DISTRIBUTED SCHEDULING

83742

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING AND THE INSTITUTE OF ENGINEERING AND SCIENCE OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE

By
Ayşegül Toptal

July, 1999

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

Assoc. Prof. İhsan Sabuncuoğlu (Principal Advisor)

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

Assoc. Prof. Ömer Benli

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

Assoc. Prof. Osman Oğuz

Approved for the Institute of Engineering and Science:

Prof. Mehmet Baray
Director of Institute of Engineering and Science

ABSTRACT

DISTRIBUTED SCHEDULING

Ayşegül Toptal
M.S. in Industrial Engineering
Advisor: Assoc. Prof. İhsan Sabuncuoğlu
July, 1999

Distributed Scheduling (DS) is a new paradigm that enables the local decision-makers make their own schedules by considering local objectives and constraints within the boundaries and the overall objective of the whole system. Local schedules from different parts of the system are then combined together to form a final schedule. Since each local decision-maker acts independently from each other, the communication system in a distributed architecture should be carefully designed to achieve better overall system performance. These systems are preferred over the traditional systems due to the ability to update the schedule, flexibility, reactivity and shorter lead times. In this thesis, we review the existing work on DS and propose a new classification framework. We also develop a number of bidding based DS algorithms. These algorithms are tested under various manufacturing environments.

Keywords: Distributed Scheduling, Distributed Intelligent Agents, Hierarchical Systems, Bidding-based Scheduling

ÖZET

DAĞITIK ÇİZELGELEME

Ayşegül Toptal
Endüstri Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. İhsan Sabuncuoğlu
Temmuz, 1999

Dağıtık Çizelgeleme yerel karar vericilerine lokal hedeflerini ve de tüm sistemin objektiflerini gözönüne alarak kendi çizelgelerini yapma olanağı sağlayan yeni bir paradigmadır. Sistemin farklı bölümlerinden gelen çizelegeler nihayi çizelgeyi oluşturmak üzere birleştirilir. Lokal karar vericiler birbirlerinden bağımsız hareket ettiklerinden, dağıtık bir yapıda tüm sistemin hedeflerine ulaşmak için haberleşme sistemi dikkatlice tasarlanmalıdır. Bu yapılar günümüz üretim sistemlerinin çabuk karar verme, kısa tedarik süresi, hızlı güncellemeye olan ihtiyacı dolayısıyla geleneksel sistemlere tercih ediliyor. Bu çalışmada Dağıtık Çizelegeleme konusundaki literatür özetlenmiş ve oluşturulan klasifikasyon sistemi çerçevesinde karşılaştırmaları yapılmıştır. Ayrıca açık artırma usulüne dayalı Dağıtık Çizelgeleme algoritmaları geliştirilerek çeşitli sanal imalat ortamlarında denenmiştir.

Anahtar sözcükler: Dağıtık Çizelgeleme, Lokal Karar Vericiler, Hiyerarşik Sistemler, Açık Artırmalı Çizelgeleme

To my parents and my sister Seda,

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Dr. İhsan Sabuncuoğlu due to his supervision, suggestions, and understanding throughout the development of this thesis.

I am also indepted to Dr. Ömer Benli and Dr. Osman Oğuz for showing keen interest to the subject matter and accepting to read and review this thesis.

I cannot fully express my gratitude and thanks to Erkan Bilhan, Salih Ergüt, Mustafa Dağtekin and Tacıbaht Türel for their morale support and encouragement. I would also like to thank to Eylem Düzgün, Sibel Koral, Meliha Yetişgen, Banu Yüksel, Gonca Yıldırım and Pelin Arun for their friendship. I would like to extend my gratitude to Berrin Çelik, Ömer Fatih Çelik and Cengiz Çelik.

My special thanks go to my parents and my sister. It is to them this study is dedicated, without whom it would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Scheduling is one of the vital components of shop floor control systems. It is a short-term decision making process that deals with allocation of scare resources (i.e., man, machine, tools, and material handling equipment) to various tasks of competing jobs (i.e., customer orders, products, parts, etc.) over a period time period (i.e., shift, day, week, etc). These jobs may represent customer orders (received via a sales department) or can be generated internally by MRP or some equivalent systems. The existing applications have demonstrated that better ways of scheduling of resources can result in significant improvements in the system performance in terms of utilization, tardiness, lead times, and other measures.

In practice, the scheduling task is usually carried out by human schedulers in an ad hoc manner with the help of some simple spreadsheet programs. As discussed in Ovacik and Uzsoy (1997), it is very difficult to survive in today's competitive manufacturing environment with these manual methods. To alleviate the problems and satisfy the needs of the industry, there are some commercial scheduling systems developed in the market place. Sadowski (1998) review these software packages and discuss their selection criteria.

The current tendency is that schedulers at some central planing office generate a schedule (in the form of Gantt Chart) for all the jobs available on the shop floor by considering various system resources and constraints. This off-line plan is then sent to the shop floor for execution. During the realization of the schedule, however, a number of stochastic or unexpected events (i.e., machine breakdowns, excessive scrap, due-date changes, order cancellations, etc.) occur that an appropriate revision of the schedule is needed. In practice, however, this revision process is not seriously undertaken due to the difficulties in generating new schedules, or a lack of scheduling knowledge or non-existence of effective shop floor data monitoring/collection systems. For that reason, in many existing applications, scheduling is viewed as one time decision making process that determines the production goals for shop floor people rather than an operational tool that plans and controls the shop floor activities continuously. Today, with the advances in the computer technology and flexible software packages, it is relatively easier to use the scheduling systems in real time shop floor control. To achieve this, one should incorporate the necessary feedback mechanisms and easy to update features into their existing planning systems.

Another important step towards the effective use of scheduling in practice is to simplify the task of scheduling. One way of doing that is to decompose the original problem into a number of small manageable problems and solve them separately. By that way, a very complex scheduling problem (i.e., NP-hard in mathematical terms) can be reduced to some tractable cases where exact or approximate algorithms can be applied for their efficient solution. Ovacik and Uzsoy (1997) discuss these decomposition methods in their recent book. In our opinion, decomposition approaches can also close the gap between the theory and practice since various algorithms developed for small problems in the literature can find the application possibilities.

In this context, a distributed approach can be a very useful paradigm to achieve the above objective. Distributed Scheduling (DS) distributes the functionality of scheduling decisions among local decision makers who have direct access to decision making tools and computing facilities and each of which is responsible for his own segment of production while coordinating with each other for eliminating conflicts and serving the global objective of the system (Kutanoglu, 1997). With this approach, we do not only divide the entire scheduling problem into some small problems but also we consider local needs, preferences, and constraints and use more detailed and accurate information. Distributed systems can also be very reactive or responsive to dynamic and unpredictable events such as machine breakdowns, new job arrivals, order cancellations since the decisions are quickly made in local decision units. Their monitoring and control system can be much simpler than the traditional (centralized) systems since any decision or action does not need to be approved by the global planner or scheduler. Hence, distributed systems are very suitable for real-time scheduling.

In the second chapter of this thesis, a review of the related research is provided and various issues related to the distributed scheduling research and practice are discussed. A new classification scheme is also proposed and the existing studies in the literature are reviewed within this proposed framework. As a result of the literature review, conclusions and future research directions are presented.

In the third chapter, three distributed scheduling algorithms designed for process team structure, are presented. First, they are compared with the centralized scheduling algorithms on the well-known job shop problem instances against the L_{max} (maximum lateness) criterion. The results indicate that the proposed algorithms perform better than the existing algorithms for the majority of the problem instances. Then a comparative study is conducted in a simulated environment with alternative machines to assess the effectiveness of the bidding methods. Makespan and tardiness related criteria are also used during comparisons. The strengths/weaknesses of each algorithm are discussed in detail. In the fourth chapter, three DS based algorithms are proposed for product team structure. They are tested in different experimental settings and compared within themselves. A conceptual comparison of all the algorithms including the ones in Chapter 3 is also provided in this chapter. Finally, the concluding remarks are made and further research directions are outlined in Chapter 5.

Chapter 2

LITERATURE REVIEW

Scheduling is one the most studied areas in Industrial Engineering (IE) and Operations Research (OR). Hence, there are already a number of excellent survey papers published in the literature. Day & Hottenstein (1970), Graham et al. (1979), Graves (1981) are the first general survey papers that summarize and classify the existing research in this area. Later, more focussed survey papers are published. For example, Sen & Gupta (1984) reviewed deterministic due-date scheduling papers. Gupta and Kyprasis (1987) summarized the single machine scheduling literature. Similarly, Cheng and Sin (1990) surveyed the parallel machine scheduling research. In another study, Koulamas (1994) focussed on the total tardiness problem and presented a comprehensive review on this measure. Stochastic and dynamic aspects of scheduling problems also led to development of another scheduling branch on dispatching (or scheduling) rules. This part of the literature is reviewed by Panwalkar & Iskander (1977), and Blackstone et al. (1982). Rachamadugu & Stecke (1994) and Basnet & Mize provided the review of scheduling in the context of flexible manufacturing systems. In addition to the traditional IE and OR tools, there have been developments in the areas of artificial intelligence, expert systems and neural networks that led to publications of a number of scheduling survey papers. For example, Stephen (1986) presented a comprehensive annotated bibliography of artificial intelligence for scheduling. Kanet & Adelsberger (1987) and Kusiak & Cheng (1988) summarized the Expert Systems related research activities in scheduling. In another paper, Szelke and Kerr (1994) provided an overview of research in the domain of the knowledge based reactive scheduling systems. Sabuncuoglu (1998) reviewed the entire literature of neural networks that are applied to various scheduling problems. All these papers provide the reader an overview of scheduling and application of various solution methods.

In the scope of this literature survey, we focus on distributed scheduling research and applications. Our objective is to summarize the current research and suggest some future research directions. We also provide a classification framework by which the existing scheduling applications can easily be summarized. Although the aim is to review the distributed scheduling research works as comprehensive as possible, our coverage is limited with the publications appeared in the scientific literature and the commercial packages in the market place.

The rest of this chapter is organized as follows. First, distributed systems will be presented from a scheduling point of view. This is followed by the proposed classification framework and the related notation in Section 2.2. Then we reviewed the distributed scheduling research within this framework in Section 2.3. Finally, conclusions and future research directions are presented in Section 2.4.

2.1. Distributed Systems: Agent-based, Team-based and Holonic Systems

Distributed scheduling aims at creating distributed environments for production scheduling. We can identify three main forms of DS applications in the literature: agent-based systems, holonic systems, and team-based systems. In the *agent-based* *systems* or *distributed artificial intelligence* applications, the local decision-makers are represented by *agents*. The agent concept was first introduced in computer science to create autonomous and intelligent agents (Rahimifard and Newman 1998). In general, the agents are expected to have autonomy (ability to operate without human intervention), social ability (ability to interact and communicate with others), reactivity (ability to react in response to changes in environment), pro-activeness (ability to take an initiative role). According to Aydin and Oztemel (1998) an agent should have three basic properties: perception (the ability to receive messages from the environment, cognition (the ability to evaluate the messages received) and action (the ability to make the required action).

In agent based systems, agents represent various entities in the system such as orders, machines, departments, group technology cells, parts, products, teams of resources. As seen in Figure 2.1, we can classify the agents into regular agents and a manager agent. *Regular agents* make their local scheduling decisions, a manager agent (mediator agent or master agent) initiates a bid, selects a bid, resolves conflicts during negotiation, overwrites decisions made by regular agents, finalises scheduling decisions considering overall system performance. *Manager agents* usually have the right to access all the information sources in the system. It is the only agent with

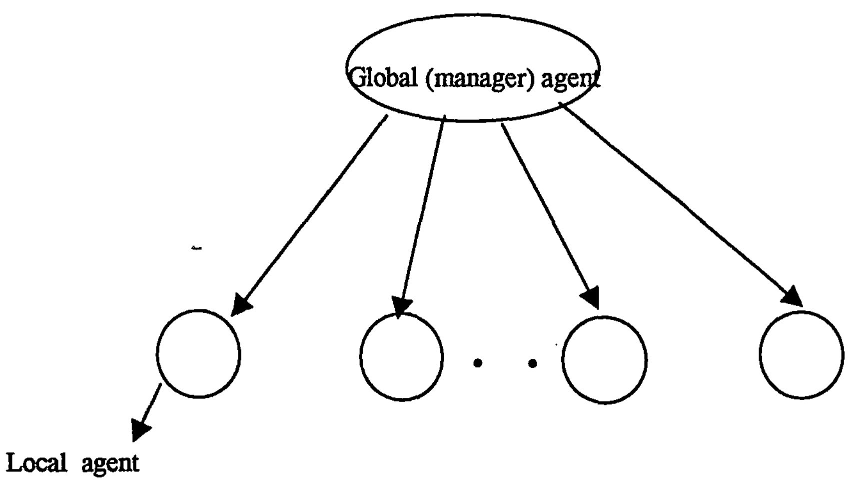


Figure 2.1. A Schematic view of agent-based systems

infinite life. It can also create other agents in the system whenever necessary. In that sense, regular agents (i.e. parts, products, and orders) are called temporary agents. Agents, multi-agent techniques, and their applications in different areas are discussed in detail in Zhang, Stanley, Smith and Gruver (1998).

Similar to the agents, other autonomous, cooperative and intelligent manufacturing entities are also defined (i.e. cells, fractals and holons) in a different distributed control paradigm. This paradigm is called holonic manufacturing, bionic manufacturing and fractal factory. In *holonic manufacturing*, intelligence is distributed over holons. A holon is an autonomous and co-operative building block of a manufacturing systems for transforming, transporting, storing and/or validating information and physical objects (Bongaerts, et al., 1996). A holon can be made up of other holons. For implementation of a holonic manufacturing system, there must be product holons, order holons and resource holons (Figure 2.2). There may also be staff holons, workstation holons, transportation holons etc., according to system needs and complexity (Bongaerts, et al., 1995). In *bionic manufacturing*, individual entities are called cells while in fractal factory they are called fractals. As discussed by Tharumarajah, Wells and Nemes (1996), cells do not communicate directly but through a higher authority. Task specifications also come from this authority. In *fractal factory*, lower level fractals also take part on goal coordination and task specification. Different than cells, fractals communicate directly and co-operate with

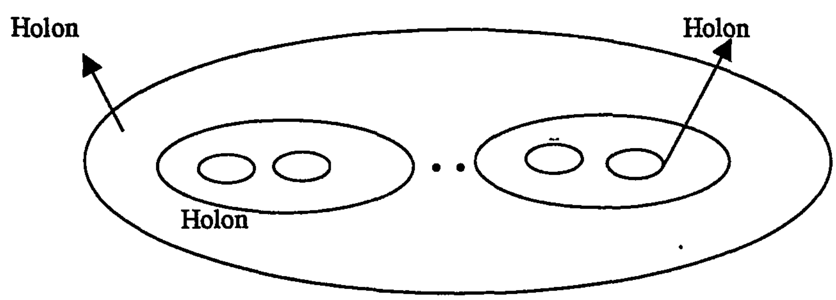


Figure 2.2 A schematic view of holonic systems

each other. In holonic manufacturing, task or goal specifications come from a higher level but rather in a colsultive manner. Holons also co-operate with each other.

Another distributed scheduling application is the self-managed team approach. Amelsvoort and Benders (1996) focused on self-directed teams, their formation and the circumstances in which they are used for better organizational effectiveness. This idea has been introduced to the production literature as a result of the need for reduced lead times, agile and flexible manufacturing systems. In this approach, teams are formed from production groups that can perform a certain or a number of different tasks. The idea is to distribute scheduling and control activities over various teams called local planners (Figure 2.3). In case of a conflict between local teams, there is a global planner who acts as a conflict resolver. These teams correspond to the agents in the agent-based system and team managers are equipped with all the necessary tools to make their own decisions by considering local objectives and constraints. Rahimifard and Newman (1998) define three basic team structures: process teams, product teams and project teams. A process team is a collection of resources, which are specialised to perform a certain process. A product team is a collection of resources with different

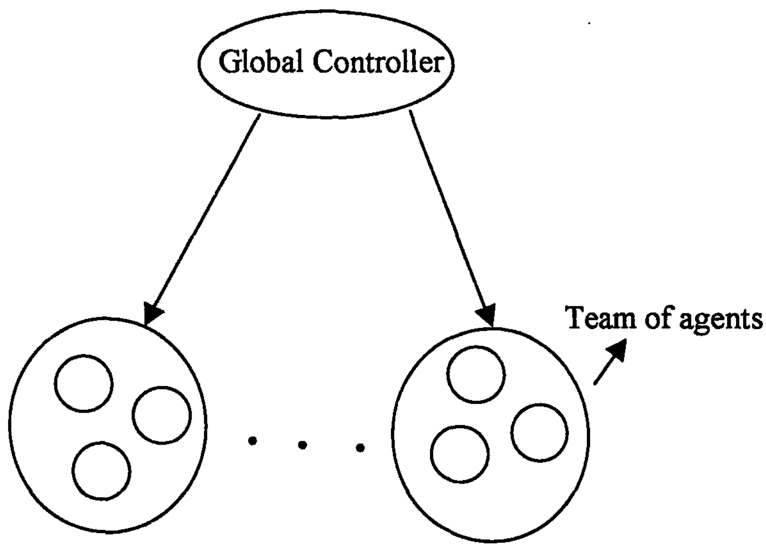


Figure 2.3. A schematic view of team based systems

skills so that a range of products can have all their processes performed within one product team. Project teams are more flexible and organised according to the needs of a certain project. The resources in a project team are collected in such a way that all the operations needed for the project can be performed within the team. The team based approach is currently being implemented in one of the Eureka projects (Rahimifard and Newman 1998; Sabuncuoglu and Toptal 1998) for manufacturing SMEs (Small Manufacturing Enterprises).

In the light of all these discussions, decomposition stated earlier in the thesis should be viewed as a mean or an approach that can be used to implement a distributed scheduling structure in practice. As discussed in Ovacik and Uzsoy (1997), the scheduling problem is divided into smaller subproblems along with some dimensions (time-based and set of scheduling entities such as operations jobs and workcenters). Each subproblem is solved in a systematic manner by a central mechanism (e.g. shifting bottleneck procedure). The aim is to get a feasible solution for the original problem by optimizing a specific performance measure. In a distributed structure, however, the local planners or agents try to optimize their conflicting goals by considering their preferences and local information sources. Solutions are obtained for decision units independently from each other. The manager agent or global planner acts only if there is a conflict between local decision units.

In general, distributed problem solving is used to tackle problems that are difficult to solve in any environment. Decker (1987) emphasizes the essentiality of control and communications in distributed problem solving. The three dimensions that control varies are; cooperation (from fully cooperative systems to antagonistic systems), organization (the amount of hierarchy and the relations between different layers), dynamics (whether the organization is static or changes from problem to problem). Communications are specified through paradigm (usage of global memory or message passing), content (relevance, timeliness and completeness of the communication) and protocol (e.g. network protocols, contract nets, atomic transactions, etc). Evolution of control architectures is further analysed by Dilts, Boyd and Whorms (1991). They present four basic forms (i.e. centralized, proper hierarchical, modified hierarchical and heterarchical). In another paper, Crowe and Stahlman (1995) classify the distributed shop floor control structures as hierarchical, pure heterarchical and quasi-heterarchical. They define four distributed control strategies (sequencing, job bidding, negotiation and co-operation). In another study, Sucur and Coskunoglu (1995) divide the AI-based scheduling systems into three categories (i.e. constraint directed scheduling systems, expert-based scheduling systems and distributed scheduling systems) and review the basic studies.

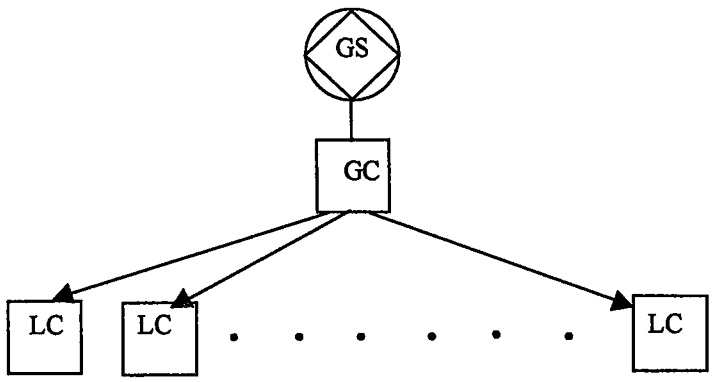
In the next section, we analyse the distributed scheduling research along two dimensions: information flow and communication mechanism between intelligent entities. We also offer a detailed classification scheme considering the most of the scheduling attributes.

2.2. Classification of Scheduling Systems

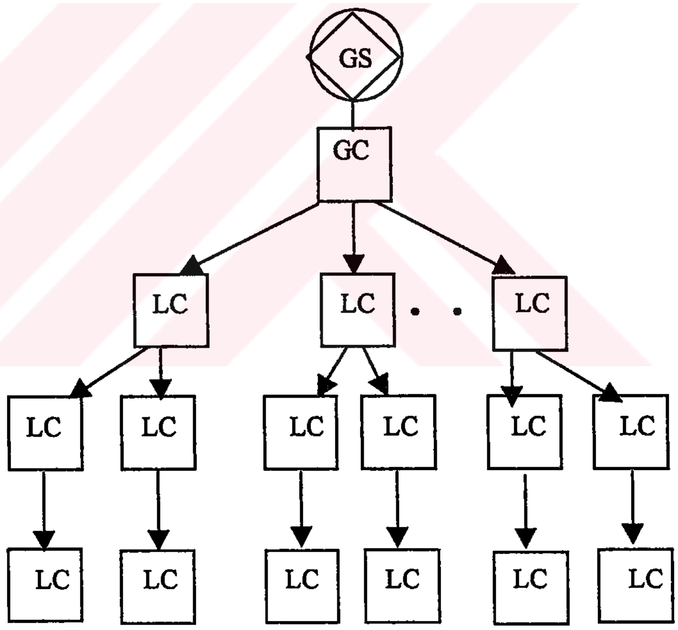
In the past, the scheduling research was classified in terms of static vs. dynamic, deterministic vs. stochastic, on-line vs. off-line. Based on the information technology, new classification criteria can be defined in terms of the flow of information and communication between decision-makers. With respect to the information flow, we can identify two types: centralized (or hierarchical) systems (Figure 2.4) and decentralized (or distributed) systems. In terms of communication between decision-makers, we can identify four mechanisms: bidding, iterative bidding, negotiation, co-operation, domination and iterative refinement.

Communication Mechanisms

In a *bidding* mechanism, a new job (part) introduced to the system broadcasts its arrival and requests bids either by itself or by the help of a manager agent. The agents (machines or cells) prepare bids considering their own capabilities. Those, which are



a. Single Layer Centralized



b. Multi-layer Centralized

Figure 2.4. Centralized Scheduling Architectures

incapable, do not bid. The best bid is selected according to some selected criteria. In a regular bidding mechanism there is no guarantee that parts with higher priorities (i.e., urgent jobs) will be processed first. Yang, Barash and Upton (1993) propose an aggregate bidding scheme where they give higher status indices to urgent parts. This forces machines to prefer processing urgent parts even by preemption.

The bidding process can be *iterative* in which case the bids are revised under the light of new information gained from the results of the first bid. According to the *negotiation* mechanism, agents at the same hierarchical level communicate with each other or exchange information to find other agents to execute processes of a certain job that they cannot do by themselves or to compete for winning the jobs. As a result of the negotiation process, the agents can prepare joint bids or some agents can withdraw their bids (Crowe and Stahlman, 1995).

In the *co-operation* mechanism, the agents collaborate to achieve a better overall system performance. This means that as opposed to the negotiation case, in which agents try to maximize their local goals and take help from each other for the processes, the agents in the co-operation case may choose a second best policy for their local goals for the sake of the overall system objectives (Crowe and Stahlman, 1995).

Iterative refinement approach operates by exchanging the schedule information between different agent types who are responsible from their own constraints and thereby solving conflicts or revising the existing schedule towards better system performance. In a *domination* mechanism, a higher level agent or module decides on the schedule and lower levels implement these decisions. In other words, the decision at each level becomes a constraint for the lower levels. Most of the existing hierarchical scheduling systems are based on this mechanism.

Information Flow structure (Centralized vs Decentralized)

In terms of the information flow, we call a system as centralized (or hierarchical) if a master scheduler develops a schedule for the entire system and local controllers implement the scheduling decision through controller. As seen in Figure 2.4, the flow of information is basically from top to bottom and there is a domination based communication mechanism in these systems. The studies proposed by Chang and Luh (1997), Hax and Meal (1975) could be good examples for this so called the hierarchical control architecture. In centralized systems, it is not only difficult to keep track of all the local needs of lower units but also to update the schedule. Today's manufacturing environments are so dynamic and stochastic that a new schedule may not satisfy the needs of the system even right after the release to the shop floor. The revision process is relatively easier in decentralized systems since they are less sensitive to random events. In centralized systems, if the master scheduler fails at some point then the system may not operate. Decentralized systems, however, are fault-tolerant and continue to function, even if they are less good, in spite of a change in some of the schedule. Therefore, it is easy to maintain and modify an existing schedule in decentralized systems. Moreover, decentralized systems are extendible since new elements can easily be added to the system for improved functionality. As also stated by Dilts, Boyd and Whorms (1991) decentralized systems are also reconfigurable and adaptable. Because control strategies can be developed whenever changes occur. For example when a machine fails, the parts that will be processed on this machine can easily be rerouted to alternative machines.

Decentralized systems also reduce the complexity of computer systems. As discussed by Chiu and Yih (1995), the complexity in a hierarchical structure increases with its size resulting high costs of development and maintenance. Decentralized systems also contribute to the motivation of the employees by giving autonomy to employees or local decision-makers and letting them influence decisions of the organisation. Moreover, they provide flexibility and reactivity in manufacturing which are helpful to customer driven product design and manufacturing (Hvolby and Hφjbjerre, 1994).

In today's competitive world, shortened development and manufacturing lead-times are gaining importance. Decentralized systems are useful in the sense that they provide shorter lead-times by reducing the complexity, deriving a feasible solution in an acceptable duration. There are some disadvantages of decentralized systems such as a lack of global view, hardware and software are not yet available for full functionality. As of now, there is not a fully implemented distributed scheduling system in practice. There is also a tendency for duplicate and potentially conflicting data, possibility of deadlock in decentralized systems. This is because agents know only a limited knowledge about each other, global information is not easily accessible and data is not located where needed (Crowe and Stahlman, 1995). In these systems local goals may be optimal but this does not guarantee the optimal solution of the entire system. These strengths and weaknesses of both centralized and decentralized systems are summarised in Table 2.1.

Table 2.1 Advantages and Disadvantages of Centralized and Decentralized Systems

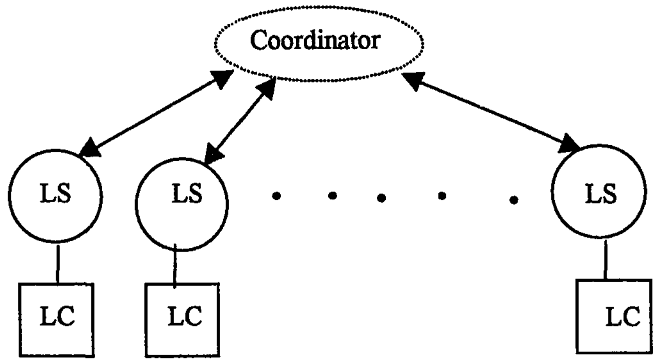
	Advantages	Disadvantages
Centralized	-Global optimization	- Difficult to keep track of local needs of lower units - Difficult to update the schedule - Not much reliable - Complex computer systems
Decentralized	- More reliable - Fault-tolerant - Modifiable - Extendable - Reconfigurable and adaptable - Contribute motivation of employees - Flexible and reactive - Shorter lead times	- Lack of analytical solutions - Unavailability of hardware and software - Tendency for duplicate or conflicting data - Does not guarantee optimal solution for the entire system

Classification of Decentralized Systems

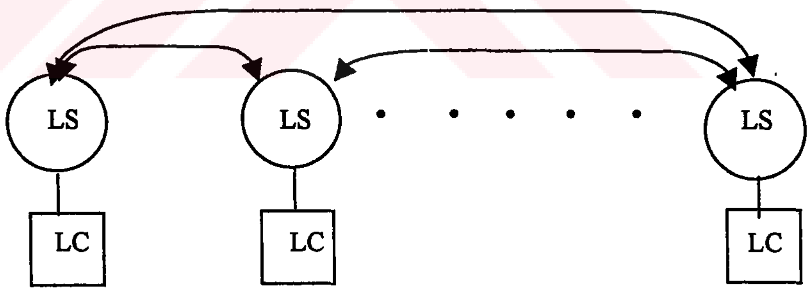
We can further classify the decentralized systems into *pure heterarchical* and *quasi-heterarchical* systems. Pure heterarchical systems contain no organisational hierarchy. Each agent communicates in pairs (Figure 2.5b) or through a coordinator agent (Figure 2.5a.). These systems utilize iterative communication mechanism, which usually results in no more than a feasible schedule (i.e. they do not try to optimize overall system objective). Agents only consider their own goals; neither take help from others, negotiate nor co-operate with other agents. As in Figure 2.5a, the coordinator in pure heterarchical systems helps the communication and extracts the necessary information for the agents.

In a quasi-heterarchical system, the local schedulers make their own schedules considering their local goals. These local schedules are then evaluated within the overall objective of the system by a manager agent. If there is one layer of manager agents then the DS architecture is called as single layer quasi-heterarchical (Figures 2.5c & 2.5d). If there are more than one such layer, then the DS architecture is multi-layer quasi-heterarchical (Figures 2.5e & 2.5f). The role of the manager agent in these systems is to solve potential conflicts, select bids, overwrite the decisions of local schedulers, and finalise the scheduling decisions. Again these schedules are implemented through the local controllers. Communication between agents can be achieved by co-operation, negotiation, bidding or any combination of the above. Since the master scheduler makes the decisions by considering the overall system objectives, there must be co-operation between agents.

The local schedulers can also develop schedules by sharing some of the operations of the jobs via negotiation. In addition, local schedules can be formed by bidding. The number of bidding decisions between layers in the hierarchy shows whether the multi–layer system is with single bid or multiple bids. If the bid decision given in the first layer is given as a bid to the upper layer then, the system is multi-layer with multiple bids.

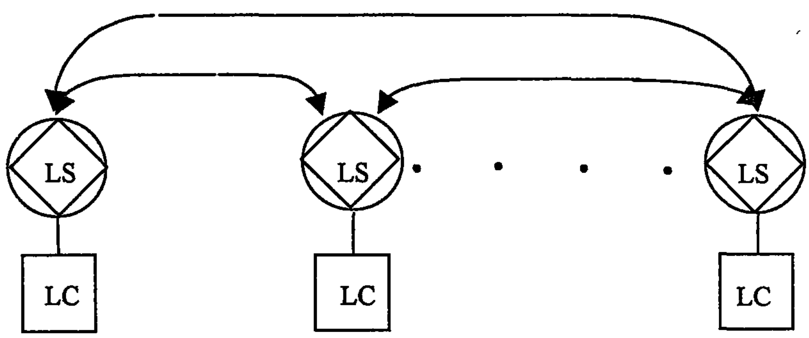


a. Pure heterarchical with coordinator

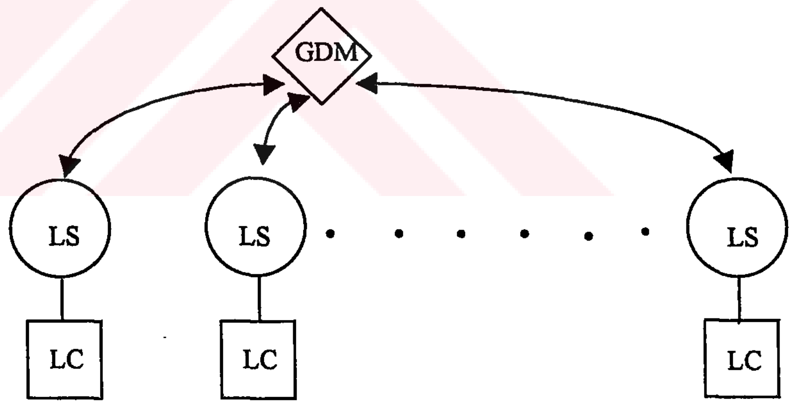


b. Pure heterarchical without coordinator

Figure 2.5. Heterarchical DS Architectures

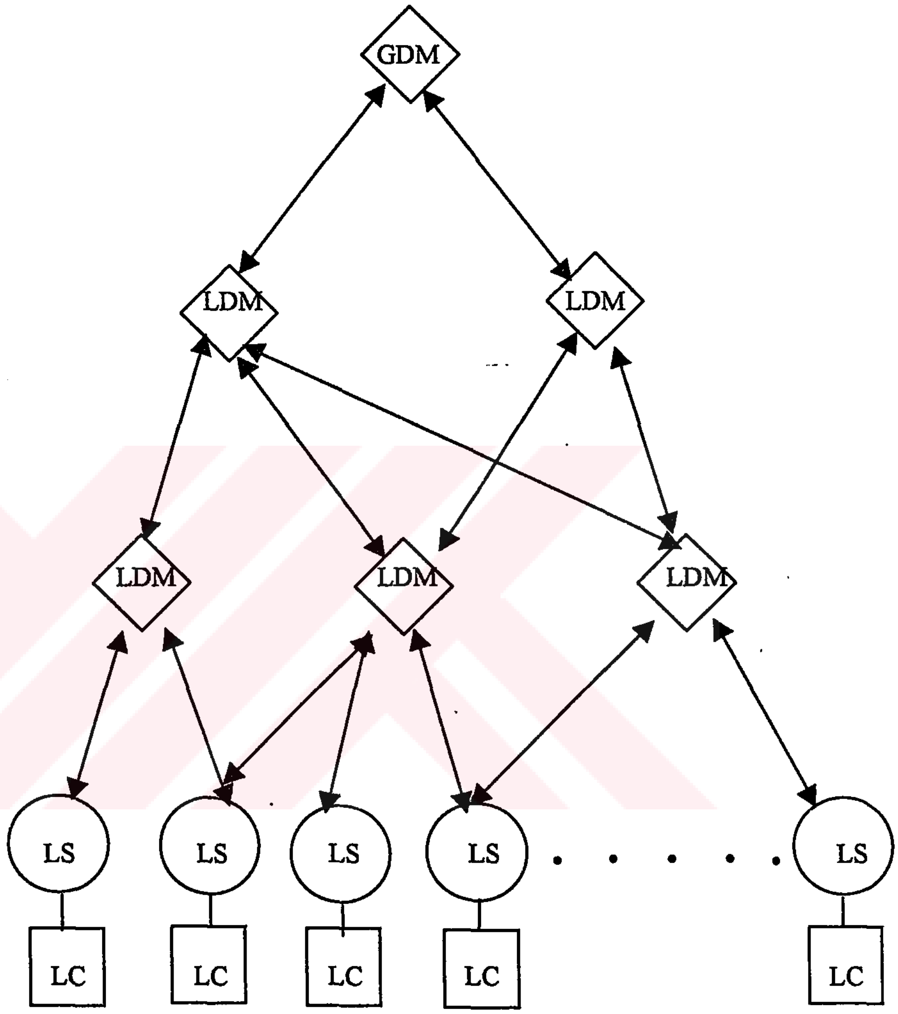


c. Single Layer Quasi-heterarchical without a Separate Manager Agent



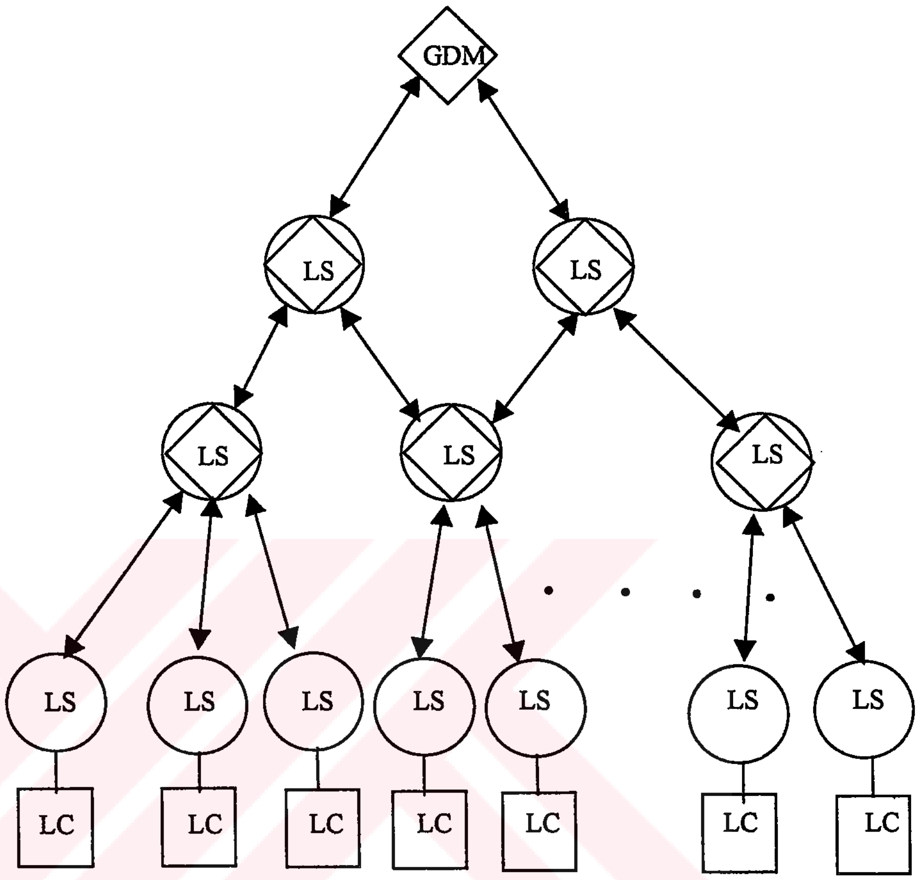
d. Single Layer Quasi-heterarchical with a Separate Manager Agent

Figure 2.5. Heterarchical DS Architectures (Con't)



e. Multi Layer Quasi-heterarchical with a Single Bid (Con't)

Figure 2.5. Heterarchical DS Architectures (Con't)

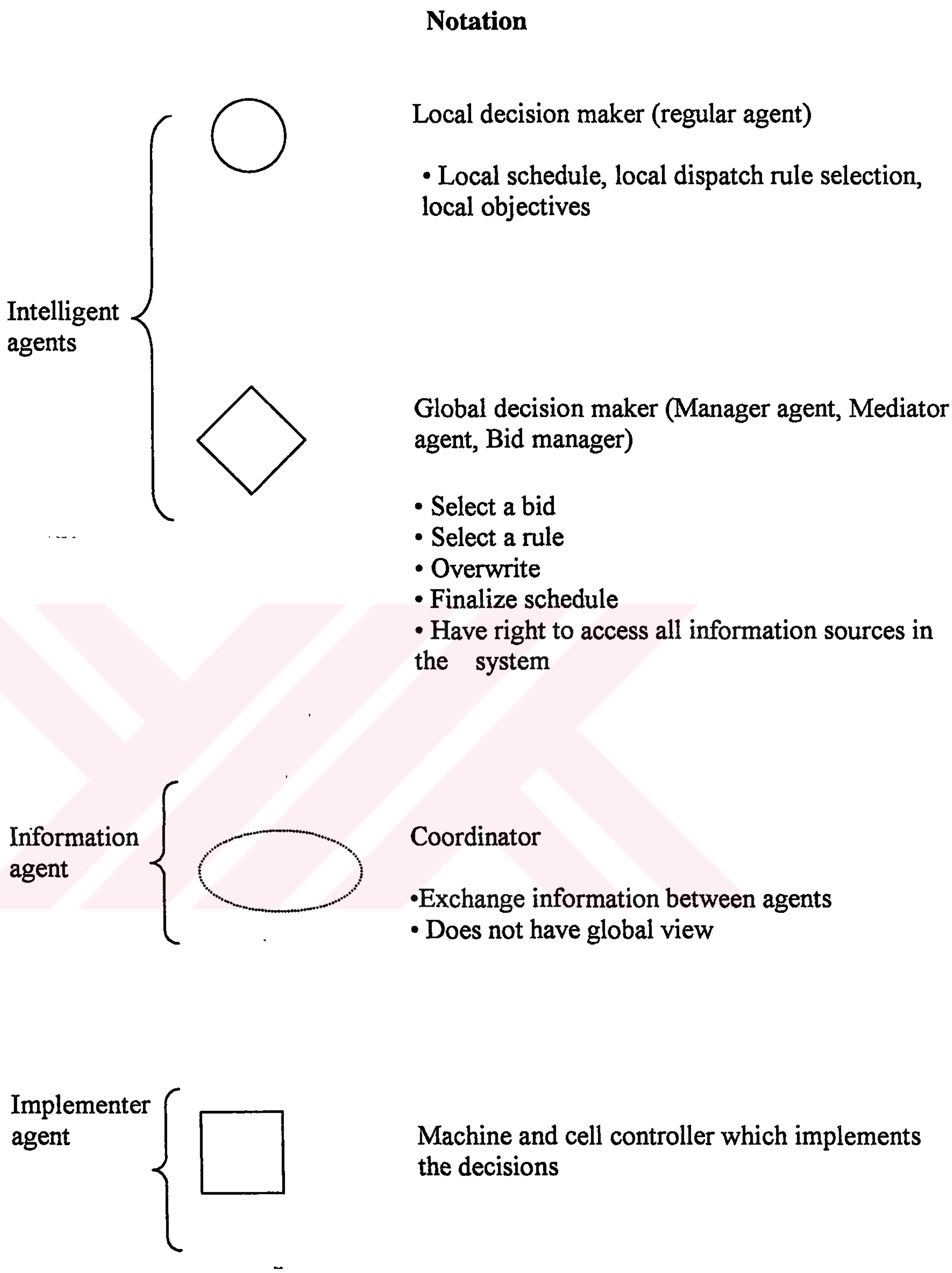


f. Multi-layer Quasi-heterarchical with Multiple Bids

Figure 2.5. Heterarchical DS Architectures (Con't)

Where,

DM = Local Decision Maker
GDM = Global Decision Maker
LS = Local Scheduler
GS = Global Scheduler
LC = Local Controller


Notation
Local decision maker (regular agent)
• Local schedule, local dispatch rule selection, local objectives
Intelligent agents
Global decision maker (Manager agent, Mediator agent, Bid manager)
• Select a bid
• Select a rule
• Overwrite
• Finalize schedule
• Have right to access all information sources in the system
Information agent
Coordinator
•Exchange information between agents
• Does not have global view
Implementer agent
Machine and cell controller which implements the decisions

From another point of view, there is no master-slave relationship between agents in pure heterarchical systems. But in the quasi-heterarchical and centralized systems, there is master-slave relationship in the sense that master scheduler has domination on the slave scheduler either by generating a general schedule (longer time period and less detail) and thereby setting constraints on the slave scheduler or changing the decision of the local schedulers. According to this classification, holarchies are quasi-heterarchical systems.

In Table 2.2 we propose a new classification framework in which the existing studies in the distributed scheduling area are summarised by using the terminology developed in this thesis. We use seven attributes to classify the previous research work: scheduling system /agent/ scheduling generation /method/ communication mechanism /objective/ environment/TBA (Team Based Manufacturing type).

In this notation, scheduling system refers to the type of system (i.e., hierarchical, pure heterarchical and quasi-heterarchical, etc.) The second entry represents the agent that acts independently, shares information with other agents, makes trade-offs between local goals and global objective of the system whenever necessary. In the third attribute, we define the scheduling generation scheme. This can be a detailed schedule in the form of a Gantt Chart or dispatching (i.e. the best dispatching rule is selected for the next time period via simulation). Dispatching rule selection can be executed either in an on-line and off-line mode. In the on-line mode, simulation is used to select the best rule whereas in the off-line mode, simulation experiments are conducted in advance and the information about the best rules and conditions are stored in the database.

The method element in the notation indicates the main procedure or tool used to make the decisions. As discussed earlier, communication mechanism can be in the form of bidding, negotiation, co-operation, domination and iterative revision. Apart from local goals of local schedulers, any system has a global objective such as reducing flowtime, tardiness, WIP, lead-time; increasing utilisation, throughput, etc.

Local goals can either be the same as the global goal or they can be different. Environment is another attribute, which defines the system where the prospective scheduling system is intended to work. The last element of the notation is TBA which refers to the type of teams as defined in (Rahimifard and Newman 1998). It takes two values: R1 refers to Team based Manufacturing with Different Production Capabilities and R2 represents to Team based Manufacturing with Similar Process Capabilities. We now give a review of the existing studies in the distributed scheduling area.

2.3. Analysis of the Existing Studies

The existing studies in the literature are summarised below. They are presented beginning from a very simple heterarchical structure (pure heterarchical system), continuing with the single layer quasi heterarchical systems in the order of publication year. Finally multi layer quasi heterarchical systems with single and multiple bids are reviewed. A list of these studies is given in Table 2.2.

Sycara, Roth, Sadeh and Fox (1991) propose a pure heterarchical system in which customer orders are represented by agents. Each order agent also monitors some of the shared resources and the resources that it only uses. In this system each agent first determines the probabilistic activity demand for each unscheduled activity of the order on each resource. If the order has to visit some resources more than once, then it sums demand of these activities on this resource. This demand reflects an agent's need for a resource over time and is presented to the monitoring agent who then calculates the overall aggregate demand in time for that resource. The peak of the aggregate demand determines the critical time interval of the resource and the activities which are scheduled within this interval are called critical activities. Each agent calculates a survivability measure for its critical activity; which represents the probability that this start time will not result in a capacity constraint violation. The monitoring agent of each resource gives reservations according to the requests of the agents based on survivability measure. Backtracking is also used whenever an agent can not find feasible schedule. The information flow between the monitoring agent and order agents in an iterative manner continues until a feasible schedule is developed. The monitoring agent decides on the final schedule of the resource.

The second pure heterarchical DS architecture is proposed by Sycara and Liu (1993). The authors employ three kinds of agents: order agents, resource agents and a manager agent. An order agent represents a customer order and a resource agent is assigned to a resource (i.e. machine). The scheduling problem is defined within capacity constraints of the resources, precedence constraints and earliest start and latest finish times. An order agent is responsible for eliminating precedence constraint violations and a resource agent eliminates capacity constraint violations. There is also a manager agent who is responsible for coordination and communication of the order and resource agents (Figure 2.5a). According to their procedure, each order agent calculates time boundaries (time between latest finish time and earliest start time) for the activities under his control. Each resource agent calculates contention ratio, which is the ratio of sum of the processing times of all activities on that resource to the sum of their time boundaries. If this contention ratio is greater than a threshold value, then the machine is declared as bottleneck. Each resource agent generates its schedule using the EDD rule and the nondelay schedule generation scheme. The resource with the greatest contention ratio is considered as primary bottleneck machine whereas other bottleneck machines are viewed as secondary bottleneck resources. In the procedure, secondary bottleneck resources update their schedules according to primary agent's schedule in such a way that precedence constraint violations are minimised. Communication between the agents (both order and resource agents) is coordinated by the manager agent (coordinator) in an iterative manner until a feasible schedule is developed. In the later studies, the authors (Sycara and Liu, 1994,1995) proposed a new threshold mechanism on regular changes of same conflicting pairs so that loop prevention is ensured. They also discuss why the proposed system minimises the

Table 2.2. List of the Studies in Distributed Scheduling and their Characteristics

Authors	Scheduling System	Agent	Scheduling Generation	Method	Communication Mechanism	Objective	Enviro.	TBM
Shaw (1987)	Single layer Q.H. without a separate MA	Machine (GT cell)	FCFS dispatching	Bidding algorithm with SPT	Bidding	Meeting due Dates	FMS	R2,R1
Shaw and Whinston(1998)	Single layer Q.H. without a separate MA	Machine (GT cell)	FCFS Dispatching	Bidding algorithm with EFT	Bidding	Meeting due Dates	FMS	R2, R1
Parunak (1988)	Multi-layer Q.H. with multiple bids	Machine (GT cell)	FCFS Dispatching	Bidding algorithm with EFT	Co-operation + Job bidding + negotiation	Meeting due dates + balancing the load	FMS	R2
Burke and Prosser (1991)	Multi-layer Q.H.	Machine	Detailed schedule	Depth first search and Bactracking	Co-operation+ Negotiation	Meeting due dates + balancing the load	Job shop	R2
Sycara, Roth, Sadeh and Fox (1991)	Pure heterarchical with coordinator	Order agent + Monitoring agent	Detailed schedule	Iteration	Iterative revision	Not explicitly stated	Not given	R1
Lin and Solberg (1992)	Pure heterarchical	Part agent+ Resource agent	Detailed schedule	Bidding	Co-operation + Negotiation	Several objectives	Not given	R1,R2
Hadavi, Hsu, Chen and Lee (1992)	Single layer Q.H. with a separate MA	Machine	Dispatch rule selection	Dispatch rule select. via simulation	Co-operation	Meeting due dates	Job shop	R1,R2
Sycara and Liu (1993, 1994, 1995)	Pure heterarchical with coordinator	Machine + Order agent	Detailed schedule	Iteration	Iterative revision	Meeting due dates	Job shop	R1
Chiu and Yih (1994)	Single layer Q.H. with a separate MA	Machine + Manager agent	Dispatching	Dispatch rule select. via GA and off-line sim.	Co-operation	Several objectives	FMS	R1, R2
Duffie and Prabhu (1994)	Single layer Q.H. without a separate MA	Machine	Dispatching	Dispatch rule select. via on-line sim.	Co-operation	Minimizing tardiness	FMS	R1, R2
Agarwal, De and Wells (1995)	Single layer Q.H. with a separate MA	Machine (GT cell)	Detailed schedule	Bidding with SPT and customer approval	Co-operation + bidding	Meeting due dates	FMS	R2
Duffie and Prabhu (1996)	Single layer Q.H. without a separate MA	Machine Agent	Dispatching	Dispatch rule select. via on-line sim, and bidding	Co-operation + bidding	Minimizing tardiness	FMS	R1, R2
Chung, Park, Kang and Park (1996)	Single layer Q.H. without a separate MA, (Centralized in LP case)	Machine	SPT dispatching	Bidding with SPT and LP model	Co-operation + bidding	Meeting due Dates	FMS	R2, R1
Maturana and Norrie (1996)	Multi-layer quasi-heterarchical with single bid	Product + part + machine	Dispatching	Bidding	Co-operation + bidding	Meeting due Dates	Job shop	R2
Kutanoglu and Wu (1998)	Single layer Q.H. with a separate MA	Job + Manager agent	Detailed schedule	Iteration and bidding	Iterative revision + bidding	Meeting due dates + minimizing total tardiness	Job shop	R1, R2
Oguz (1998)	Single layer Q.H. with a separate MA	Machine+ Manager agent	Dispatching	Decomposition	Cooperation	Meeting due dates + minimizing makespan	Job shop	R2

number of late jobs.

Another pure heterarchical system is proposed by Lin and Solberg (1992). The control modules in this system consists of part agents, resource agents, intelligence agents, monitor agents, communication agents and database management agents. In preparing a schedule, part agents and resource agents employ a priced based bidding mechanism. The objectives of a part agent represents the needs of the customer (i.e. minimization of flow time, cost of production etc.) and each resource (i.e tool, AGV, m/c) agent has a charge price for different resource time slots. Part agents try to maximize their objectives while minimizing the total cost they pay. When a part enters the system, the part agent associated with this part, announces a bid to select its resources. If the bottleneck resource is the machine and it is the only resource that can perform a particular operation, the part agent pays its price and tries to reserve other resources (i.e. AGV, tool) for that period. After the resources prepare their bids, the part agent acts as a manager agent and selects the best bid. If a certain resource is demanded by many part agents, the resource agent will charge an increased price in later periods to decrease the total demand. By that way, the system smooths itself and each resource reaches an equilibrium price. There is a strong negotiation scheme through which parts agents compete with each other to reserve resources. Part and resource agents have their own intelligence files to give simple decisions but if they need different algorithms, they can communicate with intelligence agents where different algorithms are stored. Monitor agents watch the information flow and provide global information. Communication agents arrange the arrival and departure of agents and distribute messages. Database management agents maintain information for the system entities.

The second most used DS architecture in the literature is quasi heterarchical. The earliest study in this area is due to Shaw (1987) who developed a single layer quasi heterarchical scheduling system without a separate manager agent (Figure 2.5c). In this study agents represent FMS cells. Each agent acts as a bid manager to route the job to the cell for the next task or set of operations. FMS cells, which are capable of performing this task, offer their bids to the bid manager. In cases of new job arrivals, any idle cell in the system acts as a bid manager. When cells receive a task announcement message from the communication network, they calculate the expected finishing time (EFT) using the processing time, travelling time and expected waiting time information. This information (or bid) is sent back to the bid manager who makes the final decision. If more than one task-announcement come to an FMS cell, then the cell ranks the tasks according to some local criteria (i.e., set-up time, job urgency etc). As also stated in Table 2.2, job bidding is the main communication mechanism in this approach. In the follow-up study, Shaw and Whinston (1988) employed the SPT (shortest processing time) rule as the bidding criterion instead of EFT in a Petri net-based communication protocol.

Another single layer quasi heterarchical architecture is proposed by Hadavi, Hsu, Chen and Lee (1992). In the system, which is called REDS, there are six modules: The order handler module is where the new orders are entered and preprocessing is performed. Event handler module takes the preprocessed order from the order handler and keeps it in a job shop pool. The sequencer module, which is a part of the event handler, generates a dispatch list for every machine. The sequencer and capacity watcher can release the jobs from the event handler's pool through the order watcher. Event handler passes the detailed schedule to shop floor and updates the information in database. In this system each agent represents a machine and each machine selects its best dispatching rule via simulation. A list of selected rules is stored in sequencer module. The sequencer, which employs these rules in scheduling, acts as a bid manager (Figure 2.5d). The communication is the co-operation type since the sequence decisions are made considering the overall system objectives (Table 2.2). The authors also point out that a good scheduling software must generate schedules that satisfy constraints regarding management objectives, tool availability, vendor tardiness; schedules that are flexible and must be able to present a schedule even if there is incomplete information. They should also be quite robust while meeting management objectives. Here robustness implies that schedules generated are feasible and that scheduler can react to changes on the shop floor by minimally revising existing schedules.

Chiu and Yih (1994) propose another single layer quasi heterarchical scheduling system in which scheduling rules are selected based on simulation experiments. In this study, the authors showed that the proposed heterarchical system outperforms a dispatching based centralized system, which uses a single dispatching rule for the entire horizon. Two types of dispatching rule selection are used: look-ahead approach and knowledge-based approach. The look-ahead simulation approach determines the best dispatching rule just prior to its implementation in the real system (i.e., on-line) while the knowledge-based approach retrieves the best rule combination from its database when the system reaches to a predetermined state. In this off-line approach, genetic algorithm and simulation are used to select the best combination of dispatching rules and the results and corresponding states are stored in the database. Note that this rule selection procedure is repeated for many possible states. In this application of DS, study, agents represent machines and each machine agent decides on its local goal and a set of dispatching rules to achieve this goal. But the final selection is always made centrally by the manager agent. From this point of view the scheduling system is single layer quasi heterarchical with a separate manager agent (Figure 2.5d). Since the best combination of rules is selected to maximise the overall system performance objectives, the communication mechanism can be considered as the co-operation type.

The study by Duffie and Prabhu (1994) is also an example to single layer quasi heterarchical system without a separate manager agent (Figure 2.5c). Each agent who represents a machine determines its local goal and a set of dispatching rules independent from others. Again, simulation is used to select the best combination of dispatching rules whenever a decision is needed. This system can be viewed as the on-line version of the Chu and Yih's study since the best rule is not selected from the database but from the results of simulation conducted at that point in time. Note that in the Hadavi et. al's work (Hadavi, Hsu, Chen and Lee 1992), the sequencer that acts as the manager agent has the right to change the scheduling decisions made by the machine agents. However, in the both Chiu & Yih and Duffie & Prabhu works, the best rules or combinations are selected by the manager agent given the rule candidates proposed by the local agents. Again, the communication mechanism is the co-operation type since the overall system objectives are achieved. In their later study, the authors (Duffie and Prabhu 1996) consider alternative machines and use a bidding mechanism to make the dispatching rule selection.

Agarwal, De and Wells (1995) propose a distributed scheduling system for flexible cellular job shops. Agents represent cells and they are formed according to group technology principles, i.e. each cell produces parts belonging to a certain part family. Apart from these agents (or local schedulers), there is an overall manager who requests bids and coordinates the communication between the cells. It also evaluates the offers and makes reservations. Each cell generates its own schedule in such a way that maximum utilisation within the cell is achieved. The bidding process is executed in four phases. In phase 1, each cell tries to add a new job to the end of the existing schedule. Since the machines are assumed to be versatile all the operations of a job can be performed on a single machine by a proper tooling. Each cell makes its own offer in terms of quoted deliveries. The overall manager decides on the best offer by communicating with the customer. If the bids from the cells are not acceptable in terms of due date, phase 2 is executed and cells prepare the bids by using their less utilised machines. If the bids are not still acceptable, the phase 3 of the algorithm is invoked and the operations of the job are allocated among the machines of the cell. If not (i.e., a feasible offer is still not found), in phase 4 the manager agent requests a bid for each individual operation of the job. In case of no feasible offer for the job, either the job (or customer order) is rejected or it is placed back to the bidding process after negotiating its due date with the customer. Since the overall manager has the right to make the final decision, this system is an example for a single layer quasi heterarchical system (Figure 2.5d) and utilises the co-operation and bidding type communication mechanisms.

Chung, Park, Kang and Park (1996) use both a bidding algorithm and an LP (linear Programming) model in their distributed scheduling system. Three cases are identified. The first case arises when an operation of the part is completed on a machine and is ready to be routed to one of the alternative machines. This problem is defined as the machine assignment problem. The second case arises when a machine is idle and there are parts to be processed on this machine. This problem is called the part assignment problem. The third case is rather general in the sense that there is more than one machine to process parts and there is more than one part to be processed on each machine. Third case is solved by an LP formulation. A bidding algorithm is used in the first two cases. To implement bidding, the authors develop distributed software, which consists of communication server, agent module, local scheduler, and state monitoring module and command dispatcher. Agent module acts as a bid manager when the machine requests a bid from other agents and acts as a bidder when this cell offers bids to another bid manager. Communication between cells is accomplished by communication servers. The bids are prepared by the agent module of that cell. In this system authors use SPT as the bid criterion. Scheduling in each cell is accomplished by an on-line dispatching mechanism. Since the overall system performance is optimised by an LP formulation, this system can be said to have co-operation and bidding communication mechanism. For the first two cases the structure resembles that of a single layer quasi heterarchical system without a separate manager agent (Figure 2.5c), however for the third case the LP formulation results in a global schedule and is directly implied. So the third case can be considered as centralized (Figure 2.5a). Their system is different from the Shaw's (1987) due to the SPT bidding rule (Shaw uses the FCFS rule) and due to multiple bid requests. The authors also consider an AGV based material handling system in their study.

Kutanoglu and Wu (1998) have recently proposed a single-layer quasi heterarchical system using the notion of combinatorial auction. There are two kinds of agents: Jobs are the agents who bid for discrete time slots on machines. These time slots are as large as the time unit used in expressing the processing times and are associated with prices determined by a function. Auctioneer who acts as a manager agent selects the job for a certain time slot on a machine and updates the prices associated with each time slot on each machine. Bid by a job is represented as a function of total weighted tardiness of the job and total payment in terms of resource usage (i.e. utility function). Each job agent solves its locally constrained utility maximisation problem to find the best resource time slots. Having collected these requests, the auctioneer updates the individual machine time slot prices to reduce resource conflicts. These iterations continue until either a certain limit of number of iterations or number of conflicts is reached or a feasible schedule is found. If a feasible schedule is not found until a certain time limit, then the manager agent solves the conflicts.

The first multi-layer quasi heterarchical DS system is given in Parunak (1988). According to this approach, the global scheduler prepares a schedule of the entire system for some time interval (i.e. months). Each workcell (i.e. department or GT cell) in the lower levels of the hierarchy adds the necessary details to the schedule using a bidding algorithm (i.e. workcells prepare bids for the tasks defined at this higher level). There may be several workcells embedded in a workcell. More than one workcell can also negotiate to share for the tasks of a job or operations of a task. In this system, bidders may be either individual agents (workcells) or a group of agents. Therefore there are multiple bids (Figure 2.5f). The bidding criterion is meeting due dates while preserving the load balance of the system. Apart from bidding and negotiation, there is also the co-operation type communication among the agents since a higher level workcell or the global scheduler makes decisions to optimise the overall system performance.

Another multi-layer quasi heterarchical system was introduced by Burke and Prosser (1991). In DAS (Distributed Asynchronous Scheduler) there is an hierarchy of agents through which the scheduling problem is solved by messaging. The three classes of agents are operational (O-agent), tactical (T-agent) and strategic (S-agent). Each O-agent corresponds to a machine. A T-agent represents an aggregation of similar resources and is responsible for the load balancing of its resources. The S-agent has access to the whole scheduling system and is responsible for releasing new work and resolution of conflicts. The scheduling problem is solved via a constraint satisfaction method using depth first search algorithm and backtracking. The O-agents'a goal is to schedule an operation on its resource. If an O-agent cannot schedule an operation (i.e., can not satisfy the constraints for that operation), then this conflict is informed to a higher agent in the hierarchy, basically the T-agent. If the problem is over constrained that the T-agent can not solve, then it is delivered to the S-agent. The S-agent has the maximal right to relax constraints for an operation. The messaging mechanism allows an efficient environment for co-operation between agents. There is also negotiation between resources under the same T-agent. Because to satisfy the constraints of a new operation, some of the operations may be changed between similar resources through the T-agent.

Maturana and Norrie (1996) develop a distributed task planning and coordination architecture, which has multi-layer quasi heterarchical structure with a single bid (Figure 2.5e). This system contains a template mediator, data-agent managers, and active mediators. There is one template mediator (TM) which has an infinite life and global knowledge of the system. But there are a number of data-agent managers and active mediators. A data agent manager corresponds to a first-level subtask or a product and is created by TM. An active mediator corresponds to a second-level subtask or a part and is found by DAM. Template mediator divides a task into its first-level subtasks according to a list of predetermined attributes. Each of these subtasks is assigned to a data-agent manager (DAM) who is responsible for dividing the first-level subtasks to second-level subtasks, which are assigned to active mediators. Each active mediator (AM) requests from the template mediator to search for suitable resources to accomplish the tasks. TM identifies candidate resources for this task considering overall system performance. Then AM requests the bids from these resources and makes the selection to achieve local objectives. This decision is then revised by DAM and TM using the due-date information and overall system objectives. Thus this DS architecture is based on co-operation and bidding mechanism (Table 2.2).

The workstation architecture in holonic manufacturing systems proposed by Wyns, Brussel, Valckenaers and Bongaerts (1996) can also be considered as a multi layer quasi-heterarchical system. Agents correspond to order holons, workstation holons and resource holons. An order holon defines the product to be made and manages the precedence constraints between all operations of an order. It contains the information such as order due date and importance. Each workstation holon has a resource allocation holon and a process-control holon. The resource allocation holon assures safe reservation and allocation of resources so that the resource is not assigned to more than one operation for the same time interval. The process-control holon receives a job description and the resource-allocation information stating which resources can be used. The resource holon holds resource specific information. Reactive scheduler acts as a manager agent. The reactive scheduler gives a nominal planning to order holons. Order holons request reservations for their operations from the resource allocation holons according to reactive scheduler's specifications. Order holons and resource allocation holons cooperate to accomplish these specifications. After an agreement, the order holon passes the data related with allocations to the process-control holon which controls the execution of the process.

In a recent study, Oguz (1998) developed a single layer quasi heterarchical system with a separate manager agent. In this system, a central scheduler (manager agent) generates an off-line schedule for the entire system. This solution is then used to guide the dispatching or scheduling decisions given at the machine level. Each machine agent is responsible to for the decisions at trigger points (i.e. arrival of a new job, completion of an operation, machine breakdown etc.) in an online fashion. The author proposes different ways for local schedulers to handle the scheduling task. In the first approach, each machine generates an optimum sequence for all of its unscheduled operations. If the first operation in the sequence is available, then it is assigned to the machine. If it is not in the sequence, they choose an operation that has become available before its release time. If such an operation can not be found, they use the predetermined cost function to select a job. In the second approach, each machine uses the sequence generated by the central scheduler and dispatches the first operation in the sequence if it is available. Otherwise it chooses the first operation in the sequence that is available. Alternatively, the author proposes local schedulers to use minimum slack algorithm within the boundaries generated by the global schedule or FIFO (First in First Out) algorithm. Although the original problem seems to be decomposed into single machine problems, decomposition does not bring much benefit in terms of solution complexity. Because the global scheduler should generate the entire schedule anyway. But the proposed approach enables the system to implement the schedule according to original sequence and respond to disruptions as quickly as possible. The algorithm is modelled in a distributed object oriented structure. Giving right to local schedulers to reschedule, there is some level of autonomy in the proposed scheduling systems. However, these local schedules can not incorporate their own objectives into the schedule and their decisions are limited by the global schedule. Thus the autonomy is somehow partial. Their social ability and the interaction between each other is also limited. By decreasing the dependency of local schedulers to the global schedule and letting them consider their own goals, intelligence of these entities can be increased.

2.4. Conclusions and Future Research Directions

In this chapter, we reviewed the existing studies in the distributed scheduling literature within the classification framework proposed in the thesis (Table 2.2). The strength and weaknesses of distributed systems (and centralized systems) are also discussed.

During this survey we noted that distributed paradigm is relatively a new concept in scheduling. We have only identified fourteen papers published in the last ten years. Among them, twelve of these studies use the quasi heterarchical architecture and two of them are based on the pure hetererachical architecture. Some of these distributed systems are extended and modified over the time. It seems that the quasi heterarchical system is the most dominant DS architecture in the literature.

In all these systems different combinations of communication mechanisms are utilised. In the pure heterarchical systems (Shaw, 1987, 1988), the communication is usually achieved by an iterative refinement approach. Co-operation is the preferred communication mechanism for the quasi heterarchical systems. We observed that job bidding is also used especially in the recent studies.

Another observation is that most of these systems optimise due date related performance measures. Flow time and cost based measures are not studied well. The nature of the manufacturing environment is either a job shop or an FMS.

We also noted that the proposed distributed scheduling systems have not been adequately tested neither in a simulated environment nor in a real manufacturing environment. Even, they have not been compared with centralized scheduling systems. Thus, the strengths and weaknesses discussed in Table 2.1 and the text have not been acknowledged yet.

Except for few cases, the scheduling task is conducted by dispatching. That is, a full schedule is not generated in advance (i.e., off-line scheduling), but rather operations are scheduled one at a time as the system state changes (i.e., on-line scheduling).

In almost every system, machines and/or orders are considered as the agents. Their communication is usually carried out via a manager agent.

In all these studies, the schedule generation aspect is only considered. The monitoring and control aspects of scheduling are not adequately addressed. Thus, we do not know the performance of the distributed scheduling systems in dynamic and stochastic environments that call for more reactive scheduling systems.

Since these systems are not actually tested in real manufacturing environments, the efficiency of their communication systems is not generally known. In a recent study, Veeramani and Wang (1997) discuss the importance of the communication system performance (i.e., auction throughput, auction time etc.). We believe that researchers should explore this point in the future studies.

In spite of the all theoretical works being done in the literature, there are also studies in the commercial market to design software packages that have distributed capabilities. These systems are sometimes sponsored by research institutes of different countries. For example, the next generation of PREACTOR and MMS scheduling systems are being developed as a part of Eureka projects.

Finally, we think that future distributed scheduling systems should designed by using the quasi heterarchical structure in order to achieve a global system performance. Independence of agents in heterarchical systems has certain advantages but the overall system performance or optimization of the global objectives should not be totally left out for the sake of these advantages. The bidding algorithms seem to produce reasonably good results when there are alternative machines and alternative cells. Thus, the prospective DSs should also incorporate this communication mechanism.

In the future studies, there is also a need to measure the trade-off between the complexity and the effectiveness of communication systems. The future scheduling systems must also incorporate advantages of both centralized and decentralized systems. Thus, we encourage researchers to develop more effective quasi-heterarchical systems in future DS applications.

CHAPTER 3

BIDDING BASED ALGORITHMS FOR PROCESS BASED SYSTEM

In this chapter, we describe three bidding-based distributed scheduling algorithms for systems with process teams. In the previous chapter, we reviewed the existing studies and proposed a classification framework. In this part of the thesis, we extend this study by developing three algorithms with different characteristics and testing them under various operating conditions.

3.1 Algorithm B1 (Operation initiated bidding algorithm)

This algorithm is developed for the systems where each individual resource agent or team has its own goal to achieve, in addition to overall system objective(s). The algorithm is currently designed for machine agents, but it can be easily modified to cells or departments. We consider the case where jobs can have different machine visitation sequences and each machine has different processing capabilities (i.e., the process team concept discussed in Rahimifard and Newman, 1998).

According to the steps of the algorithm, schedulable operations are first ranked by the manager agent (overall system manager). Then the bid is initiated for each operation in the list one by one until each operation is assigned to one machine. During the bidding process, only the machines (resource agents or local decision makers) that are capable of processing the operation prepare their bids and the manager agent selects the best bid using some pre-determined criteria. According to the classification scheme discussed in the previous chapter, this model can be classified as *single layer quasi-heterarchical system with a separate manager agent* (Figure 3.1).

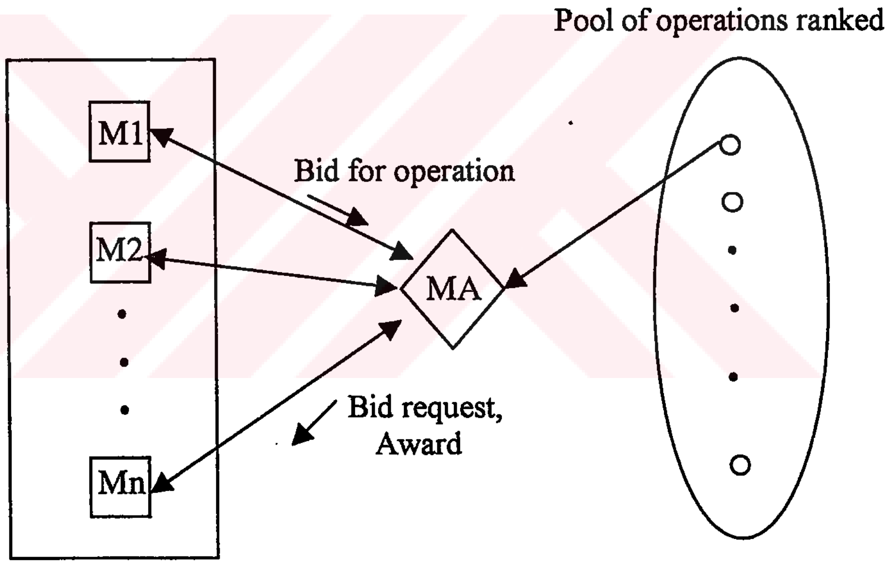


Figure 3.1. A schematic view of Algorithm B1.

During the bid preparation process, each resource agent (or machine) tries to insert this new operation into the first available position in its existing schedule to win the bid. By doing that it may affect one or more operations in the sequence. In the first version (*competitive* version) of Algorithm B1, no *right shift* on other resource agents is allowed for these affected operations. Here, right shift is defined as the movement of an operation to the right in the sequence due to the delay of its previous operation. In the second version *(collaborative* version) of Algorithm B1, right shifts on the other machines are allowed as long as the second best bids are not violated (i.e., the new completion times on these affected machines do not exceed than the completion times proposed previously by the second best bid alternatives).

In the first version, resource agents (or teams) act as real competitors to each other. Hence, one team cannot effect the promised completion time of an operation on another team (i.e. no right shift violation is allowed). In the second version, however, for the sake of achieving overall system objectives, teams can sacrifice from their local objectives to the extend that a delayed operation is completed no later than the one offered by the second best bid for the same operation on another machine.

The main logic of two versions of the proposed algorithm are given in Figure A.1 (see Appendix). There are mainly four steps: 1) Ranking schedulable operations, 2) Bid request, 3) Bid preparation, and 4) Bid selection.

Ranking schedulable operations: This step is executed whenever an operation is scheduled, and/or a new job arrives to the system. Here, the set of schedulable operations are ranked according to some criteria such as job or customer urgency, operation urgency, number of bottleneck machines, etc. Dispatching rules such Earliest Due Date (EDD), minimum slack (SLACK) can also be used for this purpose.

Bid request: A bid is requested by the manager agent from the resource agents for the first operation in the list. During the bid request process, the manager agent sends the necessary information to all the resource agents that are capable of processing the operation. This includes earliest start time, latest completion time, operation time, set-up time, material handling time, schedulable time of the operation. Then each candidate resource agent prepares its bid.

Bid preparation: Bid preparation is the most difficult part of the algorithm, which requires the following additional steps (These steps are executed by each resource agent):

1. According to some local criteria (or local dispatching rule), find the first possible position for the operation in the existing schedule and determine the range of positions (RANGE). Suppose that the bid is to be prepared for H_2 and there are already six operations scheduled on a bidder resource. Here, A_2 refers to the second operation of Job A and D_3 refers to the third operation of Job D, etc. Assume that the local criterion is EDD and H_2 has the second minimum due date operation after is A_2. In this case, the resource agent tries to insert H_2 in the first available position in the range given below (see Figure 3.2).

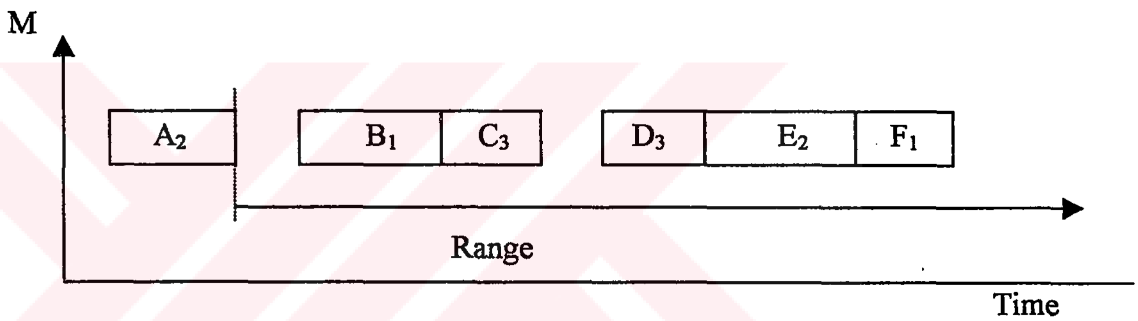


Figure 3.2. Range after Step 1 of bid preparation

2. Starting from the last position in the sequence, reduce this range by considering the factor called *Distance Constraint*. This constraint acts as a balancing mechanism that compromises local and global objectives. Specifically, we try to avoid extreme violations or changes of initial ranking for the sake of satisfying local objectives.

   As an example, consider two operations X and Y with Y having higher local priority than X. Suppose that X is already scheduled on the machine and now the bid is prepared by the same machine for Y. If X and Y were not in the same schedulable operation list, X must have had much higher global priority than Y so that it was scheduled a long time before Y. Thus, we should not allow Y to be scheduled before X in this situation. On the other hand, if they were in the same schedulable operation list, we let the local priority overwrite the global priority and hence schedule Y. Thus, the distance constraint acts as a mechanism that measures the extend to which the global objectives can be sacrificed for the sake of local objectives. In the example below, suppose that B_1 and H_2 are not in the same schedulable operation list. Hence, H_2 can not be placed before B_1 and the new range for H_2 is reduced to begin from the end of B_1 (see Figure 3.3).

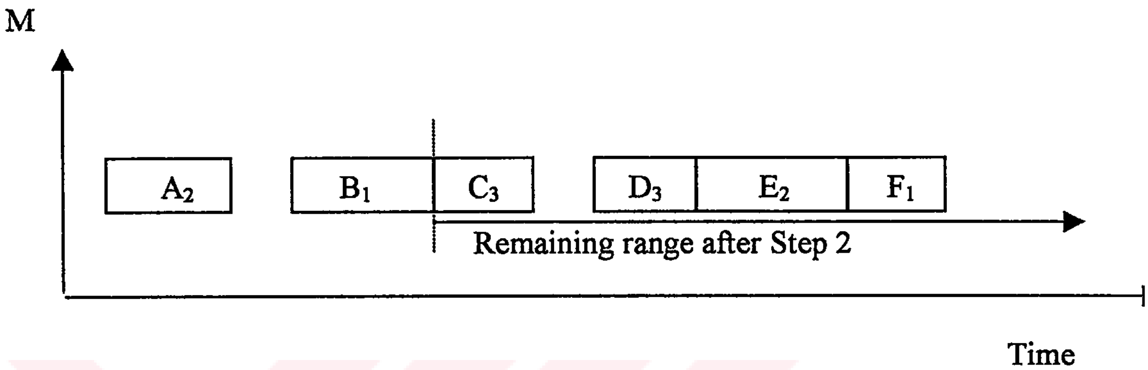


Figure 3.3 Range after Step 2 of bid preparation

3. For the affected operations on the same machine (i.e., the operations that will be moved to the right in the sequence due to the potential inserted operation), check the next position in the range from left to right to see if the second best bid of this affected operation is violated. If there is a violation, repeat the step for the next position to further reduce *RANGE*. In our example, assume that the second best bid of C_3 is violated. Hence the new range for H_2 starts after C_3 (see Figure 3.4).

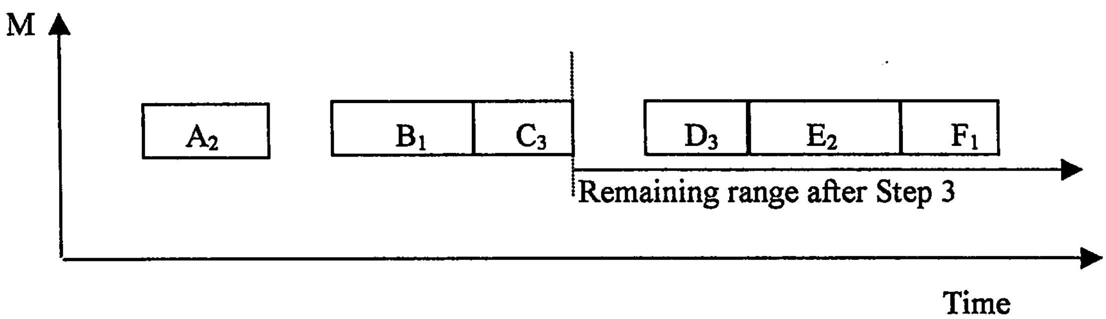


Figure 3.4 Range after step 3 of bid preparation

4. Execute either of the below:

4.1. (Competitive Version): Using the *RANGE* obtained at Step 3, check the remaining positions by considering RIGHT SLACK of the affected operations on the machine The right slack is distance between the completion time of the operation on the current machine and the start time of the next operation of the same job on another resource. If there is no right slack violation, the operation under consideration will be placed into the first available position in the sequence. Otherwise, we the return to Step 3 and check remaining positions. In the example, H_2 is inserted between E_2 and F_1 since right slacks of D_3 and E_2 are exceeded.

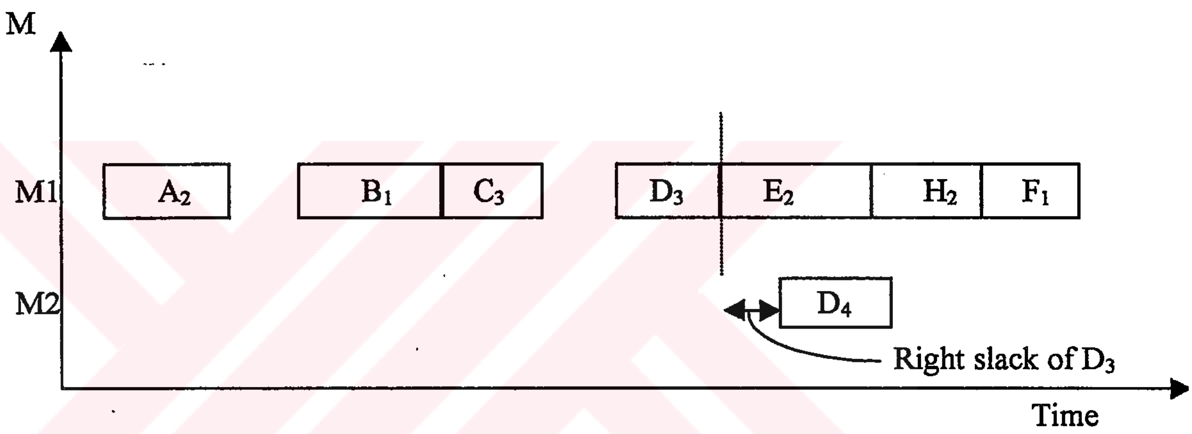


Figure 3.5 Schedule after H_2 is inserted using competitive version

4.2. (Collaborative Version) Using the reduced RANGE at Step 3, check the remaining positions considering the second best bids of all affected operations including the ones on other machines (see Figure 3.6).

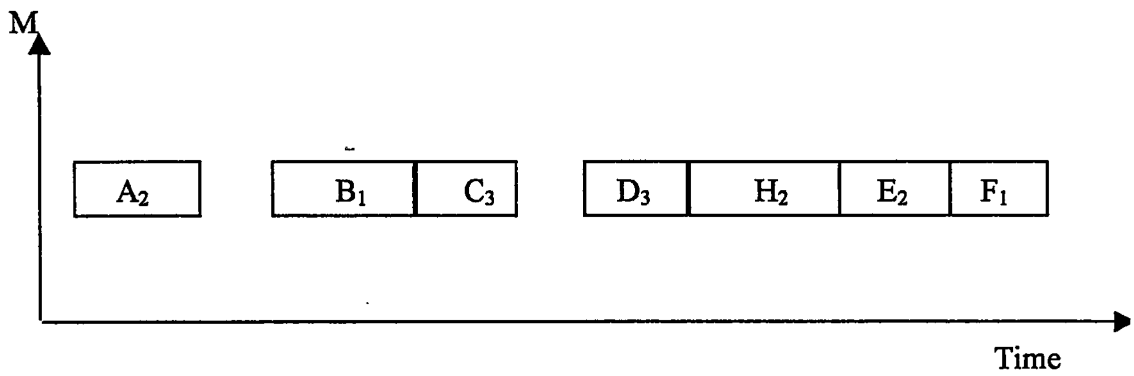


Figure 3.6. Schedule after H_2 is inserted using collaborative version

In both versions, if there is no second best bid (i.e., there is only one machine to process the operation) and right slacks of all affected operations are infinity (i.e., later operations are not scheduled yet), then latest completion time violations can be considered for these affected operations.

Bid selection: After the bids are prepared and presented to the manager agent, the best bid is selected. Here, a number of different criteria can be used for this purpose. These can be EFT (Earliest Finish Time), whether the machine is bottleneck or not (prefer the non-bottleneck alternative), least work content or other rules.

The implementation of Algorithm-B1 on the example problem is given in Table B.3 in Appendix.

3.2 Algorithm B2 (Machine initiated bidding algorithm)

Algorithm B2 is also a bidding algorithm. Instead of giving a bid for each operation in response to the request of manager agent, resource agents themselves volunteer to take the operations from the MA by considering their local objectives. As compared to Algorithm-B1, bidding is initiated by the resource agent and MA acts as a conflict resolver in case two or more resource agents want a particular operation. According to the classification framework proposed, this system can be classified as *single layer quasi heterarchical system with a separate manager agent* (Figure 3.7).

The main characteristics of the algorithm are as follows: There is a pool of operations. These operations are assigned priority by the manager agent considering the global system objectives (i.e., customer importance, due dates). Each resource agent (or machine) also ranks the operations in the pool according to its local goals and overall system objective (i.e. a composite measure of global and local objectives). This composite measure can be viewed as a weighted sum of global and local objectives. The following bidding mechanism is used to assign operations to machines. But first, we give the necessary definitions used in the algorithm.

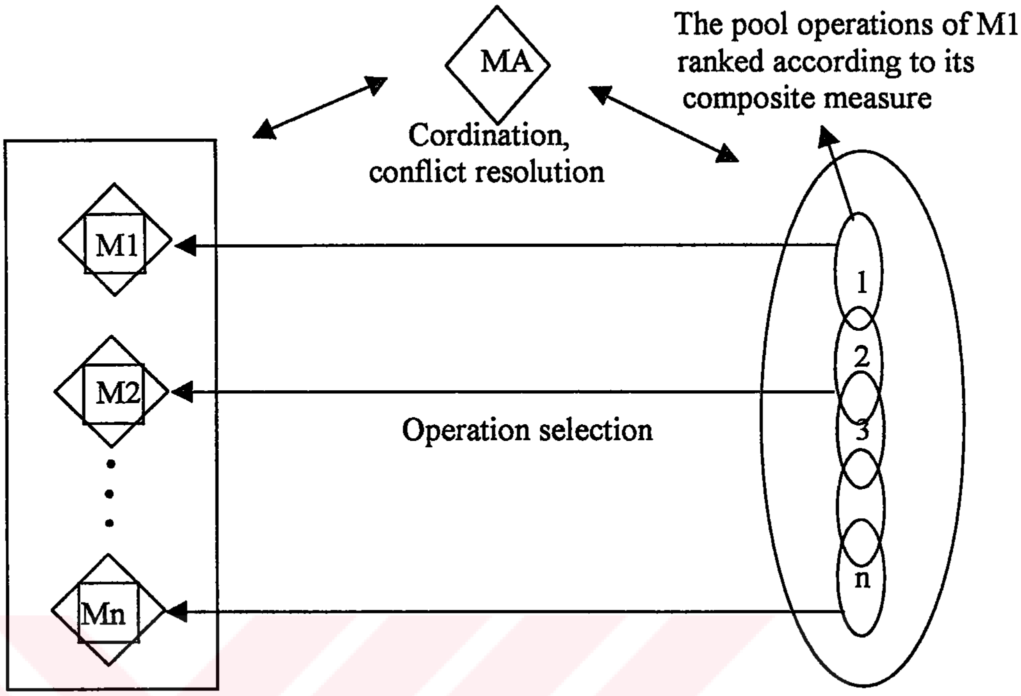


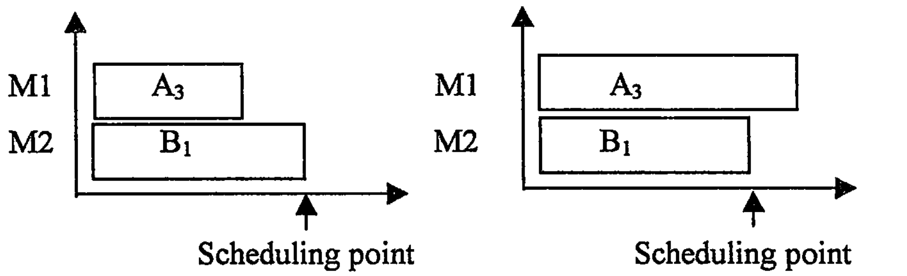
Figure 3.7. A schematic view of Algorithm-B2.

Schedulable operation: An operation, which is schedulable at the current scheduling point (i.e. the operation whose predecessors are already scheduled). For example, A_4 which is the next operation of job A, is a schedulable operation on machine M2 at the scheduling point (see Figure 3.8a).

Soon-available operation: An operation which is not currently schedulable but will soon become available (i.e. the operation whose immediate predecessor is not completed yet at the current scheduling point). In Figure 3.8b, A_4 is a soon-available operation on machine M2 at the scheduling point since A_3 is not finished yet.

In *permanent assignment,* the operation is permanently assigned to a machine whereas in *temporary assignment,* the assignment is done temporarily.

Scheduling point: It is the time at which the scheduling decision is to be made. It is determined by the completion time of an operation on a given machine. But if the



(a) A_4 is a schedulable operation (b) A_4 is a soon-available operation

Figure 3.8. Illustration of Schedulable and Soon-available operations

machine has a temporary assignment, the scheduling point is advanced to the next available time of the machine. As seen in Figure 3.9, t_2 is not a scheduling point since the machine M_1 has a temporary assignment (displayed by the dashed rectangular box for operation F_2). However, t_3 is the scheduling point for machine M_2.

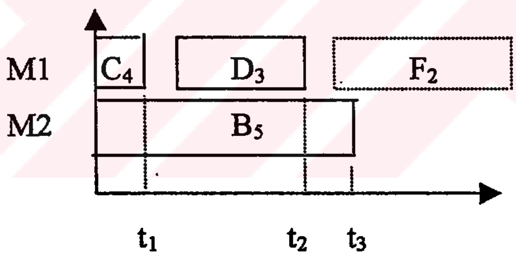


Figure 3.9. Illustration of scheduling points

There are two steps of the algorithm (see Figure A.2 in Appendix): 1) Initial assignment and 2) Assignment of consecutive operations.

Initial assignment: Initially all machines choose their operations according their composite measures. The initial assignment is only made among schedulable operations. If more than one machine request a particular operation, the conflict is resolved by the manager agent (i.e. one of these machines is selected or awarded by a bidding mechanism).

Assignment of consecutive operations: An operation assignment decision is made whenever a machine becomes available at some point in time (i.e., a scheduling point). The steps of this assignment are as follows:

1. Clock (or time) is advanced to the scheduling point. If there are temporary assignments from the previous iterations with starting times before or at this time point, these assignments are made permanent.
2. Resource Agents (RAs) of all the available machines check to see if there are schedulable operations in the list. If there is no such schedulable operation, Step 3 is executed. Otherwise, each RA selects the highest ranked operation according to its composite measure. In case of a conflict, MA makes the final decision. Note that all these assignments at this stage are permanent.
3. Step 2 is executed for soon-available operations with the following change: if the operation has already been temporarily assigned to another machine with an earlier completion time, then no assignment is made. Otherwise, the machine takes over this operation via the manager agent from the other machine
4. The above steps are repeated until all the operations are assigned to the machines.

As explained above, operations are assigned to machines either permanently or temporarily. Permanent assignments can be made only for schedulable operations or soon-available operations with only one alternative machine. Temporary assignments are made only for soon-available operations with several alternative machines to process. Temporary assignment acts as a look-ahead mechanism to make better scheduling decisions. In a way, it alleviates the problem of myopic decisions if one considers only the schedulable operations. But it has also some drawbacks due to inserted idleness and possibility of loosing some schedulable operations because of this temporary assignment. In the algorithm, we use a time window with appropriate size (i.e. average operation time) to overcome these drawbacks.

The implementation of Algorithm-B2 on the example problem is given in Table B.4 .

3.3. Algorithm C (Job initiated bidding algorithm)

Algorithm-C basically applies to the same problem that Algorithm-B1 is used. It differs from Algorithm-B1 in that it schedules operations job by job (i.e., bidding for the next operation can not be started before bidding of the previous operation of the same job). Moreover, Algorithm-C takes into account the material handling and transportation times to calculate the time at which the job becomes available for processing at a machine. Algorithm-C uses H2-Bidding mechanism to take advantage of set-up and uses H3-Bidding mechanism to take advantage of set-up and transportation times. If set-up and transportation times are not important (or can be ignored) then we use Algorithm C with H1-bidding mechanism. In this case, the algorithm looks similar to Algorithm B1 except that jobs are scheduled (or put in bid) instead of operations. These bidding mechanisms will be explained in sequel.

If a machine is capable of processing two consecutive operations of a job, then Algorithm-C is more likely to assign these operations to the same machine to obtain the set-up time saving. In Algorithm-B1, however, it is quite unlikely to initiate the bids for two consecutive operations of the same job one after another. Hence, two consecutive operations of the same job can be assigned to different resource agents in Algorithm B1.

As the effects of material handling and set-up are explicitly considered, one can naturally expect that Algorithm C works better in such systems where Total material handling time/Total operation time and Total set-up time/Total operation time are high. Alternatively, Algorithm-B1 is expected to work better in systems with higher routing flexibility (number of alternative m/c's per operation) and higher due-date tightness.

Algorithm-C has also competitive and collaborative versions. In the competitive version, no right shift of operations on other machines is allowed. The collaborative version, however, allows the right shift as long as this shift does not make the affected job completion times to exceed their second best bids.

When Algorithm-C is implemented with the H1-Bidding mechanism, the decisions are made for each operation of the job one after another other. This system can be classified as *single layer quasi-heterarchical system with a separate manager agent*. When Algorithm-C is used with H2 and H3, the machine that prepares the bid for an operation selects the previous machine that can process for the preceding operation. In this case, the final machine assignment is made after the manager agent selects the machine for the last operation of the job. Because of this structure, the resulting system can be classified as *multi-layer quasi heterarchical system with multiple bids* (Figure 3.10).

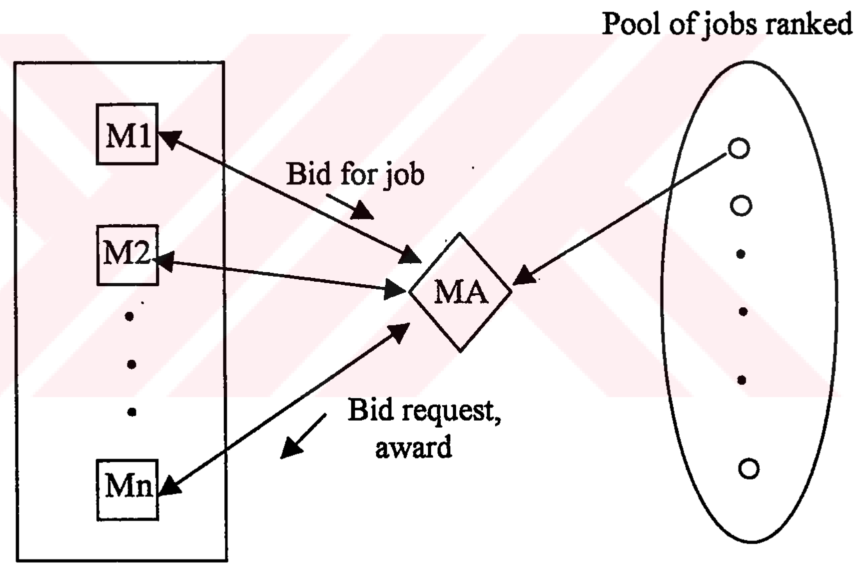


Figure 3.10. A schematic view of Algorithm C

The proposed algorithm is depicted in Figure A.3 (see Appendix). There are mainly four steps: 1) Ranking schedulable jobs, 2) Bid request, 3) Bid preparation, and 4) Bid selection.

Ranking schedulable jobs: This step is executed whenever a new job enters the system. Schedulable jobs are ranked according to some criteria such as, job or customer urgency, whether jobs visit bottleneck machines. The rules such as EDD and SLACK can also be used for this purpose.

Bid request: Starting from the first job in the list, the manager agent opens the bids for each operation by sending the necessary information to resource agents. This information can include earliest start time, latest completion time, operation time, set-up time and average distance from a station to another.

The bidders may use different bidding mechanisms. In this study, we propose three mechanisms:

H1- Bidding Mechanism: In this mechanism, operations of the job are placed in the bid one at a time according to the precedence relations. For each operation, the resource agents (i.e. machines) prepare their bids and the manager agent selects the best bid based on Earliest Finish Time (EFT) criterion. During the bid preparation process, each resource agent calculates the availability time (or arrival time of the job) on this m/c by using EFT of the previous operation and material handling time between the current machine and the machine which had won the bid of the previous operation of that job. Then the procedure given in bid preparation is invoked to find the earliest possible position in the existing schedule of the m/c (i.e. it determines the EFT).

As shown in Figure 3.11, resource agents that will prepare bid for A_{k+1} uses the bid which offers the EFT for A_k.

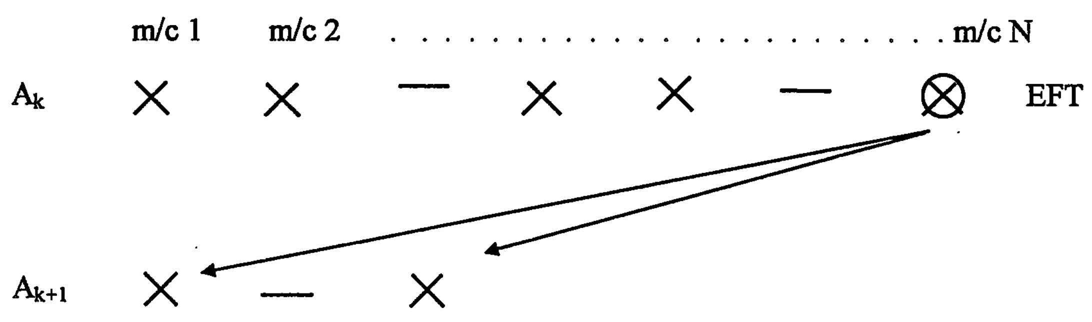


Figure 3.11. Illustration of H1

× A machine that can process the operation (i.e. A_k and A_{k+1})

— A machine which can not process the operation

⊗ The machine which wins the bid for A_k with EFT

H2 Bidding Mechanism: H2 is different than H1 in two aspects. First, bid selection is not finalised until the last operation is assigned to a machine. Second, the bid is prepared either for current operation A_{k+1} or for the combined operation A_k and A_{k+1} provided that, A_{k+1} and A_k can be also processed on the same machine. The bid for A_{k+1} is prepared by using the procedure in Section 2 and considering the EFT of A_k and the material handling time information. The bid for combined operation A_k and A_{k+1} is prepared to take advantage of set-up and material handling time. In an attempt to prepare such a bid for the combined operation $A_k + A_{k+1}$, algorithm first checks if this combined operation can be placed in the first available position in the existing schedule of that machine, such that starting time of A_{k+1} on the machine does not exceed its lower bound. If it exceeds this lower bound, bid is prepared for only A_{k+1} as if the m/c is only capable of processing A_{k+1}. LB (lower bound) is the sum of EFT of A_k, material handling time and set-up time. LB is a kind of threshold value after which availability times of A_{k+1} in both cases (i.e. whether it comes from the machine processing A_k in the earliest time or A_k is processed on the same m/c) are equal. Hence, the bid is prepared for the combined operation.

Because of the above reasoning in the bidding process, at the last operation, machines which propose bid for A_N might have used different bids which come from different m/c's. Same thing is true for A_{N-1}, A_{N-2},.. Hence, the final selection can be made after determining the bid for A_N and repeating this process all the way to A_1 (Figure 3.12).

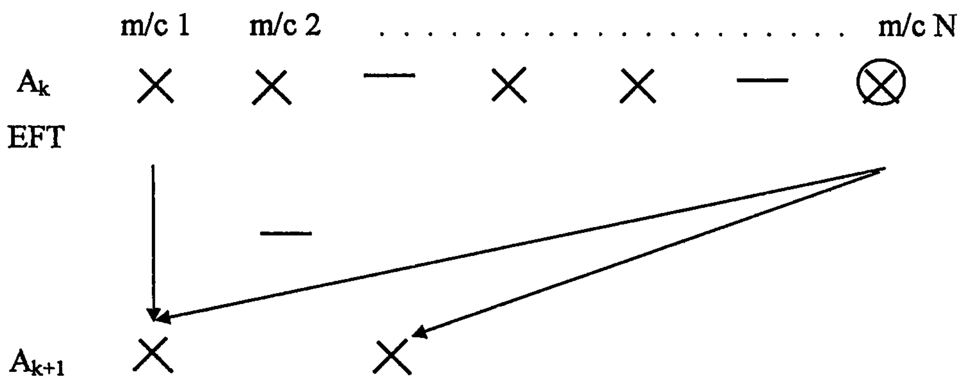


Figure 3.12. Illustration of H2

<u>H3-Bidding Mechanism:</u> It is almost the same as H2 except that earliest available time is calculated by using the following procedure:

1. Calculate the availability time of A_{k+1} by considering the previous best bid (a_1=EFT for A_k + MHT (Material Handling Time)). Repeat the same calculation if A_k is processed on the same machine (a_2 =Completion time for A_k on the same machine).
2. Find the minimum of a_1 and a_2.
3. Eliminate the alternative bids for A_k if their completion time > min(a_1, a_2).
4. For the remaining elements, find the bid with minimum availability (a_3)

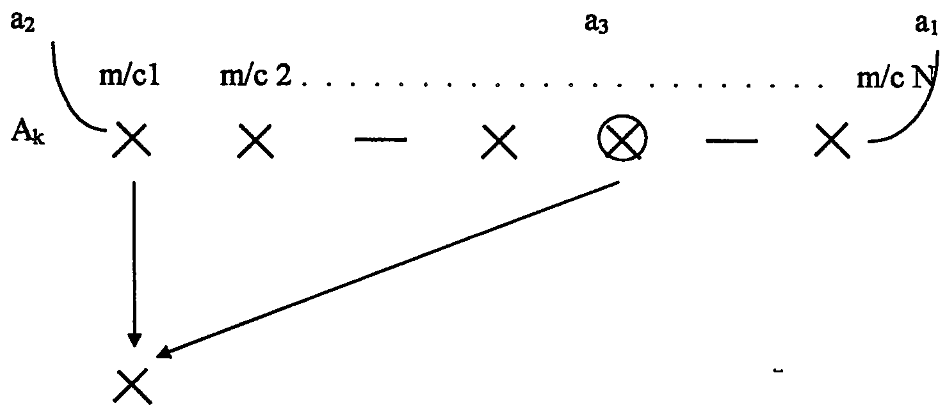


Figure 3.13. Illustration of H3

Here ⊗ shows the Earliest Availability Time for the previous operation

Bid preparation: The bid is prepared by resource agents using any of H1, H2, and H3. Execute the following steps for each operation in the bid:

1. According to some local criteria, find the first possible position for the operation in the existing schedule and determine the range of positions *(RANGE)*.
2. Check the positions from left to right. If there is a job whose last operation is scheduled on this machine and completion time is exceeded by some threshold value (indifference amount that is determined before to show the tolerance in lateness of a job), then reduce the range by considering positions after this operation until the end and repeat this step. If this indifference amount is zero, then we do not allow any delay of any job from its committed delivery time in the bid.
3. (Only in the competitive version), check the remaining positions from left to right by considering the second best bids of operations on the same machine. If there is a violation, move to the next position and repeat the same procedure starting from Step 2.
4. Either execute 4.1 or 4.2

4.1. (Competitive version): Using the reduced *RANGE* at Step 3, further check the remaining positions by considering operations' right slacks to prevent right shift of operations on other machines.

4.2. (Collaborative version): Using the reduced *RANGE* at Step 2, the remaining positions are checked considering the second best bids of all affected operations (including those on other machines) by the manager agent.

Some of the information types that can be used in the above steps are: Second best bids of affected operations and jobs, right operation slacks of affected operations, set-up times, and transportation times.

Bid selection: After the bids are prepared and presented to the manager agent, the best bid is selected. Several criteria can be used for this purpose, such as EFT, whether the machine is bottleneck or not (prefer the nonbottleneck alternative), least work content, minimum number of set-up.

Implementation of Algorithm-C with the H3-colloborative bidding mechanism on the example problem is given in Table B.5 (Appendix)

3.4 Experimental Design

3.4.1 Classical Job Shop Environment with Sequence Dependent Setup Times

A classical job shop environment where jobs are available for processing at time zero, is used in this study. The data regarding this job shop environment is previously generated by Demirkol, Mehta and Uzsoy (1998) to test a number of algorithms designed for the single machine Lmax problem. We used this data set to test 4 proposed bidding algorithms (B1-Competitive Version; B1-Collaborative Version, C-Competitive Version and C-Collaborative Version) developed for distributed scheduling systems. Although the proposed algorithms are constructed for decentralised systems in our mind, comparisons with those proposed by Ovacik and Uzsoy (1994) enables us to see their effectiveness in a centralised environment. By that way, we also gain insights into the performance of our algorithms in the classical job shop environment. It is important to state that there is no data set in the literature for distributed scheduling problems. We will attempt to generate such data sets later in the thesis.

Although the proposed bidding algorithms do not have an internal logic to optimise a global criterion, we used EDD to rank the operations (in versions of B1 Algorithm) or jobs (in versions of C Algorithm) to find an order in which they are put in bid. Also we used EDD as the local rule for all machines in the system.

In the traditional shop used in this study, it is assumed that each job has to be processed at each machine exactly once. The sequence in which jobs visit machines is pre-determined and the routing of each job is a random permutation of the machines. All machines have sequence-dependent setup times. Besides, setup and processing times at each machine are a priori known and uniformly distributed between 1 and 200.

The job due dates are uniformly distributed on an interval determined by the expected workload of the system and the parameters R and τ. The mean μ for the interval is given by

$$\mu=(1-\tau)E[Cmax]$$

where τ denotes the percentage of jobs expected to be tardy. The expected makespan is calculated by estimating the total setup and processing time required by all jobs at all machines (or workcenters) and dividing the result by the number of workcenters available. The interval from which the due dates are generated is then given by $\mu \pm \mu R/2$, where R is a parameter determining the range of the interval. In the shop, preemption is not allowed.

Four factors are considered in the first part of the experiments. These are:

- Percentage of jobs expected to be tardy (τ)
- Due date range parameter (R)
- Number of jobs (n)
- Number of machines (m)

The values used for various parameters are summarised in the table below. As can be seen in this table, there are 320 problem instances resulted from 32 problems with 10 replications (There are two problems per combination in the original data set and two levels of τ and R).

Table 3.1. Experimental factors and their levels

	Values Used				Total
Due Date Range	1.5	2.5			2
% Tardy Jobs	0.3	0.6			2
# Of Machines	5	10	15	20	4
# Of Jobs	10	20			2

3.4.2 Classical Job Shop Environment with Sequence Dependent Setup Times and Alternative Machines

At this stage, the previous data set is modified by adding alternative machines to the job shop problem. Five problem instances are generated at each experimental conditions. It is assumed that every operation can be performed on one alternative machine whose operation time is 10 percent higher than the ideal (or original) machine. For simplicity, we used the machine numbers as the basis in determining the alternative machines. Specifically, we set the first two machines are alternative to each other, then the next two machines, and so forth. If the number of machines in the problem is odd, then the last machine has no alternative. The set up time of the alternative machine is assumed to be equal to the maximum set-up time that the operation incurs before or after an operation on its ideal machine. But if two operations of the same job can be processed on the same machine, set-up time between these two operations is set to zero.

3.5 Computational Results

As of now, we have presented Algorithms B1, B2 and two versions of Algorithm C. We have initially tested all the algorithms on well known makespan data, however since B2 did not perform well in the pilot runs we have omitted it in further experimental testing.

3.5.1 Classical Job Shop Environment with Sequence Dependent Setup Times

The results of the algorithms for the Lmax criterion are summarised in Table 3.2. Each cell is the average of 10 replications (or instances) for the given experimental setting. Even though the proposed algorithms are designed to minimise Lmax measure, we also report the statistics for makespan, number of tardy jobs and average tardiness criteria. All these results together with the detailed comparisons for each problem instance can be found in Table C.1 and Table C.2 in Appendix.

When the global ranking rule and the local scheduling rules are the same (i.e. EDD), both Algorithm-B1 and its collaborative version (B1-Col) produce the same schedule. To see this, let's take a very simple example. Suppose that there are two operations X and Y that will be put in a bid in that order (i.e., X has higher global priority than Y). Assume that a machine which was just awarded for the operation X is also in the process of bidding for Y. Regardless of the type of the algorithms (B1 or B1-Col) is use, the initial range for Y begins from the end of X since Y has lower local priority than X. In this circumstance, B1-Col does not attempt to insert Y before X if it causes a right shift. Hence, the part of B1-Col that differentiates it from B1 is not executed and both of the algorithms generate the same schedule. For that reason, the results of B1-Col are omitted in the tables. It will be considered when the local rules are different from the global ranking rule.

As can be seen in Table 3.2, the best Lmax results of the proposed algorithms is considerably better than the best results reported for these problem instance in the literature. On the average, the Lmax performance is improved over 18.77 percent. Specifically, the proposed algorithms produced smaller Lmax values in 280 of the 320 problems. In these 280 instances, the improvement percentage is about 22.51% whereas the deterioration is only 7.43 % in the remaining 40 instances. Our best results yield smaller Lmax in 72 out of 80 problems for R=1.5, τ=0.3. When R=1.5, τ=0.6, the results are better in 56 of 80 instances. When R=2.5, the statistics are 79 and 73 for τ=0.3 and τ=0.6, respectively.

Table 3.2. Summary of results for each experimental setting

T=0.3, R=1.5	B1				C				C-Col				Best of All			
Size	APD	NBS	BIA	WIA	APD	NBS	BIA	WIA	APD	NBS	BIA	WIA	APD	NBS	BIA	WIA
n=10, m=5	34.26	9	39.77	-15.3	22.82	8	30.16	-6.52	21.58	8	29.91	-11.8	36.43	9	41.22	-6.715
n=20, m=5	-6.82	3	21.44	-18.9	0.08	3	23.64	-10	-18.3	1	5.955	-21	1.784	3	23.64	-7.58
n=10, m=10	22.31	9	25.4	-5.49	18.9	9	21.32	-2.92	23.77	10	23.77	0	27.01	10	27.01	0
n=20, m=10	21.08	8	26.85	-2.03	16.22	9	18.35	-2.97	16.86	9	19.38	-5.78	24.32	10	24.32	0
n=10, m=15	23.32	10	23.32	0	19.02	10	19.02	0	24.73	10	24.73	0	29.1	10	29.1	0
n=20, m=15	18.59	10	18.59	0	9.779	9	11.59	-6.51	16.07		18	-1.23	21.21	10	21.21	0
n=10, m=20	26.83	10	26.83	0	20.75	10	20.75	0	25.75	10	25.75	0	27.67	10	27.67	0
n=20, m=20	23.8	10	23.8	0	14.93	10	14.93	0	20.29	10	20.29	0	24.55	10	24.55	0
T=0.3, R=2.5	B1				C				C-Col				Best of All			
n=10, m=5	15.91	8	21.02	-4.54	17.52	8	23.53	-6.53	10.77	7	20.86	-12.8	22.3	9	24.94	-1.451
n=20, m=5	22.18	10	22.18	0	17.53	10	17.53	0	21.23	10	21.23	0	24.76	10	24.76	0
n=10, m=10	28.09	10	28.09	0	23.23	10	23.23	0	25.41	10	25.41	0	29.9	10	29.9	0
n=20, m=10	24.95	9	27.93	-1.93	15.27	8	19.2	-0.44	24.8	10	24.8	0	27.09	10	27.09	0
n=10, m=15	33.6	10	33.6	0	24	9	26.88	-1.92	31.22	10	31.22	0	34.4	10	34.4	0
n=20, m=15	16.88	8	21.26	-0.63	11.55	7	20.81	10.05	16.79	10	16.79	0	21.5	10	21.5	0
n=10, m=20	35.87	10	35.87	0	27.64	10	27.64	0	34.64	10	34.64	0	36.48	10	36.48	0
n=20, m=20	26.29	10	26.29	0	20.8	10	20.8	0	24.34	10	24.34	0	27.22	10	27.22	0

Table 3.2 Cont'd.

T=0.6, R=1.5	B1				C				C-Col				Best of All			
Size	APD	NBS	BIA	WIA	APD	NBS	BIA	WIA	APD	NBS	BIA	WIA	APD	NBS	BIA	WIA
n=10, m=5	-9.07	1	0.624	-10.2	-13.8	1	0.624	-15.4	-10.6	1	1.236	-11.9	-5.849	2	0.93	-7.544
n=20, m=5	-16.9	0	0	-16.9	-12.5	2	4.335	-16.7	-17.8	0	0	-17.8	-8.244	2	4.335	-11.39
n=10, m=10	8.852	8	12.1	-4.14	2.828	7	9.21	-13.7	3.776	7	11.55	-14.4	11.74	8	15.73	-4.137
n=20, m=10	-0.68	3	11.73	-6	-8.8	2	9.678	-13.4	-7.89	4	4.549	-16.2	0.34	5	8.169	-7.484
n=10, m=15	17.83	10	17.83	0	3.484	7	8.18	-10.8	12.24	10	12.24	0	19.11	10	19.11	0
n=20, m=15	2.853	8	4.24	-4.63	0.218	6	5.015	-6.99	1.204	7	5.522	-8.88	5.839	9	7.002	-4.633
n=10, m=20	21.83	10	21.83	0	10.15	9	11.39	-1.03	17	9	19.25	-3.23	22.89	10	22.89	0
n=20, m=20	10.25	10	10.25	0	-2.06	3	5.263	-5.202	4.088	8	5.757	-2.59	10.48	10	10.48	0
T=0.3, R=2.5	B1				C				C-Col				Best of All			
n=10, m=5	6.965	9	9.268	-13.8	8.193	9	10.45	-12.1	4.038	7	8.071	-5.39	9.564	9	11.97	-12.12
n=20, m=5	-2.6	3	9	-7.59	-9.25	2	11.41	-14.4	-4.9	2	16.21	-10.2	4.235	5	11.9	-3.433
n=10, m=10	12.14	10	12.14	0	7.711	8	12.11	-9.87	12.25	8	15.48	-0.68	17.37	10	17.37	0
n=20, m=10	7.055	8	9.82	-4.01	2.351	5	12.27	-7.65	5.837	7	11.1	-6.46	11.76	10	11.76	0
n=10, m=15	23.95	10	23.95	0	14.62	10	14.62	0	22.84	10	22.84	0	26.25	10	26.25	0
n=20, m=15	11.7	9	13.55	-4.95	7.428	8	11.15	-7.47	11.95	9	14.16	-7.95	13.28	9	15.31	-4.95
n=10, m=20	26.48	10	26.48	0	15.94	10	15.94	0	24.44	10	24.44	0	28.17	10	28.17	0
n=20, m=20	15.11	10	15.11	0	7.323	9	9.397	-11.3	15.32	10	15.32	0	17.82	10	17.82	0

APD: Average Percent Deviation, NBS: # of problems (out of 10 replications) that the proposed algorithm is better than the best result in the literature
BIA/WIA: Average Percent Deviation in problems that the proposed algorithm is better/worse than the best result reported in the literature

In general, the results indicate that the proposed algorithms perform better as τ decreases when R is kept constant. In the other cases where τ is kept constant our algorithms yield better results when R increases. Hence, the worst scenario for our algorithms is the condition where τ=0.6, R=1.5, even though the proposed algorithms are still better than the existing ones.

In terms of the success rate of our algorithms as compared to the best Lmax reported in the literature, Algorithm B1 yields better results in 165 out of 280 problem instances.Statistics for C-Col and C are 70 and 45, respectively. Table 3.3 summarises the success rates of the algorithms for different values of τ ,R. Note that the proposed algorithms perform better when the dispersion around the mean due-date value increases (i.e., job due dates become more distinct). This is due to the fact operations or jobs are ranked according to their latest completion times and put in bid in that order. Besides, the proposed algorithms attempt to schedule the high operations before the lower priority operations.

Table 3.3. Distribution of problems where our best Lmax result is better than the best result reported in the literature

	τ=0.3, R=1.5	τ=0.3, R=2.5	τ=0.6, R=1.5	τ=0.6, R=2.5
B1	43	55	32	35
C	10	13	9	13
C-Col	19	11	15	25

If we compare the performance of our algorithms individually with respect to the best results reported in the literature, B1 is better in 263 problems out of 320. The related performance statistics are 238 and 251 for C and C-Col, respectively (Table 3.4).

Table 3.4. Comparison of each algorithm with the best results reported in the literature

	τ=0.3, R=1.5	τ=0.3, R=2.5	τ=0.6, R=1.5	τ=0.6, R=2.5
B1	69	75	50	69
C	68	72	37	61
C-Col	67	77	44	63

When the proposed algorithms are compared with each other, B1 is definitely a winner. As seen in Table 3.5, it produces better results in more than half of the instances. The success of B1 can be due to the fact that it allows operations of different jobs to be scheduled one after another. Moreover, we note that collaborative version of Algorithm-C dominates its competitive version since it allows more flexibility. However, it does not perform as well as B1 since there is no alternative machine in the benchmark problems and the second best bid constraint which is the only limitation in C-Col is satisfied in all cases. This causes a lower priority operation to be scheduled before higher priority operations. However, as the due-date becomes more tight (τ=0.6) the number of problems that C-Col produced the best results increases from 31 to 50 (see Table 3.5). This indicates that when the global objectives are difficult to achieve, any collaboration between agents improves the overall system performance.

Table 3.5. Comparison of our algorithms within themselves

	τ=0.3, R=1.5	τ=0.3, R=2.5	τ=0.6, R=1.5	τ=0.6, R=2.5	Total
B1	45	55	45	37	182
C	15	14	12	16	57
C-Col	20	11	23	27	81

As stated earlier, the results are obtained for the other performance measures even though the algorithms are designed to minimize Lmax. Analysing these results, Algorithm B1 looks the best for the makespan criterion. This is because it allows operations of jobs to be scheduled simultaneously. Algorithm C is the best for the average tardiness and number of tardy jobs criteria. Since it schedules all operations of the most urgent jobs first.

We also observed that the Lmax performance of the proposed algorithms gets better for as the number of jobs/number of machines ratio decreases. As this ratio increases, variability of jobs on a given machine increases that makes the sequence dependent set-up optimisation problem more important. Our algorithms use the sequence dependent set-up time information during the bid preparation, but they are not specially designed to optimise sequence dependent set-up. In the traditional centralized scheduling systems, the sequence dependent set-up time minimization can be achieved by iterative scheduling. But such an iterative method may not feasible for a bidding-based scheduling system since the promised delivery dates can not be delayed beyond their second best bids.

3.5.2 Classical Job Shop Environment with Sequence Dependent Setup Times and Alternative Machines

In this section, we measure the performance of the proposed algorithms in a job shop with alternative machines. In the experiments, we used the previous data set by adding one alternative machine for each operation. But this time, we solved 5 problem instances (not 10) at each condition to save the computation time). The results of 160 problem instances indicate an improvement of 63.75% (102/160) over the no alternative machine case for the Lmax criterion when Algorithm B1 is used (see Table C.3 in Appendix). The improvement rate is 47.5 % (76/160) and 27.5% (44/160) for Algorithm C and C-Col, respectively (Table 3.6). Thus, we can conclude that B1 is the most effective algorithm to improve Lmax when alternative machines exist in the system.

Table 3.6. Number of problems that algorithms yield the best Lmax

	τ=0.3, R=1.5	τ=0.3, R=2.5	τ=0.6, R=1.5	τ=0.6, R=2.5	Total
B1	22	28	25	27	102
C	21	22	14	19	76
C-Col	12	15	6	11	44

We observed that the performance of the algorithms C and C-Col is worsened when alternative machines are used. This may be due to the fact that jobs spend more time in the system for longer set-up and processing times on alternative machines. Apparently, these two algorithms (C and C-Col) may not have fully utilized the opportunities of using alternative machines to reduce Lmax. To see why this happens, we repeat the experiments by taking smaller setup times on the alternative machines. Instead of the maximum set-up time, we use the average (of minimum and maximum) set-up time of an operation that can be processed on its ideal machine. We consider two cases: n=20, m=5 and n=10, m=10 for τ=0.3 and R=1.5. When the set-up is taken as the maximum value on the alternative resources, we observe the improvement of 50% over the no alternative machine case for Algorithm B1. The statistics are 40% and 30% for Algorithm C and C-Col, respectively. On the other hand, when the average set-up time is used on the alternative resources, the percentage improvement goes up to 90% for B1 and 80% for both Algorithms C and C-Col. From these experiments we can conclude that the system performance can be improved significantly by the alternative machines as long as set-up times (or operation times) on alternative resources are not too much higher than the ones on their ideal machines.

In the alternative machine case, Algorithm B1 is still the best as it yields the minimum Lmax for 123 of the 160 problem instances (Table 3.7). This is followed by the algorithms C and C-Col whose best results are recorded for 23 and 14 times, respectively. The outstanding performance of Algorithm B1 can be attributed to the fact that it can schedule operations of different jobs one after another and thus utilises the alternative machines in more effective way.

Table 3.7. The best of all results for Lmax criterion in one alternative machine case

	τ=0.3, R=1.5	τ=0.3, R=2.5	τ=0.6, R=1.5	τ=0.6, R=2.5	Total
B1	29 (21)	31 (22)	34 (21)	29 (15)	123
C	7 (10)	4 (8)	4 (6)	8 (6)	23
C-Col	4 (9)	5 (10)	2 (13)	3 (19)	14

Another observation is that the collaborative version of Algorithm C loses its advantages in the alternative machine case. Recall that C-Col improves the system performance (Table 3.5) when global objectives are difficult to satisfy (i.e. tight due-dates). On the other hand, alternative machines create favourable conditions to achieve global objectives and hence C-Col can not find enough opportunities to show its improved performance.

In addition to the Lmax criterion, we computed the values of makespan, average tardiness and number of tardy jobs. A summary table below is prepared to show best of all results for these measures. Note that the results are also reported for the no alternative machine case in parentheses. As can be seen in Table 3.8, B1 is the best algorithm for makespan (i.e. 93/160 in the alternative machine case and 66/160 in no alternative machine case). For the average tardiness criteria, Algorithm C yields the best results (i.e. 70/160 in the alternative machine case and 89/160 in the no alternative machine case). Note that C-Col is even better than B1 in the tardiness case. This improved performance of both C and C-Col over B1 is due to that fact they schedule the operations of the most urgent jobs first.

Table 3.8. Overall best of results for makespan and average tardiness in 160 problem instances

	Makespan	Av. Tardiness
B1	93 (66)	35 (33)
C	40 (38)	70 (89)
C-Col	27 (56)	55 (38)

3.6. Distributed environment where agents have different local rules than the global objective

In order to test how the bidding based algorithms designed for the process team structure perform in a distributed environment in which each agent has different local objectives, we perform additional experiments. We choose the experimental condition with R=2.5, τ=0.3, n=20 and m=10 for this case (This is the condition where due dates are distinct and the size of the problems is large enough to measure the effect of the different local objectives).

We took 5 problem instances in this condition. We solved these problems when all the local scheduling rules are EDD (i.e., the case where both the local and global objectives are the same). Then we changed the local rules of some machines different from the global rule EDD. We assumed that local decision-makers try to lower their mean flow time. Thus, we used SPT as the local scheduling rule. In the experiments, our aim is to determine how the performance of the local and global objectives change with the number of local decision-makers (k). We first took k=0 (there is no local decision-maker with a different local scheduling rule than the global ranking rule) and increased it to k=1, k=2, k=4 and k=10 (all the machines have local rules different than the global ranking rule) in an incremental fashion. For k=1, we selected the machine which has the highest mean flow time record in the k=0 case. We determined the machines for k=2, k=3, k=4 and k=10 in the similar way. At every value of k, we also recorded the changes in global performance criterion (Lmax). We solved these problems by four algorithms and hence we took 100 runs.

As can be seen in Table C.4, in all of the 5 replications of k=1 there is an improvement in mean flow time of the bottleneck machine. Only B1-Col gives no improvement for the selected 5 problems. In this algorithm, although the machines attempt to insert the operations into the most suitable positions according to their local goals, they usually could not achieve this due to the fact that the constraints imposed by B1-Col (distance constraint, second best bid constraint etc.) are too tight that the machine can not perform the insertion. Note that these constraints are for the sake of not violating global ranking. Thus, the constraints can be loosened to increase the performance of B1-Col in the distributed environment. On the other hand for Algorithm C & C-Col such a distinction can not be made. The performance of local decision-makers differs according to the characteristics of the problem when any of these two algorithms is in use.

We also observed that as the local decision-makers that have the same local scheduling rule (different than the global ranking rule) increases, the average performance of a machine which satisfies its local goal decreases. This may be because of the fact that after a point, the cumulative trade-off that can be sacrificed from the global objective becomes shared among all these local agents. We also noted that significant improvements in local objectives are accompanied with no or small changes in global performance. This shows that the bidding algorithms proposed in this study enable the local agents to improve their local goals within the overall objectives of the system

3.7 Conclusion

In this study, we developed the bidding-based algorithms for distributed systems where there is a competition between alternative resources. Some of the proposed algorithms utilize operation bidding whereas some carry out the bidding on the job basis. In these systems, resource agents (or machines) are independent from each other and have no knowledge about their competitors. They give their best bids to get as many operations or jobs as possible.

The results of computational experiments in centralized systems indicate that the proposed algorithms even perform better than the heuristics designed for this environment.

It is important to note that the proposed algorithms represent a new approach for distributed scheduling and we tried to present them in a very general way. The algorithms can be modified to optimize local different objectives. In addition, the insert capabilities can be used in rescheduling and reactive scheduling areas (i.e. interruption, machine breakdown, due-date changes etc.). When they are used in a centralised environment, multiple objectives can also be considered.

CHAPTER 4

BIDDING BASED ALGORITHMS FOR PRODUCT BASED SYSTEMS

In the previous chapter we described three algorithms designed for process team structure. Here, we propose 2 algorithms for product team structure. They are also tested in different experimental conditions.

4.1 Product Team-based algorithms

4.1.1 Algorithm D1 (Job initiated bidding algorithm)

Algorithm D1 applies to the situations in which the jobs are allocated among the teams (or agents) by bidding and each team is capable of performing all or most of the operations of the job. A team of resources can represent a factory of a multi-national company or a part of the factory such as a department or a group technology cell. We assume that these teams have similar processing capabilities (e.g. product teams discussed in Rahimifard and Newman, 1998).

In Algorithm D1, schedulable jobs are first ranked according to some criteria by the manager agent (overall manager) and then bidding is carried out for each job in the list. Here, teams are the bidders for the jobs. In order to prepare a bid, the team managers (resource group agent) also request bids for the operations of the job from the machines which belong to the same team. If the machines (i.e. resource agent) in a team are capable of processing the operation, the team manager selects the best bid among the bids of these machines. If there is no machine in the team that can process a particular operation, the team manager requests a bid from other teams. Hence, a team manager both plays the roles of bidder (for the job) and bid manager (for the operations of the job). From this point of view, this system can be classified under *the multi-layer quasi-heterarchical systems with multiple bids* (Figure 4.1).

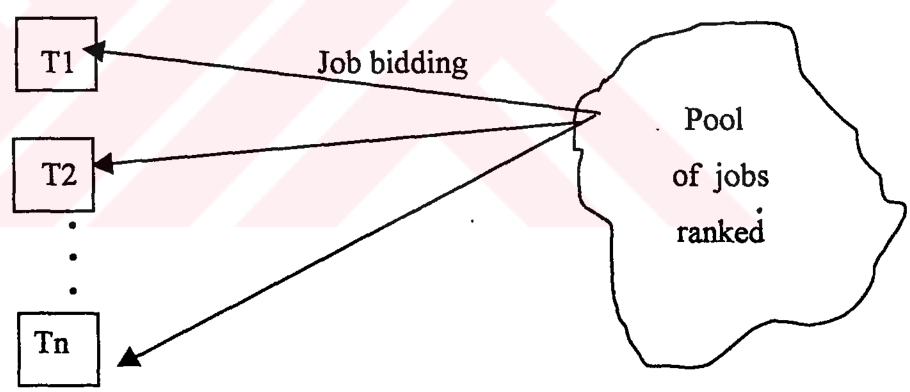


Figure 4.1. A schematic view of Algorithm D1

If a team has to prepare one bid for a job and another bid for a particular operation of the same job for another team, these bids are prepared separately by using the existing schedule. Although the team is competing with other teams to win this job, it tries to prepare as good bid as possible for these operations of the same job for which others team requested the help. This is for the sake of optimising overall system objectives. On the contrary, if a team behaves selfishly and does not do its best while helping other teams, it may loose the operations. Even though this mechanism gives us a land of collaboration between teams, it brings its own limitations or constraints on future bids. Since the dependency among the teams increases (or the flexibility of the teams decreases) due to the commitments made by the teams (e.g., right shift on the other team are not allowed for checking possible positions to insert a new operation in the existing schedule.

There are mainly four steps: 1) Ranking schedulable operations, 2) Bid request, 3) Bid preparation, and 4) Bid selection (see Figure A.4)

Ranking schedulable jobs: Schedulable jobs are ranked according to a predetermined criteria. This can be done by job or customer urgency or by assigning higher priority to the jobs which have to visit bottleneck machines. Rules such as EDD (Earliest Due Date), Slack/NOP (Number of Operations) can also be used to rank the jobs or break ties.

Bidding: A new job (part) broadcasts its arrival and requests bids either by the itself or by the help of a manager agent. The product teams prepare the bids considering their own capabilities (those which are incapable do not bid). The best bid is selected according to some criteria. The steps of the bidding mechanism used here are as follows:

Bid request: Manager agent sends the necessary bid information regarding the job to the team managers. This information can include earliest start time and latest completion times of the job, operation times, set-up times and material handling times.

Bid preparation: Bidding for the jobs is initiated by the overall system manager (i.e. manager agent). The teams are the bidders for the jobs. A team manager having received the bid request initiates bidding for the operations of the job. For the operations that can not be performed within the team, the bid is requested from other teams. In this case, the team preparing the bid for the job acts as bid manager while other teams are bidders.

Bid preparation for a job: During this process, team managers employs a bidding mechanism to assign the operations of the current job to machines. If the material handling is negligible within the team, H2 and H3 can be used as the bidding mechanism. But if the operations can not be processed by the team, the team manager has to request bids from other teams, H3 can also used for this purpose since H3 explicitly consider the material handling. In our current structure of the algorithm, H1 is implemented.

Bid preparation for an operation: Bid is prepared by machines to the manager agent of the product team. Any machine preparing a bid for the operation uses the availability time of the operation and executes the following steps:

1. Find the first possible place for the operation in the sequence according to the local objectives by using a dispatching or scheduling rule and determine the range (RANGE) of possible positions (i.e., from this position to the end of schedule).
2. Check each position in the range (RANGE) by considering the right slacks of the operations that the team has already promised to deliver for another machine and the right slacks of those operations whose successors are processed by other teams. If there is a violation, repeat Step 2 for the remaining positions.
3. From the first possible position at the end of Step 2, check the possible positions considering the second best bids of all affected jobs (including those on other machines).

During these steps of the algorithm, the following information is required: Second best bids for the affected jobs, right operation slacks for operations whose successors are processed by other teams and for operations which are processed for other teams, set-up time and transportation time matrices are required for bid preparation.

Bid Selection: The following criteria can be used to select the best bid for a job: EFT, number of bottleneck machines used, least work content, minimum number of set-ups or some other criteria according the overall objectives of the firm (see Appendix B for the numerical example).

4.1.2 Algorithm D2 (Team initiated bidding algorithm)

In algorithm D1, jobs are ranked and bid is initiated by the manager agent for each job in the work list. Then teams prepare bids for the job (sometimes by taking a help from the other teams. In Algorithm D2, local decision makers (managers of product teams) can process all the operations of the jobs. As compared to D1, teams choose the jobs they want to process. Manager agent resolves possible conflicts among teams (see Figure 4.2).

In this algorithm, alternative teams are gathered into groups and each group of teams is associated with a pool of jobs. If there is only one team with a certain production capability, it is also assigned to the pool of jobs it can process. The teams in each group can have different priorities. These priorities are determined by the manager agent according to the teams' expertise, the quality of work done in the past, the importance of the team for achieving overall system objective etc. Each team ranks the jobs in its pool according to local goals and overall system objectives (i.e. using a composite measure of global and local objectives).

Every group executes the following steps until there is no unassigned job in its pool (see Figure A.5):

1. Starting from the most expert team in a group, each team requests a job
2. If two or more teams are volunteer to take the same job, MA resolves the conflict among these teams and makes the final assignment decision.

3. After all teams in a group request a job, the current iteration is completed and all assignments become permanent.
4. If there is any unassigned job, another iteration is initiated and all the steps starting from Step 1 are repeated. If there is no unassigned job then that group stops executing the algorithm.

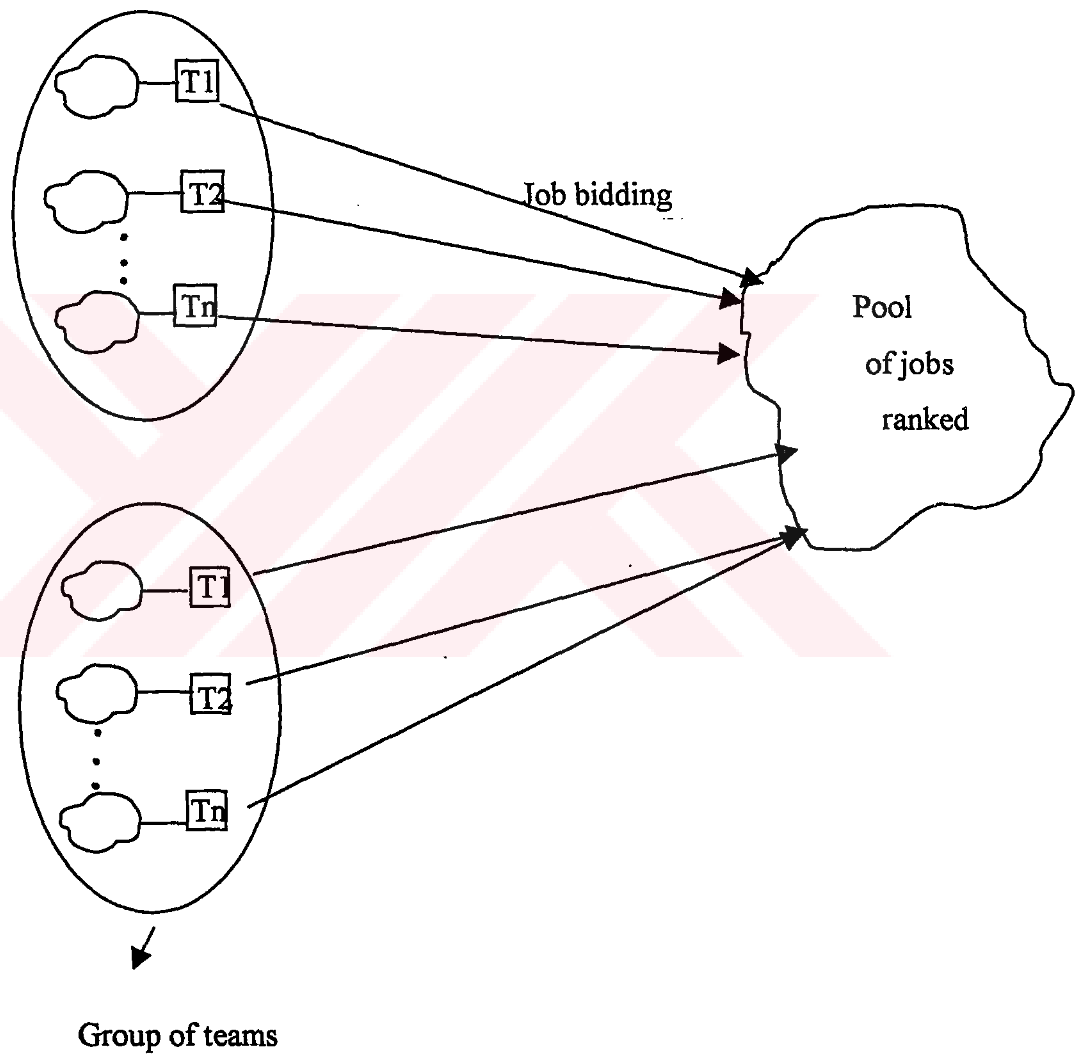


Figure 4.2 Algorithm D1

4.1.3 Experimental Setting

In order to test the performance of the algorithms we used the benchmark problems in the literature. Specifically, we modified the problem sets given in

Lawrence (1984) to generate new problem instances suitable for product team structure. In the current application, we consider two product teams alternative to each other. We also assume that all the operations of a job can be processed within the teams. To generate the problems with these characteristics, we chose two problems (LA01 through LA10 and LA16 through LA20 from Lawrence (1984) and combined into one problem instance. We assumed that machines which have the same numbers in the two problems are alternative to each other but they are in different teams. If a job X has an operation on machine Y in one problem instance, the same job is assumed to be processed on a machine with same number (i.e., machine Y) in the second problem instance. But, the machines specified in the original problem instance are considered to be the ideal machines for that job.

Consider for example five problems: LA01 through LA05 from Lawrence (1984). Each problem instance has 10 jobs (n=10) and five machines (m=5) with 5 operations. We choose two of out of 5 problems and generate new problems with 10 machines and 20 jobs, each with five operations. Note that 5 machines are in the first team and the other 5 are in the second team. Taking 5 to 2 different combinations from LA01 through LA05, we obtain 10 problem instances. We generate the different size problems (i.e., 20/20 and 30/10) in a similar manner (e.g., 10 problems using LA06 through LA10 and 10 problems using LA16 through LA20). As a result, we use a total of 30 problems in the computational experiments. The due dates are added to the problem sets using the method proposed by Ovacik and Uzsoy (1994).

The values used for various parameters are summarised in the table below. As can be seen in Table 4.1, there are 120 problem instances resulted from 30 problems with 10 replications.

The processing time on alternative machine is taken as 11/10 of its original processing time. In addition, set-up time is assumed to take 20 % of the operation time. In the current application, global objective is taken as the minimisation of Lmax

Table 4.1 Experimental factors and their levels

	Values Used		Total
Due Date Range (R)	1.5	2.5	2
% Tardy Jobs (τ)	0.3	0.6	2

and jobs are ranked according to EDD rule to determine an order in which jobs are put in bid. However, we also report the statistics for makespan, average tardiness and number of tardy jobs (see Table D.1 in Appendix). Teams have local goals and they aim to take as many of their ideal jobs as possible. In Algorithm D1, in preparing bid for a job, teams append an operation to the end of the schedule or place it into a position that does not lead to a shift in the schedule, if the operation belongs to a job which is not ideal for the team. On the other hand, to win an ideal job, a team may insert the operations into the schedule even though this insert can cause a shift of other operations scheduled. We did not explicitly described again but; if we run competitive version of Algorithm C in a team we get competitive version of Algorithm D1. This is also considered in the experimentals.

4.1.4 Computational Results

The detailed results are presented in Table D.1 (see the appendix). A summary table (Table 4.2) is also prepared to highlight the important findings. As can be seen in Table 4.2, the collaborative version of Algorithm D1 (D1-Col) is the worst among all three algorithms when Lmax is the global criteria. Hence, we decided to focus on the comparison between D1 and D2 in the rest of the analysis. As seen in Table 4.2, D1 yields better Lmax results than D2 in 75 out of 120 problems. D2 is better in 43 problems and they produce the same results in the remaining two problem instances. Thus we can conclude that D1 is generally better than D2.

When analyzing the results in detail, it can be observed that as due-dates become more tight (τ=0.6), D1 gets significantly better than D2 (e.g., D1 is better in 23

Table 4.2 Comparison of D1 & D2 for Lmax criterion

Size	τ=0.3, R=1.5	τ=0.3, R=2.5	τ=0.6, R=1.5	τ=0.6, R=2.5	Total
n=20, m=10	7 D1, 3 D2	3 D1, 7 D2	8 D1, 52 D2	5 D1, 5 D2	23 D1, 17
n=30, m=10	4 D1, 6 D2	1 D1, 8 D2	6 D1, 4 D2	7 D1, 3 D2	18 D1, 21
n=20, m=20	8 D1, 1 D2	8 D1, 2 D2	9 D1, 1 D2	9 D1, 1 D2	34 D1, 5
Total	19 D1, 10 D2	12 D1, 17 D2	23 D1, 7 D2	21 D1, 9 D2	75 D1, 43

instances out of 30 for R=1.5). However, when the ratio of the number of jobs to the number of machines increases ratio, the performance of D2 becomes better (D2 is better in 21 instances out of 30 for n=30, m=10). Note that the set-up time increases considerably when this ratio increases. The improvement in the performance of D2 in such an environment is due to the fact that it minimizes the set-up and processing times since more jobs are assigned to their ideal machines.

In terms of local performance measures (i.e., the number of jobs that are assigned to their ideal machines or the number times the teams are awarded with the jobs that they requested), D2 performs significantly better than D1 and D1-Col since it gives an ability to the teams to select their jobs. When D1 and D1-Col are compared within each other, D1-Col displays better performance in terms of the local objectives (see Table 4.3 for the comparison of D1 and D1-Col in terms of the local performance criterion). For example, when D1-Col is used for n=20 and m=10, on the average a team is awarded 9.15 out of its 10 ideal jobs on the average, for τ=0.3 and R=1.5. The improved performance of D1-Col over D1 is due to the fact that teams in D1-Col can delay the promised completion times of jobs they were awarded previously, as long as this delay does not exceed the second best bid offered and hence can propose better completion time for the current job in bid. They can also use this opportunity to better satisfy their local goals.

Table 4.3 Comparison of D1&D1-Col for local performance criterion

Size	τ=0.3, R=1.5	τ=0.3, R=2.5	τ=0.6, R=1.5	τ=0.6, R=2.5
n=20, m=10	9.15/10 D1-Col, 7.5/10 D1	9.8/10 D1-Col, 7.85/10 D1	9.5 /10 D1-Col, 7.55/10 D1	9.35/10 D1-Co 7.7/10 D1
n=30, m=10	14.5/15 D1-Col, 11.7/15 D1	14.4/15 D1-Col, 11.55 D1	13.9/15 D1-Col, 10.75/15 D1	14/15 D1-Col 11.7/15 D1
n=20, m=20	9.9/10 D1-Col, 7.4/10 D1	9.7/10 D1-Col, 7.55/10 D1	9.75/10 D1-Col, 6.75/10 D1	9.85/10 D1-Cc 7.7/10 D1

4.2 Conceptual Comparison of the Algorithms

The algorithms presented are based on different architectures and teams in distributed scheduling. We can summarise the essential differences as follows:

1. In Algorithms B1, C, D1 jobs or operations choose the teams they will be processed by, whereas in B2 and D2 teams select the operations or jobs.
2. In Algorithms B1, B2 and C, operations are put in bid assuming both scenario 1 and scenario 2 mentioned by Rahimifard and Newman (1998) can exist in the system. In B1 and B2 manager agent ranks the operations. In C, no operation of a new job is put in bid unless all the operations of a higher priority job are assigned.
3. In Algorithm B1, resource agents have some kind of control over their actions but the autonomy they have in Algorithm B2 is more. Because in B2, resource agents determine priorities by themselves and choose their own tasks. The same is true for Algorithms D1 and D2.
4. Especially in collaborative version of Algorithm B1, there are communication and collaboration between the teams, but in B2 there is not a direct communication between teams. So social ability of teams in B1 is higher than that of teams in B2. The same is true for D1 and D2.
5. As teams do not decide which operation/job to process in B1 and D2 themselves, they are less pro-active as compared to their roles in B2 and D1.

6. Algorithm A (see Appendix E) is different from all others in the sense that there are both resource agents and order agents in the system. There is no bidding mechanism. The algorithm is based on an iterative mechanism through which each agent solves its conflicts independent from all others.

These differences are summarised in the table below:

Table 4.4 Distributed System characteristics of the algorithms

	B1	B2	C	D1	D2	A
Autonomy	**	***	**	****	***	*****
Social Ability	***	**	***	***	****	**
Pro-activeness	**	***	**	***	**	**
Reactivity	***	***	***	***	***	***
Team type	Process	Process	Process	Product	Product	Process

Chapter 5

CONCLUSION AND FUTURE RESEARCH WORK

This thesis is mainly composed of three parts. In the first part, we reviewed the existing studies in distributed scheduling within the classification framework developed. We have identified the characteristics of these studies as well as the differences from traditional systems. DS is different from centralized systems due to the following facts: 1) Orders, products, resources, etc. are given intelligence because; they can act without the intervention of a human being. They use this intelligence in making their own schedules. The local schedulers in a DS System are called *agents* in distributed artificial intelligence; *holons* in holonic manufacturing; *teams* in team based systems. 2) The scheduling problem is divided into sub-problems where each sub-problem is assigned to a local decision-maker. 3) Local schedulers act independently from each other and they do not have an overall view of the whole system. But they should have the social ability to communicate with each other. The most important disadvantage of a DS System is that; it sacrifices from global objectives. Thus the communication ability should be used to overcome conflicts in the schedule and keep the global objectives as much as possible.

As a result of this literature survey, we found out that; a quasi-heterarchical system where the manager agent considers overall objectives of the system should eliminate if not all but some of the disadvantages of a DS architecture. We also observed that most of the existing studies are being implemented in the on-line mode (i.e., they simply append a new operation to the end of the current schedule). Of course this limits an agent's ability to put an operation in a position that would better optimize local objectives. Thus in order to cope up with these problems we developed quasi-heterarchical distributed systems that schedules the operations ahead of time (i.e., off-line). Bidding is used as a communication mechanism and agents have the ability to insert an operation to a position to satisfy their local goals better. But this insertion ability is limited with the constraints that keep global performance of the system. One of the other deficiencies of the existing studies in the literature is that, these systems have not been compared with centralized scheduling systems. But as will be discussed later, we compared some of our algorithms with the best found results of some well known data sets for the centalized environment.

In the second part, we proposed three bidding based algorithms (i.e. B1, B2 and C) for the process team architecture. In the bidding based systems, local planners (or resource agents) generate their own schedules under the supervision a global planner (manager agent) that has the overall system view. In the first method (Algorithm B1), the bid is requested for each operation by the manager agent. It is basically an operation initiated bidding algorithm. In the second method (Algorithm B2), the resource agents initiate the bid. We tested B2 on the known job shop problems with optimum makespan criterion. Since it gives more emphasis on achieving local goals, it did not perform well under these pilot runs. Hence we did not pursue the testing of this algorithm in the later part of the thesis due the time limitation. However it can be undertaken as a future research topic and be tested under highly distributed systems with alternative machines. In the third method (Algorithm C), the bidding is performed for all the operations of a job before another job. We tested two versions (Competitive and collaborative) of the algorithm.

In the third part of the thesis, we presented two new algorithms (i.e., Algorithm D1 and Algorithm D2) for the product team architecture. Algorithm D1 applies to the situations in which the jobs are allocated among the teams (or agents) by bidding and each team is capable of performing all or most of the operations of the job. Algorithm D2 gives the teams the ability to volunteer for their ideal jobs.

Even though the results of the study are discussed in detail in each related chapter, we summarize the major findings below:

- In the second part, the algorithms were tested under some well-known classical job shop problems in the literature. Our best of all results for the L_{max} criterion are significantly better than the best of all results found in the literature.
- It turns out that the bidding algorithms perform better in the loose and distinct due-date cases. However even though, due-dates are very tight and near to each other, our results are still better than the results found in the literature.
- Algorithm B1 and B1-Col give the same schedule when the global ranking rule and scheduling rules of local decision-makers are the same.
- Under the classical job shop problems with sequence dependent set-up times, Algorithm B1 performed better than others. This is followed by the collaborative and competitive versions of C. The difference between two versions of C is especially observed in the favour of C-Col when due-dates are tight. Thus, the collaboration of local decision-makers should be encouraged when global objectives are hard to be satisfied.
- The results indicate that Algorithm B1 is the best for the makespan criterion and Algorithm C is the best for average tardiness and number of tardy jobs criteria.
- When alternative machines exist, B1 is the most effective to improve L_{max}.
- In the alternative machine case, Algorithm C is the best for average tardiness and Algorithm B1 is the worst.
- As the local decision-makers that have the same local scheduling rule (different than the global ranking rule) increases, the average performance of the machines that satisfy their local goal decreases.
- Competitive version of Algorithm D1 performs better than both its collaborative version and Algorithm D2 when the global objectives are considered. As due-dates become more tight, D1 gets significantly better than D2. However, when the ratio of the number of jobs to the number of machines increases, the performance of D2 becomes better.

- If the local objectives are considered for product team structure Algorithm D2 is the best among all the algorithms tested in Chapter 4.

As we pointed out earlier in this thesis, the algorithms presented in this study are not designed to optimise a certain performance criterion. They are general approaches, but we used them in some specific cases to test their performance. For different global and local rules, the algorithms can easily be modified.

We presented Algorithm B2 in Chapter 3. We did not include it in our experiments since its performance was poor in our pilot runs. But we believe that in distributed systems where local goals are very important, it may give better results and hence it requires further investigation.

We could not use H_2 and H_3 bidding mechanisms because of lack of data. They can be tested in future studies when the required data is collected.

In this study, we also worked on the purely heterarchical and distributed system. As a result we developed a new algorithm which we call Algorithm A. We give the details of this algorithm in appendix (Appendix E). The coding and implementation of this algorithm requires major effort. Hence, it is left for further research.

Finally, as can be noted we tested all the algorithms proposed in this study under deterministic and static environment. However, they should be tested under dynamic and stochastic environment. They should also be compared with several reactive scheduling approaches. In fact this is a challenging research topic since it requires a very detailed experimentation. Results of such an analysis can also provide very useful information about the comparisons of (or strengths and weaknesses) distributed and centralized systems.

Bibliography

[1] Agarwal R., De P., Wells, C. E., 1995. "Cooperative Distributed Problem Solving: An Investigation in the Domain of Jobshop Scheduling," Proceedings of the 28th Annual Hawaii International Conference on System Sciences.

[2] Amelsvoort, P., Benders, J., 1996. "Team time: a model for developing self-directed work teams," International Journal of Operations & Production Management, Vol. 1, No.2, 159-170.

[3] Aydin, M.E., Oztemel, E., 1998. "Dynamic Job-Shop Scheduling Using Intelligent Agents," 2nd International Symposium on Intelligent Manufacturing Systems.

[4] Basnet , C. and Mize, J.H. 1994. "Scheduling and Control of Flexible Manufacturing Systems: A Critical Review", International Journal of Computer Integrated Manufacturing, Vol. 7, No. 6, 340-355.

[5] Blackstone, J.H., Phillips, D. T., and Hogg, G.L.1982. "A State-of-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations", International Journal of Production Research, Vol. 20, No. 1, 27-45.

[6] Bongaerts L., Valckenaers P., Brussel H. V., Wyns J. "Schedule Execution for a Holonic Shop Floor Control System,"

[7] Bongaerts L., Wyns J., Detand J., Brussel H. V., Valckenaers P. "Identification of Manufacturing Hollons,"

[8] Burke, P., Prosser, P.,, 1991. "A Distributed Asynchronous System for Predictive and Reactive Scheduling", Artificial Intelligence in Engineering, Vol.6, No.3, 106-124.

[9] Chang, J.W. Luh, Y.P, 1997. "Integration of Scheduling and Control in a Job Shop," Journal of the Chinese Institute of Engineers, Vol. 20, No. 1, pp. 67-76.

[10] Cheng, T.C..E. and Sin, C.C.S. 1990. " A State-of Art Review of Parallel machine Scheduling Research," European Journal of Operational Research, Vol. 47, 271-292

[11] Chiu, C. Yih, Y., 1995. "A Learning-Based Methodology for Dynamic Scheduling in Distributed Manufacturing Systems," Int. J. Prod. Res., Vol. 33, No. 11, 3217-3232.

[12] Chung, D., Park, C., Kang, S., Park, J., 1996. "Developing a Shop Floor Scheduling and Control Software for an FMS," Computers and Industrial Engineering, Vol. 30, No. 3, pp. 557-568.

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

[14] Day, Hottenstein, 1979. "Review of Sequencing Research. Naval Res. Logist. Quart. 17, 14-40.

[15] Decker, K.S., 1987. "Distrbuted Problem-Solving Techniques: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Vol. Smc-17, No. 5, 729-740.

[16] Demirkol E., Mehta, S., Uzsoy, R., 1998. "Theory and Methodology, Benchmarks for Shop Scheduling Problems", European Journal of Operational Research, 109, 137-141.

[17] 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.

[18] 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.

[19] Duffie, N.A., Prabhu, V.V., 1996. "Heterarchical Control of Highly Distributed Manufacturing Systems," International Journal of Computer Integrated Manufacturing, Vol. 9. No 4. 270-281.

[20] Day, J.E. and Hottenstein, M.P. 1970, " Review of Sequencing Research", Naval Research Logistics Quarterly, Vol.17, No.1, 11-39.

[21] Graham R.L., Lawler E. L., Lenstra, J. K. and Rinnoy, K. AHG, 1979, "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey", Ann. Discrete Math. 5, 287-326.

[22] Graves, S. C., 1981. "A Review of Production Scheduling," Operation Research, Vol. 29, No. 4, 646-675.

[23] Gupta, S.K. and Kyparisis, J. "Single Machine Scheduling Research," OMEGA, Vol. 15 207-227.

[24] Hadavi, K., Hsu, W-L., Chen T., Lee, C.N., 1992. "An Architecture for Real-Time Distributed Scheduling," In Artificial Intelligence Applications in Manufacturing, A. Famili et al. (Ed.) (Cambridge USA : AAAI Press), 215-234.

[25] Hax, A.C., and H.C. Meal, 1975. "Hierarchical Integration of Production Planning and Scheduling in Studies in Management Sciences, Vol I., Logistics, M.A. Geisler (ed.) North Holland-American Elsevier, New York.

[26] Hvolby, H.H., Hϕjbjerre, P., 1994. "A More Flexible Planning Architecture for Centralized and Decentralized Planning," Production Management Methods, C. Walter (Ed.) (Holland : Elsevier Science Publishers B. V.), 317-324.

[27] Kanet J., J. and Adelsberger, H.H., 1987. "Expert Systems in for Production Scheduling", European Journal of Operational Research, Vol. 29, 51-59.

[28] Koulamas, C. 1`994 "The Total tardiness Problem: Review and Extensions", Operations Research, Vol. 42, No.6, 1025-1041.

[29] Kusiak, A. and Chen, M. 1988. "Expert Systems for Planning and Scheduling Manufacturing Systems", European Journal of Operational Research, Vol. 34, 113-130.

[30] Kutanoglu, E., Wu, S.D., "An Auction-Theoretic of Production Scheduling to Achieve Distributed Decision Making," Department of Industrial and Manufacturing System Engineering, Lehigh University, 97W-001.

[31] Lawrence, S., "Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques", GSIA, Carnegie Mellon University, 1984.

[32] Lee, K.I., Kang, M., Park, J.H., 1996. "A collaborative scheduling system for make-to-order manufacturing," Annals of CIRP, Vol. 45,1,461-464.

[33] Lin, G. Y, and Solberg, J.J. 1992. "Integrated Shop Floor Control Using Autonomous Agents," IIE Transactions, Vol.24, Number .3

[34] Liu, J., Sycara, K. P., 1993. "Distributed Constraint Satisfaction Through Constraint Partition and Coordinated Reaction," Proceedings of the 12th International Workshop on Distributed AI.

[35] Liu, J., Sycara, K.P., 1994. "Distributed Problem Solving through Coordination in a Society of Agents," Proceedings of the 13th International Workshop on Distributed AI.

[36] Liu, J., Sycara, K.P., 1995. "Exploiting Problem Structure for Distributed Constraint Optimization," Proceedings of the First International Conference on Multiagent Systems, San Fransisco, California, June.

[37] Maturana, F.P., Norrie, D.H., 1996. "Multi-agent Mediator Architecture for Distributed Manufacturing," Journal of Intelligent Manufacturing, 7, 257-270.

[38] MMS Soft Corporation, 1994, 1998, San Diego, CA 92117.

[39] Oguz, S., June, 1998. "Object Oriented Design of a Distributed Scheduling System", Unpublished Master Thesis, Bosphorous University.

[40] Ovacik, I.M., Uzsoy, R., Decomposition Methods for Complex Factory Scheduling Problems, Kluwer Academic Publishers.

[41] Ovacik and Uzsoy, R., 1994. "Exploiting Shop Floor Status Information to Schedule Complex Job Shops", Journal of Manufacturing systems, 13, 2, 73-84.

[42] Panwalkar, S.S., and Iskender, W. 1977. " A survey of Scheduling Rules", Operations Research, Vol.25, No.1, 45-61.

[43] Parunak, H.V.D., 1988. "Distributed Artifical Intelligence Systems," A. Kusiak (Ed.), Artificial Intelligence: Implications for CIM, IFS, UK., 225-251.

[44] Preactor, The CIMulation Centre, Bumper way, Chippenham, Wilshire, England SN146RA.

[45] Rachamadugu, R. and Stecke, K. 1994. "Classification and Review of FMS Scheduling Procedures," Production Planning and Control, Vol. 5, No. 1, 2-20.

[46] Rahimifard, S., Newman, S.T., 1998. "Reference Architectures for Team Based Distributed Production Planning and Control," Eureka – Factory: Project No 1629, January.

[47] Sabuncuoglu, I. and Toptal, A., 1998. "Distributed Scheduling Algorithms," Eureka- Factory: Project No 1629, October.

[48] Sabuncuoglu, I., 1998. "Scheduling with Neural Networks: A review of the Literature and New Research Directions", Production Planning and Control, Vol. 9, No. 1, 2-12.

[49] Sen, T. and Gupta, S.K., 1984. "A State-of Art Survey of Static Scheduling Research Involving Due Dates," OMEGA, Vol.12, No. 1, 63-76.

[50] Shaw, M.J., 1987. "A Distributed Scheduling Method for Computer Integrated Manufactruing: The Use of Local Area Networks in Cellular Systems," Int. J. Prod. Res., Vol. 25, No. 9, pp. 1285-1303.

[51] Shaw, M.J,1988."A Distributed Knowledge-Based Approach to Flexible Automation: The Contract Net Framework," The International Journal of Flexible Manufacturing System, 1, 85-104.

[52] Shaw, M.J., Solberg, J.J., Woo, T.C., 1992. "System Integration in Intelligent Manufacturing: An Introduction," IIE Transactions, Vol.24, No. 3, 2-6.

[53]Steffen, M.S., 1986. "A Survey of Artificial Intelligence-Based Scheduling Systems", Proceedings of Fall Industrial Engineering Conference, Dec 7-1, Boston, MA, 395-405.

[54] Sucur, M., Coskunoglu, O., 1995. "A Critical Review of Research in Knowledge-Based Scheduling Systems", Institute of Industrial Engineersa, 4^{th} Industrial Engineering Research Conference Proceedings.

[55] Suresh, V., Chaudhuri, D., 1993. "Dynamic Scheduling-A Survey of Research," International Journal of Production Economics, 32. 53-63.

[56] Sycara, K.P., Roth, S.F., Sadeh, N., Fox, M.S., 1991. "Resource Allocation in Distributed Factory Control," IEEE Expert, February, 29-40.

[57] Szelke, E. and R.M. Kerr, 1994. "Knowledge-based Reactive Scheduling," Production Planning and Control, Vol.5., No.2., 124-145.

[58] Tharumarajah, A., Wells, A.J., Nemes, L., 1996. "Comparison of the Bionic, Fractal and Holonic Manufacturing System Concepts," Int. J. Computer Integrated Manufacturing, Vol. 9, No.3, 217-226.

[59]Veeramani, D., Wang, K. J., 1997. "Performance Analysis of Auction-Based Distributed Shop-Floor Control Schemes from the Perspective of the Communication System," The International Journal of Flexible Manufacturing Systems, 9. 121-143.

[60]Yang, E.H., Barash, M.M., Upton, D.M., 1993. "Accomodation of Priority Parts in a Distributed Computer-Controlled Manufacturing System with Aggregate Bidding Schemes," 2^{nd} Industrial Engineering Research Conference Proceedings.

[61] Zhang T., Stanley, K, Smith, M.H., Gruver, W.A., 1998."Multi-Agent Techniques in Holonic Manufacturing Systems," 2^{nd} International Symposium on Intelligent Manufacturing Systems.

Appendix A

Flow Charts of the Proposed Algorithms

Figure A.1 Algorithm B1

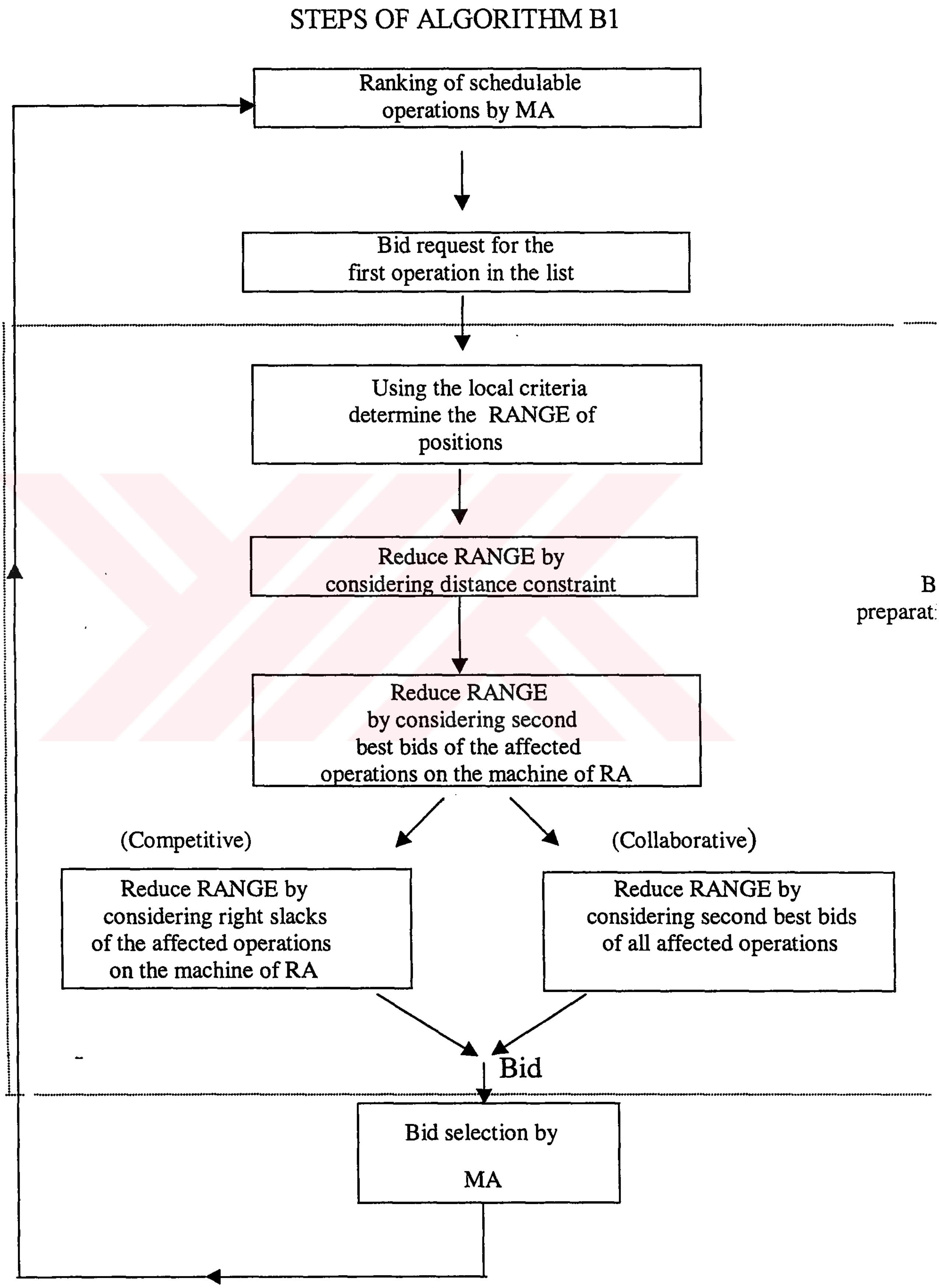


Figure A.2 Algorithm B2

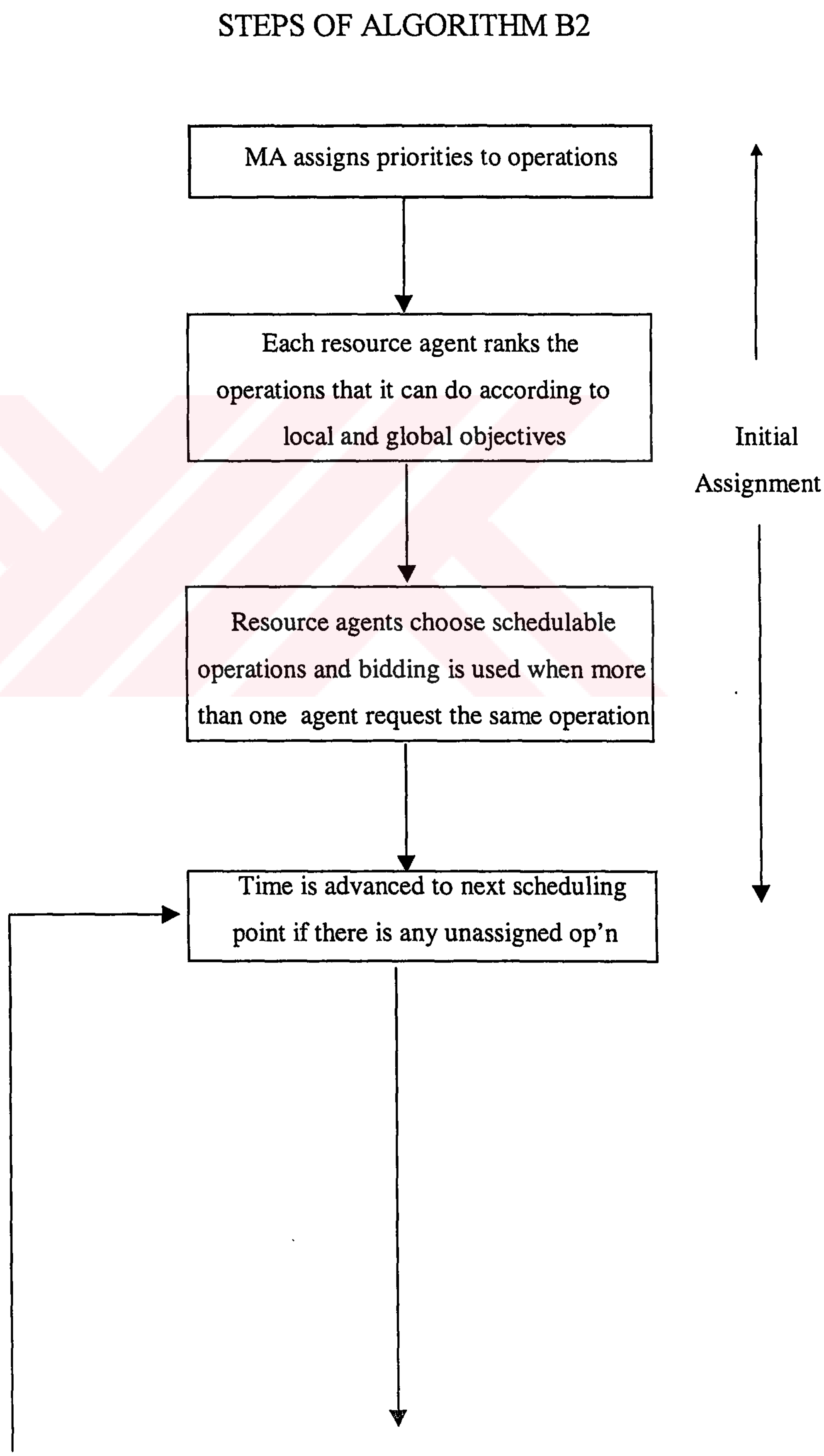


Figure A.2 (Cont)

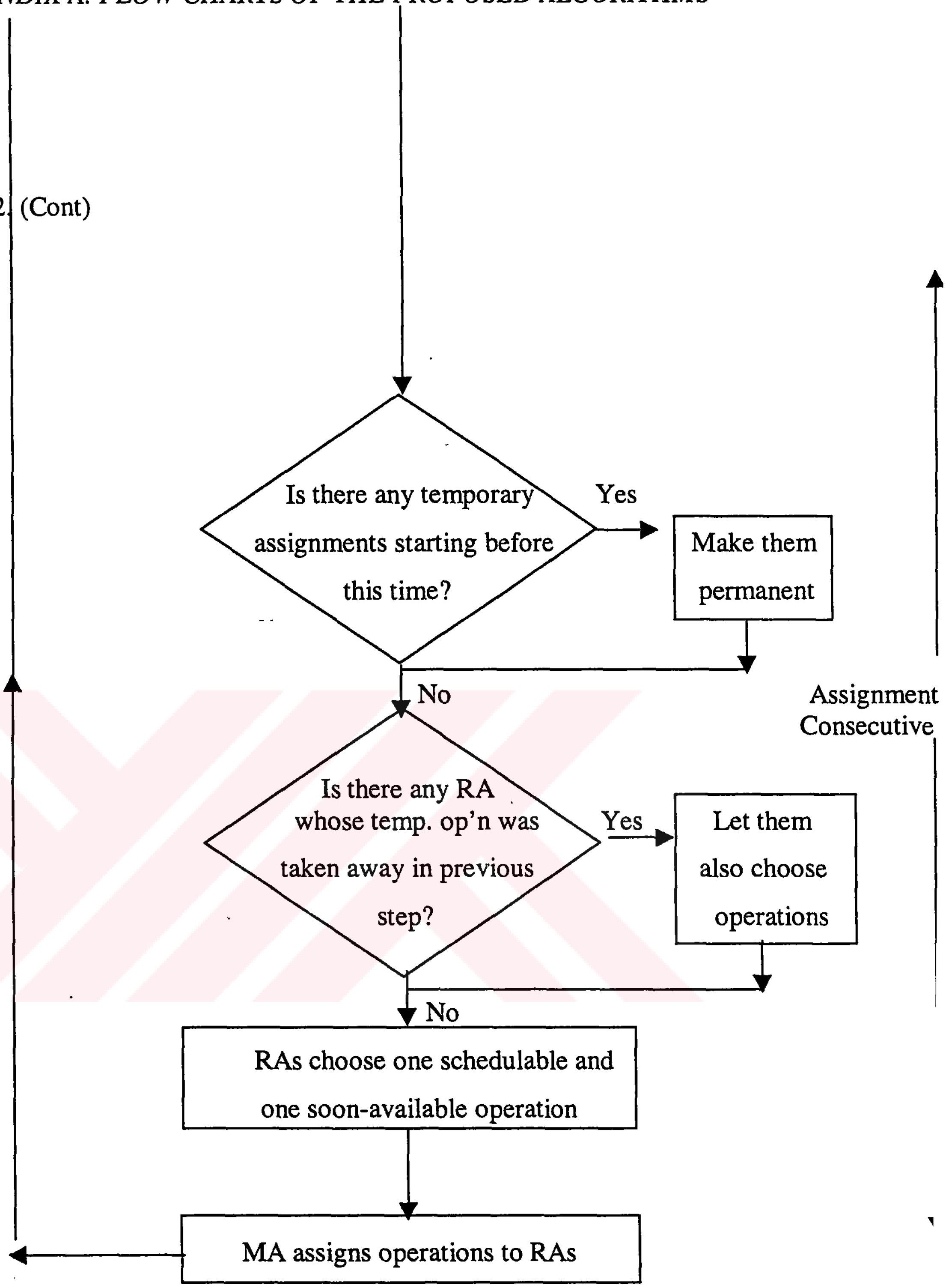


Figure A.3 STEPS OF ALGORITHM C

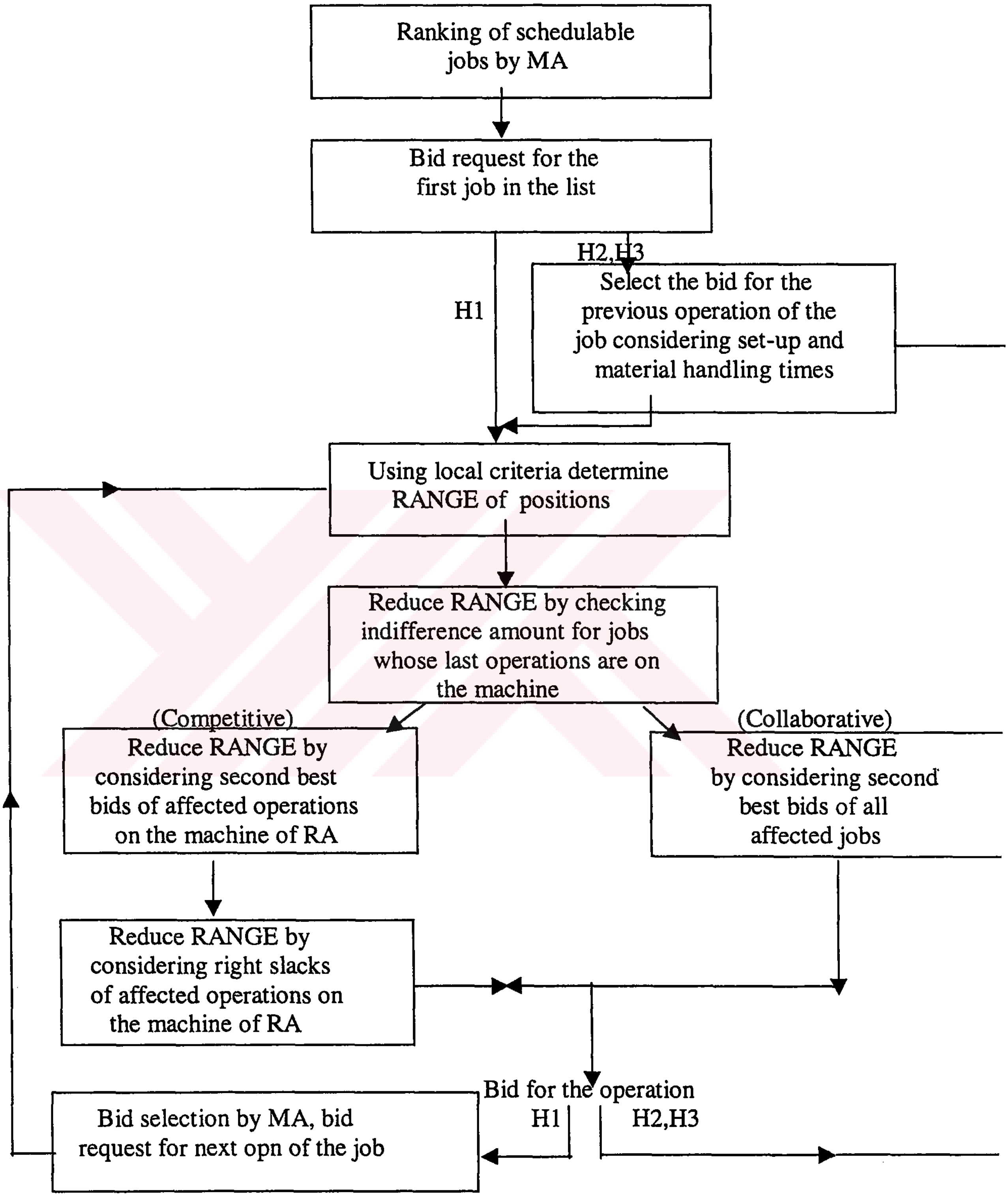


Figure A.4

STEPS OF ALGORITHM D1

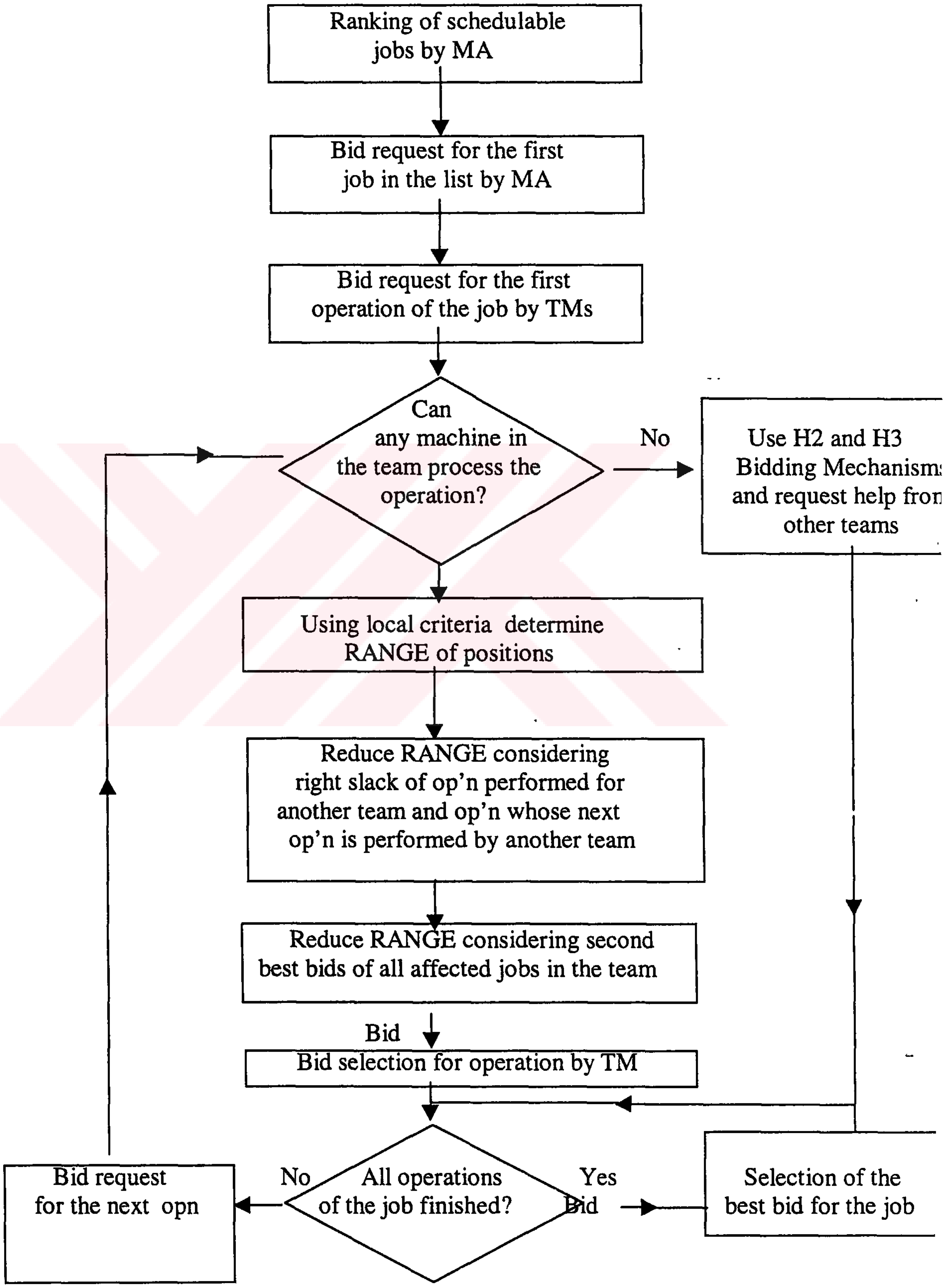
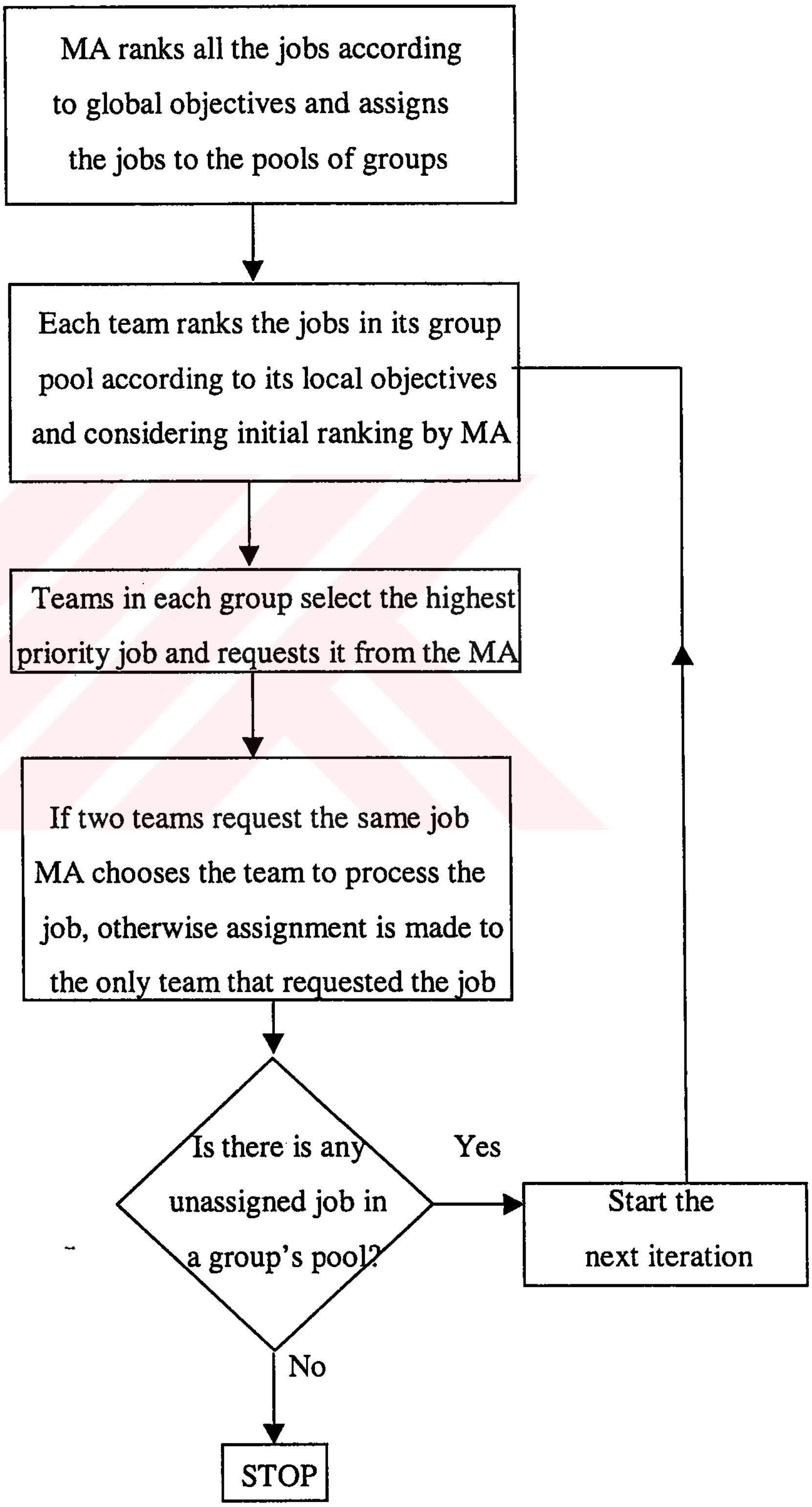


Figure A.5

STEPS OF ALGORITHM D2



Appendix B

The Example Problem

EXAMPLE PROBLEM

Suppose that there are four resource agents (or machines) to process four jobs. The job and system related information are given in Table B.1 and Table B.2.

Table B.1 Job information

Jobs	ES- P-LC (Alternative machines)				Due-dates
A	0-4-8 (I-II)	4-6-14 (II)	10-2-16 (II-III)		16
B	0-7-13 (II-III)	7-3-16 (III-IV)	10-5-21 (III)	15-4-25 (III-IV)	25
C	0-8-14 (IV)	8-6-20 (I-IV)	14-3-23 (IV)		23
D	0-5-9 (I-III)	5-4-13 (II)	9-7-20 (I-IV)	16-6-26(I)	26

ES: Earliest start time, P: Processing time

LC: Latest completion time which is found by subtracting the operation times from the due-date of the job

Assume that Resource Agent 1 (RA_1) and RA_3 schedule (or dispatch) the operations according to the SPT rule, RA_2 utilises the LPT rule and RA_4 makes the decision based on utilisation. The transportation and set-up times are given below:

Table B.2 System related information: a. Transportation time matrix, b. Set-up time matrix

(a) Transportation time matrix (m/c × machines)

m/c	I	II	III	IV
I	0	2	1	1
II	2	0	2	2
III	1	1	0	1
IV	3	1	3	0

(b) Set-up time matrix (Jobs × Jobs)

Jobs	A	B	C	D
A	0	1	0.5	1.5
B	1	0	2	0.5
C	1	3	0	1
D	2	1	2	0

Latest completion time of operations is used by manager agent to rank S (the set of schedulable operations). Job urgency is used to break ties, if necessary. Bid selection is made in such a way that the bidder who offers the best EFT (Earliest Finish Time) for the operation is awarded. In addition, MOP (Maximum number of Operations of the same job that can be Processed), workload and utilisation of bidders are used to break ties.

Table B.3 Iterations of Algorithm B1 on the example problem

Iteration	Set of schedulable	Operation in bid	Bids prepared	Bid selected	Selection criteria operations
1	$\{A_1,D_1, B_1,C_1\}$	A_1	RA_1 (4), RA_2 (4) *	RA_2	EFT/ MOP**
2	$\{D_1,B_1, A_2,C_1\}$	D_1	RA_1 (5), RA_3 (5)	RA_1	EFT/ MOP
3	$\{B_1, D_2,A_2,C_1\}$	B_1	RA_2 (12), RA_3 (7)	RA_3	EFT
4	$\{D_2,A_2,C_1,B_2\}$	D_2	RA_2 (11)	RA_2	_
5	$\{A_2,C_1,B_2,D_3\}$	A_2	RA_2 (10)	RA_2	_
6	$\{C_1,A_3,B_2,D_3\}$	C_1	RA_4 (8)	RA_4	_
7	$\{A_3,B_2,C_2,D_3\}$	A_3	RA_2 (19.5), RA_3 (14)	RA_3	EFT
8	$\{B_2,C_2,D_3\}$	B_2	RA_3 (10), RA_4 (14)	RA_3	EFT
9	$\{C_2,D_3,B_3\}$	C_2	RA_1 (17), RA_4 (14)	RA_4	EFT
10	$\{D_3,B_3,C_3\}$	D_3	RA_1 (24.5), RA_4 (24.5)	RA_1	EFT/MOP
11	$\{B_3,C_3,D_4\}$	B_3	RA_3 (20)	RA_3	_
12	$\{C_3,B_4,D_4\}$	C_3	RA_4 (17)	RA_4	_
13	$\{B_4,D_4\}$	B_4	RA_3 (24), RA_4 (25)	RA_3	EFT

* RA_2 offers the bid for A_1 with EFT of 4

**Since RA_1 and RA_2 offer the same bid (tie), bid is selected using the second criterion MOP.

Table B.4 Iterations of Algorithm B2 on the example problem

Time	P_1	P_2	Machines to choose ops	Operations chosen	Permanent assignments	Temporary assignments
0	$\{A_1, B_1, C_1, D_1\}$	-	M1,M2,M3,M4	M1-A_1,M2-B_1 M3-D_1-M4-C_1	M1-A_1,M2-B_1 M3-D_1-M4-C_1	-
4	$\{A_2\}$	$\{B_2,D_2,C_2\}$	M1	M1-C_2	-	M1-C_2
5	$\{D_2,A_2\}$	$\{B_2,C_2\}$	M3	M3-B_2	-	M3-B_2
7	$\{D_2,A_2,B_2\}$	$\{C_2\}$	M2	M2-A_2	M2-A_2	-
8	$\{D_2,B_2,C_2\}$	$\{A_3\}$	M4	M4-C_2	M4-C_2	-
12	$\{B_3,D_2\}$	$\{A_3,C_3\}$	M1,M3	M3-B_3,M3-A_3	M3-B_3	M3-A_3
14	$\{A_3,C_3,D_2\}$	$\{B_4\}$	M1,M2,M4	M2-A_3 M4-C_3,M4-B_4	M2-A_3,M4-C_3	M4-B_4
16	$\{D_2\}$	$\{B_4\}$	M1,M2	M2-D_2	M2-D_2	-
17	$\{B_4]$	$\{D_3\}$	M1,M3,M4	M3-B_4,M1-D_3 M4-B_4	M3-B_4	M1-D_3
21	-	$\{D_3\}$	M3,M4	M4-D_3	-	-
21.5	$\{D_3\}$	-	M2,M3,M4	-	M1-D_3	-
30.5	$\{D_4\}$	-	M1,M2,M3,M4	M1-D_4	M1-D_4	-

P_1 : Schedulable operations in the pool

P_2 : Soon-available operations in the pool

Table B.5 Iterations of Algorithm-C on the example problem

Iteration	Operation in bid	Bids prepared	Using the bid of	Final assignment
1	D_1	RA_1 (5)	_	
		RA_3 (5)	_	
2	D_2	RA_2 (10)	RA_3	
3	D_3	RA_1 (19)	RA_2	
		RA_4 (19)	RA_2	
4	D_4	RA_1 * (25)	RA_1	D_1-RA_3, D_2-RA_2, D_3-RA_1, D_4-RA_1
5	B_1	RA_2 (7)	_	
		RA_3 (13)	_	
6	B_2	RA_3 (12)	RA_2	
		RA_4 (12)	RA_2	
7	B_3	RA_3 (17)	RA_3	
8	B_4	RA_3* (21)	RA_3	B_1-RA_2, B_2-RA_3, B_3-RA_3, B_4-RA_3
		RA_4 (22)	RA_3	
9	A_1	RA_1 (4)	_	
		RA_2 (17.5)	_	
10	A_2	RA_2 (19.5)	RA_1	
11	A_3	RA_2* (21.5)	RA_2	A_1-RA_1, A_2-RA_2, A_3-RA_2
		RA_3 (24)	RA_2	
12	C_1	RA_4 (8)	_	
13	C_2	RA_1 (34.5)	RA_4	
		RA_4 (14)	RA_4	
14	C_3	RA_4* (17)	RA_4	C_1-RA_4, C_2-RA_4, C_3-RA_4

* Bid selected by the manager agent for the last operation of a job

Solution of the Sample Problem using Algorithm D1

In the application of the Algorithm D1, jobs are ranked by MA according to criticality ratio and put in bid in this order. It turns out that, D is the first job to be scheduled. We assume that resource agents who can process at least 40% of the operations of a job, can only give bids.

Iteration 1:

1. Bid request: Bid is requested for Job D which has the highest criticality ratio.
2. Bid preparation: Team 1 is the only team that can offer the bid because M1 performs most of the operations of the job except the second operation. Since RA_1 is initially empty, the first operation of D (D_1) is loaded on M1 with a completion time of 5 time units. For the next operation (D_2), a bid is requested from other teams. This operation is assigned to RA_2 because M2 is the only machine that can process D_2. M2 receives the job at 7 after 2 units of transportation time and makes it available on M1 at time 13. Remaining two operations of Job D can be processed by RA_1. Hence these operations are loaded on the machine one after another with the completion times of 20 and 26, for the operations D_3 and D_4, respectively.
3. Bid selection: Since RA_1 is the only resource agent that prepares bid for Job D, it is awarded the job (see Figure B.1) .

Iteration 2:

1. Bid request: The next job with the highest criticality ratio is Job B. Hence the bid is requested for Job B.
2. Bid preparation: RA_3 and RA_4 prepare bids for Job B. RA_3 can process all the operations of Job B. Since it is initially idle, operations of Job B are loaded on M3 one after the other with the job completion time of 19 (=7+3+5+4).

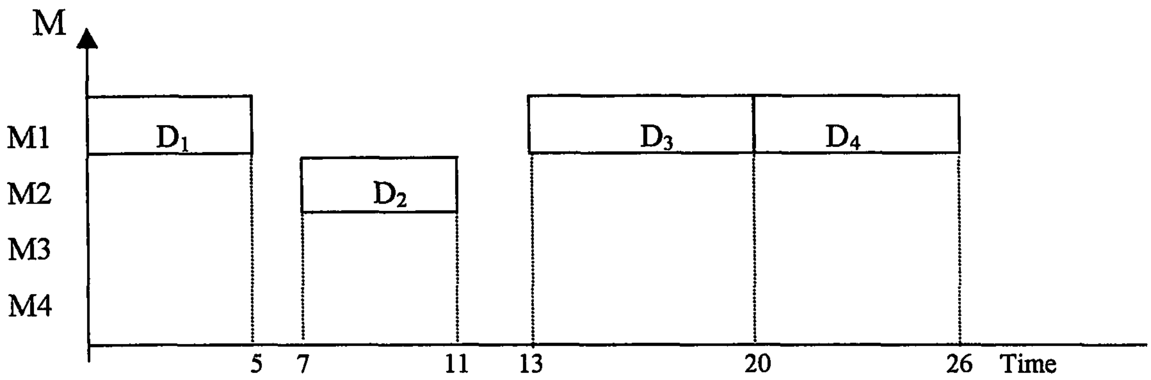


Figure B.1 Gannt chart after iteration 1

RA_4 can also prepare a bid for Job B even though it can not perform first and third operations. First it opens the bid for operation B_1. There are two candidate machines that can process B_1: M2 and M3. Earliest availability time (or bid) by RA_2 is 9. This is obtained by collaborating with RA_1 to shift its operations by 0.5 unit. On the other hand, the bid from RA_3 is 8 (=7+1). Thus, the bidder RA_4 chooses RA_3 to process B_1. The second operation B_2 is performed by RA_4 with the ending time of 11. Since RA_4 is not capable of processing B_3, it asks help from the other teams. It appears that third team is the only team that can offer help with the scheduled completion time on M3 is 19 time units (=11+3+5). This makes the job available on M4 at time 20. Hence, the last operation of the Job B_4 can be scheduled on M4 with the finish time of 24.

3.Bid selection: MA chooses RA_3 to process Job B, because it finishes the job earlier (see Figure B.2).

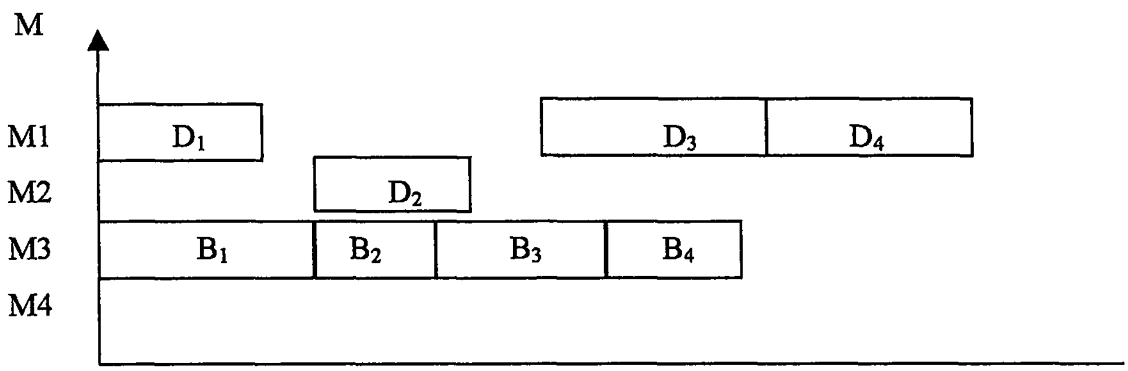


Figure B.2 Gannt chart after iteration 2

Iteration 3:

1. Bid request: Bid is requested for Job A.

2. Bid preparation: Team 2 is the only team that can offer the bid because M2 performs all operations of Job A. Currently there is only one operation (i.e. D_2) scheduled on M2. M2 can process A_1 before D_2 with a completion time of 4 time units. However A_2 can not be processed just after A_1 is completed. Because due-date of job D is exceeded. Thus A_2 is scheduled after D_2 with a completion time of 19 time units. The last operation of Job A (A_3) is also assigned to the end of A_2 with no violation and is completed at time 21.

3. Bid selection: RA_2 is awarded with the job (see Figure B.3).

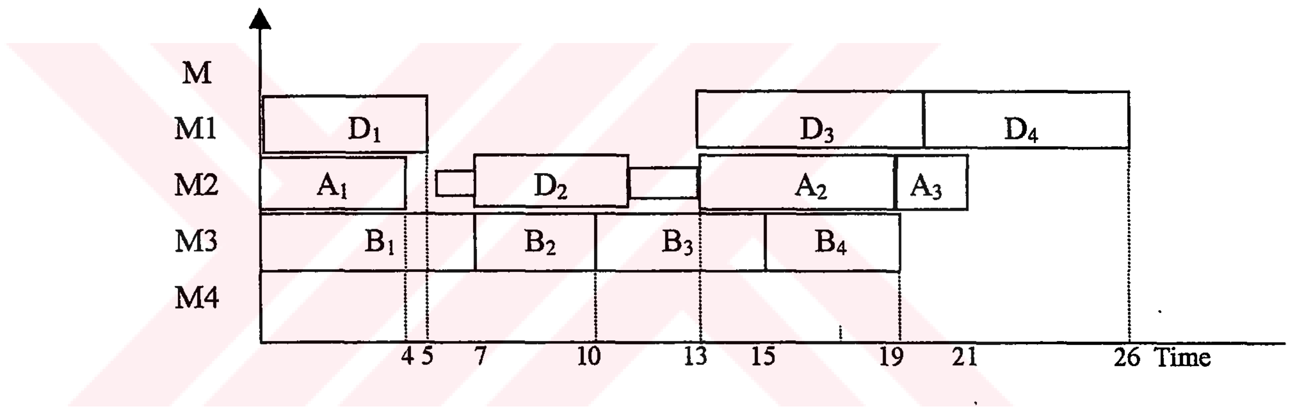


Figure B.3 Gannt chart after iteration 3

Iteration 4:

1. Bid request: Bid is requested for Job C.

2. Bid preparation: Team 4 is the only team that can do all of the operations of Job C. Initially no operation is scheduled on M4, thus it can process the operations of Job C one after another. Thus, as seen in Figure 4, Job C can be completed at 17.

3. Bid selection: RA_4 is assigned to the job (see Figure B.4).

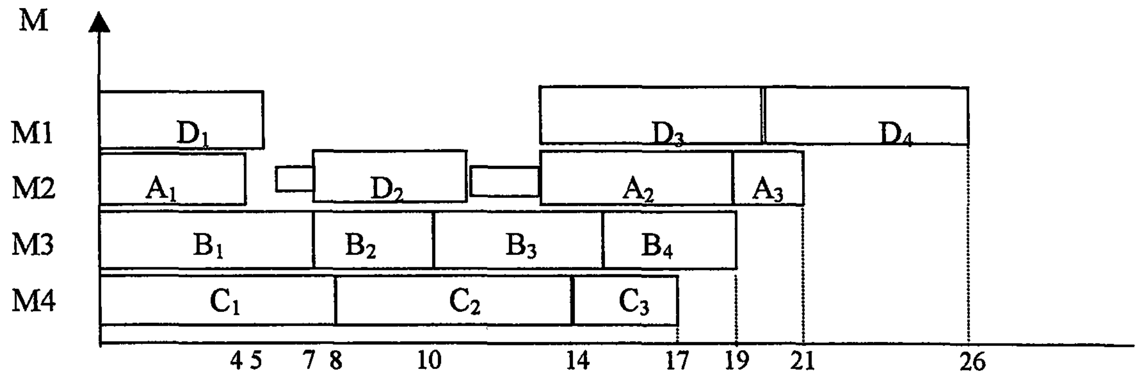


Figure B.4. Gannt chart after iteration 4

Solution of the Sample Problem using Algorithm D2

According to the construction of the example problem, there are four groups of teams each with one m/c or RA (i.e., groups 1,2,3 and 4 correspond to RA_1, RA_2, RA_3 and RA_4, respectively). Every group is also associated with a pool of jobs (i.e., pools of RA_1, RA_2, RA_3 and RA_4 consist of jobs D, A, B and C, respectively).

As there is only one job in the pool for each group, each RA would take the job in its pool. Hence, MA will assign Job A to RA_2, Job B to RA_3 and Job C to RA_4. However, M1 can not process the second operation of Job D, although it wants to process the other operations of the same job. Thus, MA will give a second chance to RA_1 to communicate with other agents and take help from them. D_2 can only be processed by M_2, thus RA_1 will collaborate with RA_2. However, RA_2 does not want the completion time of Job A to exceed its due-date. For that reason it appends D_2 to the end of its schedule in which case Job D is completed by RA_1 at time 32.5. Since there is no better alternative, MA confirms this assignment (see Figure B.5).

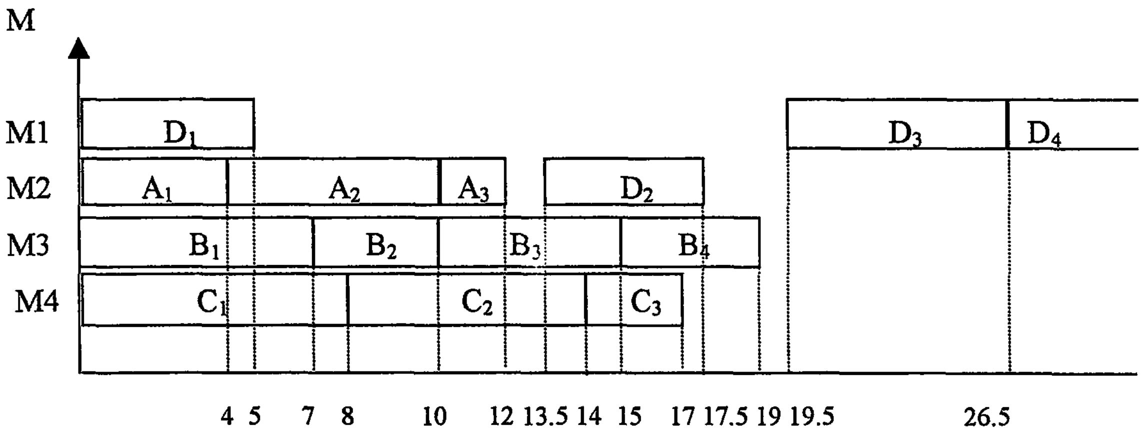


Figure B.5 Gannt chart of schedule

Appendix C

Detailed Results of Chapter 3

Table C.1 Detailed comparison of each algorithm with the best results reported in the literature

T=0.3, R=1.5, 10 jobs, 5 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P161	500	**248**	50.4	486	2.8	541	-8.2	248	50.4
P162	1042	**743**	28.69	875	16.03	952	8.637	743	28.69
P163	863	635	26.42	**530**	38.59	736	14.72	530	38.59
P164	839	**399**	52.44	403	51.97	399	52.44	399	52.44
P168	834	962	-15.3	**890**	-6.71	962	-15.3	890	-6.715
P170	1112	**760**	31.65	760	31.65	1067	4.047	760	31.65
P173	1121	**725**	35.33	1192	-6.33	834	25.6	725	35.33
P176	1408	**971**	31.04	1006	28.55	971	31.04	971	31.04
P179	1102	**789**	28.4	842	23.59	789	28.4	789	28.4
P180	786	208	73.54	408	48.09	**201**	74.43	201	74.43
A.P.D.			34.26		22.82		21.58		36.43
N.B.S.			9		8		8		9
N.W.S			1		2		2		1
B.I.A.			39.77		30.16		29.91		41.22
W.I.A.			-15.3		-6.52		-11.8		-6.715

T=0.3, R=1.5, 20 jobs, 5 machines

P185	1050	1098	-4.57	**1067**	-1.62	1331	-26.8	1067	-1.619
P187	1299	1636	-25.9	**1409**	-8.47	2044	-57.4	1409	-8.468
P188	1016	**1078**	-6.1	1128	-11	1078	-6.1	1078	-6.102
P189	1008	813	19.35	**776**	23.02	1185	-17.6	776	23.02
P190	587	**358**	39.01	358	39.01	616	-4.94	358	39.01
P191	1132	1388	-22.6	**1136**	-0.35	1216	-7.42	1136	-0.353
P192	1388	1609	-15.9	**1501**	-8.14	1767	-27.3	1501	-8.141
P194	1015	1375	-35.5	1148	-13.1	**1081**	-6.5	1081	-6.502
P195	1058	995	5.955	**964**	8.885	995	5.955	964	8.885
P198	978	**1192**	-21.9	1246	-27.4	1317	-34.7	1192	-21.88
A.P.D			-6.82		0.08		-18.3		1.784
N.B.S.			3		3		1		3
N.W.S.			7		7		9		7
B.I.A.			21.44		23.64		5.955		23.64
W.I.A.			-18.9		-10		-21		-7.58

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.3, R=1.5, 10 jobs, 10 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P203	2447	1973	19.37	**1704**	30.36	1768	27.75	1704	30.36
P204	1889	**1449**	23.29	1480	21.65	1449	23.29	1449	23.29
P205	1932	**1414**	26.81	1904	1.449	1536	20.5	1414	26.81
P207	2208	**1251**	43.34	1459	33.92	1399	36.64	1251	43.34
P209	2565	2018	21.33	2640	-2.92	**1945**	24.17	1945	24.17
P211	2440	2574	-5.49	**2026**	16.97	2435	0.205	2026	16.97
P213	1897	**1278**	32.63	1355	28.57	1278	32.63	1278	32.63
P215	2306	1732	24.89	1626	29.49	**1610**	30.18	1610	30.18
P218	1971	**1580**	19.84	1829	7.204	1580	19.84	1580	19.84
P220	2851	2363	17.12	2215	22.31	**2209**	22.52	2209	22.52
A.P.D			22.31		18.9		23.77		27.01
N.B.S.			9		9		10		10
N.W.S.			1		1		0		0
B.I.A.			25.4		21.32		23.77		27.01
W.I.A.			-5.49		-2.92		0		0

T=0.3, R=1.5, 20 jobs, 10 machines

P221	2304	1574	31.68	**1503**	34.77	1523	33.9	1503	34.77
P223	1766	1396	20.95	**1213**	31.31	1250	29.22	1213	31.31
P224	2282	**1463**	35.89	1750	23.31	1656	27.43	1463	35.89
P226	2044	2114	-3.42	1914	6.36	**1864**	8.806	1864	8.806
P227	2211	**1781**	19.45	2072	6.287	1980	10.45	1781	19.45
P228	2215	**1328**	40.05	1670	24.6	2343	-5.78	1328	40.05
P230	3254	**3117**	4.21	3184	2.151	3216	1.168	3117	4.21
P233	2224	2238	-0.63	2290	-2.97	**2088**	6.115	2088	6.115
P234	2078	**1288**	38.02	1768	14.92	1308	37.05	1288	38.02
P235	2349	**1772**	24.56	1846	21.41	1873	20.26	1772	24.56
A.P.D			21.08		16.22		16.86		24.32
N.B.S.			8		9		9		10
N.W.S.			2		1		1		0
B.I.A.			26.85		18.35		19.38		24.32
W.I.A.			-2.03		-2.97		-5.78		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.3, R=1.5, 10 jobs, 15 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P241	2950	2335	20.85	2640	10.51	**2193**	25.66	2193	25.66
P243	3324	2732	17.81	2811	15.43	**2681**	19.34	2681	19.34
P244	3188	**2079**	34.79	2504	21.46	2636	17.31	2079	34.79
P247	2726	**2375**	12.88	2431	10.82	2401	11.92	2375	12.88
P250	2691	2064	23.3	2465	8.398	**1870**	30.51	1870	30.51
P252	3388	**2435**	28.13	2671	21.16	2573	24.06	2435	28.13
P254	3366	2245	33.3	**1848**	45.1	2061	38.77	1848	45.1
P257	3657	3265	10.72	2505	31.5	**2078**	43.18	2078	43.18
P259	3318	**2505**	24.5	2969	10.52	2643	20.34	2505	24.5
P260	3289	**2405**	26.88	2787	15.26	2757	16.18	2405	26.88
A.P.D			23.32		19.02		24.73		29.1
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			23.32		19.02		24.73		29.1
W.I.A.			0		0		0		0

T=0.3, R=1.5, 20 jobs, 15 machines

P262	3402	**2929**	13.9	3148	7.466	3134	7.878	2929	13.9
P264	3055	**2216**	27.46	2677	12.37	2436	20.26	2216	27.46
P265	2999	2782	7.236	2567	14.4	**2343**	21.87	2343	21.87
P267	3010	2878	4.385	3206	-6.51	**2775**	7.807	2775	7.807
P269	2671	**1963**	26.51	2174	18.61	2089	21.79	1963	26.51
P271	3074	**2628**	14.51	2898	5.725	2831	7.905	2628	14.51
P272	3864	**2807**	27.36	3524	8.799	3028	21.64	2807	27.36
P274	3732	2960	20.69	3196	14.36	**2804**	24.87	2804	24.87
P276	3176	**2544**	19.9	2599	18.17	3215	-1.23	2544	19.9
P278	2573	1956	23.98	2460	4.392	**1854**	27.94	1854	27.94
A.P.D			18.59		9.779		16.07		21.21
N.B.S.			10		9		9		10
N.W.S.			0		1		1		0
B.I.A.			18.59		11.59		18		21.21
W.I.A.			0		-6.51		-1.23		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)
T=0.3, R=1.5, 10 jobs, 20 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P281	4302	3193	25.78	3453	19.74	**3153**	26.71	3153	26.71
P283	4377	**3081**	29.61	3361	23.21	3309	24.4	3081	29.61
P285	4117	**2759**	32.99	3195	22.39	2809	31.77	2759	32.99
P287	3613	**2496**	30.92	2505	30.67	2672	26.04	2496	30.92
P290	4091	3033	25.86	**2970**	27.4	2989	26.94	2970	27.4
P291	4277	**3582**	16.25	3853	9.913	3737	12.63	3582	16.25
P293	4231	3352	20.78	3564	15.76	**3304**	21.91	3304	21.91
P295	4079	3019	25.99	3341	18.09	**2889**	29.17	2889	29.17
P297	4209	2878	31.62	3204	23.88	**2812**	33.19	2812	33.19
P299	4535	**3241**	28.53	3789	16.45	3411	24.79	3241	28.53
A.P.D			26.83		20.75		25.75		27.67
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			26.83		20.75		25.75		27.67
W.I.A.			0		0		0		0

T=0.3, R=1.5, 20 jobs, 20 machines

P301	4479	**3245**	27.55	3614	19.31	3325	25.76	3245	27.55
P303	4439	**3285**	26	3475	21.72	3583	19.28	3285	26
P305	4527	**3067**	32.25	3294	27.24	3436	24.1	3067	32.25
P309	4093	**3567**	12.85	3833	6.352	3844	6.084	3567	12.85
P311	4711	**3107**	34.05	3410	27.62	3144	33.26	3107	34.05
P312	4242	3612	14.85	4105	3.23	**3379**	20.34	3379	20.34
P314	4100	3512	14.34	**3431**	16.32	3479	15.15	3431	16.32
P317	4918	**3567**	27.47	4167	15.27	3627	26.25	3567	27.47
P318	4244	**3253**	23.35	4078	3.911	3591	15.39	3253	23.35
P319	4541	**3393**	25.28	4161	8.368	3755	17.31	3393	25.28
A.P.D			23.8		14.93		20.29		24.55
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			23.8		14.93		20.29		24.55
W.I.A.			0		0		0		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)
T=0.3, R=2.5, 10 jobs, 5 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P321	1351	**1035**	23.39	1035	23.39	1035	23.39	1035	23.39
P322	1572	1385	11.9	**1210**	23.03	1210	23.03	1210	23.03
P324	1116	**1000**	10.39	1246	-11.6	1175	-5.29	1000	10.39
P325	545	557	-2.2	**344**	36.88	683	-25.3	344	36.88
P326	1139	**763**	33.01	763	33.01	1023	10.18	763	33.01
P328	1241	**970**	21.84	1118	9.911	970	21.84	970	21.84
P329	1585	1694	-6.88	**1608**	-1.45	1708	-7.76	1608	-1.451
P330	1308	**1086**	16.97	1086	16.97	1086	16.97	1086	16.97
P333	1423	**818**	42.52	1015	28.67	818	42.52	818	42.52
P335	1973	1812	8.16	**1649**	16.42	1812	8.16	1649	16.42
A.P.D			15.91		17.52		10.77		22.3
N.B.S.			8		8		7		9
N.W.S.			2		2		3		1
B.I.A.			21.02		23.53		20.86		24.94
W.I.A.			-4.54		-6.53		-12.8		-1.451

T=0.3, R=2.5, 20 jobs, 5 machines

P341	1975	**1673**	15.29	1844	6.633	1673	15.29	1673	15.29
P343	1141	**712**	37.6	1053	7.713	871	23.66	712	37.6
P344	1656	**1342**	18.96	1645	0.664	1342	18.96	1342	18.96
P346	2311	1939	16.1	**1903**	17.65	1939	16.1	1903	17.65
P348	1498	1215	18.89	**984**	34.31	1215	18.89	984	34.31
P350	1801	**1359**	24.54	1359	24.54	1369	23.99	1359	24.54
P351	2251	**1941**	13.77	1979	12.08	2027	9.951	1941	13.77
P352	2254	2054	8.873	1910	15.26	**1856**	17.66	1856	17.66
P355	980	**570**	41.84	681	30.51	570	41.84	570	41.84
P357	1985	**1470**	25.94	1470	25.94	1470	25.94	1470	25.94
A.P.D			22.18		17.53		21.23		24.76
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			25.99		21.11		24.07		24.76
W.I.A.			0		0		0		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.3, R=2.5, 10 jobs, 10 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P361	1685	1224	27.36	1523	9.614	**1128**	33.06	1128	33.06
P362	1836	**1088**	40.74	1088	40.74	1088	40.74	1088	40.74
P366	2927	**1922**	34.34	2420	17.32	1978	32.42	1922	34.34
P369	2280	**1462**	35.88	1585	30.48	1893	16.97	1462	35.88
P370	2775	**2103**	24.22	2518	9.261	2254	18.77	2103	24.22
P371	2773	**2115**	23.73	2185	21.2	2115	23.73	2115	23.73
P374	2638	1874	28.96	**1769**	32.94	1874	28.96	1769	32.94
P375	2047	**1310**	36	1310	36	1310	36	1310	36
P376	1572	**1378**	12.34	1431	8.969	1401	10.88	1378	12.34
P378	2791	2308	17.31	**2072**	25.76	2441	12.54	2072	25.76
A.P.D			28.09		23.23		25.41		29.9
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			28.09		23.23		25.41		29.9
W.I.A.			0		0		0		0

T=0.3, R=2.5, 20 jobs, 10 machines

P382	3113	**2185**	29.81	2267	27.18	2267	27.18	2185	29.81
P383	2946	**1996**	32.25	2457	16.6	1996	32.25	1996	32.25
P385	3739	3811	-1.93	3756	-0.45	**3592**	3.932	3592	3.932
P388	3248	**2378**	26.79	2446	24.69	2402	26.05	2378	26.79
P390	2711	1824	32.72	2154	-0.43	**1789**	34.01	1789	34.01
P392	3173	**2357**	25.72	3129	1.387	2357	25.72	2357	25.72
P394	4132	3561	13.82	**3190**	22.8	3462	16.21	3190	22.8
P395	3041	2146	29.43	2571	15.46	**1986**	34.69	1986	34.69
P397	2552	**1696**	33.54	1967	22.92	1800	29.47	1696	33.54
P399	2471	**1796**	27.32	1914	22.54	2014	18.49	1796	27.32
A.P.D			24.95		15.27		24.8		27.09
N.B.S.			9		8		10		10
N.W.S.			1		2		0		0
B.I.A.			27.93		19.2		24.8		27.09
W.I.A.			-1.93		-0.44		0		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)
T=0.3, R=2.5, 10 jobs, 15 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P401	4219	**3431**	18.68	4300	-1.92	4031	4.456	3431	18.68
P404	3892	2488	36.07	2937	24.54	**2289**	41.19	2289	41.19
P406	2741	**1567**	42.83	1915	30.13	1667	39.18	1567	42.83
P408	3651	**2980**	18.38	3586	1.78	3126	14.38	2980	18.38
P409	3585	**2233**	37.71	2298	35.9	2242	37.46	2233	37.71
P411	3911	**2821**	27.87	3600	7.952	2955	24.44	2821	27.87
P412	3059	**1527**	50.08	1527	50.08	1527	50.08	1527	50.08
P414	3386	2612	22.86	2726	19.49	**2517**	25.66	2517	25.66
P415	4194	**2449**	41.61	2847	32.12	2708	35.43	2449	41.61
P417	3670	**2204**	39.95	2204	39.95	2204	39.95	2204	39.95
A.P.D			33.6		24		31.22		34.4
N.B.S.			10		9		10		10
N.W.S.			0		1		0		0
B.I.A.			33.6		26.88		31.22		34.4
W.I.A.			0		-1.92		0		0

T=0.3, R=2.5, 20 jobs, 15 machines

P421	4219	4234	-0.36	**2837**	32.76	3965	6.02	2837	32.76
P422	3341	**2797**	16.28	3689	-10.4	2797	16.28	2797	16.28
P424	3892	**2528**	35.05	2740	29.6	2679	31.17	2528	35.05
P425	3883	**2707**	30.29	3208	17.38	2816	27.48	2707	30.29
P426	2741	1954	28.71	2007	26.78	**1718**	37.32	1718	37.32
P427	3120	3148	-0.9	3299	-5.74	**3009**	3.558	3009	3.558
P428	3651	**3086**	15.48	3393	7.067	3353	8.162	3086	15.48
P429	3585	**3265**	8.926	3380	5.718	3284	8.396	3265	8.926
P430	2923	**2692**	7.903	3331	-14	2818	3.592	2692	7.903
P431	3911	**2837**	27.46	2881	26.34	2896	25.95	2837	27.46
A.P.D			16.88		11.55		16.79		21.5
N.B.S.			8		7		10		10
N.W.S.			2		3		0		0
B.I.A.			21.26		20.81		16.79		21.5
W.I.A.			-0.63		10.05		0		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.3, R=2.5, 10 jobs, 20 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P442	4581	**2675**	41.61	3097	32.39	2675	41.61	2675	41.61
P444	4416	2801	36.57	**2691**	39.06	2801	36.57	2691	39.06
P446	4034	**2256**	44.08	2903	28.04	2256	44.08	2256	44.08
P448	4638	**3166**	31.74	3259	29.73	3259	29.73	3166	31.74
P450	4096	**2539**	38.01	2922	28.66	2539	38.01	2539	38.01
P452	4616	**2845**	38.37	3013	34.73	2911	36.94	2845	38.37
P454	4457	3135	29.66	**2975**	33.25	3026	32.11	2975	33.25
P456	4667	**2962**	36.53	3798	18.62	2965	36.47	2962	36.53
P458	4756	**3347**	29.63	3613	24.03	3777	20.58	3347	29.63
P460	5059	**3413**	32.54	4662	7.847	3526	30.3	3413	32.54
A.P.D			35.87		27.64		34.64		36.48
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			35.87		27.64		34.64		36.48
W.I.A.			0		0		0		0

T=0.3, R=2.5, 20 jobs, 20 machines

P461	4668	**2994**	35.86	3220	31.02	3551	23.93	2994	35.86
P463	5963	4869	18.35	**4810**	19.34	4910	17.66	4810	19.34
P465	4072	**2661**	34.65	3088	24.17	2902	28.73	2661	34.65
P467	5465	4078	25.38	4083	25.29	**4004**	26.73	4004	26.73
P469	4975	**3606**	27.52	4326	13.05	3327	33.13	3606	27.52
P471	5267	**4130**	21.59	4648	11.75	4474	15.06	4130	21.59
P473	5359	4294	19.87	4184	21.93	**4081**	23.85	4081	23.85
P475	4692	**3761**	19.84	3993	14.9	3696	21.23	3761	19.84
P477	5135	3724	27.48	**3570**	30.48	4074	20.66	3570	30.48
P479	5347	**3615**	32.39	4489	16.05	3615	32.39	3615	32.39
A.P.D			26.29		20.8		24.34		27.22
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			26.29		20.8		24.34		27.22
W.I.A.			0		0		0		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)
T=0.6, R=1.5, 10 jobs, 5 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P481	1501	1788	-19.1	1791	-19.3	**1638**	-9.13	1638	-9.127
P483	1616	1671	-3.4	1751	-8.35	**1659**	-2.66	1659	-2.661
P485	1505	1717	-14.1	1753	-16.5	**1711**	-13.7	1711	-13.69
P487	1635	1833	-12.1	2036	-24.5	**1833**	-12.1	1833	-12.11
P489	1617	**1628**	-0.68	1771	-9.52	1689	-4.45	1628	-0.68
P491	1861	1854	0.376	1959	-5.27	**1838**	1.236	1838	1.236
P493	1328	1719	-29.4	1941	-46.2	**1578**	-18.8	1578	-18.83
P495	1441	1625	-12.8	**1486**	-3.12	1752	-21.6	1486	-3.123
P497	1443	**1434**	0.624	1434	0.624	1764	-22.2	1434	0.624
P499	1491	**1493**	-0.13	1577	-5.77	1531	-2.68	1493	-0.134
A.P.D.			-9.07		-13.8		-10.6		-5.849
N.B.S.			1		1		1		2
N.W.S			9		9		9		8
B.I.A.			0.624		0.624		1.236		0.93
W.I.A.			-10.2		-15.4		-11.9		-7.544

T=0.6, R=1.5, 20 jobs, 5 machines

P502	2265	2302	-1.63	**2288**	-1.02	2731	-20.6	2288	-1.015
P504	2121	2454	-15.7	2421	-14.1	**2259**	-6.51	2259	-6.506
P506	1966	2271	-15.5	**1872**	4.781	2427	-23.4	1872	4.781
P508	1955	2503	-28	**2144**	-9.67	2303	-17.8	2144	-9.668
P510	2442	2866	-17.4	2580	-5.65	**2521**	-3.24	2521	-3.235
P512	1596	**1775**	-11.2	1819	-14	1819	-14	1775	-11.22
P514	1734	**2144**	-23.6	2521	-45.4	2460	-41.9	2144	-23.64
P516	2246	2439	-8.59	2446	-8.9	**2427**	-8.06	2427	-8.059
P518	1617	**2066**	-27.8	2185	-35.1	2085	-28.9	2066	-27.77
P520	2057	2464	-19.8	**1977**	3.889	2345	-14	1977	3.889
A.P.D			-16.9		-12.5		-17.8		-8.244
N.B.S.			0		2		0		2
N.W.S.			10		8		10		8
B.I.A.			0		4.335		0		4.335
W.I.A.			-16.9		-16.7		-17.8		-11.39

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)
T=0.6, R=1.5, 10 jobs, 10 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P521	2742	2428	11.45	2443	10.9	**2288**	16.56	2288	16.56
P523	2479	**2271**	8.39	2332	5.93	2362	4.72	2271	8.39
P525	2745	**2928**	-6.67	3136	-14.2	3091	-12.6	2928	-6.667
P527	3012	**2509**	16.7	2789	7.404	2734	9.23	2509	16.7
P529	2732	2530	7.394	2467	9.7	**2398**	12.23	2398	12.23
P531	2683	2301	14.24	**2126**	20.76	2549	4.994	2126	20.76
P533	2562	2286	10.77	2899	-13.2	**1967**	23.22	1967	23.22
P535	2062	**1631**	20.9	1873	9.166	1858	9.893	1631	20.9
P537	2419	**2251**	6.945	2404	0.62	2625	-8.52	2251	6.945
P539	2304	**2341**	-1.61	2507	-8.81	2810	-22	2341	-1.606
A.P.D			8.852		2.828		3.776		11.74
N.B.S.			8		7		7		8
N.W.S.			2		3		3		2
B.I.A.			12.1		9.21		11.55		15.73
W.I.A.			-4.14		-13.7		-14.4		-4.137

T=0.6, R=1.5, 20 jobs, 10 machines

P542	3297	**3628**	-10	3794	-15.1	4092	-24.1	3628	-10.04
P544	3307	**3473**	-5.02	4129	-24.9	3828	-15.8	3473	-5.02
P546	3405	**3711**	-8.99	3905	-14.7	3749	-10.1	3711	-8.987
P548	3249	**3513**	-8.13	3595	-10.6	3745	-15.3	3513	-8.126
P550	3377	**2702**	19.99	2797	17.18	2924	13.41	2702	19.99
P552	2919	**3073**	-5.28	3935	-34.8	3567	-22.2	3073	-5.276
P554	3273	3326	-1.62	3317	-1.34	**3159**	3.483	3159	3.483
P556	3366	**3108**	7.665	3507	-4.19	3333	0.98	3108	7.665
P558	3413	**3156**	7.53	3472	-1.73	3402	0.322	3156	7.53
P560	3630	3737	-2.95	**3551**	2.176	3979	-9.61	3551	2.176
A.P.D			-0.68		-8.8		-7.89		0.34
N.B.S.			3		2		4		5
N.W.S.			7		8		6		5
B.I.A.			11.73		9.678		4.549		8.169
W.I.A.			-6		-13.4		-16.2		-7.484

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)
T=0.6, R=1.5, 10 jobs, 15 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P561	3719	3672	1.264	3585	3.603	**3576**	3.845	3576	3.845
P563	3609	**3374**	6.511	3394	5.957	3536	2.023	3374	6.511
P565	3656	**2637**	27.87	3396	7.112	2768	24.29	2637	27.87
P567	3798	**2962**	22.01	3418	10.01	3245	14.56	2962	22.01
P569	3598	**3020**	16.06	3632	-0.94	3313	7.921	3020	16.06
P571	3776	2870	23.99	3109	17.66	**2595**	31.28	2595	31.28
P573	3387	2824	16.62	2964	12.49	**2723**	19.6	2723	19.6
P575	3651	**3243**	11.18	3965	-8.6	3551	2.739	3243	11.18
P577	3375	**2625**	22.22	4147	-22.9	3280	2.815	2625	22.22
P579	3882	**2697**	30.53	3477	10.43	3363	13.37	2697	30.53
A.P.D			17.83		3.484		12.24		19.11
N.B.S.			10		7		10		10
N.W.S.			0		3		0		0
B.I.A.			17.83		8.18		12.24		19.11
W.I.A.			0		-10.8		0		0

T=0.6, R=1.5, 20 jobs, 15 machines

P582	4389	4186	4.625	4336	1.208	**4023**	8.339	4023	8.339
P584	3921	**3852**	1.76	4090	-4.31	4251	-8.42	3852	1.76
P586	4618	**4345**	5.912	4555	1.364	4504	2.469	4345	5.912
P588	4714	**4420**	6.237	4805	-1.93	4531	3.882	4420	6.237
P590	4166	**4359**	-4.63	4731	-13.6	4915	-18	4359	-4.633
P592	4391	4297	2.141	4121	6.149	**4048**	7.811	4048	7.811
P594	4474	4508	-0.76	**4343**	2.928	4467	0.156	4343	2.928
P596	4790	**4389**	8.372	4449	7.119	4801	-0.23	4389	8.372
P598	4187	4030	3.75	**3713**	11.32	3950	5.66	3713	11.32
P600	4418	4368	1.132	4776	-8.1	**3961**	10.34	3961	10.34
A.P.D			2.853		0.218		1.204		5.839
N.B.S.			8		6		7		9
N.W.S.			2		4		3		1
B.I.A.			4.24		5.015		5.522		7.002
W.I.A.			-4.63		-6.99		-8.88		-4.633

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.6, R=1.5, 10 jobs, 20 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P601	4142	3442	16.9	3746	9.561	**3341**	19.34	3341	19.34
P603	4602	**3996**	13.17	4154	9.735	3968	13.78	3996	13.17
P605	4758	**3334**	29.93	4752	0.126	3566	25.05	3334	29.93
P607	4837	3575	26.09	3762	22.22	**3516**	27.31	3516	27.31
P609	4752	**3540**	25.51	4801	-1.03	4101	13.7	3540	25.51
P611	4865	3715	23.64	**3695**	24.05	3907	19.69	3695	24.05
P613	4571	3397	25.68	4075	10.85	**3097**	32.25	3097	32.25
P615	4523	**3803**	15.92	4484	0.862	4669	-3.23	3803	15.92
P617	4903	**3953**	19.38	4403	10.2	4304	12.22	3953	19.38
P619	5047	**3933**	22.07	4296	14.88	4546	9.927	3933	22.07
A.P.D			21.83		10.15		17		22.89
N.B.S.			10		9		9		10
N.W.S.			0		1		1		0
B.I.A.			21.83		11.39		19.25		22.89
W.I.A.			0		-1.03		-3.23		0

T=0.6, R=1.5, 20 jobs, 20 machines

P622	5455	5210	4.491	**5160**	5.408	5377	1.43	5160	5.408
P624	5274	4577	13.22	5402	-2.43	**4538**	13.96	4538	13.96
P626	5818	**5099**	12.36	5941	-2.11	5810	0.138	5099	12.36
P628	5025	**4535**	9.751	5368	-6.83	5217	-3.82	4535	9.751
P630	5125	**4839**	5.58	5597	-9.21	5194	-1.35	4839	5.58
P632	5869	**5152**	12.22	6314	-7.58	5245	10.63	5152	12.22
P634	5643	**5157**	8.612	5877	-4.15	5246	7.035	5157	8.612
P636	5414	**4720**	12.82	5325	1.644	5075	6.262	4720	12.82
P638	5437	4994	8.148	**4962**	8.736	5423	0.257	4962	8.736
P640	5505	**4660**	15.35	5731	-4.11	5156	6.34	4660	15.35
A.P.D			10.25		-2.06		4.088		10.48
N.B.S.			10		3		8		10
N.W.S.			0		7		2		0
B.I.A.			10.25		5.263		5.757		10.48
W.I.A.			0		5.202		-2.59		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.6, R=2.5, 10 jobs, 5 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P641	1763	**1560**	11.51	1613	8.508	1560	11.51	1560	11.51
P643	1824	2075	-13.8	**2045**	-12.1	2075	-13.8	2045	-12.12
P645	1574	**1385**	12.01	1404	10.8	1547	1.715	1385	12.01
P647	1768	1623	8.201	**1548**	12.44	1623	8.201	1548	12.44
P649	1356	1248	7.965	1272	6.195	**1225**	9.661	1225	9.661
P651	1110	**976**	12.07	1043	6.036	976	12.07	976	12.07
P653	1450	**1306**	9.931	1306	9.931	1306	9.931	1306	9.931
P655	1466	1378	6.003	**1152**	21.42	1416	3.411	1152	21.42
P657	1355	1151	15.06	**1116**	17.64	1377	-1.62	1116	17.64
P659	1211	1203	0.661	**1198**	1.073	1220	-0.74	1198	1.073
A.P.D			6.965		8.193		4.038		9.564
N.B.S.			9		9		7		9
N.W.S.			1		1		3		1
B.I.A.			9.268		10.45		8.071		11.97
W.I.A.			-13.8		-12.1		-5.39		-12.12

T=0.6, R=2.5, 20 jobs, 5 machines

P662	1614	1654	-2.48	1622	-0.5	**1187**	26.46	1187	26.46
P664	2081	2397	-15.2	**2288**	-9.95	2415	-16	2288	-9.947
P666	2436	2440	-0.16	2875	-18	**2291**	5.952	2291	5.952
P668	1610	1771	-10	1792	-11.3	**1644**	-2.11	1644	-2.112
P670	2129	2494	-17.1	**2146**	-0.8	2474	-16.2	2146	-0.798
P672	1730	**1756**	-1.5	2173	-25.6	1926	-11.3	1756	-1.503
P674	1745	**1696**	2.808	2240	-28.4	1902	-9	1696	2.808
P676	1836	**1710**	6.863	1737	5.392	1915	-4.3	1710	6.863
P678	1712	1826	-6.66	2067	-20.7	**1760**	-2.8	1760	-2.804
P680	1962	**1620**	17.43	1620	17.43	2346	-19.6	1620	17.43
A.P.D			-2.6		-9.25		-4.9		4.235
N.B.S.			3		2		2		5
N.W.S.			7		8		8		5
B.I.A.			9		11.41		16.21		11.9
W.I.A.			-7.59		-14.4		-10.2		-3.433

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)
T=0.6, R=2.5, 10 jobs, 10 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P682	2985	2104	29.51	2658	10.95	**2053**	31.22	2053	31.22
P684	2782	2464	11.43	**1831**	34.18	2078	25.31	1831	34.18
P686	3006	2676	10.98	2976	0.998	**2500**	16.83	2500	16.83
P688	2924	2550	12.79	**2250**	23.05	2631	10.02	2250	23.05
P690	2403	**2245**	6.575	2632	-9.53	2334	2.871	2245	6.575
P692	2691	**2229**	17.17	2966	-10.2	2551	5.203	2229	17.17
P694	2678	2701	-0.86	**2602**	2.838	2701	-0.86	2602	2.838
P696	2635	**2085**	20.87	2418	8.235	2085	20.87	2085	20.87
P698	2749	2531	7.93	2553	7.13	**2433**	11.5	2433	11.5
P700	3232	3069	5.043	**2926**	9.468	3248	-0.5	2926	9.468
A.P.D			12.14		7.711		12.25		17.37
N.B.S.			10		8		8		10
N.W.S.			0		2		2		0
B.I.A.			12.14		12.11		15.48		17.37
W.I.A.			0		-9.87		-0.68		0

T=0.6, R=2.5, 20 jobs, 10 machines

P701	3491	3420	2.034	3540	-1.4	**3320**	4.898	3320	4.898
P703	3458	**3315**	4.135	3999	-15.6	3549	-2.63	3315	4.135
P705	3521	3272	7.072	**3168**	10.03	3328	5.481	3168	10.03
P707	3403	3601	-5.82	**3109**	8.639	3373	0.882	3109	8.639
P709	3305	**2899**	12.28	3122	-0.43	3829	-15.9	2899	12.28
P710	3467	3543	-2.19	**3046**	12.14	3055	11.88	3046	12.14
P712	3191	**3028**	5.108	3473	-8.84	3218	-0.85	3028	5.108
P714	3729	3406	8.662	4178	-12	**2983**	20.01	2983	20.01
P716	3105	**2459**	20.81	2626	15.43	2639	15.01	2459	20.81
P718	3147	2566	18.46	2655	15.63	**2532**	19.54	2532	19.54
A.P.D			7.055		2.351		5.837		11.76
N.B.S.			8		5		7		10
N.W.S.			2		5		3		0
B.I.A.			9.82		12.27		11.1		11.76
W.I.A.			-4.01		-7.65		-6.46		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.6, R=2.5, 10 jobs, 15 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P721	3596	2737	23.89	2918	18.85	**2699**	24.94	2699	24.94
P723	3558	**2499**	29.76	2973	16.44	2632	26.03	2499	29.76
P724	4219	**3171**	24.84	3554	15.76	3586	15	3171	24.84
P726	3970	**2985**	24.81	3645	8.186	3219	18.92	2985	24.81
P727	3647	**2795**	23.36	3305	9.378	2954	19	2795	23.36
P729	3903	**2909**	25.47	3062	21.55	3144	19.45	2909	25.47
P731	3810	3096	18.74	3367	11.63	**2777**	27.11	2777	27.11
P733	3790	**2654**	29.97	3216	15.15	2748	27.49	2654	29.97
P735	3595	2994	16.72	2835	21.14	**2506**	30.29	2506	30.29
P737	3396	**2652**	21.91	3119	8.157	2710	20.2	2652	21.91
A.P.D.			23.95		14.62		22.84		26.25
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			23.95		14.62		22.84		26.25
W.I.A.			0		0		0		0

T=0.6, R=2.5, 20 jobs, 15 machines

P741	4588	3939	14.15	**3816**	16.83	4039	11.97	3816	16.83
P743	4386	**3584**	18.29	3976	9.348	3616	17.56	3584	18.29
P745	4598	4124	10.31	**3990**	13.22	4078	11.31	3990	13.22
P747	4126	3617	12.34	3800	7.901	**3426**	16.97	3426	16.97
P749	4101	**4304**	-4.95	4616	-12.6	4427	-7.95	4304	-4.95
P751	4016	**3711**	7.595	4110	-2.34	3823	4.806	3711	7.595
P753	4322	3822	11.57	3770	12.77	**3682**	14.81	3682	14.81
P755	4638	**4164**	10.22	4419	4.722	4168	10.13	4164	10.22
P757	4397	3565	18.92	3624	17.58	**3473**	21.01	3473	21.01
P759	5043	4109	18.52	4700	6.802	**4092**	18.86	4092	18.86
A.P.D			11.7		7.428		11.95		13.28
N.B.S.			9		8		9		9
N.W.S.			1		2		1		1
B.I.A.			13.55		11.15		14.16		15.31
W.I.A.			-4.95		-7.47		-7.95		-4.95

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.1 (Cont'd)

T=0.6, R=2.5, 10 jobs, 20 machines

	Uzsoy -1994	B1	Percent Deviation	C	Percent Deviation	CCol	Percent Deviation	Best of all	Percent Deviation
P762	4461	**3274**	26.61	3805	14.71	3694	17.19	3274	26.61
P764	4525	**2974**	34.28	3419	24.44	3089	31.73	2974	34.28
P766	4725	3573	24.38	3896	17.54	**3465**	26.67	3465	26.67
P768	4638	**3339**	28.01	4100	11.6	3773	18.65	3339	28.01
P770	4734	3329	29.68	4295	9.273	**3166**	33.12	3166	33.12
P772	5012	4416	11.89	4472	10.77	**4167**	16.86	4167	16.86
P774	4750	**3416**	28.08	3718	21.73	3800	20	3416	28.08
P776	5096	4139	18.78	4312	15.38	**3944**	22.61	3944	22.61
P778	5154	**3277**	36.42	3782	26.62	3682	28.56	3277	36.42
P780	4692	3440	26.68	4348	7.332	**3330**	29.03	3330	29.03
A.P.D			26.48		15.94		24.44		28.17
N.B.S.			10		10		10		10
N.W.S.			0		0		0		0
B.I.A.			26.48		15.94		24.44		28.17
W.I.A.			0		0		0		0

T=0.6, R=2.5, 20 jobs, 20 machines

P781	5445	5094	6.446	5063	7.016	**5010**	7.989	5010	7.989
P783	5516	**4279**	22.43	4330	21.5	4737	14.12	4279	22.43
P785	5817	5108	12.19	5506	5.346	**4821**	17.12	4821	17.12
P787	5679	**4808**	15.34	5065	10.81	4886	13.96	4808	15.34
P789	5412	4843	10.51	6026	-11.3	**4100**	24.24	4100	24.24
P791	5837	**5225**	10.48	5507	5.654	5398	7.521	5225	10.48
P793	5777	**4678**	19.02	5330	7.738	4845	16.13	4678	19.02
P795	6242	5268	15.6	5568	10.8	**4835**	22.54	4835	22.54
P797	5765	**4781**	17.07	5470	5.117	5124	11.12	4781	17.07
P799	5748	**4484**	21.99	5139	10.59	4686	18.48	4484	21.99
A.P.D			15.11		7.323		15.32		17.82
N.B.S.			10		9		10		10
N.W.S.			0		1		0		0
B.I.A.			15.11		9.397		15.32		17.82
W.I.A.			0		-11.3		0		0

A.P.D	Average Percent Deviation
N.B.S.	Number of Better Solutions
N.W.S.	Number of Worse Solutions
B.I.A.	Better in Average
W.I.A.	Worse in Average

Table C.2 Detailed results of algorithms for each performance criterion

Prob.	B1				C				CCol			
T=0.3, R=1.5, 10 jobs, 5 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P161	2632	248	6	64.9	2870	486	8	189.9	2747	541	5	119.6
P162	2524	743	10	545.5	2572	875	10	571.3	2524	952	10	579.6
P163	2961	635	10	324.6	2869	530	7	192.3	2926	736	9	398.8
P164	2737	399	8	190.8	2741	403	8	199	2737	399	8	190.8
P168	3149	962	10	503.8	3077	890	8	405.7	3149	962	10	465.5
P170	3128	760	10	620.5	3173	760	10	642.2	3371	1067	10	874.7
P173	2924	725	9	412.9	2743	1192	8	466.2	2910	834	9	451.8
P176	2738	971	10	637.2	2742	1006	10	587.1	2654	971	10	581.1
P179	3038	789	10	429.2	3030	842	10	551.4	3038	789	10	479.2
P180	2513	208	7	99.8	2714	408	6	133.5	2415	201	7	94
T=0.3, R=1.5, 20 jobs, 5 machines												
P185	5085	1098	17	531.15	5431	1067	17	475.55	4873	1331	15	571.1
P187	5131	1636	18	832.4	5147	1409	18	739.9	5354	2044	20	1356.85
P188	4905	1078	18	377.25	4870	1128	17	308.45	4898	1078	19	381.55
P189	4741	813	17	158.3	4865	776	13	180.95	4759	1185	12	347.9
P190	4731	358	8	94.8	4872	358	8	78.05	4696	616	9	162.8
P191	5074	1388	17	494	4977	1136	18	522.6	4969	1216	18	503.7
P192	5498	1609	17	710	4872	1501	17	668.55	5027	1767	16	760.65
P194	5268	1375	18	624.9	5184	1148	18	517.65	5117	1081	18	564.2
P195	4801	995	18	521.6	4927	964	18	589.8	4874	995	18	566.2
P198	4918	1192	15	392.3	5252	1246	16	429.55	5043	1317	15	460.95

Table C.2 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=1.5, 10 jobs, 10 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P203	3992	1973	10	1394.7	3894	1704	10	1306.6	3787	1768	10	1257.8
P204	3630	1449	9	890.9	3661	1480	10	858.3	3630	1449	9	890.9
P205	3624	1414	10	885.7	3800	1904	9	999.4	3746	1536	10	971.8
P207	3414	1251	10	924.6	3425	1459	10	794.2	3461	1399	10	1054
P209	3766	2018	10	1559.4	3717	2640	10	1562.5	3693	1945	10	1587.4
P211	4531	2574	10	1849.8	4025	2026	10	1514.4	4348	2435	10	1931.7
P213	3667	1278	10	1104.1	3548	1355	10	933.3	3667	1278	10	1104.1
P215	3664	1732	10	1510.7	3488	1626	10	1241.8	3472	1610	10	1314.3
P218	3438	1580	10	1189.6	3577	1829	10	1105.5	3759	1580	10	1152
P220	4386	2363	10	1755.7	4238	2215	10	1502	3686	2209	10	1603.1
T=0.3, R=1.5, 20 jobs, 10 machines												
P221	6216	1574	20	1164.1	6145	1503	20	938.1	5758	1523	20	1059.5
P223	6136	1396	19	619.65	5953	1213	14	374.05	5776	1250	19	561.4
P224	5960	1463	20	1044.9	6395	1750	20	1233.65	6035	1656	20	1229.2
P226	5930	2114	20	1057.65	5716	1914	19	932.45	5680	1864	19	1151.95
P227	5908	1781	20	961.85	6207	2072	20	1063.4	6017	1980	20	1232.2
P228	6134	1328	20	900.25	6466	1670	20	1084.1	6903	2343	20	1437.65
P230	7063	3117	20	2115.85	6250	3184	20	1934.4	6814	3216	20	2248.6
P233	6232	2238	20	1429.05	6229	2290	20	1416.7	6082	2088	20	1487.75
P234	5958	1288	20	823.65	5970	1768	19	860.7	5528	1308	20	691.75
P235	5787	1772	20	1096.85	6222	1846	20	1034.4	5773	1873	20	1192.3

Table C.2 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=1.5, 10 jobs, 15 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P241	4601	2335	10	1877.8	4239	2640	10	1564.5	4244	2193	10	1831.7
P243	4867	2732	10	2041.7	5014	2811	10	2008.4	4816	2681	10	2096.1
P244	4194	2079	10	1786.4	4469	2504	10	1868.3	4666	2636	10	2060.4
P247	4806	2375	10	2022	4862	2431	10	1711.1	4621	2401	10	1940.1
P250	4272	2064	10	1571.4	4867	2465	10	1896.5	4233	1870	10	1539.9
P252	4611	2435	10	2072.5	4757	2671	10	2028.1	4448	2573	10	2029
P254	4680	2245	10	1856.4	4219	1848	10	1620.7	4306	2061	10	1563.3
P257	5528	3265	10	2580.7	4858	2505	10	1814.1	4338	2078	10	1923.5
P259	4665	2505	10	2011.9	4916	2969	10	2089.4	4731	2643	10	2018.8
P260	4486	2405	10	1984.4	4868	2787	10	2073	4838	2757	10	2091
T=0.3, R=1.5, 20 jobs, 15 machines												
P262	7281	2929	20	2028.65	7208	3148	20	1880.75	7454	3134	20	2087.3
P264	6950	2216	20	1792.45	7280	2677	20	1738.4	7044	2436	20	1858.9
P265	7414	2782	20	1911.6	6907	2567	20	1688.95	7005	2343	20	1695.85
P267	7378	2878	20	2017.75	7139	3206	20	1765.1	7207	2775	20	1798.05
P269	6796	1963	20	1230.65	6885	2174	19	1152.6	6888	2089	19	1318.2
P271	7225	2628	20	1852.65	7619	2898	20	2012.8	6913	2831	20	1784.4
P272	7035	2807	20	2152.95	7392	3524	20	2511.55	7080	3028	20	2402.15
P274	7088	2960	20	2129.75	7225	3196	20	2213.8	6932	2804	20	2174.5
P276	7074	2544	20	1776.5	7016	2599	20	1659.45	7427	3215	20	1857.2
P278	6688	1956	20	1307.8	7307	2460	20	1410.85	6701	1854	20	1235.45

Table C.2 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=1.5, 10 jobs, 20 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P281	5386	3193	10	2843.1	5791	3453	10	2923.4	5491	3153	10	2828.9
P283	5373	3081	10	2638.8	5526	3361	10	2707.8	5688	3309	10	3079.1
P285	4917	2759	10	2403.2	5351	3195	10	2399	5054	2809	10	2313.7
P287	4613	2496	10	2145.3	4676	2505	10	1995.5	4776	2672	10	2255.7
P290	5273	3033	10	2563.9	5317	2970	10	2377.2	5229	2989	10	2480.8
P291	4922	3582	10	3010.5	5391	3853	10	2526.8	5148	3737	10	2704
P293	5528	3352	10	3080.5	5424	3564	10	2739.5	5578	3304	10	3013.8
P295	4702	3019	10	2599.8	5404	3341	10	2615.3	4572	2889	10	2524.7
P297	5080	2878	10	2532.7	5535	3204	10	2506.2	5014	2812	10	2537.6
P299	5371	3241	10	2863.6	5919	3789	10	2893.7	5484	3411	10	2984.2
T=0.3, R=1.5, 20 jobs, 20 machines												
P301	7827	3245	20	2730.55	7970	3614	20	2770.5	7817	3325	20	2637.2
P303	7826	3285	20	2602.65	8109	3475	20	2486.8	8152	3583	20	2677.3
P305	7839	3067	20	2599.2	7882	3294	20	2445.7	8208	3436	20	2626.8
P309	7765	3567	20	3008.9	7835	3833	20	2891.35	8078	3844	20	2987.05
P311	7985	3107	20	2265.25	8288	3410	20	2146.2	8006	3144	20	2295.85
P312	7694	3612	20	2829.75	7992	4105	20	2818.35	7921	3379	20	2800.95
P314	8127	3512	20	2579.75	8153	3431	20	2363.65	8330	3479	20	2583.15
P317	8235	3567	20	3058.05	8114	4167	20	2944.1	7710	3627	20	2937.6
P318	7932	3253	20	2404.4	8757	4078	20	2381.65	8195	3591	20	2924.6
P319	8161	3393	20	2700.6	8656	4161	20	2586.55	8338	3755	20	2929.8

Table C.2 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=2.5, 10 jobs, 5 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P321	3026	1035	8	417.7	3026	1035	8	417.7	3026	1035	8	417.7
P322	2612	1385	9	1018.5	2431	1210	9	866	2431	1210	9	866
P324	2636	1000	8	526.4	2802	1246	9	685.9	2811	1175	8	582.2
P325	2836	557	5	166.7	3088	344	5	78.5	2769	683	5	220.3
P326	2880	763	3	190.2	2836	763	4	199.3	2972	1023	5	279
P328	2704	970	5	419.4	2764	1118	5	442.7	2704	970	5	419.4
P329	2849	1694	9	1104.4	2892	1608	10	1062.2	2842	1708	9	1103.3
P330	2968	1086	9	744.9	2968	1086	9	744.9	2890	1086	9	634.7
P333	3345	818	10	458.2	3220	1015	9	462.9	3198	818	9	421.5
P335	3075	1812	10	1242.9	2679	1649	8	1085	3075	1812	10	1242.9
T=0.3, R=2.5, 20 jobs, 5 machines												
P341	5074	1673	16	1673	4955	1844	16	1012.3	5069	1673	17	1005.4
P343	4833	712	4	104.85	4841	1053	4	123.75	4762	871	10	214.65
P344	4743	1342	17	784.55	4530	1645	16	759.25	4604	1342	16	819.7
P346	5341	1939	14	934.75	4943	1903	13	861.85	5341	1939	14	934.75
P348	5098	1215	6	310.1	4797	984	6	253.9	5098	1215	6	310.1
P350	5107	1359	6	269.8	5185	1359	7	286.55	5018	1369	7	360.45
P351	5325	1941	15	846.7	5029	1979	11	862.65	5325	2027	15	873.15
P352	4341	2054	13	905.9	4501	1910	14	986.45	4926	1856	12	825.35
P355	5008	570	4	91.85	4816	681	5	96.4	5093	570	4	74.6
P357	5048	1470	15	725.7	4889	1470	15	730.95	4883	1470	16	736.5

Table C.2 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=2.5, 10 jobs, 10 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P361	3923	1224	10	919	3972	1523	10	879.6	3271	1128	10	829.9
P362	3616	1088	10	499.5	3744	1088	10	648.2	3558	1088	10	544.2
P366	4053	1922	10	1287.9	4260	2420	10	1499.3	4102	1978	10	1394.5
P369	4020	1462	10	1158.3	4196	1585	10	1085.1	3819	1893	10	1129.7
P370	4010	2103	10	1432.8	4249	2518	10	1600.7	4161	2254	10	1575.5
P371	3566	2115	10	1203.8	3656	2185	10	1151	3566	2115	10	1203.8
P374	3663	1874	10	1460.7	4151	1769	10	1575.3	3663	1874	10	1460.7
P375	3384	1310	9	576.7	3607	1310	8	508.9	3487	1310	9	585.5
P376	4177	1378	10	871.6	4230	1431	10	798.7	4200	1401	10	832.4
P378	3587	2308	10	1561.1	4044	2072	10	1496.7	3629	2441	10	1605.1
T=0.3, R=2.5, 20 jobs, 10 machines												
P382	6030	2185	19	1342.4	6170	2267	18	1326.6	6081	2267	19	1355.4
P383	5357	1996	19	1030.1	6364	2457	20	1200.8	5312	1996	19	1024.2
P385	6037	3811	19	2245.7	6520	3756	20	2139.25	6320	3592	20	2237.75
P388	6320	2378	10	724.9	6470	2446	18	790	6591	2402	17	795.7
P390	6257	1824	17	979.45	5976	2154	15	987.95	6222	1789	16	960.35
P392	6263	2357	20	1460.35	6240	3129	20	1482.5	6263	2357	20	1468.35
P394	6754	3561	20	2589.8	6530	3190	20	2205.4	6986	3462	20	2515.3
P395	6118	2146	17	1123.6	7090	2571	19	1242.95	6254	1986	18	1006.9
P397	6197	1696	19	849.1	6229	1967	17	858.05	6470	1800	19	930.2
P399	5712	1796	13	700.6	6207	1914	14	736.9	5862	2014	14	810.05

Table C.2 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=2.5, 10 jobs, 15 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P401	4473	3431	10	2889.4	5324	4300	10	2881.4	5046	4031	10	3174.4
P404	5383	2488	10	2217.2	5366	2937	10	2214	5184	2289	10	2105.9
P406	4291	1567	10	1192.7	5044	1915	10	1243.9	4766	1667	10	1332.5
P408	4680	2980	10	2498.9	5508	3586	10	2551.1	4792	3126	10	2631.8
P409	4569	2233	10	1964.9	4775	2298	10	1844.1	4970	2242	10	1965.8
P411	4771	2821	10	2521.8	4745	3600	10	2619.9	4608	2955	10	2598.1
P412	4555	1527	10	1217.5	4436	1527	10	1120.5	4384	1527	10	1180.3
P414	4814	2612	10	1984.7	4904	2726	10	2044.9	4991	2517	10	2041.8
P415	4882	2449	10	2143.2	5424	2847	10	2257.3	5141	2708	10	2260.3
P417	4766	2204	10	1867.3	4610	2204	10	1734.3	4610	2204	10	1748.2
T=0.3, R=2.5, 20 jobs, 15 machines												
P421	7053	4234	20	3076.7	6904	2837	20	1858.15	6923	3965	20	3034.8
P422	7134	2797	20	1229.55	7212	3689	18	1180.1	7141	2797	17	1157.95
P424	6585	2528	20	1943.1	6750	2740	20	2028.5	6894	2679	20	2108.45
P425	7622	2707	20	2094.8	7579	3208	20	2149.2	7757	2816	20	2165.05
P426	7809	1954	20	1460.75	6953	2007	19	1295.6	7069	1718	20	1235.45
P427	7436	3148	19	2113.25	7515	3299	20	2075.55	7039	3009	19	1894.4
P428	7006	3086	20	2339.95	7353	3393	20	2328.1	7206	3353	20	2253.55
P429	7544	3265	20	2936.6	6932	3380	20	2659	7530	3284	20	2899.65
P430	7096	2692	20	1852.35	7063	3331	20	1703.35	6823	2818	20	1822.85
P431	6904	2837	20	1858.15	7529	2881	20	1940.45	6563	2896	20	1861.5

Table C.2 (Cont'd)

Prob.	B1				C				CCol			
	T=0.3, R=2.5, 10 jobs, 20 machines											
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P442	5087	2675	10	2396	4841	3097	10	2341	5087	2675	10	2396
P444	5706	2801	10	2306.6	5313	2691	10	2213.8	5706	2801	10	2306.6
P446	4683	2256	10	1932.9	5436	2903	10	2050.5	4683	2256	10	1932.9
P448	5292	3166	10	2265.9	5508	3259	10	2353	5656	3259	10	2486.4
P450	5154	2539	10	2227.6	5811	2922	10	2420.4	5154	2539	10	2227.6
P452	5583	2845	10	2434.9	6045	3013	10	2520	5219	2911	10	2442.6
P454	5577	3135	10	2666.9	5249	2975	10	2428	5468	3026	10	2580.7
P456	5002	2962	10	2627.5	5114	3798	10	2689.1	4670	2965	10	2540.6
P458	5660	3347	10	3017.6	5498	3613	10	2720.8	5621	3777	10	2932.1
P460	5447	3413	10	2837.9	5625	4662	10	3197	5458	3526	10	2915.6
	T=0.3, R=2.5, 20 jobs, 20 machines											
P461	8694	2994	20	2582.95	9167	3220	20	2582.5	8531	3551	20	2740.55
P463	7911	4869	20	2985.4	7646	4810	20	2696.95	7899	4910	20	3056.75
P465	7917	2661	20	2098.9	7870	3088	20	1957.5	7673	2902	20	1961.3
P467	8557	4078	20	3275.5	8359	4083	20	3067.95	8417	4004	20	3088.6
P469	7842	3606	20	2829.2	8050	4326	20	3026.75	8259	3327	20	2795
P471	8467	4130	20	3550.7	9736	4648	20	3739.9	9278	4474	20	3775.55
P473	8204	4294	20	3362.2	8284	4184	20	3120.9	7679	4081	20	3070.3
P475	8411	3761	20	3040.6	8913	3993	20	3070.9	9384	3696	20	3290.25
P477	8157	3724	20	2400.6	8081	3570	20	2332.15	8646	4074	19	2639.75
P479	8090	3615	20	2806	8175	4489	20	3146.45	8566	3615	20	2849.7

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
T=0,6, R=1,5, 10 jobs, 5 machines												
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P481	3043	1788	10	1151.4	3080	1791	10	1007.7	2893	1638	10	999.3
P483	2926	1671	10	1191.9	2978	1751	10	1305	2926	1659	10	1095.8
P485	3048	1717	10	1175.7	3084	1753	10	1072.2	3042	1711	10	1076.2
P487	2802	1833	10	1300.8	2945	2036	10	1305.8	2802	1833	10	1300.8
P489	2905	1628	10	1002	3048	1771	10	985.6	2959	1682	10	1057.9
P491	3014	1854	10	1267.2	3182	1959	10	1129.5	2790	1838	10	1348
P493	3048	1719	10	1227.9	3202	1941	10	1153.5	2907	1578	10	1157
P495	2984	1625	10	1003.2	2806	1486	10	884.4	3111	1752	10	1270
P497	2659	1434	10	927.7	2659	1434	10	914.1	2989	1764	10	1007.2
P499	2784	1493	10	1016.6	2881	1577	10	1110.2	2695	1531	10	1028.6
T=0,6, R=1,5, 20 jobs, 5 machines												
P502	4820	2302	20	1730.35	4356	2288	20	1475.75	4890	2731	20	1701.65
P504	4724	2454	20	1433.4	4682	2421	20	1318.65	4560	2259	20	1431.95
P506	5010	2271	20	1459.95	4611	1872	20	1193.85	5038	2427	20	1596.3
P508	5069	2503	19	1246.65	4788	2144	17	1014.6	4947	2303	19	1220.25
P510	5166	2866	20	1890.7	4908	2580	20	1681.15	4782	2521	20	1740.8
P512	4418	1775	20	1154.5	4553	1819	20	1120.25	4445	1819	20	1237.3
P514	4558	2144	19	1193.05	5097	2521	18	1144.7	5036	2460	18	1521.6
P516	5111	2439	20	1604.85	5088	2446	19	1464.5	5099	2427	20	1531.85
P518	4861	2066	20	1223.8	4980	2185	18	1273.7	4880	2085	20	1243.4
P520	5017	2464	20	1586.9	4621	1977	19	1238.2	4898	2345	20	1506.55

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
T=06, R=1,5, 10 jobs, 10 machines												
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P521	3771	2428	10	1967.6	3786	2443	10	1707.8	3631	2288	10	1872.6
P523	3457	2271	10	1804.6	3561	2332	10	1626.6	3366	2362	10	1803.5
P525	3812	2928	10	2368.4	4020	3136	10	2109.8	4036	3091	10	2565.4
P527	3587	2509	10	2122	3783	2789	10	2168.4	3728	2734	10	2304.9
P529	3803	2530	10	1917.8	3830	2467	10	1816.9	3761	2398	10	1917.3
P531	3241	2301	10	1654.8	3243	2126	10	1586.1	3489	2549	10	1893.9
P533	3645	2286	10	1689.2	4258	2899	10	1573.6	3326	1967	10	1687.5
P535	2958	1631	10	1379.2	3259	1873	10	1250.8	3185	1858	10	1664.1
P537	3347	2251	10	1937.5	3385	2404	10	1817.4	3346	2625	10	1953.1
P539	3599	2341	10	1707.2	3805	2507	10	1797.8	4068	2810	10	2052.7
T=0,6, R=1,5, 20 jobs, 10 machines												
P542	6325	3628	20	2348.55	6491	3794	20	2106.35	6789	4092	20	2544
P544	5999	3473	20	2452.4	6808	4129	20	2305.9	6318	3828	20	2569.55
P546	6503	3711	20	2533.1	6697	3905	20	2258.4	6541	3749	20	2513
P548	6152	3513	20	2483.4	6234	3595	20	2252	6052	3745	20	2840.85
P550	5335	2702	20	29016.5	5525	2797	20	1923.3	5689	2924	20	2052.7
P552	5739	3073	20	2036.75	6703	3935	20	1907.4	6335	3567	20	1998.9
P554	6124	3326	20	2248.65	6080	3317	20	2081.95	5931	3159	20	2202.3
P556	5765	3108	20	2191.85	6159	3507	20	2200.45	5990	3333	20	2277.4
P558	5718	3156	20	2206.1	6040	3472	20	2272.25	5964	3402	20	2527.85
P560	6150	3737	20	2600.1	5964	3551	20	2380.95	6392	3979	20	2679.55

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
	T=0,6, R=1,5, 10 jobs, 15 machines											
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P561	5043	3672	10	2058.7	4906	3585	10	2588	4947	3576	10	3004
P563	4616	3374	10	3019.5	4734	3394	10	2757.5	4876	3536	10	3252.5
P565	3708	2637	10	2252.3	4633	3396	10	2390.1	3926	2768	10	2416.1
P567	4039	2962	10	2443.4	4514	3418	10	2322	4228	3245	10	2461.9
P569	4242	3020	10	2814	4999	3632	10	2764.7	4535	3313	10	3003.4
P571	4138	2870	10	2445.2	4250	3109	10	2289.6	3812	2595	10	2267
P573	4174	2824	10	2396.4	4182	2964	10	2295.4	4073	2723	10	2234.2
P575	4439	3243	10	2914.4	4800	3965	10	2786.3	4748	3551	10	3036.5
P577	3968	2625	10	2295.4	5466	4147	10	2637.5	4624	3280	10	2446.7
P579	3923	2697	10	2397.2	4444	3477	10	2439.7	4478	3363	10	2815.9
	T=0,6, R=1,5, 20 jobs, 15 machines											
P582	6733	4186	20	3345.6	6973	4336	20	3148.3	6610	4023	20	3149.8
P584	6600	3852	20	2788.75	6838	4090	20	2714.35	6970	4251	20	3074.85
P586	7025	4345	20	3288	7066	4555	20	3110.35	7184	4504	20	3191.3
P588	6850	4420	20	3339.25	7019	4805	20	2931.6	6999	4531	20	3401.55
P590	6990	4359	20	3423.55	7354	4731	20	3118.65	7546	4915	20	3475.2
P592	7061	4297	20	3094.65	6788	4121	20	2937.15	6715	4048	20	3141.6
P594	6849	4508	20	3370.15	6684	4343	20	2917.3	6808	4467	20	3128.75
P596	6719	4389	20	3530.25	6794	4449	20	3150.9	7468	4801	20	3735.4
P598	6770	4030	20	2963.5	6453	3713	20	2535.3	6690	3950	20	2849.2
P600	6785	4368	20	3400.2	7475	4776	20	3160.65	6596	3961	20	3249.15

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
T=0,6, R=1,5, 10 jobs, 20 machines												
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P601	4805	3442	10	3074	5127	3746	10	2837.1	4704	3341	10	2979.8
P603	5106	3996	10	3347.8	5255	4154	10	3061.9	4832	3968	10	3472.6
P605	4662	3334	10	2909.2	6080	4752	10	3232.6	4894	3566	10	3090.8
P607	4808	3575	10	3313.5	5077	3762	10	3072.6	4736	3516	10	3309.7
P609	4842	3540	10	3236.3	6093	4801	10	3291.2	5403	4101	10	3627.7
P611	4775	3715	10	3206.5	4725	3695	10	3023.6	4967	3907	10	3304.1
P613	4695	3397	10	3080.8	5443	4075	10	2677.3	4259	3097	10	2766.8
P615	4708	3803	10	3406	5527	4484	10	3064.7	5816	4669	10	3519.8
P617	5042	3953	10	3519.4	5248	4403	10	3111.9	5469	4304	10	3838.8
P619	5237	3933	10	3591.1	5600	4296	10	3352	5850	4546	10	3641.7
T=0,6, R=1,5, 20 jobs, 20 machines												
P622	7575	5210	20	3952.4	7893	5160	20	3650.3	8083	5377	20	4000.5
P624	7173	4577	20	3866.75	7995	5402	20	3787.55	7170	4538	20	3721.8
P626	7641	5099	20	4288.25	8483	5941	20	4133.35	7936	5810	20	4731.05
P628	7051	4535	20	3733.25	7883	5368	20	3486.6	7411	5217	20	3897.1
P630	7516	4839	20	3781.5	7739	5597	20	3664.1	7871	5194	20	3932.25
P632	7713	5152	20	4128.7	8875	6314	20	4020.15	7789	5245	20	3937.05
P634	7888	5157	20	4223	8301	5877	20	3769.45	7977	5246	20	3923.5
P636	7301	4720	20	3773.5	7888	5325	20	3336.45	7578	5075	20	3709.55
P638	7637	4994	20	3953.25	7719	4962	20	3615.75	8213	5423	20	3460.85
P640	7187	4660	20	3964	8295	5731	20	3732.4	7581	5156	20	4124.95

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
	T=0,6, R=2,5, 10 jobs, 5 machines											
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P641	2835	1560	10	1213.1	2780	1613	10	1163.4	2825	1560	10	1213.1
P643	2974	2075	10	1525.3	3007	2045	10	1381.5	2974	2075	10	1542.8
P645	2362	1385	10	988.5	2494	1404	10	993.3	2463	1547	10	1090.8
P647	3282	1623	10	1141.5	2798	1548	10	1037.3	3282	1623	10	1141.5
P649	2695	1248	10	969.8	2851	1272	10	932.6	2511	1225	10	1019.6
P651	2605	976	10	736.7	2729	1043	10	691.5	2605	976	10	729.2
P653	2982	1306	10	1098.4	2982	1306	10	1098.4	2982	1306	10	1098.4
P655	2967	1378	10	1109.1	2630	1152	10	880.5	2692	1416	10	1063.2
P657	2690	1151	10	864.4	2655	1116	10	789.6	2916	1377	10	936.7
P659	2524	1203	10	836.3	2461	1198	10	835.6	2541	1220	10	844.8
	T=0,6, R=2,5, 20 jobs, 5 machines											
P662	5070	1654	20	1035.7	5038	1622	18	963.05	4603	1187	19	825.5
P664	4823	2397	20	1574.8	4922	2288	20	1549.65	4829	2415	20	1520.5
P666	4880	2440	20	1746.7	5101	2875	20	1816.55	4467	2291	20	1601.15
P668	4997	1771	20	1346.25	4904	1792	20	1338.55	4782	1644	20	1312.3
P670	5050	2494	20	1592.55	5255	2146	20	1386.45	5030	2474	20	1569.8
P672	4925	1756	20	1180	5027	2173	20	1369.95	5412	1926	20	1296.1
P674	4948	1696	20	1165.35	5714	2240	20	1067.55	4767	1902	20	1341.1
P676	5022	1710	20	1198.8	4894	1737	20	1256.65	5227	1915	20	1356.65
P678	4795	1826	20	1107.1	5315	2067	20	1224.65	5052	1760	19	1100.5
P680	4763	1620	10	1009.85	5004	1620	20	1078.7	4664	2346	18	1338.5

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
T=0,6, R=2,5, 10 jobs, 10 machines												
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P682	3778	2104	10	1975	4343	2658	10	2092.1	3727	2053	10	1925.8
P684	4051	2464	10	2114.6	3575	1831	10	1693.3	3655	2078	10	1933.9
P686	3983	2676	10	2162.4	4283	2976	10	2180.4	3753	2500	10	2103.2
P688	3930	2550	10	2144.2	3868	2250	10	1955.1	4011	2631	10	2248.5
P690	3947	2245	10	1689.5	4334	2632	10	1814.2	3954	2334	10	1846.8
P692	3911	2229	10	2006.6	4679	2966	10	2115.3	4234	2551	10	2187
P694	4134	2701	10	2061.4	4035	2602	10	1987.7	4134	2701	10	2061.4
P696	3632	2085	10	1727.5	4025	2418	10	2006.7	3632	2085	10	1727.5
P698	3293	2531	10	2073.9	3382	2553	10	1919.9	3195	2433	10	2017.1
P700	4202	3069	10	2363.1	4059	2926	10	2143	4082	3248	10	2600.8
T=0,6, R=2,5, 20 jobs, 10 machines												
P701	6376	3420	20	2513.65	6285	3540	20	2706.95	5924	3320	20	2473.2
P703	6147	3315	20	2683.2	6177	3999	20	2685.95	5645	3549	20	2777.1
P705	6110	3272	20	2835.15	5971	3168	20	2761.75	6168	3328	20	2805.15
P707	6352	3601	20	2374.2	5915	3109	20	2054.75	6124	3373	20	2241.1
P709	5987	2899	20	2427.5	6153	3122	20	2264.8	6758	3829	20	2687.55
P710	6354	3543	20	2762.6	6001	3046	20	2420.3	6010	3055	20	2529.05
P712	5987	3028	20	2522.05	6745	3473	20	2680.55	5986	3218	20	2538.25
P714	6274	3406	20	2629	6470	4178	20	2875.3	5851	2983	20	2471.85
P716	5847	2459	20	2072.25	6052	2626	20	2139.25	5781	2639	20	2006.5
P718	5898	2566	20	1954.15	6164	2655	20	2029.85	5912	2532	20	1956.95

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
	T=0,6, R=2,5, 10 jobs, 15 machines											
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P721	4406	2737	10	2435.8	4453	2918	10	2364	4364	2699	10	2366.9
P723	3987	2499	10	2210.8	4491	2973	10	2282.7	4120	2632	10	2326.1
P724	4605	3171	10	2804.7	4669	3554	10	2984.4	4599	3586	10	3144.6
P726	4401	2985	10	2559.2	4284	3645	10	2493.9	4640	3219	10	2745.5
P727	4313	2795	10	2422.7	4778	3305	10	2279.4	4472	2954	10	2475.9
P729	4213	2909	10	2421.4	4346	3062	10	2370.3	4428	3144	10	2523.8
P731	4513	3096	10	2640.2	4826	3367	10	2599.1	4257	2777	10	2403.6
P733	4317	2654	10	2340.1	4644	3216	10	2243.1	4468	2748	10	2361.8
P735	4680	2994	10	2430	4521	2835	10	2357.2	4192	2506	10	2309
P737	4057	2652	10	2357.6	4483	3119	10	2416.6	4115	2710	10	2516.1
P739	4368	3026	10	2638.5	4552	3113	10	2345	4792	3353	10	2453.6
	T=0,6, R=2,5, 20 jobs, 15 machines											
P741	7355	3939	20	3512.7	7232	3816	20	3055.9	6865	4039	20	3567.5
P743	6744	3584	20	2957.55	7136	3976	20	2744.7	6515	3616	20	2842.75
P745	7257	4124	20	3123.15	7077	3990	20	3025.2	7001	4078	20	3231.25
P747	6766	3617	20	2940.2	7209	3800	20	2907.55	6831	3426	20	2913.1
P749	7800	4304	20	3388.45	7982	4616	20	3311.7	7424	4427	20	3461.6
P751	6768	3711	20	3022.3	7080	4110	20	2878.7	6891	3823	20	2849.2
P753	6940	3822	20	3029.2	7328	3770	20	2843.6	6934	3682	20	2999.45
P755	7283	4164	20	3511.2	7362	4419	20	3375.2	7038	4168	20	3560.2
P757	6998	3565	20	3118.95	7039	3624	20	2940.4	6729	3473	20	2921.65
P759	7320	4109	20	3177.4	7496	4700	20	3248.15	6934	4092	20	3190.6

Table C.2 (Cont'd)

Prob.	B1				C				Ccol			
T=0,6, R=2,5, 10 jobs, 20 machines												
	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.	Makespan	Lmax	#tardy	Av. Tard.
P762	4875	3274	10	2773.4	5407	3805	10	2729.6	5279	3694	10	3121.4
P764	4323	2974	10	2530.1	5050	3419	10	2791.3	4576	3089	10	2681.2
P766	5003	3573	10	3354.1	5297	3896	10	3153.2	4966	3465	10	3141.5
P768	5085	3339	10	3050.4	5628	4100	10	3127.4	5519	3773	10	3241.5
P770	4924	3329	10	3081.6	5392	4295	10	3069.6	4596	3166	10	2943.7
P772	5708	4416	10	3848.6	5702	4472	10	3379.3	5646	4167	10	3723.4
P774	5092	3416	10	3072.5	5319	3718	10	3041.3	5215	3800	10	3187.7
P776	5263	4139	10	3751.7	5491	4312	10	3322.6	5068	3944	10	3628.2
P778	5051	3277	10	2989.9	5193	3782	10	3099	5456	3682	10	3098.7
P780	4757	3440	10	3080	5362	4348	10	3204.9	4803	3330	10	3000.1
T=0,6, R=2,5, 20 jobs, 20 machines												
P781	7956	5094	20	4201.4	8040	5063	20	3889.8	7828	5010	20	4181.15
P783	7613	4279	20	3758.25	7748	4330	20	3534.95	7806	4737	20	3964.6
P785	8179	5108	20	4084.3	8746	5506	20	4114.9	7892	4821	20	3830.2
P787	7885	4808	20	4133.4	7759	5065	20	3956.6	7849	4886	20	3983.35
P789	8261	4843	20	3705.3	9565	6026	20	3642.1	7408	4100	20	3310.9
P791	8109	5225	20	4369.6	8439	5507	20	4223	8219	5398	20	4512.1
P793	8183	4678	20	3928.1	7963	5330	20	4081.55	8350	4845	20	4031.25
P795	8390	5268	20	4305.8	8715	5568	20	4132.85	8165	4835	20	4053.9
P797	8110	4781	20	4198.1	8799	5470	20	4014.6	8245	5124	20	4191.05
P799	7675	4484	20	3800.75	8010	5139	20	3709.15	7666	4686	20	3863.15

Table C.3 Detailed results of algorithms for each performance criterion in no alterntaive and one alternative machine cases

Prob.	B1				C				CCol			
	T=0.3, R=1.5, 10 jobs, 5 machines											
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P161	2632	248	6	64.9	2870	486	8	189.9	2747	541	5	119.6
	2586	202	4	41.1	2865	430	7	160.9	2865	430	7	160.9
P163	2961	635	10	324.6	2869	530	7	192.3	2926	736	9	398.8
	3132	715	8	294.5	3047	664	7	312.2	2985	618	7	279.6
P168	3149	962	10	503.8	3077	890	8	405.7	3149	962	10	465.5
	3384	1224	10	744.4	3177	990	8	398.9	3121	978	9	535.5
P173	2924	725	9	412.9	2743	1192	8	466.2	2910	834	9	451.8
	2860	779	10	485.3	3062	1276	10	578	3062	1276	10	578
P179	3038	789	10	429.2	3030	842	10	551.4	3038	789	10	479.2
	3010	1008	10	581.2	3260	1011	10	593.5	3364	1115	10	647.9
	T=0.3, R=1.5, 20 jobs, 5 machines											
P185	5085	1098	17	531.15	5431	1067	17	475.55	4873	1331	15	571.1
	5919	1414	18	715.75	5160	1077	18	491.4	5173	1084	18	555.85
P188	4905	1078	18	377.25	4870	1128	17	308.45	4898	1078	19	381.55
	5714	935	19	672.9	3260	1011	10	593.5	5737	1526	18	768.9
P190	4731	358	8	94.8	4872	358	8	78.05	4696	616	9	162.8
	5661	889	14	301.4	5741	980	11	180.8	5620	867	9	201.2
P192	5498	1609	17	710	4872	1501	17	668.55	5027	1767	16	760.65
	5552	1654	20	995.5	5039	1608	17	725.25	5359	1585	19	677.55
P195	4801	995	18	521.6	4927	964	18	589.8	4874	995	18	566.2
	5322	1047	18	641.2	5105	1495	18	700.1	5105	1495	18	695.1

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=1.5, 10 jobs, 10 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P203	3992	1973	10	1394.7	3894	1704	10	1306.6	3787	1768	10	1257.8
	3488	1433	10	1151	4118	1896	10	1197.1	4111	1889	10	1157.9
P205	3624	1414	10	885.7	3800	1904	9	999.4	3746	1536	10	971.8
	3588	1350	9	859.1	3425	1459	10	794.2	3494	1511	10	884.2
P209	3766	2018	10	1559.4	3717	2640	10	1562.5	3693	1945	10	1587.4
	3905	2037	10	1620.5	3536	2167	10	1451.6	4189	2321	10	1512.1
P213	3667	1278	10	1104.1	3548	1355	10	933.3	3667	1278	10	1104.1
	3291	876	10	790	3728	1439	10	950.3	3686	1314	10	914.2
P218	3438	1580	10	1189.6	3577	1829	10	1105.5	3759	1580	10	1152
	3712	1532	10	1223.9	3824	1605	10	1058.9	4036	1817	10	1161.7
T=0.3, R=1.5, 20 jobs, 10 machines												
P221	6216	1574	20	1164.1	6145	1503	20	938.1	5758	1523	20	1059.5
	6490	2151	20	1314.6	6248	1786	20	1101.1	6717	2075	20	1383.25
P224	5960	1463	20	1044.9	6395	1750	20	1233.65	6035	1656	20	1229.2
	6404	2061	20	1219.5	6380	1942	19	1200.95	6821	2239	19	1299.65
P227	5908	1781	20	961.85	6207	2072	20	1063.4	6017	1980	20	1232.2
	6710	1860	20	1131.45	6051	2065	20	1051.6	6556	2222	19	1114.7
P230	7063	3117	20	2115.85	6250	3184	20	1934.4	6814	3216	20	2248.6
	7356	3188	20	2096.5	6838	3148	20	1879.65	6853	3703	20	1967.35
P234	5958	1288	20	823.65	5970	1768	19	860.7	5528	1308	20	691.75
	5898	1076	20	726.7	5824	1816	19	813.45	6552	1710	19	809.65

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=1.5, 10 jobs, 15 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P241	4601	2335	10	1877.8	4239	2640	10	1564.5	4244	2193	10	1831.7
	4164	1890	10	1624	5014	2811	10	2008.4	4193	2124	10	1473.4
P244	4194	2079	10	1786.4	4469	2504	10	1868.3	4666	2636	10	2060.4
	4149	2216	10	1867.9	5086	2971	10	1925.1	5211	2841	10	1928.2
P250	4272	2064	10	1571.4	4867	2465	10	1896.5	4233	1870	10	1539.9
	4568	2360	10	1694.3	4419	2017	10	1561	4337	2129	10	1551
P254	4680	2245	10	1856.4	4219	1848	10	1620.7	4306	2061	10	1563.3
	4144	1709	10	1566.6	4324	1889	10	1390.2	4506	2071	10	1500.2
P259	4665	2505	10	2011.9	4916	2969	10	2089.4	4731	2643	10	2018.8
	4252	2162	10	1950	5019	2859	10	1888.9	5254	3094	10	1937.5
T=0.3, R=1.5, 20 jobs, 15 machines												
P262	7281	2929	20	2028.65	7208	3148	20	1880.75	7454	3134	20	2087.3
	7214	2625	20	1807.6	7801	3056	20	1789.05	7158	2648	20	1662.6
P265	7414	2782	20	1911.6	6907	2567	20	1688.95	7005	2343	20	1695.85
	7056	2247	20	1537.55	6739	2327	20	1553.7	7190	2528	20	1539.75
P269	6796	1963	20	1230.65	6885	2174	19	1152.6	6888	2089	19	1318.2
	6480	1647	20	1014.25	7213	2398	18	1008.65	7728	2881	19	1135.6
P272	7035	2807	20	2152.95	7392	3524	20	2511.55	7080	3028	20	2402.15
	7113	2917	20	2243.35	7307	3206	20	2171.95	7022	3098	20	2159.05
P276	7074	2544	20	1776.5	7016	2599	20	1659.45	7427	3215	20	1857.2
	7139	2802	20	2019.75	7281	2867	20	1697.3	7070	3077	20	1787.3

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
	T=0.3, R=1.5, 10 jobs, 20 machines											
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P281	5386	3193	10	2843.1	5791	3453	10	2923.4	5491	3153	10	2828.9
	5499	3161	10	2773.8	5415	3196	10	2561.6	5616	3278	10	2594.3
P285	4917	2759	10	2403.2	5351	3195	10	2399	5054	2809	10	2313.7
	4670	2425	10	2180.8	5308	3063	10	2153.1	5308	3063	10	2153.1
P290	5273	3033	10	2563.9	5317	2970	10	2377.2	5229	2989	10	2480.8
	4568	2369	10	2178.2	4971	2770	10	2189.8	4741	2650	10	2136.7
P293	5528	3352	10	3080.5	5424	3564	10	2739.5	5578	3304	10	3013.8
	5419	3041	10	2559.9	5131	3202	10	2357.5	5073	3101	10	2349.4
P297	5080	2878	10	2532.7	5535	3204	10	2506.2	5014	2812	10	2537.6
	4705	2374	10	2206.4	5198	3045	10	2391.7	4969	2740	10	2161.1
	T=0.3, R=1.5, 20 jobs, 20 machines											
P301	7827	3245	20	2730.55	7970	3614	20	2770.5	7817	3325	20	2637.2
	7363	2758	20	2258.3	7899	3229	20	2308.6	7933	3343	20	2281
P305	7839	3067	20	2599.2	7882	3294	20	2445.7	8208	3436	20	2626.8
	7763	2991	20	2270.6	8724	3952	20	2360.75	8544	3772	20	2192.75
P311	7985	3107	20	2265.25	8288	3410	20	2146.2	8006	3144	20	2295.85
	7850	3027	20	2197.95	7365	2871	20	1770.15	7441	3107	20	1818.95
P314	8127	3512	20	2579.75	8153	3431	20	2363.65	8330	3479	20	2583.15
	8323	3691	20	2425.5	7585	2953	20	1906.5	8534	3683	20	2119.15
P318	7932	3253	20	2404.4	8757	4078	20	2381.65	8195	3591	20	2924.6
	7708	3139	20	2326.3	8143	3523	20	1959.8	8108	3673	20	1968.2

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=2.5, 10 jobs, 5 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P321	3026	1035	8	417.7	3026	1035	8	417.7	3026	1035	8	417.7
	3300	907	10	549.5	3279	1035	10	508.4	3279	1035	10	508.4
P324	2636	1000	8	526.4	2802	1246	9	685.9	2811	1175	8	582.2
	2868	989	8	579.9	2919	1283	9	535.6	2769	683	5	220.3
P326	2880	763	3	190.2	2836	763	4	199.3	2972	1023	5	279
	2984	830	5	239.7	2888	763	4	191.6	2814	763	4	200.8
P329	2849	1694	9	1104.4	2892	1608	10	1062.2	2842	1708	9	1103.3
	3207	1860	10	1196.1	3239	2010	10	1263.2	3240	2036	10	1384
P333	3345	818	10	458.2	3220	1015	9	462.9	3198	818	9	421.5
	3313	818	10	421.5	3067	1022	7	322.9	3067	1022	7	322.9
T=0.3, R=2.5, 20 jobs, 5 machines												
P341	5074	1673	16	1673	4955	1844	16	1012.3	5069	1673	17	1005.4
	5491	2004	17	1215.3	5217	1743	17	1022.7	4838	1948	17	1079.35
P344	4743	1342	17	784.55	4530	1645	16	759.25	4604	1342	16	819.7
	5019	1470	18	986.9	4943	1903	13	861.85	5035	1391	18	870.85
P348	5098	1215	6	310.1	4797	984	6	253.9	5098	1215	6	310.1
	5411	1146	6	286.1	5271	1080	7	272.8	5556	1080	8	272.45
P351	5325	1941	15	846.7	5029	1979	11	862.65	5325	2027	15	873.15
	5646	2261	19	1122.5	5590	2110	19	962.8	5326	2220	18	976.85
P355	5008	570	4	91.85	4816	681	5	96.4	5093	570	4	74.6
	5444	570	5	79.8	5666	694	4	80.15	5600	808	4	78.3

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=2.5, 10 jobs, 10 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P361	3923	1224	10	919	3972	1523	10	879.6	3271	1128	10	829.9
	3793	1257	10	902	3876	1427	10	701.5	3732	1283	10	687
P366	4053	1922	10	1287.9	4260	2420	10	1499.3	4102	1978	10	1394.5
	4024	1891	10	1350.2	4411	2155	10	1444.9	4411	2155	10	1444.9
P370	4010	2103	10	1432.8	4249	2518	10	1600.7	4161	2254	10	1575.5
	3744	1904	10	1405.7	4088	2699	10	1557.5	4088	2699	10	1557.5
P374	3663	1874	10	1460.7	4151	1769	10	1575.3	3663	1874	10	1460.7
	3660	1703	10	1440	3709	1742	10	1387.3	3709	1742	10	1400.8
P376	4177	1378	10	871.6	4230	1431	10	798.7	4200	1401	10	832.4
	3732	933	10	668.3	4242	1490	9	733.2	4311	1559	9	782.2
T=0.3, R=2.5, 20 jobs, 10 machines												
P382	6030	2185	19	1342.4	6170	2267	18	1326.6	6081	2267	19	1355.4
	6139	2271	19	1492.1	6089	2354	19	1368.75	6304	2200	20	1381
P385	6037	3811	19	2245.7	6520	3756	20	2139.25	6320	3592	20	2237.75
	6691	3391	20	2240.75	6759	3743	19	2150.15	6679	3776	19	2268.45
P390	6257	1824	17	979.45	5976	2154	15	987.95	6222	1789	16	960.35
	6534	1731	20	1101.15	6380	1893	16	943.75	6387	1906	19	995.65
P394	6754	3561	20	2589.8	6530	3190	20	2205.4	6986	3462	20	2515.3
	6900	3325	19	2378.95	6973	3670	20	2338.5	6830	3370	20	2362.7
P397	6197	1696	19	849.1	6229	1967	17	858.05	6470	1800	19	930.2
	6982	1523	20	1120.05	6526	1816	20	1137.45	6573	1777	18	971.15

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
	T=0.3, R=2.5, 10 jobs, 15 machines											
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P401	4473	3431	10	2889.4	5324	4300	10	2881.4	5046	4031	10	3174.4
	4751	2884	10	2523.6	4188	3315	10	2308.5	4812	3275	10	2379.9
P406	4291	1567	10	1192.7	5044	1915	10	1243.9	4766	1667	10	1332.5
	4423	1294	10	991.8	4710	1802	10	1171.8	5001	1943	10	1264.5
P409	4569	2233	10	1964.9	4775	2298	10	1844.1	4970	2242	10	1965.8
	4189	1735	10	1411.9	4759	2281	10	1682.5	4666	2281	10	1692.6
P412	4555	1527	10	1217.5	4436	1527	10	1120.5	4384	1527	10	1180.3
	4156	1331	10	1040.7	4271	1616	10	1089	4609	1854	10	1236.6
P415	4882	2449	10	2143.2	5424	2847	10	2257.3	5141	2708	10	2260.3
	5001	2473	10	2189	5078	2484	10	2136.5	4485	2645	10	2158
	T=0.3, R=2.5, 20 jobs, 15 machines											
P421	7053	4234	20	3076.7	6904	2837	20	1858.15	6923	3965	20	3034.8
	6980	3753	20	2876.95	7212	4116	20	2736.7	7249	4290	20	2829.55
P424	6585	2528	20	1943.1	6750	2740	20	2028.5	6894	2679	20	2108.45
	6831	3586	19	2281.85	6800	2526	20	1884.55	6875	2537	20	1940.95
P426	7809	1954	20	1460.75	6953	2007	19	1295.6	7069	1718	20	1235.45
	7856	1781	20	1477.05	7582	2171	20	1329.35	7562	1912	20	1133.35
P428	7006	3086	20	2339.95	7353	3393	20	2328.1	7206	3353	20	2253.55
	7722	2825	20	2434.4	7490	2989	20	2197	7491	3047	20	2276.7
P430	7096	2692	20	1852.35	7063	3331	20	1703.35	6823	2818	20	1822.85
	7239	2733	20	1917	7450	2691	20	1849	7741	2672	20	1992.95

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0.3, R=2.5, 10 jobs, 20 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P442	5087	2675	10	2396	4841	3097	10	2341	5087	2675	10	2396
	4835	2532	10	2246.7	5170	2623	10	2216.1	5102	3192	10	2195.5
P446	4683	2256	10	1932.9	5436	2903	10	2050.5	4683	2256	10	1932.9
	4593	1917	10	1707.4	5238	2617	10	1771.3	5349	2898	10	1797.5
P450	5154	2539	10	2227.6	5811	2922	10	2420.4	5154	2539	10	2227.6
	5195	2218	10	1988.1	5461	2733	10	2101.2	5400	2702	10	2044.9
P454	5577	3135	10	2666.9	5249	2975	10	2428	5468	3026	10	2580.7
	5045	2291	10	2031.8	4947	2701	10	2050.4	5172	2701	10	2106
P458	5660	3347	10	3017.6	5498	3613	10	2720.8	5621	3777	10	2932.1
	5224	3089	10	2558.1	5505	3916	10	2767.3	5953	3916	10	2812.1
T=0.3, R=2.5, 20 jobs, 20 machines												
P461	8694	2994	20	2582.95	9167	3220	20	2582.5	8531	3551	20	2740.55
	8465	2970	20	2375.45	8689	3087	20	2327.05	7899	4910	20	3056.75
P465	7917	2661	20	2098.9	7870	3088	20	1957.5	7673	2902	20	1961.3
	7449	2057	20	1661.15	7339	2634	20	1502.15	8417	4004	20	3088.6
P469	7842	3606	20	2829.2	8050	4326	20	3026.75	8259	3327	20	2795
	7799	3279	20	2732.85	7974	3455	20	2586.9	9278	4474	20	3775.55
P473	8204	4294	20	3362.2	8284	4184	20	3120.9	7679	4081	20	3070.3
	8396	3633	20	3200.2	8913	3993	20	3070.9	8134	3930	20	3001.5
P477	8157	3724	20	2400.6	8081	3570	20	2332.15	8646	4074	19	2639.75
	8071	3101	20	2393.9	8175	4489	20	3146.45	8791	3563	20	2362.3

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0,6, R=1,5, 10 jobs, 5 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P481	3043	1788	10	1151.4	3080	1791	10	1007.7	2893	1638	10	999.3
	2919	1664	10	1089.2	3433	2144	10	1046	3459	2170	10	1131.4
P485	3048	1717	10	1175.7	3084	1753	10	1072.2	3042	1711	10	1076.2
	3177	1846	10	1206	2877	1711	10	934.8	2878	1712	10	943.7
P489	2905	1628	10	1002	3048	1771	10	985.6	2959	1682	10	1057.9
	3122	1845	10	1212.7	3215	1938	9	1116.9	3185	1908	9	1131.3
P493	3048	1719	10	1227.9	3202	1941	10	1153.5	2907	1578	10	1157
	3253	1991	10	1329.4	3459	2188	10	1275	3669	2340	10	1285.5
P497	2659	1434	10	927.7	2659	1434	10	914.1	2989	1764	10	1007.2
	3037	1646	10	1131.7	3108	1717	10	1082.5	2998	1607	10	1055.2
T=0,6, R=1,5, 20 jobs, 5 machines												
P502	4820	2302	20	1730.35	4356	2288	20	1475.75	4890	2731	20	1701.65
	5329	2685	20	1952.95	5567	2923	20	1783.45	5712	3068	20	1911.95
P506	5010	2271	20	1459.95	4611	1872	20	1193.85	5038	2427	20	1596.3
	5237	2498	20	1602.7	5141	2402	20	1430.2	5668	2929	20	1582.6
P510	5166	2866	20	1890.7	4908	2580	20	1681.15	4782	2521	20	1740.8
	5821	3445	20	2165.85	5238	2943	20	1869.2	5340	3079	20	1982.2
P514	4558	2144	19	1193.05	5097	2521	18	1144.7	5036	2460	18	1521.6
	5089	2513	18	1382.05	5207	2631	18	1210.15	5180	2600	17	1369.35
P518	4861	2066	20	1223.8	4980	2185	18	1273.7	4880	2085	20	1243.4
	4745	1950	19	1127.1	5160	2365	19	1221.8	5208	2413	19	1291.8

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
	T=06, R=1,5, 10 jobs, 10 machines											
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P521	3771	2428	10	1967.6	3786	2443	10	1707.8	3631	2288	10	1872.6
	3556	2213	10	1686.9	3843	2706	10	1718.6	3843	2706	10	1718.6
P525	3812	2928	10	2368.4	4020	3136	10	2109.8	4036	3091	10	2565.4
	3865	2837	10	2376.9	3736	2790	10	1923.1	3669	2903	10	1899.4
P529	3803	2530	10	1917.8	3830	2467	10	1816.9	3761	2398	10	1917.3
	3612	2285	10	1830.2	4193	2822	10	1768.3	4193	2822	10	1767.2
P533	3645	2286	10	1689.2	4258	2899	10	1573.6	3326	1967	10	1687.5
	3344	1985	10	1538.7	3753	2394	10	1518.9	3573	2214	10	1393.7
P537	3347	2251	10	1937.5	3385	2404	10	1817.4	3346	2625	10	1953.1
	3247	2207	10	1898.9	3527	2410	10	1706.4	3969	2852	10	1937.3
	T=0,6, R=1,5, 20 jobs, 10 machines											
P542	6325	3628	20	2348.55	6491	3794	20	2106.35	6789	4092	20	2544
	6037	3436	20	2295.85	6393	4103	20	2166.45	7012	4747	20	2303.8
P546	6503	3711	20	2533.1	6697	3905	20	2258.4	6541	3749	20	2513
	6256	3482	20	2482.1	6504	3712	20	2246.9	6827	4035	20	2230.3
P550	5335	2702	20	29016.5	5525	2797	20	1923.3	5689	2924	20	2052.7
	5718	2980	20	2213.75	6399	3634	20	2069.95	6455	3690	20	2054.05
P554	6124	3326	20	2248.65	6080	3317	20	2081.95	5931	3159	20	2202.3
	5928	3300	20	2298.6	6791	3993	20	2120.85	6164	3469	20	2253.8
P558	5718	3156	20	2206.1	6040	3472	20	2272.25	5964	3402	20	2527.85
	5902	3487	20	2348.35	6053	3624	20	2200.1	6562	3994	20	2172.55

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
	T=0,6, R=1,5, 10 jobs, 15 machines											
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P561	5043	3672	10	2058.7	4906	3585	10	2588	4947	3576	10	3004
	5902	3487	10	2348.35	4038	3091	10	2180.1	4005	2678	10	2149.6
P565	3708	2637	10	2252.3	4633	3396	10	2390.1	3926	2768	10	2416.1
	4160	2942	10	2390.5	4132	3081	10	2204.6	4135	2960	10	2119.7
P569	4242	3020	10	2814	4999	3632	10	2764.7	4535	3313	10	3003.4
	4116	2749	10	2550.3	4989	3622	10	2671.8	5112	3745	10	2705.8
P573	4174	2824	10	2396.4	4182	2964	10	2295.4	4073	2723	10	2234.2
	3808	2658	10	2295.1	4457	3239	10	2071	4934	3584	10	2143.2
P577	3968	2625	10	2295.4	5466	4147	10	2637.5	4624	3280	10	2446.7
	3982	2634	10	2296.6	5111	3718	10	2443.2	4589	3241	10	2381.7
	T=0,6, R=1,5, 20 jobs, 15 machines											
P582	6733	4186	20	3345.6	6973	4336	20	3148.3	6610	4023	20	3149.8
	6994	4357	20	3299.75	7404	4792	20	3120.65	7069	4457	20	3214.75
P586	7025	4345	20	3288	7066	4555	20	3110.35	7184	4504	20	3191.3
	6667	3987	20	3184.65	7624	4929	20	3114.35	7454	4759	20	2982.4
P590	6990	4359	20	3423.55	7354	4731	20	3118.65	7546	4915	20	3475.2
	6996	4373	20	3494.05	7311	4688	20	3145	7722	5099	20	3136.5
P594	6849	4508	20	3370.15	6684	4343	20	2917.3	6808	4467	20	3128.75
	6385	4044	20	3013.7	7284	4937	20	2809.15	7819	5472	20	2872.7
P598	6770	4030	20	2963.5	6453	3713	20	2535.3	6690	3950	20	2849.2
	6372	3685	20	2780.4	7357	4617	20	2645.6	7427	4687	20	2818.55

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0,6, R=1,5, 10 jobs, 20 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P601	4805	3442	10	3074	5127	3746	10	2837.1	4704	3341	10	2979.8
	4109	2839	10	2576.7	5540	4159	10	2463.2	5375	3994	10	2669
P605	4662	3334	10	2909.2	6080	4752	10	3232.6	4894	3566	10	3090.8
	4238	2910	10	2652.5	5540	4212	10	2783.8	5540	4212	10	2796.2
P609	4842	3540	10	3236.3	6093	4801	10	3291.2	5403	4101	10	3627.7
	4651	3480	10	3200	5571	4269	10	3042.3	5896	4594	10	3128.7
P613	4695	3397	10	3080.8	5443	4075	10	2677.3	4259	3097	10	2766.8
	4179	2811	10	2612	5052	3684	10	2726.6	4905	3855	10	2681.7
P617	5042	3953	10	3519.4	5248	4403	10	3111.9	5469	4304	10	3838.8
	4872	3515	10	3284.9	5343	4413	10	3198.8	5343	4413	10	3198.8
T=0,6, R=1,5, 20 jobs, 20 machines												
P622	7575	5210	20	3952.4	7893	5160	20	3650.3	8083	5377	20	4000.5
	7555	5089	20	4123.9	8088	5355	20	3467.2	7762	5261	20	3433.55
P626	7641	5099	20	4288.25	8483	5941	20	4133.35	7936	5810	20	4731.05
	7171	4854	20	3996.55	8721	6313	20	3997.95	8794	6386	20	3979.05
P630	7516	4839	20	3781.5	7739	5597	20	3664.1	7871	5194	20	3932.25
	7237	4577	20	3625.85	7768	5091	20	3256.7	7750	5073	20	3254.65
P634	7888	5157	20	4223	8301	5877	20	3769.45	7977	5246	20	3923.5
	7394	4735	20	3963.45	8047	5523	20	3584	7947	5581	20	3596.2
P638	7637	4994	20	3953.25	7719	4962	20	3615.75	8213	5423	20	3460.85
	6977	4187	20	3394.1	8237	5447	20	3409.25	8019	5229	20	3273.55

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0,6, R=2,5, 10 jobs, 5 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P641	2835	1560	10	1213.1	2780	1613	10	1163.4	2825	1560	10	1213.1
	3026	1328	10	1142.8	3033	1613	10	1216.8	3218	1613	10	1287.3
P645	2362	1385	10	988.5	2494	1404	10	993.3	2463	1547	10	1090.8
	2881	1677	10	1229.4	2568	1364	10	994.5	2559	1355	10	962.1
P649	2695	1248	10	969.8	2851	1272	10	932.6	2511	1225	10	1019.6
	2694	1157	10	917.9	2666	1479	10	887.7	2651	1349	10	847.5
P653	2982	1306	10	1098.4	2982	1306	10	1098.4	2982	1306	10	1098.4
	2958	1204	10	885.4	2915	1161	10	881.9	2958	1204	10	905.2
P657	2690	1151	10	864.4	2655	1116	10	789.6	2916	1377	10	936.7
	2973	1434	10	987.9	3089	1550	10	994.3	2997	1458	10	972.5
T=0,6, R=2,5, 20 jobs, 5 machines												
P662	5070	1654	20	1035.7	5038	1622	18	963.05	4603	1187	19	825.5
	4786	1370	20	962.15	5514	2098	19	1130.45	5394	1978	19	1028.2
P666	4880	2440	20	1746.7	5101	2875	20	1816.55	4467	2291	20	1601.15
	5275	2838	20	2024.45	5349	2866	20	2014.35	5618	3109	20	2066.45
P670	5050	2494	20	1592.55	5255	2146	20	1386.45	5030	2474	20	1569.8
	5704	2782	20	1867.65	5755	2681	20	1821.5	5763	2748	20	1885.3
P674	4948	1696	20	1165.35	5714	2240	20	1067.55	4767	1902	20	1341.1
	5275	1801	20	1278.75	5585	2174	20	1334	5663	2189	20	1259.2
P678	4795	1826	20	1107.1	5315	2067	20	1224.65	5052	1760	19	1100.5
	5606	2376	20	1374.15	5604	2740	20	1368.8	5511	2721	20	1408.25

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0,6, R=2,5, 10 jobs, 10 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P682	3778	2104	10	1975	4343	2658	10	2092.1	3727	2053	10	1925.8
	3955	2270	10	1905.9	3751	2111	10	1798.8	3751	2111	10	1798.8
P686	3983	2676	10	2162.4	4283	2976	10	2180.4	3753	2500	10	2103.2
	3530	2579	10	2006.8	3744	2651	10	1817.6	3744	2651	10	1817.6
P690	3947	2245	10	1689.5	4334	2632	10	1814.2	3954	2334	10	1846.8
	3637	1935	10	1470.3	3619	1917	10	1369	3604	1902	10	1346.1
P694	4134	2701	10	2061.4	4035	2602	10	1987.7	4134	2701	10	2061.4
	3574	2083	10	1711.3	3968	2690	10	1859.1	3765	2487	10	1774.6
P698	3293	2531	10	2073.9	3382	2553	10	1919.9	3195	2433	10	2017.1
	3613	2417	10	2045.9	3496	2734	10	1782.6	3496	2734	10	1782.6
T=0,6, R=2,5, 20 jobs, 10 machines												
P701	6376	3420	20	2513.65	6285	3540	20	2706.95	5924	3320	20	2473.2
	6576	3171	20	2615.4	6854	3654	20	2590.15	7199	3999	20	2722.15
P705	6110	3272	20	2835.15	5971	3168	20	2761.75	6168	3328	20	2805.15
	6634	3794	20	2840.45	6035	3369	20	2591	6323	3483	20	2615.25
P709	5987	2899	20	2427.5	6153	3122	20	2264.8	6758	3829	20	2687.55
	6251	2935	20	2214.75	6652	3541	20	2408.7	6771	3560	20	2494.2
P712	5987	3028	20	2522.05	6745	3473	20	2680.55	5986	3218	20	2538.25
	6727	3276	20	2614.3	6404	3792	20	2537.9	5851	2983	20	2471.85
P716	5847	2459	20	2072.25	6052	2626	20	2139.25	5781	2639	20	2006.5
	6433	2841	20	2124.9	6693	3205	20	2126.65	6315	2953	20	2009.3

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
T=0,6, R=2,5, 10 jobs, 15 machines												
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P721	4406	2737	10	2435.8	4453	2918	10	2364	4364	2699	10	2366.9
	4569	2900	10	2283.4	4309	2731	10	2159	4667	3002	10	2254
P724	4605	3171	10	2804.7	4669	3554	10	2984.4	4599	3586	10	3144.6
	4301	3014	10	2727.4	4516	3539	10	2751.3	4664	3244	10	2691.5
P727	4313	2795	10	2422.7	4778	3305	10	2279.4	4472	2954	10	2475.9
	3956	2554	10	2118.4	4805	3332	10	2031.7	4444	2926	10	1979.3
P731	4513	3096	10	2640.2	4826	3367	10	2599.1	4257	2777	10	2403.6
	4143	2673	10	2417.9	4300	3179	10	2306.2	4693	3179	10	2367.8
P735	4680	2994	10	2430	4521	2835	10	2357.2	4192	2506	10	2309
	4397	2703	10	2424.7	4376	2677	10	2171.3	4581	2882	10	2232.2
T=0,6, R=2,5, 20 jobs, 15 machines												
P741	7355	3939	20	3512.7	7232	3816	20	3055.9	6865	4039	20	3567.5
	7340	3891	20	3375.9	7516	4351	20	3266.85	7471	4390	20	3175
P745	7257	4124	20	3123.15	7077	3990	20	3025.2	7001	4078	20	3231.25
	6859	3911	20	2921.5	7092	3845	20	2771.75	7191	4302	20	2890.3
P749	7800	4304	20	3388.45	7982	4616	20	3311.7	7424	4427	20	3461.6
	7806	4130	20	3318.25	7859	4998	20	3063.3	8015	5154	20	3204.4
P753	6940	3822	20	3029.2	7328	3770	20	2843.6	6934	3682	20	2999.45
	7063	3693	20	3068.8	7023	4017	20	2821.05	7305	4285	20	2925.55
P757	6998	3565	20	3118.95	7039	3624	20	2940.4	6729	3473	20	2921.65
	7052	3700	20	3057.75	6915	3681	20	2880.3	7709	4212	20	2907.6

Table C.3 (Cont'd)

Prob.	B1				C				CCol			
	T=0,6, R=2,5, 10 jobs, 20 machines											
	Makespan	Lmax	# tardy	Av. Tard.	Makespan	Lmax	# tardy	Av.Tard.	Makespan	Lmax	# tardy	Av. Tard.
P762	4875	3274	10	2773.4	5407	3805	10	2729.6	5279	3694	10	3121.4
	4483	2889	10	2579.9	5433	3680	10	2475.2	5023	3270	10	2330.3
P766	5003	3573	10	3354.1	5297	3896	10	3153.2	4966	3465	10	3141.5
	4717	3136	10	2987.7	5080	3987	10	2968.3	5085	3992	10	2973.3
P770	4924	3329	10	3081.6	5392	4295	10	3069.6	4596	3166	10	2943.7
	4674	3213	10	2945	4830	3365	10	2708	4779	3602	10	2725.2
P774	5092	3416	10	3072.5	5319	3718	10	3041.3	5215	3800	10	3187.7
	4600	3037	10	2701.9	5705	4104	10	2735.5	5705	4104	10	2735.4
P778	5051	3277	10	2989.9	5193	3782	10	3099	5456	3682	10	3098.7
	4748	3023	10	2745	5654	3880	10	2957.5	5923	4149	10	3012.7
	T=0,6, R=2,5, 20 jobs, 20 machines											
P781	7956	5094	20	4201.4	8040	5063	20	3889.8	7828	5010	20	4181.15
	7262	4506	20	3782.65	7758	4940	20	3427.6	7774	4912	20	3472.8
P785	8179	5108	20	4084.3	8746	5506	20	4114.9	7892	4821	20	3830.2
	7649	4192	20	3573.1	8428	5237	20	3736.85	7970	5190	20	3681.85
P789	8261	4843	20	3705.3	9565	6026	20	3642.1	7408	4100	20	3310.9
	7436	3938	20	3226.85	8692	5343	20	3346.15	8484	5135	20	3282.05
P793	8183	4678	20	3928.1	7963	5330	20	4081.55	8350	4845	20	4031.25
	7581	4337	20	3716.65	7726	4778	20	3565	7938	4874	20	3551.7
P797	8110	4781	20	4198.1	8799	5470	20	4014.6	8245	5124	20	4191.05
	7398	4069	20	3631.65	8158	4970	20	3668.5	8419	5090	20	3705.15

Note: The first line belonging to each problem is the results of one alternative machine case and the second line is the results of two alternative machine case.

Table C.4 Results of Algorithms for Distributed Environment

Problem1

Alg-B1	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	425	398	707	506	288	442	276	321	297	369	2271	0
1 SPT	3	434	406	689	506	288	442	276	321	314	369	2271	18
2 SPT	3&4	434	406	689	506	288	442	276	321	314	369	2271	9
4 SPT	1,3,4&6	434	406	689	506	288	442	276	321	314	369	2271	2.25
All SPT	all	379	300	705	596	328	411	476	166	301	339	2271	2.7

Alg-B1Col	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	425	398	707	506	288	442	276	321	297	369	2271	0
1 SPT	3	425	398	707	506	288	442	276	321	297	369	2271	0
2 SPT	3,4	425	398	707	506	288	442	276	321	297	369	2271	0
4 SPT	1,3,4&6	425	398	707	506	288	442	276	321	297	369	2271	0
All SPT	all	426	408	716	523	258	437	269	321	301	352	2271	1.8

Alg-C	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	327	214	850	534	304	401	266	499	295	246	2200	0
1 SPT	3	309	187	812	495	345	407	436	403	271	268	2200	38
2 SPT	3&4	340	252	580	534	338	577	330	533	238	240	2225	135
4 SPT	3,4,6&8	335	252	580	534	340	574	345	492	266	240	2225	26
All SPT	all	421	208	832	303	342	568	342	300	259	349	2200	1.2

Alg-Ccol	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	318	193	642	626	315	586	342	516	244	259	2200	0
1 SPT	3	318	193	614	652	315	586	342	516	244	259	2200	28
2 SPT	3,4	300	256	1020	421	242	475	303	470	255	244	2200	-86.5
4 SPT	3,4,6&8	300	256	1020	421	242	475	303	470	255	244	2200	-4
All SPT	all	300	256	1020	421	242	475	303	470	255	244	2200	5.5

Table C.4 (Cont'd)
Problem 2

Alg-B1	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	386	364	270	493	301	185	235	164	806	830	3317	0
1 SPT	10	386	366	260	607	319	185	235	164	705	825	3317	5
2 SPT	9&10	386	366	260	607	319	185	235	164	705	825	3317	53
4 SPT	1,4,9&10	430	272	284	576	328	190	235	168	705	836	3250	-8
All SPT	all	430	272	284	576	328	190	235	168	705	836	3250	-3.2

Alg-B1Col	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	all	386	364	270	493	301	185	235	164	806	830	3317	0
1 SPT	10	386	364	245	502	319	185	231	164	812	834	3317	-4
2 SPT	9&10	386	364	245	502	319	185	231	164	812	834	3317	-5
4 SPT	1,4,9&10	324	378	270	471	328	190	231	168	812	834	3250	18.5
All SPT	all	324	378	270	471	328	190	231	168	812	834	3250	2.8

Alg-C	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	362	381	423	438	231	277	191	282	707	743	3743	0
1 SPT	10	410	341	580	457	314	356	178	271	642	652	3517	91
2 SPT	9&10	410	341	580	457	314	356	178	271	642	652	3517	78
4 SPT	3,4,9 &10	388	357	542	441	466	334	191	236	620	634	3517	18.5
All SPT	all	373	371	521	575	338	274	305	321	816	455	3997	-31.4

Alg-Ccol	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	418	445	319	501	257	206	200	290	861	637	3776	0
1 SPT	9	418	445	319	501	257	206	200	290	861	637	3776	0
2 SPT	9&10	506	335	314	499	174	275	248	316	610	795	3776	46.5
4 SPT	2,4,9&10	348	399	487	270	223	303	186	363	622	737	3886	104
All SPT	all	361	384	474	259	265	237	160	369	638	759	3869	19.7

Table C.4 (Cont'd)
Problem 3

Alg-B1	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	714	203	254	420	282	312	586	436	373	392	1601	0
1 SPT	1	542	257	321	287	307	325	660	468	376	327	1601	172
2 SPT	1&7	651	257	321	287	307	325	682	352	376	327	1601	-16.5
4 SPT	1,4,7,&8	538	403	241	509	323	312	317	449	390	324	1404	85.74
All SPT	all	512	202	366	285	536	214	279	473	517	582	1657	0.6

Alg-B1Col	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	714	203	254	420	282	312	586	436	373	392	1601	0
1 SPT	1	714	203	254	420	282	312	586	436	373	392	1601	0
2 SPT	1&7	714	203	254	420	282	312	586	436	373	392	1601	0
4 SPT	1,4,7,&8	544	347	247	395	280	314	692	353	326	457	1638	43
All SPT	all	426	202	366	285	536	214	279	473	498	664	1657	2.9

Alg-C	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	729	446	325	317	356	371	284	364	289	331	1893	0
1 SPT	1	729	446	325	317	356	371	284	364	289	331	1893	0
2 SPT	1&2	729	446	325	317	356	371	284	364	289	331	1893	0
4 SPT	1,2,6&8	713	478	474	278	259	346	213	378	278	424	1893	-1.25
All SPT	all	684	428	430	377	260	254	305	412	363	360	1893	-6.1

Alg-Ccol	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	652	415	310	477	334	399	358	369	380	365	1906	0
1 SPT	1	652	415	310	477	334	399	358	369	380	365	1906	0
2 SPT	1&4	558	539	352	383	408	350	456	342	390	322	1994	94
4 SPT	1,2,4&6	777	352	281	505	484	275	605	359	332	382	2202	8.5
All SPT	all	748	423	284	496	469	246	266	474	302	331	2050	0.7

Table C.4 (Cont'd)
Problem 4

Alg-B1	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	298	197	496	462	386	587	451	209	482	711	3324	0
1 SPT	10	227	337	490	530	395	571	382	173	678	516	3324	195
2 SPT	6&10	291	242	551	220	436	575	368	231	803	595	3254	64
4 SPT	3,6,9&10	291	242	551	220	436	575	368	231	803	595	3254	-62
All SPT	all	320	251	503	472	436	444	370	244	616	573	3289	12.5

Alg-B1Col	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	298	197	496	462	386	587	451	209	482	711	3324	0
1 SPT	10	298	197	496	462	386	587	451	209	482	711	3324	0
2 SPT	6&10	339	328	387	357	434	454	452	291	714	663	3199	90.5
4 SPT	3,4,9&10	339	328	387	357	434	454	452	291	714	663	3199	7.5
All SPT	all	312	322	393	414	322	523	448	266	690	636	3176	-4.7

Alg-C	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	188	356	312	326	411	423	355	295	929	628	3670	0
1 SPT	9	188	356	312	326	411	423	355	295	929	628	3670	0
2 SPT	9&10	188	356	312	326	392	423	379	295	929	640	3670	-6
4 SPT	5,6,9&10	188	356	312	326	294	423	379	295	1025	644	3670	1.25
All SPT	all	226	369	323	370	303	377	331	410	952	680	3673	-10.9

Alg-Ccol	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	310	277	369	433	257	501	371	429	700	621	3370	0
1 SPT	9	310	277	369	433	257	501	371	429	700	621	3370	0
2 SPT	9&10	310	277	369	433	257	501	371	429	700	621	3370	0
4 SPT	4,6,9&10	310	277	369	433	257	501	371	429	700	621	3370	0
All SPT	all	310	277	369	433	257	501	371	429	700	621	3370	0

Table C.4 (Cont'd)
Problem 5

Alg-B1	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	262	379	224	321	309	725	295	431	592	579	1523	0
1 SPT	6	261	362	221	340	287	636	300	428	568	695	1523	89
2 SPT	6&9	261	362	221	340	287	636	300	428	568	695	1523	56.5
4 SPT	6,8,9&10	261	362	221	340	287	636	300	428	568	695	1523	0
All SPT	all	261	365	174	370	287	636	300	428	579	695	1523	2.2

Alg-B1Col	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	262	379	224	321	309	725	295	431	592	579	1523	0
1 SPT	6	261	362	221	325	287	727	300	428	568	606	1523	-2
2 SPT	6&9	261	362	221	325	287	727	300	428	568	606	1523	11
4 SPT	6,8,9&10	261	362	221	325	287	727	300	428	568	606	1523	-0.5
All SPT	all	261	362	221	325	287	727	300	428	568	606	1523	3.2

Alg-C	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	571	216	229	541	417	440	604	354	374	408	1816	0
1 SPT	7	508	321	313	492	442	440	406	397	356	391	1816	198
2 SPT	1,7	508	321	313	492	442	440	406	397	356	391	1816	130.5
4 SPT	1,4,6&7	680	240	233	390	178	656	473	211	438	468	1816	-10.75
All SPT	all	552	427	207	369	582	392	360	265	360	534	1732	10.6

Alg-Ccol	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	403	451	222	637	414	449	327	309	412	384	1777	0
1 SPT	4	403	451	222	637	414	449	327	309	412	384	1777	0
2 SPT	2,4	403	451	222	637	414	449	327	309	412	384	1777	0
4 SPT	2,4,5&6	397	452	485	429	433	367	311	313	411	383	1777	67.5
All SPT	all	436	408	222	516	252	551	324	361	432	388	1777	11.8

P221

Alg-B1	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	506	817	361	417	568	248	201	291	268	201	2577	0
1 SPT	2	611	573	405	471	467	380	194	258	182	395	2591	
2 SPT	2&5	611	573	405	471	467	380	194	258	182	395	2591	
4 SPT	1,2,5&4	520	497	376	396	466	433	171	329	230	411	1793	
All SPT	all	520	408	364	396	469	449	171	346	261	492	1793	

Alg-B1Col	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	506	817	361	417	568	248	201	291	268	201	2577	0
1 SPT	2	506	817	361	417	568	248	201	291	268	201	2577	0
2 SPT	2&5	506	817	361	417	568	248	201	291	268	201	2577	0
4 SPT	1,2,5&4	506	817	361	417	568	248	201	291	268	201	2577	0
All SPT	all	506	817	361	417	568	248	201	291	268	201	2577	0

Alg-C	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	364	658	509	241	614	307	286	248	304	280	1786	0
1 SPT	2	364	658	509	241	614	307	286	248	304	280	1786	0
2 SPT	2&5	364	657	506	241	606	307	286	248	304	272	1786	
4 SPT	1,2,3&5	558	495	189	762	395	333	207	248	326	271	1786	
All SPT	all	483	653	443	404	400	616	253	248	303	233	1966	

Alg-Ccol	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	384	630	557	332	439	433	344	371	391	240	2075	0
1 SPT	2	384	630	557	332	439	433	344	371	391	240	2075	0
2 SPT	2&3	384	630	557	332	439	433	344	371	391	240	2075	0
4 SPT	2,3,5&6	422	658	335	505	594	423	263	284	386	329	2172	
All SPT	all	248	816	322	514	551	419	230	223	340	261	1961	

P224

Alg-B1	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	642	402	845	427	463	319	277	134	187	308	1941	0
1 SPT	3	344	305	464	419	374	847	207	236	453	290	1734	
2 SPT	1&3	344	305	464	419	374	847	207	236	453	290	1734	
4 SPT	1,3,4&5	517	510	434	336	289	578	185	235	577	306	1823	
All SPT	all	312	237	360	352	482	797	255	178	606	242	1505	

Alg-B1Col	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	642	402	845	427	463	319	277	134	187	308	1941	0
1 SPT	3	316	286	372	499	437	826	220	259	487	282	1835	
2 SPT	1&3	316	286	372	499	437	826	220	259	487	282	1835	
4 SPT	1,3,4&5	327	304	350	731	433	570	335	264	295	409	1761	
All SPT	all	388	274	430	856	365	420	316	229	338	420	1807	

Alg-C	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	361	432	497	667	346	490	215	379	258	314	1942	
1 SPT	4	392	436	716	563	346	490	215	304	195	314	1942	
2 SPT	3&4	377	351	626	659	214	524	241	339	274	381	2112	
4 SPT	2,3,4&6	393	347	622	659	209	524	255	339	274	381	2112	
All SPT	all	420	426	552	753	345	318	317	342	306	315	2146	

Alg-Ccol	SPT mc's	MF(m=1)	MF(m=2)	MF(m=3)	MF(m=4)	MF(m=5)	MF(m=6)	MF(m=7)	MF(m=8)	MF(m=9)	MF(m=10)	Lmax	LO change
All EDD	no	312	472	654	824	198	470	271	346	236	311	2239	
1 SPT	4	312	472	654	824	198	470	271	346	236	311	2239	
2 SPT	3&4	312	476	637	824	198	470	271	346	236	311	2239	
4 SPT	2,3,4&6	328	475	649	824	198	470	262	346	236	311	2239	
All SPT	all	321	483	473	1052	198	470	262	327	222	316	2239	

Appendix D

Detailed Results of Chapter 4

Table D.1 Detailed results of algorithms for each performance criterion

Prob.	D1-Col						D1-Comp						D2					
T=0.3, R=2.5, 20 jobs, 10 machines																		
	M.span	Lmax	#id.j*	#id.j**	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av.Tard.
L12***	1437	1434	10	10	15	507.75	1013	896	8	8	19	399.45	1082	589	10	10	19	389.15
L13	1319	1337	10	10	16	442	1017	571	8	8	18	224.5	1037	557	10	10	19	246.6
L14	1269	1264	10	10	14	475.35	1029	639	9	9	19	243.1	1198	513	10	10	19	260.55
L15	1310	1337	10	10	13	412.1	1134	710	8	9	17	212.8	1126	589	10	10	16	269.75
L23	1295	1370	10	10	16	619.4	1067	924	7	7	20	482.35	1036	775	10	10	20	412.65
L24	1474	1538	10	10	14	357.25	1031	588	7	8	17	155.05	1153	384	10	10	17	208.8
L25	1425	1325	10	8	15	458.75	889	601	8	9	19	285.05	1279	568	10	10	20	353.2
L34	1263	1257	10	10	18	434.85	957	474	7	8	20	301.95	1090	618	10	10	20	354.7
L35	1321	1188	10	9	12	359.25	911	387	7	6	18	201	1039	506	10	10	20	235.9
L45	1536	1471	10	9	16	360.45	888	480	8	8	18	250.85	988	489	10	10	19	249.5
T=0.3, R=2.5, 30 jobs, 10 machines																		
L67	1644	1712	14	15	23	611.967	1439	899	12	12	30	362.133	1556	633	15	15	30	384.967
L68	1946	2046	13	13	20	673.367	1393	877	13	12	27	357.8	1435	877	15	15	27	342.6
L69	1845	1894	15	14	18	555.567	1283	939	14	12	24	219.167	1549	681	15	15	22	204.433
L610	1575	1568	15	14	17	477.733	1492	741	11	10	24	248.1	1571	566	15	15	28	294.4
L78	1763	1501	14	14	16	408.133	1354	588	12	12	29	200.7	1534	510	15	15	26	198.2
L79	1686	1572	15	14	16	519.133	1311	596	12	9	24	209.467	1519	668	15	15	23	275.933
L710	1740	1724	15	13	23	621.233	1337	764	10	10	28	357.333	1570	691	15	15	27	331.133
L89	1610	1700	14	14	18	444.9	1367	596	12	12	21	166.267	1472	554	15	15	21	156.233
L810	1636	1578	14	15	16	512.3	1428	886	11	12	16	165.467	1488	402	15	15	21	164.833
L910	1981	2117	13	13	20	562.933	1444	1062	12	11	27	244.6	1554	613	15	15	26	282.9

* Number of ideal jobs the first team is awarded
** Number of ideal jobs the second team is awarded
*** Problem generated using LA01 and LA02

Table D.1 (Cont'd)

Prob.	D1-Col						D1-Comp						D2					
T=0.3, R=2.5, 20 jobs, 20 machines																		
	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av.Tard.
L1617	2296	2261	10	9	19	1107.9	1364	1043	7	7	20	745.8	1573	1209	10	10	20	803.35
L1618	2065	2126	10	10	18	872.15	1424	853	8	7	20	614.8	1550	882	10	10	20	712.15
L1619	2195	2190	8	10	18	901.55	1571	928	7	7	20	623.8	1637	1008	10	10	20	742.05
L1620	2015	2003	9	10	17	846	1540	886	6	8	20	586.15	1664	1007	10	10	20	605.35
L1718	1994	1989	10	9	19	884.2	1369	968	8	7	20	665.95	1445	991	10	10	20	614.45
L1719	1915	1880	10	9	19	961.35	1367	974	9	8	20	717.6	1497	1051	10	10	20	761.1
L1720	1893	1937	10	10	19	734.55	1465	984	8	7	20	488.2	1589	1008	10	10	20	514.3
L1819	1664	1566	10	10	18	697.45	1455	1018	8	8	20	565.45	1586	805	10	10	20	618.75
L1820	2477	2522	10	10	17	942.45	1452	868	8	8	20	615.2	1703	1006	10	10	10	641.8
L1920	1960	2038	10	10	17	904.45	1438	781	7	8	20	549.35	1375	780	10	10	20	536.95
T=0.3, R=1.5, 20 jobs, 10 machines																		
	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av.Tard.
L12	1306	1168	10	10	12	394.8	899	536	8	7	20	252.4	1108	595	10	10	20	350.85
L13	1146	1030	9	10	15	414.4	929	534	8	10	19	224.9	1227	627	10	10	19	275.5
L14	1245	1144	10	9	15	412.65	1029	616	9	9	20	290.95	1110	490	10	10	19	317.65
L15	1252	980	7	9	15	397.6	890	666	8	9	19	261.3	1056	784	10	10	17	268.5
L23	1232	1127	9	10	16	412.95	999	475	6	7	20	281.2	1154	644	10	10	19	303.3
L24	1294	1031	7	9	16	420.1	1003	755	6	6	20	325.45	1244	642	10	10	19	306.7
L25	1399	1305	10	10	15	448.5	938	471	9	10	18	184.15	1063	491	10	10	18	171.3
L34	1296	1123	8	9	16	444.4	886	482	6	5	20	229.7	1021	424	10	10	20	272.4
L35	1082	880	9	10	15	349.3	862	419	7	7	20	257.3	1071	559	10	10	20	302.55
L45	1053	842	9	9	17	369.3	945	490	6	7	20	296.7	1089	564	10	10	20	302.35

Table D.1 (Cont'd)

Prob.	D1-Col						D1-Comp						D2					
T=0.3, R=1.5, 30 jobs, 10 machines																		
	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av.Tard.
L67	1687	1462	13	14	24	409.867	1378	673	12	11	29	200.833	1421	567	15	15	28	231.133
L68	1426	1233	12	13	21	380.233	1289	820	11	11	27	269.133	1423	635	15	15	27	245
L69	2209	1821	14	15	24	557.2	1410	752	11	12	28	236.367	1628	823	15	15	27	278.567
L610	1636	1448	15	15	18	411.9	1312	735	13	13	29	271.533	1516	729	15	15	29	285.033
L78	1677	1509	15	15	20	495.9	1327	573	13	13	28	201.367	1461	524	15	15	30	241.667
L79	1824	1635	13	14	22	451.333	1426	546	10	10	30	233.9	1489	508	15	15	29	281.567
L710	1410	1196	14	13	18	361.533	1299	638	12	11	28	235.5	1529	778	15	15	27	283.7
L89	1752	1560	15	15	22	497	1378	684	11	11	29	285.8	1523	552	15	15	29	303.867
L810	1858	1627	15	14	23	553.867	1480	513	13	12	25	191.1	1620	738	15	15	26	243.5
L910	1554	1361	22	14	14	429.333	1501	616	12	12	28	273.633	1508	638	15	15	27	294.767
T=0.3, R=1.5, 20 jobs, 20 machines																		
L1617	1966	1833	10	10	19	857.1	1607	1042	8	8	20	591.3	1607	1042	10	10	20	713.2
L1618	2012	1862	10	10	18	783.2	1438	1040	8	8	20	624.9	1684	1091	10	10	20	677.8
L1619	2035	1934	10	10	18	832.25	1564	988	7	7	20	623.8	1618	1011	10	10	20	631.9
L1620	2270	2106	10	10	18	880.85	1603	1007	7	7	20	567.95	1557	1039	10	10	20	632.55
L1718	1649	1428	10	10	18	610	1390	820	8	7	20	499.7	1650	1080	10	10	20	529.85
L1719	1671	1570	10	10	19	782.8	1318	825	7	7	20	608.4	1394	1003	10	10	20	631.55
L1720	1816	1689	9	9	18	817.1	1351	891	8	8	20	647.65	1576	1232	10	10	20	753.75
L1819	1934	1812	10	10	19	870.35	1435	842	5	6	20	574	1510	1048	10	10	20	631.15
L1820	1827	1703	10	10	19	788.6	1457	893	6	7	20	576.25	1564	968	10	10	20	598.95
L1920	1824	1707	10	10	10	801.65	1517	906	10	9	20	633.65	1415	804	10	10	20	605.45

Table D.1 (Cont'd)

Prob.	D1-Col						D1-Comp						D2					
T=0.6, R=2.5, 20 jobs, 10 machines																		
	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av.Tard.
L12	1156	1180	9	9	19	588.55	1034	911	7	7	20	433.3	1277	777	10	10	20	442.75
L13	1379	1366	10	9	17	545.1	929	534	8	10	19	224.9	1043	784	10	10	20	463.65
L14	1382	1411	9	9	19	674.65	1029	937	9	10	20	516.65	1104	818	10	10	20	475.35
L15	1186	1216	10	10	18	490.85	1046	715	6	8	20	439.95	1108	692	10	10	20	471.1
L23	1033	1036	10	10	17	456.8	954	626	7	5	20	418.75	1085	698	10	10	20	450.6
L24	1548	1517	10	9	18	617.6	1024	647	9	9	20	343.2	1063	597	10	10	20	355.35
L25	1226	1274	9	9	18	564.05	1112	953	6	6	20	501	1066	699	10	10	20	450.05
L34	1081	1111	10	10	17	445.65	951	564	7	7	20	369.6	1030	707	10	10	20	418
L35	1323	1298	10	9	16	531.05	806	449	8	8	20	330.8	863	575	10	10	20	340.55
L45	1071	1070	9	10	19	487.3	906	545	7	7	20	393.75	993	703	10	10	20	398.5
T=0.6, R=2.5, 30 jobs, 10 machines																		
L67	1599	1497	13	14	27	627.6	1385	827	12	12	30	481.467	1557	854	15	15	30	490.4
L68	1759	1798	15	15	27	726.8	1516	908	9	9	30	557.167	1590	891	15	15	30	596.6
L69	1719	1542	14	13	28	598.6	1351	596	11	11	29	373.233	1551	826	15	15	30	426.633
L610	2081	2036	15	13	26	702.233	1389	714	12	11	30	483.333	1507	907	15	15	30	520.7
L78	1643	1519	13	13	28	725.967	1343	791	10	11	30	508.033	1468	867	15	15	30	571.033
L79	1647	1591	14	13	26	629.3	1389	815	10	10	30	465.333	1499	927	15	15	30	441.8
L710	1773	1853	15	13	28	725.433	1298	1213	12	12	30	553.9	1478	827	15	15	30	591.167
L89	1888	1875	15	15	25	779.467	1402	1057	11	13	30	490.9	1416	760	15	15	30	483.133
L810	1611	1682	15	14	27	605.1	1324	740	9	10	30	488.733	1442	768	15	15	30	491.9
L910	1848	1785	14	12	28	684.567	1402	751	10	10	30	544.8	1658	975	15	15	30	536.033

Table D.1 (Cont'd)

Prob.	D1-Col						D1-Comp						D2					
T=0.6, R=2.5, 20 jobs, 20 machines																		
	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av.Tard.
L1617	1915	1917	10	10	20	998.4	1486	1084	5	5	20	700.3	1670	1268	10	10	20	771.8
L1618	1938	1922	10	9	20	975.8	1439	1009	7	7	20	803.05	1607	1144	10	10	20	913.6
L1619	2231	2259	10	10	20	1091.85	1384	1221	8	9	20	833.2	1561	1274	10	10	20	900.8
L1620	2076	2040	10	10	20	1059.75	1618	1115	7	6	20	780.55	1609	1164	10	10	20	852
L1718	1953	1938	10	10	19	1017.9	1548	1176	6	6	20	815.85	1619	1247	10	10	20	890.05
L1719	1843	1845	10	10	20	961.15	1484	1082	6	7	20	726.25	1623	1324	10	10	20	804.9
L1720	2139	2174	10	10	20	1098.7	1381	1022	7	6	20	741.5	1621	1229	10	10	20	774.15
L1819	1889	1873	9	8	20	991.65	1359	1142	6	7	20	755.3	1445	1187	10	10	20	859.25
L1820	2168	2085	10	9	20	1019.75	1570	1195	8	8	20	690.95	1535	1067	10	10	20	738.2
L1920	1827	1805	10	10	20	1050.65	1299	1027	7	7	20	785.7	1549	1273	10	10	20	879.05
T=0.6, R=1.5, 20 jobs, 10 machines																		
	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av. Tard.	M.span	Lmax	#id.j	#id.j	tardy	Av.Tard.
L12	1412	1334	10	10	20	603.25	1006	743	8	7	20	504	1118	749	10	10	20	482.3
L13	1217	1101	10	9	18	532.75	1015	674	6	7	20	383.4	1057	685	10	10	20	389.35
L14	1291	1203	10	10	19	578.6	956	811	8	8	20	445.3	1260	906	10	10	20	433.5
L15	1202	1124	10	10	19	482.15	895	598	7	8	20	378.75	1185	820	10	10	20	409.95
L23	1183	1120	10	10	18	454.5	938	651	9	8	19	421.1	1034	703	10	10	20	434.35
L24	1261	1172	8	9	19	483.35	997	763	8	8	20	396	1025	724	10	10	19	424.45
L25	1279	1213	8	8	18	497.8	911	845	10	10	20	451.1	980	641	10	10	20	419.1
L34	1168	1114	9	9	19	506.7	958	751	7	6	20	392.6	1072	754	10	10	20	450
L35	1155	1089	8	9	18	556.95	865	678	6	6	20	430.45	990	759	10	10	20	443.2
L45	1124	1011	10	10	20	477.75	923	673	9	8	20	404.35	1142	826	10	10	20	407.95

Table D.1 (Cont'd)

Prob.	D1-Col						D1-Comp						D2					
							T=0.6, R=1.5, 30 jobs, 10 machines											
L67	1844	1762	15	14	29	741.233	1416	934	11	10	30	506.433	1546	989	15	15	30	549.1
L68	1828	1727	15	15	25	603	1194	815	13	13	30	516.233	1401	857	15	15	30	554.267
L69	1603	1368	14	12	26	633.3	1486	906	12	12	30	473	1635	1062	15	15	30	495.633
L610	1510	1343	14	13	28	616.367	1373	943	12	11	30	557.233	1504	926	15	15	30	548.933
L78	1694	1541	14	14	24	581.867	1219	791	12	12	30	446.033	1504	926	15	15	30	548.933
L79	1700	1598	14	14	29	730.467	1296	966	13	13	30	545.7	1416	913	15	15	30	522.033
L710	1854	1648	15	12	25	683.667	1542	1006	10	10	30	442.8	1741	1205	15	15	30	489.733
L89	1646	1530	14	15	26	654.133	1362	860	10	11	30	508.267	1551	1055	15	15	30	511.1
L810	1819	1721	15	14	26	615.9	1259	1109	14	15	30	492.4	1366	850	15	15	30	475.933
L910	1904	1792	14	13	26	632.7	1460	1054	10	10	30	511.767	1508	936	15	15	30	548.767
							T=0.6, R=1.5, 20 jobs, 20 machines											
L1617	1898	1840	10	10	20	1028.92	1422	1071	7	6	20	755.55	1506	1149	10	10	20	815.95
L1618	1964	1860	9	9	20	959.9	1500	1136	7	8	20	718.3	1555	1275	10	10	20	740.1
L1619	2144	2056	10	10	20	1091.85	1503	1286	8	8	20	790.2	1717	1363	10	10	20	839.6
L1620	2415	2347	10	10	20	1091.35	1477	1108	8	7	20	737.5	1514	1133	10	10	20	795.1
L1718	1469	1373	9	10	20	886.45	1434	1083	7	8	20	757.8	1330	1029	10	10	20	740.8
L1719	1636	1568	10	10	20	814.1	1171	962	8	8	20	711.2	1685	1335	10	10	20	802.1
L1720	1848	1763	10	10	20	981.75	1457	1111	8	7	20	676	1595	1249	10	10	20	715.4
L1819	1776	1597	10	10	20	877.9	1250	993	8	9	20	728.6	1516	1243	10	10	20	781.05
L1820	1868	1804	10	10	20	953.45	1530	1235	8	7	20	783.95	1578	1308	10	10	20	874.9
L1920	1907	1816	10	10	20	1128.5	1398	1099	9	8	20	761.55	1394	1178	10	10	20	786.75

Appendix E

Algorithm A

ALGORITHM A

In this algorithm machines are resource agents. Each order/job is associated with an order agent. There is also a manager agent responsible for coordination of information and conflict resolution.

The following definitions and notation are used in the algorithm:

Precedence violation: The violation in the current schedule in which case an operation of a certain job is scheduled before the previous operation of the same job is not completed.

Resource conflict: The violation in the current schedule in which case two operations are scheduled on a resource for the same time interval.

Time boundry: The interval between earliest start time and latest completion time of an operation.

Hard constraint: The constraint ensuring no precedence violation.

Soft constraint: The constraint ensuring no resource conflict.

Critical conflict pair: It is a pair of operations which are in conflict and at least one of them is processed on a bottleneck machine.

Advantageous operation: It is one of the operations in a conflict pair, whose shift offers an easier solution to the violation than the other operation in the pair does.

ES_{ij}: Earliest Start time of Operation j of Job i

LC_{ij}: Latest Completion time of Operation j of Job i

p_{ij}: processing time of Operation j of Job i

n: number of operations of Job i

LRS_{ij}: Left Resource Slack of Operation j of Job i

= (Current start time of the operation)-(Current completion time of previous operation on the resource)

RRS_{ij}:Right Resource Slack of Operation j of Job i

= (Current start time of next operation on the resource) – (Current completion time of the operation)

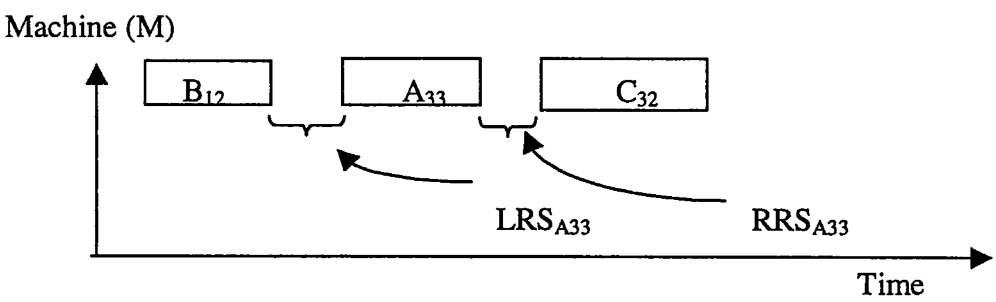


Figure E.1 LRS and RRS of an operation

LS_{ij}: Left Slack of Operation j of Job i relating the resource

= min(Current start time of the operation on the resource-ES_{ij}, LRS_{ij})

RS_{ij}: Right Slack of Operation j of Job I relating the resource

= min (LC_{ij}-Current completion time of the operation, RRS_{ij})

LOS_{ij}: Left Order Slack of Operation j of Job i

= (Current start time of the operation)-(Current completion time of previous operation of the same job)

ROS_{ij}: Right Order Slack of Operation j of Job i

= (Current start time of the next operation of the same job)- (Current completion time of the operation)

LS_{ij}' : Left Slack of Operation j of Job i relating the order

= min(Current start time of the operation -ES_{ij}, LOS_{ij})

RS_{ij}' : Right Slack of Operation j of Job i relating the order

= min(LC_{ij}-Current completion time of the operation, ROS_{ij})

The basic steps of the algorithm are as follows (see the flow charts at the end of appendix) :

1. Manager agent calculates the earliest start times and latest completion times of operations and allocates the operations to machines (i.e. Algorithm A1).

2. Resource agents generate their initial schedules (i.e. Algorithm A2) and send the related information to order agents through the manager agent.

3.1. If there is no precedence violation, feasible schedule is found.

3.2. If there is a precedence violation, each order agent uses Algorithm A3 to eliminate its own precedence violations. The related information is sent to the resource agents via the manager agent.

4.1. If there is no resource conflict, feasible schedule is found.

4.2. If there is a resource conflict, each resource agent uses A4 to eliminate its resource conflicts. After manager agent sends the related information to order agents the steps starting from 3.1 are executed until a feasible schedule is found.

Step 1: ES_{ij} and LC_{ij} are found as follows:

$$ES_{ij} := (\text{Release date of Job i}) + \sum_{k+1}^{j-1} p_{ik}$$

$$LC_{ij} = (\text{Due date of Job i}) - \sum_{k=j+1}^{n} p_{ik}$$

Algorithm A1 (Allocation of operations to machines): There may be alternative machines in the system so the manager agent should allocate the operations among the machines considering their alternatives. The steps of this algorithm are:

1. Assign the operations which has no alternatives to the only machine they can be processed by.
2. If the operations in the neighborhood of assigned operations can also be processed on the same machine, allocate them, too.
3. Among the remaining operations, combine any consecutive operations of the same job that can be processed on the same machine as if one operation.
4. Find the LC of each operation. Beginning with the operation which has the smallest LC, assign the operation to the machine which has the least sum of processing time in the required time boundry.

Step 2: Shifting bottleneck procedure is applied. Each resource agent generates its own initial schedule using the operations assigned by the manager agent. If there is more than one schedulable operation at some point, then the operation with the smallest EDD is scheduled first.

Step 3.2 (Algorithm A3)*:* Starting from earlier critical conflict pairs, each order agent executes the following steps to solve its precedence violations:

Let A_i and A_{i+1} be activities in conflict such that A_i is the prior operation and A_{i+1} is the subsequent operation (See Figure E.2).

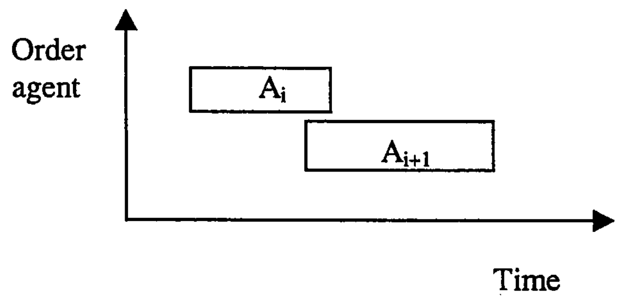


Figure E. 2 Two operations having precedence violation

1. Calculate LS for A_i and RS for A_{i+1}
2. Find the adventageous operation

2.1 If $LS_{Ai} > RS_{Ai+1}$, A_i is advantageous. If A_i can be moved to the left without violating time boundary, shift A_i. Else go 2.1.1

2.1.1. Try to shift A_{i+1} to the right without violating time boundary. If it can not be shifted go to 2.1.2

2.1.2. Try to shift A_i and A_{i+1} simultaneously within their time boundaries. If they can not be shifted, go to 2.1.3.

2.1.3. Shift A_{i+1} to the right.

2.2. If $RS_{Ai+1} > LS_{Ai}$ choose A_{i+1} as adventagous operation and repeat the same steps.

Step 4 (Algorithm A4): Each resource agent solves its conflicts starting from initial conflict pairs using this algorithm. There may be different scenarios in the formation of a resource conflict. These scenarios and the conflict solution procedure in each case are as follows:

1. If both of the operations are shifted from left: Suppose that A_i and B_j are two operations having a resource conflict. Assume that B_j is the adventagous operation i.e. $RS_{Bj}' \geq RS_{Ai}'$.

   1.1 Shift B_j to the right with no order violation. If it can not be shifted, go to 1.2.

   1.2 Shift A_i to the right with no order violation. If it can not be shifted, go to 1.3.

   1.3 Shift B_j to the right without LC violation. If it can not be shifted, shift A_i to the right without LC violation. If this can not be achieved go to 1.4.

   1.4 Shift B_j to the right.

2. If both of the operations are shifted from right: Suppose that A_i and B_j are two operations having a resource conflict. Assume B_j is the advantegous operation.

   2.1 Shift B_j as far as to the left in such a way that it causes no conflict, then append A_i to the end. If this makes no conflict for A_i, implement the solution.Else go to 2.2

2.2 Shift A_i as far as to the left in such a way that it causes no conflict, then append B_j to the end. If this makes no conflict for B_j, implement the solution. Else go to 2.3

2.3 If shifting B_j to the left causes only precedence violation, append A_i to the end and implement the solution. Else go to 2.4.

2.4 If shifting A_i to the left causes only precedence violation, append B_j to the end and implement the solution. Else go to 2.5

2.5 Shift B_j to the leftmost with no violation, append A_i to the end.

3. If one operation is shifted from left and the other from right: Assume A_i is shifted from left and B_j is shifted from right. Move B_j to the left without any violation and append A_i to the end.
4. If one operation is shifted from left and the other remained in the same position:
   4.1. If there is a solution which provides minimum amount of shift with no violation, implement it; else go to 4.2
   4.2. If there is a solution which provides minimum amount of soft constraint violation with no hard constraint violation, implement it; else go to 4.3
   4.3. Choose the solution with minimum hard constraint violation.
5. If one operation is shifted from right and the other remained in the same position, the procedure in 4 is applied.

Optional step:

This step may be applied instead of the last steps in each scenario of the above algorithm in order to shift the operations on bottleneck machines as small as possible.A time window is determined (i.e. average time boundaries of operations in the system). Time slots before and after the operations on bottleneck machines, which fit to this window are weighted. The total weight should sum up to the size of the time window. The time slots nearer to the operation on bottleneck machine should have more weight than the ones which are not close. If an RA is solving a conflict pair whose operations come just before or after the operation which is on the bottleneck, then it does the following: It finds the length of the intersection of time boundary of each operation with the related time window. Then it sums the weights within the intersection and divides the first value by the second. The operation which has the higher ratio is shifted.

A4′. Handling of Alternative Machine Case

If there are alternative machines in the system, the algorithm works the same. However we allow the change of an operation from one m/c to an alternative one. For coordination and conflict resolution we assign a team manager to each group of alternative machines. The steps of modified A4 are as follows:

1. Each RA solves its conflicts using A4 but only fixes those conflicting pairs which have a solution with no violation. It selects one operation from every conflicting pair whose solution has either a hard constraint or a soft constraint violation. Every RA informs these selected operations to its team manager.
2. Each team manager ranks all the operations and requests bids from all the machines in the team.
3. Each RA in the team gives two bids; one best case bid (the bid prepared assuming all of its operations that it informed to the team manager are given to another RA) and a worst case bid (the bid prepared assuming none of the operations it informed to the team manager are given to another RA).
4. After collecting the bids, the team manager gives an operation to another machine if the second machine's worst case bid is better than the operation's current position in the first machine. If the worst case bid has no conflict it is fixed otherwise best case bids are checked by Step 5.
5. After the fourth step, some of the operations may have been given to other machines, thus some of the best case bids may be realized. The team manager asks

the RA's whether their best case bids are still valid. If they are, then the related operations are given to these machines.

Figure E.3 Algorithm A

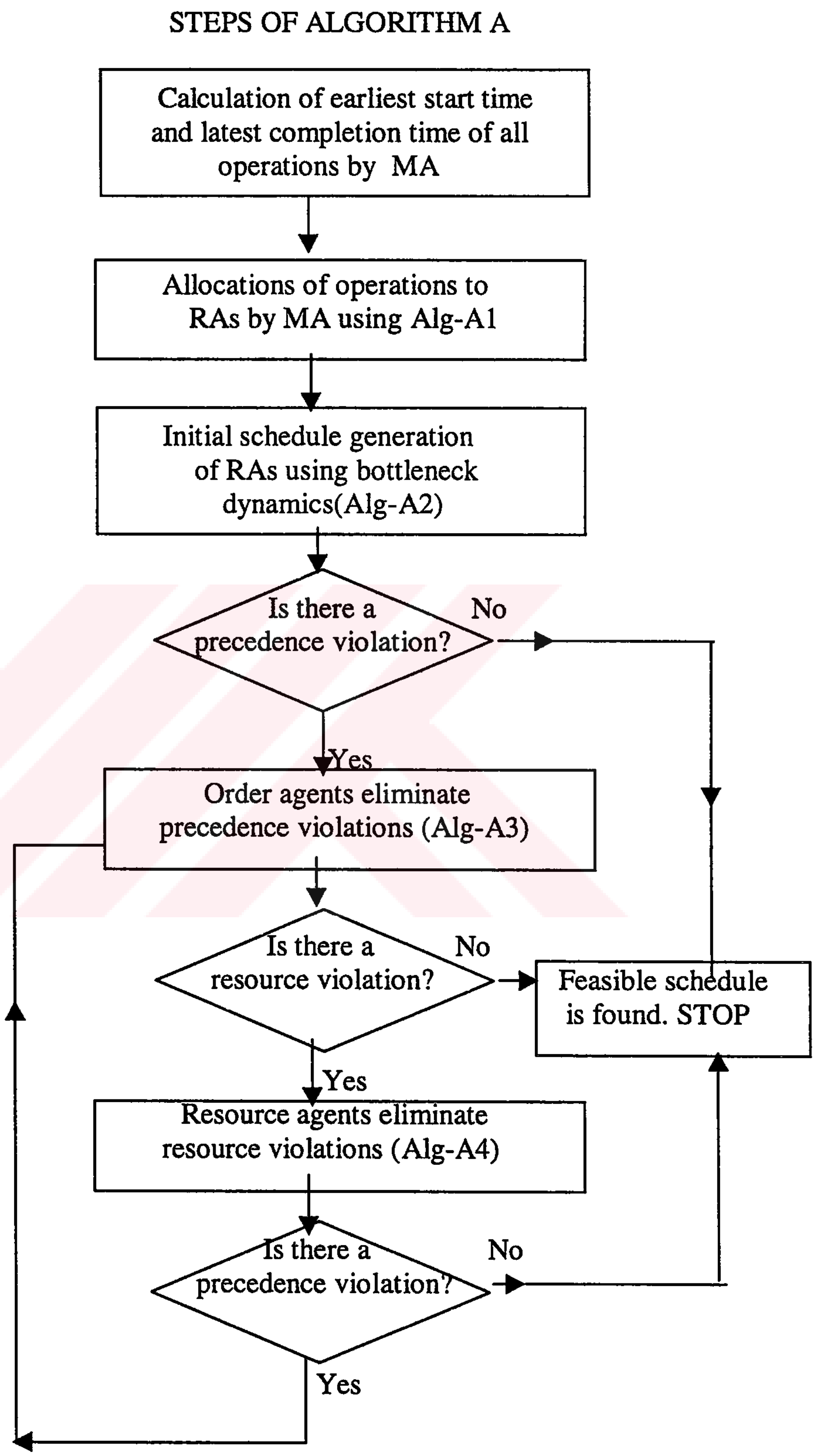


Figure E.4 Algorithm A3

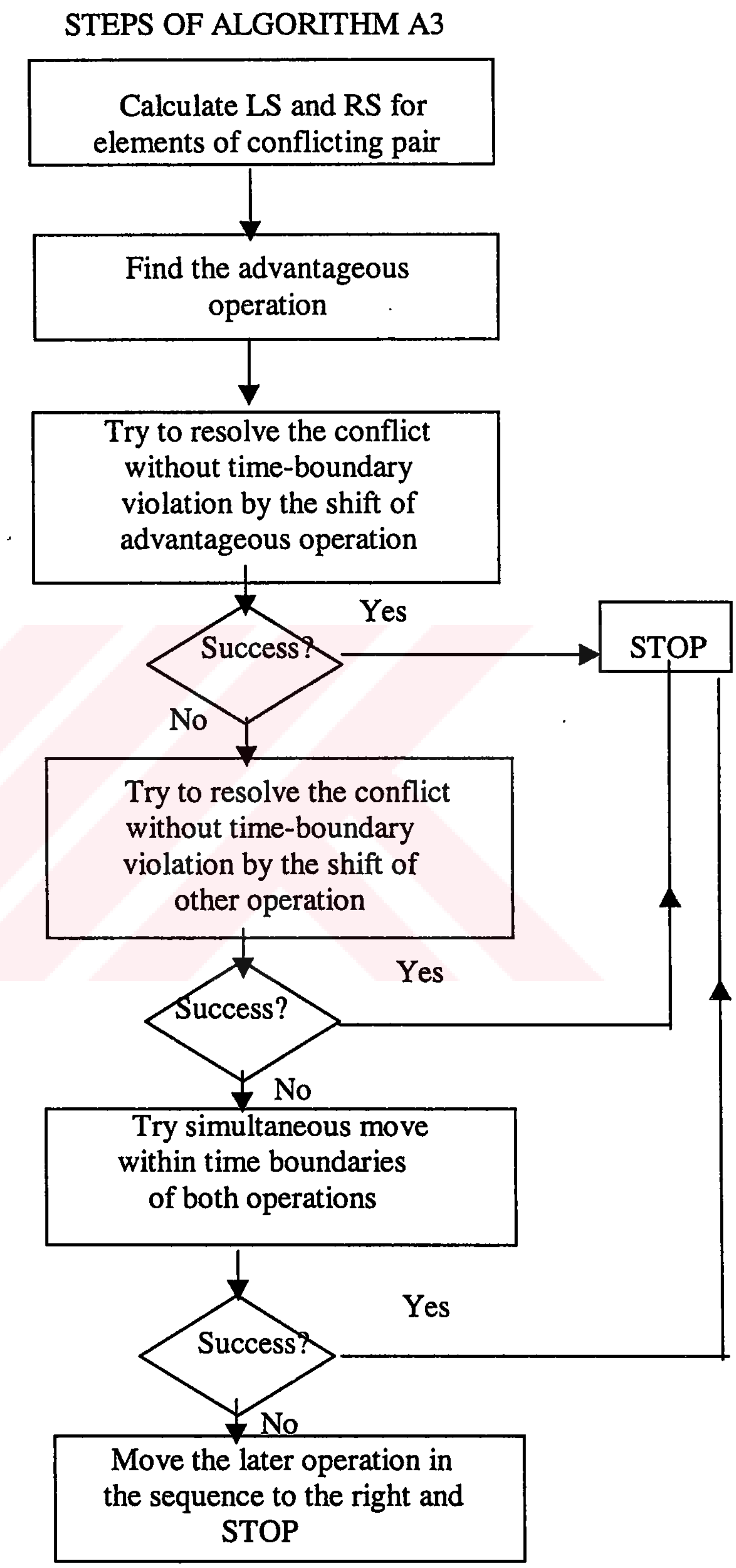


Figure E.5 Algorithm A4

ALGORITHM A4

(If both operations are shifted from left)

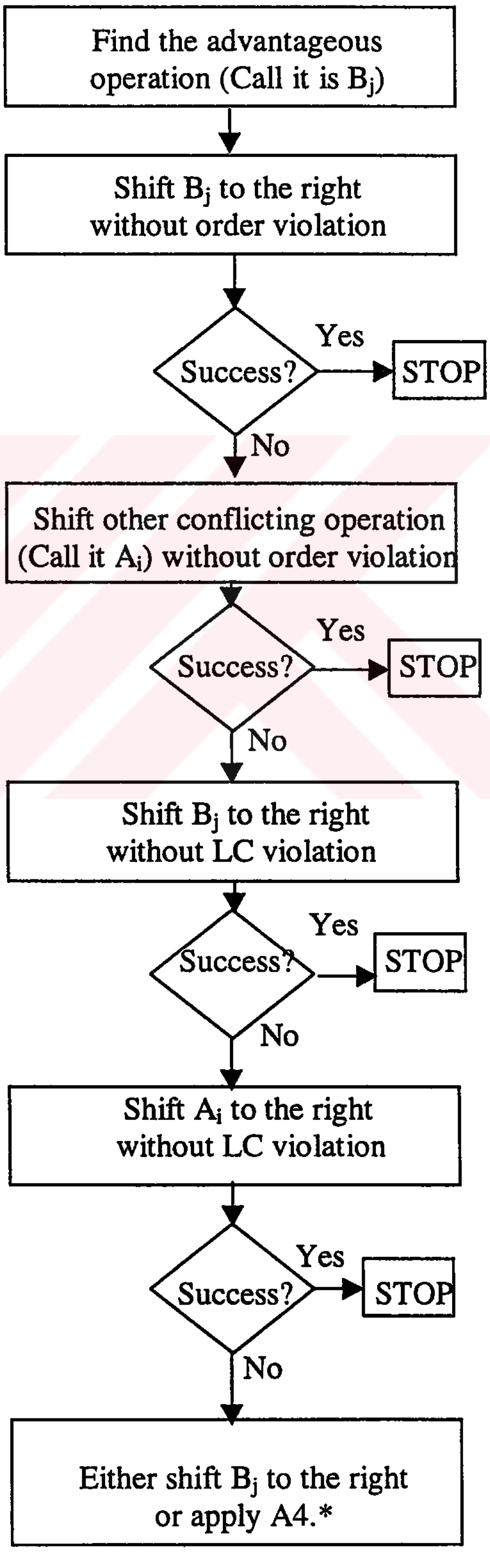


Figure E.5 (Cont'd)

ALGORITHM A4
(If both operations are shifted from right)

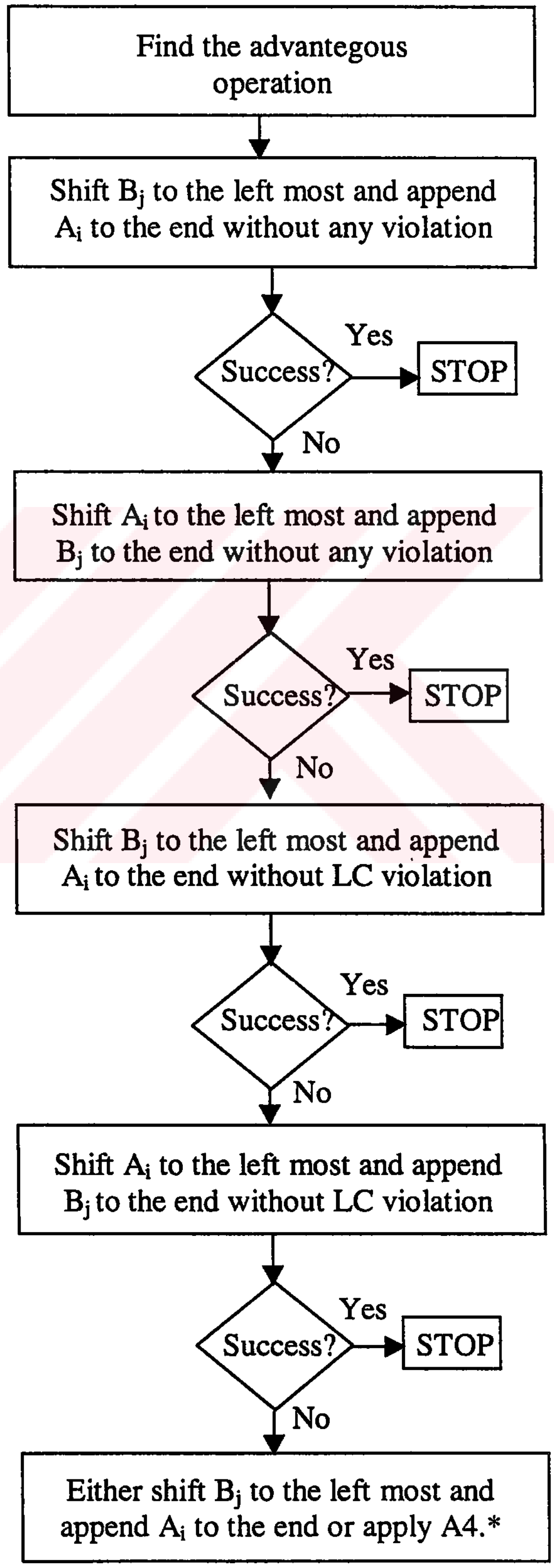
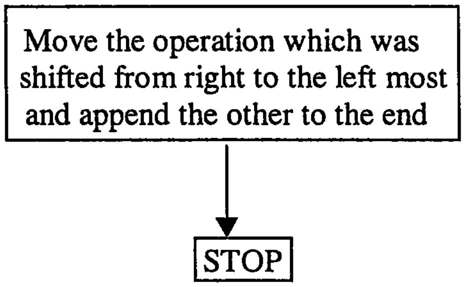


Figure E.5 (Cont'd)

ALGORITHM A4

(If one operation is shifted from left and the other from right)



(If one of the operations remained in the same position)

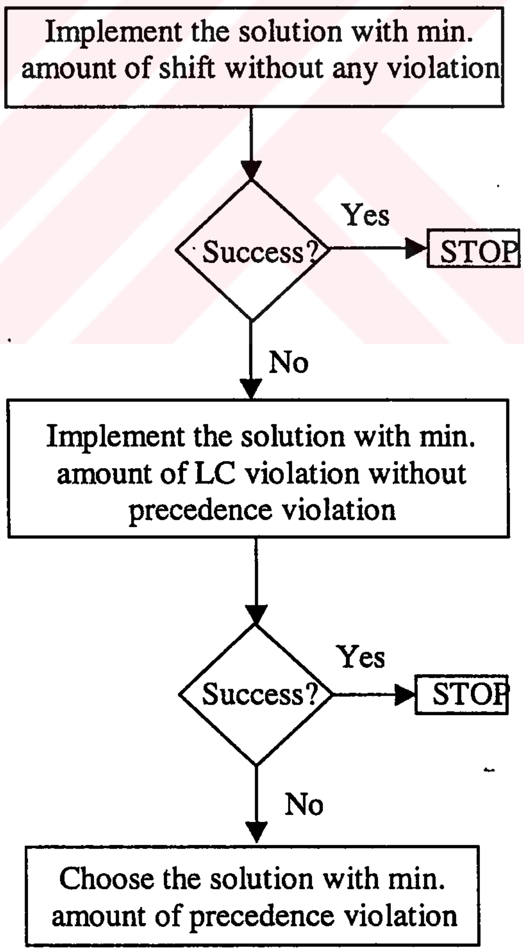


Figure E.5 (Cont'd)

ALGORITHM A4.*

(Optional for last step)