38, No. 2, pp. 103-113.

Subramaniam, V., Lee, G.K., Ramesh, T., Hong, G.S., Wong, Y.S. (2000-1), "Machine Selection Rules in a Dynamic Job Shop", International Journal of Advanced Manufacturing Technology, Vol. 16, No. 12, pp. 902-908

Subramaniam, V., Lee, G.K., Hong, G.S., Wong, Y.S., Ramesh, T., (2000-2), "Dynamic Selection of Dispatching Rules for Job Shop Scheduling", Production Planning and Control, Vol. 11, No. 1, pp. 73-81.

Ünver, H.Ö., (1996), "An object oriented approach to design of a modular shop floor controller", Masters Thesis, Graduate School of Natural and Applied Sciences, Department of Mechanical Engineering, Middle East Technical University

Ünver, H.Ö., (2000), "A systems framework and structured methodology for design and development of manufacturing control systems using n-tier client/server technology", Ph.D. Thesis, Graduate School of Natural and Applied Sciences, Department of Mechanical Engineering, Middle East Technical University

Wang, Y., Usher, J.M., (2002), "An agent-based approach for flexible routing in dynamic job shop scheduling", Proceedings of Industiral Engineering Research Conference

Xue, D., Sun, J., Norrie, D.H., (2001), “An Intelligent Optimal Production Scheduling Approach using Constraint-based Search and Agent-based Collaboration”, Computers in Industry, Vol. 46, No. 2, pp. 209-231

Yi-Chi Wang, (2003), “Application of Reinforcement Learning to Multi-Agent Production Scheduling”, Ph.D. Thesis, Faculty of Mississippi State University, Department of Industrial Engineering, Mississippi State University

Yücel, N.D., (2005), “Simulation of a Flexible Manufacturing System: A Pilot Implementation”, Masters Thesis, Graduate School of Natural and Applied Sciences, Department of Mechanical Engineering, Middle East Technical University

APPENDIX A

COMPONENTS OF MODELED SYSTEM

Individual hardware components of the flexible manufacturing cell in Computer Integrated Manufacturing Laboratory (CIMLAB) were explained in Section 3.1.1. The figure demonstrating the locations of the resources of the system was also given. In order to help the reader to understand the main characteristics of the system better, the photographs of the system are provided in this appendix. The first picture shows the overall operating system and the succeeding figures are the pictures of the individual system components.

Figure A.1 The general view of the system under operation

Figure A.2 The CNC turning machine

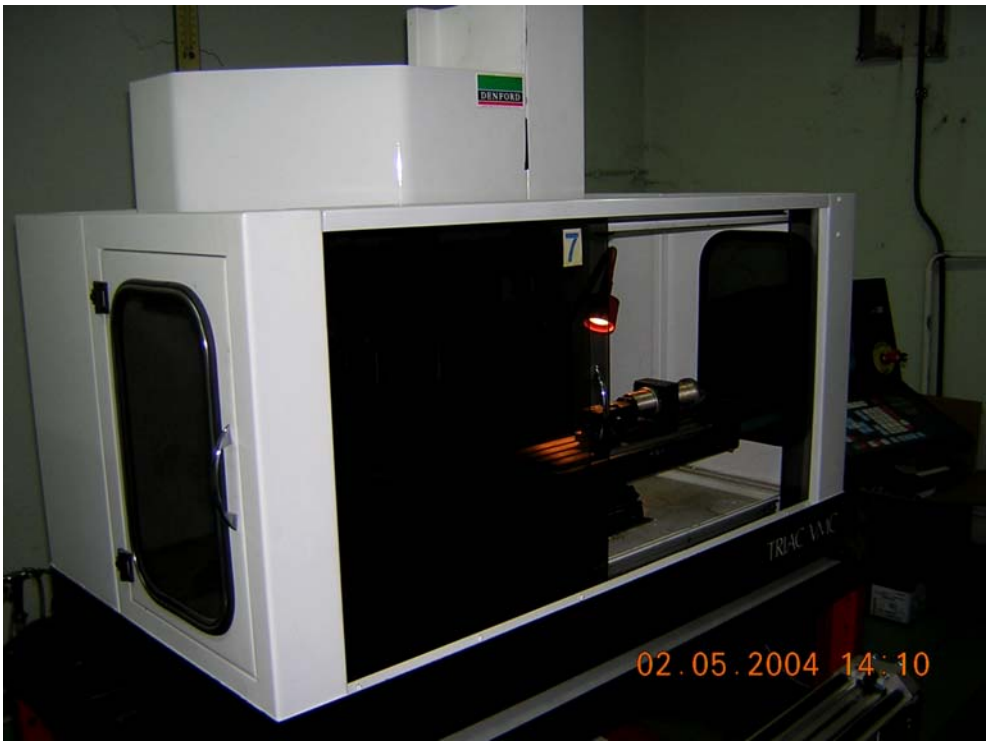

Figure A.3 The CNC milling machine

Figure A.4 The Robot on PLRD and the conveyor

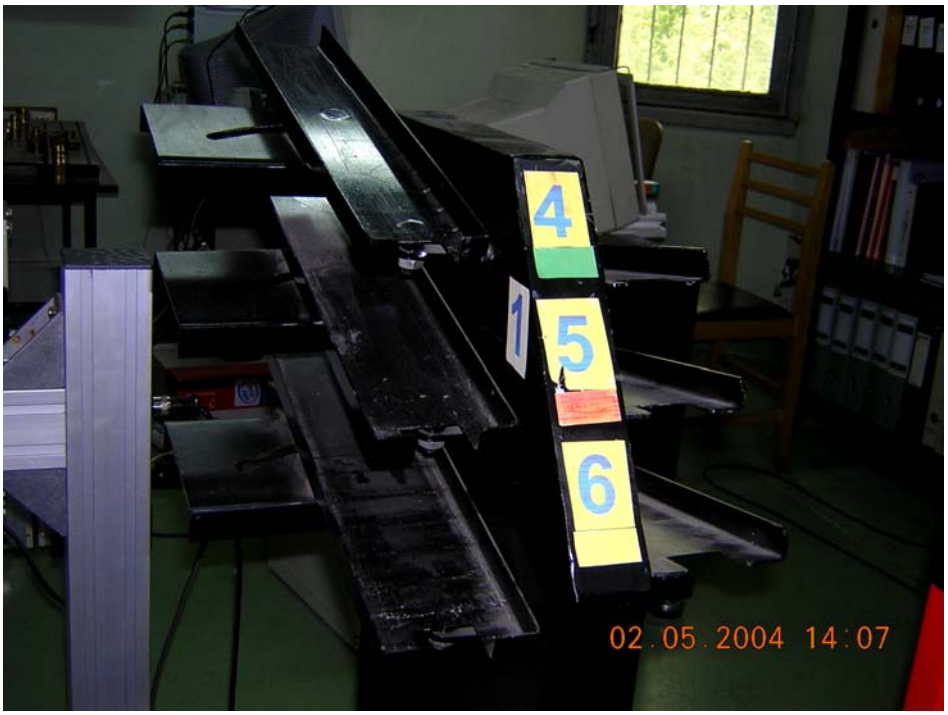

Figure A.5 The Stationary buffer modeled as AGV

APPENDIX B

STATISTICAL DISTRIBUTIONS

Computer simulation is one of the most important and frequently used tools for the analysis of complex flexible manufacturing systems which cannot be easily analyzed by more conventional methods. Such simulations use generated values from various statistical distributions to mimic the behavior of the real system. This appendix is intended as a reference to give the basics about the statistical distributions which are used throughout the study.

B.1 Uniform Distribution

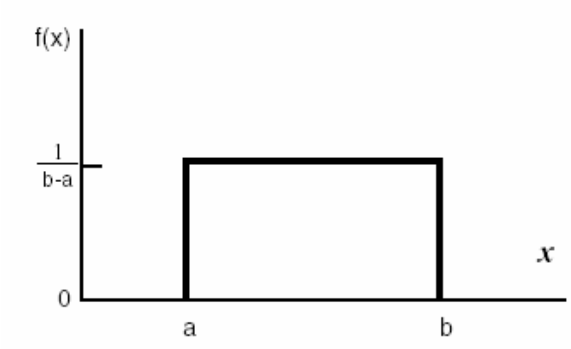

Figure B.1 Uniform probability density function (Kelton 2004)

The uniform distribution is used for the cases where the probability of occurrence of any value over a finite range is considered to be equally likely. Since it needs only the range of data to be known (maximum value b and minimum value a), uniform distribution has a large variance in the generated numbers.

Random numbers (r) from the triangular distribution are generated by the following pseudo code:
r = a+RND()*(b-a)

A random integer between a and b is created by the pseudo code:
r = INT((b-a+1)*RND()+a)

B.2 Exponential Distribution

Figure B.2 Exponential probability density function (Kelton 2004)

The exponential distribution describes the interval between events when the average number of events per unit of continuum has a Poisson distribution.

Exponential distribution is commonly used for modeling random inter-arrival times (the times between the two successive parts entering to the system) in queuing theory. It is also well suited for reliability theory for modeling the constant hazard rate portion of the bathtub curve.

The mean of the exponential distribution is given by β. If the exponential is used as the distribution of inter-arrival times then β is the mean inter-arrival time.

Exponentially distributed random numbers (r) are generated by the pseudo code:
r = - β * ln(RND())

B.3 Triangular Distribution

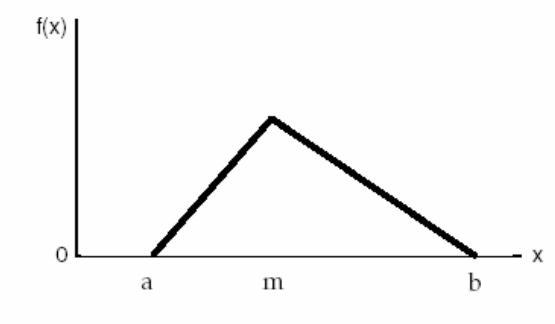

Figure B.3 Triangular probability density function (Kelton 2004)

The Triangular Distribution is used for the cases where there is limited sample data. The number generation is based on the specification of maximum (b), minimum (a) and most likely (m) values. The most likely value (m) is assigned as the modal value of the distribution.

Random numbers (r) from the triangular distribution are generated by the following pseudo code:

```
u=RND()
if u <= (m-a)/(b-a) then
r = a+sqr(u*(b-a)*(m-a))
else
r = max-sqr((1-u)*(b-a)*(b-m))
end if
```

Storing the value from the random number function in the variable u is important because most random number function return a new value each time they are called. Without the use of the u variable, the statement would use one value for branching and another for calculation.