Modeling and Optimization of

Advanced Planning and Scheduling (APS)

by

CHEN Kejia

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Department of Industrial and Systems Engineering

The Hong Kong Polytechnic University

August 2006

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_______________________________(Signed)

_______Chen Kejia_______(Name of student)

ABSTRACT

Production planning and control systems based on the Material Requirements Planning (MRP) logic have been extensively implemented in the manufacturing industry. Despite its widespread use, MRP ignores capacity constraints, assumes that lead times are fixed, and does not consider operation sequences of items. All of these create many problems on the shop floor for later production. Unquestionably, MRP and operations scheduling are closely interrelated and intertwined together. Consequently, they should be integrated together to generate realistic production schedules for the shop. This integration leads to the problem of Advanced Planning and Scheduling (APS) and this thesis mainly focuses on the modeling and optimization of APS.

In this thesis, a Mixed Integer Programming (MIP) model for the APS, with the objective of minimizing cost of both production idle time and tardiness or earliness penalty of an order, is formulated. The proposed mathematical model explicitly considers capacity constraints, operation sequences, lead times and due dates in a multi-order environment and generates production schedules with operation starting time and finish time for the shop floor. Numerical results indicate that the established APS model can favorably produce optimal schedules. Since the APS problem has been proved to be NP-hard, a genetic algorithm (GA) is built to tackle it more efficiently. A series of computational tests demonstrate that the suggested GA approach is satisfactory in creating effective production plans and schedules. In order to cope with the Dynamic Advanced Planning and Scheduling

(DAPS) problem where new orders arrive on a continuous basis, both the MIP and the GA are further extended by incorporating a periodic policy with a frozen interval. The objective of the offered methodology is to find a schedule such that both production idle time and penalties on tardiness and earliness of both original orders and new orders are minimized at each rescheduling point. The provided dynamic mechanism is confirmed to be capable of improving the schedule stability while retaining efficiency. Furthermore, a prototype of the advanced planning and scheduling decision support system is designed to assist production planners to make effective decisions. Finally, a real APS problem arising from a specialist light source manufacturing company is illustrated to validate the applicability of the developed methods and system.

ACKNOWLEDGMENTS

This research would have not been so successful without the help of many people who took great support. I would like to acknowledge my supervisor, Dr. P. Ji of Department of Industrial and Systems Engineering in The Hong Kong Polytechnic University, for his guidance throughout my research project. His encouragement, helpful suggestions and countless discussions have supported the development of the project. His deep understanding and wide knowledge have broadened my view in the field about this research.

Besides, I have to express my sincere gratitude to my family and friends for their tremendous support and encouragement. Especially my parents, I owe the most to them.

Definitely, the financial support from The Hong Kong Polytechnic University enables this project possible. I would like to acknowledge all parties from the heart including those mentioned above together with those who helped me indeed but missed to thank before.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Due to stiff global competition, manufacturing companies become more and more customer-driven. The success of the manufacturing companies will rely on the ability to offer quality, cost effective products to increasingly demanding customers with incredible speed and accuracy. Meanwhile, over the past several decades, there have been rapid improvements in information technology. Nowadays, computer-based information management systems are very common to manufacturing companies, from small and mid-size to large corporations. All the changes in the environment have caused the evolution of manufacturing production planning and control systems, from Material Requirements Planning (MRP) to Manufacturing Resource Planning (MRP-II), and to Enterprise Resource Planning (ERP). Despite the developments that the manufacturing production planning and control systems have gone through, one thing has never been changed, that is, MRP is always the focal point of all manufacturing applications.

1.1 MATERIAL REQUIREMENTS PLANNING (MRP)

Material Requirements Planning (MRP) first developed in the 1960's is a set of procedures which transform a Master Production Schedule (MPS) into time-phased net requirements. It follows a top-down hierarchical approach and its basic ideas are that the demands for components depend on the demands for the subassemblies or final products they constitute; therefore, it generates the demands for components from the actual demands of the dependent final products.

1.1.1 Inputs and outputs of MRP

There are five main inputs to an MRP system:

- Master Production Schedule (MPS)
- Bills of materials (BOM)
- Inventory records
- Lot sizing rules
- Planned lead time

The primary outputs of an MRP system include:

- Order release notices, for placement of planned orders
- Replanning notices, when there are changes in open orders
- Cancellation notices, when there is cancellation or suspension of open orders
- Backup data

Figure 1.1 illustrates in more detail how MRP functions within the overall framework of a production planning and control system. The front end in Figure 1.1 represents the long term planning portion. Resource planning and production planning take a long term view to make decisions for the foreseeable future. The MPS which contains the detailed requirements for final products by date and quantity can then be created for an extended period on the basis of directions set by the production and marketing departments. The feasibility of MPS is verified by use of a rough cut capacity planning (RCCP) tool. The engine portion describes the MRP system and its associated inputs and outputs. Based on the routing file which defines how to produce an item (machines, toolings, setup times, etc), the capacity requirements planning (CRP) module checks the plans generated by the MRP for feasibility. If infeasible, adjustments should be made in the MPS and/or to production capacity before a new MRP is run. If feasible, the time-phased MRP plans, which take the form of orders, are delivered to the shop floor. The back end in Figure 1.1 depicts the detailed shop floor and vendor control system [Orl69, Vol88, Vos03].

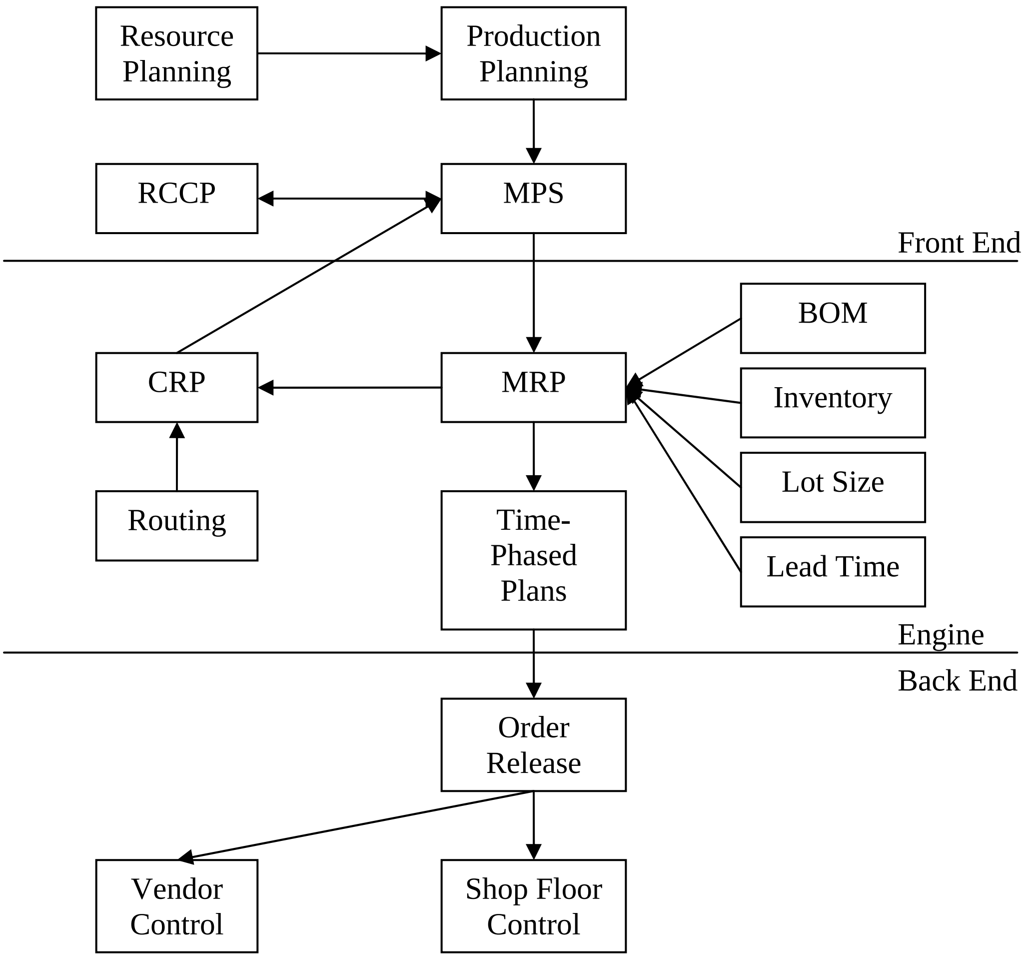


Figure 1.1 MRP within the production planning and control system [Vol88]

1.1.2 MRP logic

An MRP system operates in the following manner. From the gross requirements for end items as specified in the MPS, MRP considers scheduled receipts and on-hand inventory to determine net requirements. The net requirements are grouped into orders according to lot-sizing rules. The orders are then offset by the necessary lead time for fulfilment. The resulting planned order releases provide the gross requirements for the next level's items in the BOM. The process repeats itself for all items, one by one.

The basic MRP record is displayed in Table 1.1. The record includes the following:

- Gross Requirements: the total amount needed in each period
- Scheduled Receipts: existing replenishment orders due in each period
- On-Hand Inventory: inventory status after the production and demand have occurred in each period
- Net Requirements: Max { Gross Requirements – Scheduled Receipts – On-Hand Inventory, 0}
- Planned Order Receipts: replenishment orders scheduled to arrive in each period
- Planned Order Releases: generated from Planned Order Receipts by offsetting the lead time

A simple example is used to illustrate the MRP logic. It is assumed that there is a two-level final product (F1) that has a bill of materials (BOM) as shown in Figure 1.2 and the properties presented in Figure 1.3. In the MRP system, BOM is the central element, which identifies the components that are combined to make other subassemblies and ultimately the final products, and also reflects the production procedure. Usually, the items in the BOM are sorted in low level code order. In such a way, the list begins with the final products and no child item appears in the list before the parent item. As seen in Figure 1.2 and 1.3, the final product (F1) is assembled from several components, including two units of component C1. The lead times of F1 and C1 are 2 time buckets and 1 time bucket, respectively. While the lot size for F1 is 1, that is, it can be ordered in any quantity, the lot size for C1 is 40. The MRP records for F1 and C1 are listed in Table 1.2. From the table, it could be found that the planned orders for the parent (F1) become the gross requirements for the child component (C1). Records for other items would be filled exactly in the same way. According to the logic, the components are coordinated to arrive together for assembly.

Table 1.1 A MRP record

Period	1	2	3	4	5	6	7
Gross Requirements							
Scheduled Receipts							
On-Hand Inventory							
Net Requirements							
Planned Order Receipts							
Planned Order Releases							

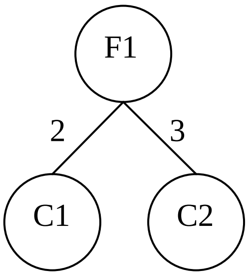


Figure 1.2 BOM for a simple MRP example

1. F1:

    Lead Time: 2 time buckets

    Lot Size: 1

    Components: 2 C1, 3 C2

    Initial Inventory: 50

2. C1:

    Lead Time: 1 time bucket

    Lot Size: 40

    Components: N/A

    Initial Inventory: 225

3. C2:

    Lead Time: 1 time bucket

    Lot Size: 100

    Components: N/A

    Initial Inventory: 0

Figure 1.3 Data for a simple MRP example

Table 1.2 (a) A simple example of MRP

F1	1	2	3	4	5	6	7
Gross Requirements	15	25	120		65		20
Scheduled Receipts							
On-Hand Inventory(50)	35	10	0	0	0	0	0
Net Requirements			110		65		20
Planned Order Receipts			110		65		20
Planned Order Releases	110		65		20		

Table 1.2 (b) A simple example of MRP

C1	1	2	3	4	5	6	7
Gross Requirements	220		130		40		
Scheduled Receipts	40						
On-Hand Inventory(225)	45	45	35	35	35	35	35
Net Requirements			85		5		
Planned Order Receipts			120		40		
Planned Order Releases		120		40			

1.2 SCHEDULING

Scheduling, which concerns execution of material plans, is a significant activity in production planning and operation systems. The function of scheduling in a manufacturing environment is depicted with the flow of information in Figure 1.4.

As illustrated in the figure, scheduling interacts with many other aspects of an organization and has an immediate effect on the company's performance.

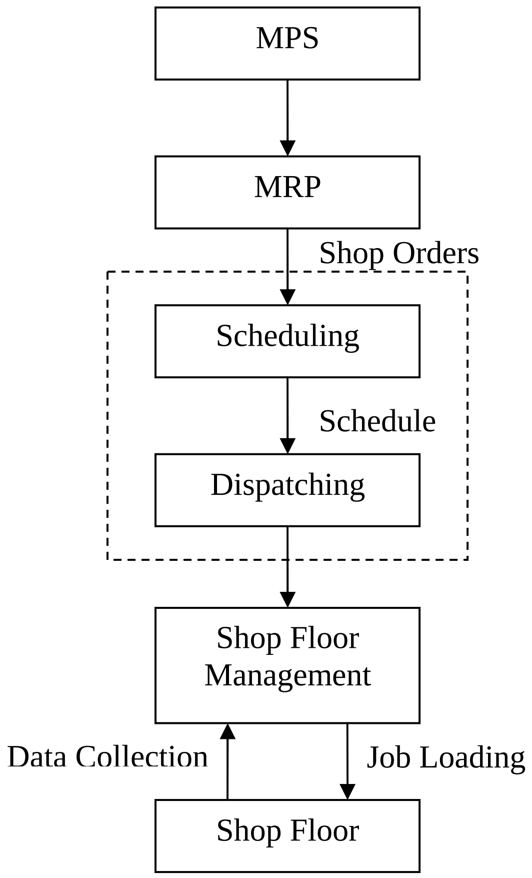


Figure 1.4 Scheduling within the production planning and control system

Shop orders with certain production objectives are produced as the results of MRP. These shop orders have to be processed on the work centers in a given sequence. In view of operation sequential constraints and resource capacity constraints, a detailed schedule of the tasks to be performed should be developed to achieve the objectives. The performance of the scheduling directly impacts the operating efficiency and production control. A good scheduling decision can reduce the idle time of machines or work centers and increase the throughput of production, while the poor one may affect capacity utilization, work-in-process (WIP) inventory, shop floor control, and so on. Ultimately, poor performance of the scheduling will lead to decisions that adversely influence a company's sales and profitability.

1.3 PROBLEMS

Despite the wide spread use of MRP, it ignores capacity constraints, assumes that lead times are fixed, and does not consider operation sequences of items [Bil83, Kra87, Sum93, Taa97, Vos03]. This creates many problems on the shop floor for later production, such as varying workloads, changing bottleneck, etc. Moreover, there is no guarantee that a feasible production schedule exists for the generated production plan. When it is not feasible, a great deal of adjustments should be made to the production plan for the capacity levels, lot sizes, MPS, etc. The unreliable planning process drives planners to lengthen planned lead times in order to get better performance. However, longer planned lead times normally cause

- higher forecast error,
- longer queues,
- more work-in-process (WIP),
- lower machine utilization,
- less throughput,
- higher production costs,
- more unreliable planned lead times.

According to the standard MRP doctrines, the above problems are alleviated by closed-loop capacity planning. However, typical MRP-based manufacturing planning and control systems only utilize capacity planning techniques such as rough cut capacity planning (RCCP) and capacity requirements planning (CRP). RCCP is designed to estimate capacity requirements to ensure the feasibility of the given MPS before MRP generates its plans, which is only approximate and based on infinite loading assumptions. CRP checks the plans created by MRP for feasibility by translating the plans into shop hours by work centre by period, which still does not take into account lead times, operation sequence, etc. Neither RCCP nor CRP provides any true closed loop feedback to the production planning and control process.

Unquestionably, MRP and production scheduling are closely interrelated, and they should be integrated together to generate realistic production schedules for the shop floor, which leads to the problem of Advanced Planning and Scheduling (APS). In addition, Advanced Planning and Scheduling (APS) creates a unified solution space that covers both the production planning solution space and the scheduling solution space (although such a space may be complex), and provides a base to effectively combine the solution attempts on both production planning and shop floor scheduling. Tremendous savings in solution efforts would be anticipated when the two functions are successfully integrated. Consequently, Advanced Planning and Scheduling (APS) is the study focus of this project.

1.4 RESEARCH OBJECTIVES

In this project, five distinctive objectives associated with the optimization of Advanced Planning and Scheduling (APS) are to be achieved. First of all, a mathematical model for the Advanced Planning and Scheduling (APS), with the objective of minimizing cost of both production idle time and tardiness or earliness penalty of an order, is to be built. The proposed model will explicitly consider capacity constraints, operation sequences, lead times and due dates in a multi-order environment and generate realistic operation schedules for the shop floor, which will overcome the principal difficulty inherent in the existing MRP procedures.

The second objective is to develop a genetic algorithm to solve the APS problem effectively and efficiently. The APS problem has been proved to be NP-hard [Faa87, Moo03]. Any exact optimization approach is highly impossible to solve this kind of problem efficiently, and heuristic methods are often adopted to tackle this issue. Besides, the GA-based method is to be exploited to find good solutions to the APS problem due to its simplicity and flexibility.

The third objective is to investigate the Dynamic Advanced Planning and Scheduling (DAPS) problem where new orders arrive on a continuous basis, and provide a dynamic strategy to enrich both the mathematical model and the GA approach. Traditional APS problems always include static environment assumptions, such as the availability of all orders. However, in practice, any plans and schedules are always subject to new conditions and constraints due to the highly dynamic environment. In other words, plans and schedules generation is only one aspect of the production process. Dynamic control is equally important for the successful implementation of the APS system.

The fourth objective is to construct a seamless decision support system for APS. Conventional decision support systems in production planning and control are structured on the basis of the hierarchical production planning (HPP) principle. Most of these systems suffer from incompatibility of decisions at different levels. To be effective, an APS-based production decision support system is to be designed.

The fifth objective is to apply the established system to a real-life industrial case. Many manufacturing firms have products with a multi-level structure, and encounter the APS problem. A practical APS problem arising from a light source manufacturer is to be solved to test the developed methods and system.

Overall, the objectives of this research are:

- to establish a mathematical model for APS, with the integration of production planning and scheduling
- to offer a modern heuristic approach for the optimization of APS
- to introduce a Dynamic Advanced Planning and Scheduling (DAPS) mechanism
- to develop an interactive Advanced Planning and Scheduling Decision Support System (APSDSS)
- to apply the designed system to a real case

These five research objectives form a guideline for this thesis.

1.5 SCOPE OF THIS THESIS

This project is mainly devoted to Advanced Planning and Scheduling (APS) so as to minimize the total costs of both production idle time and tardiness or earliness penalty of an order. The structure of the thesis is organized as follows.

In Chapter 2, an extensive literature review is conducted to demonstrate what have been studied on MRP, scheduling as well as APS in the past 50 years. Since the mathematical programming and genetic algorithm are adopted to tackle the APS problem in this project, the fundamental concepts and procedures of the pertinent exact algorithms and heuristic methods are also surveyed.

In Chapter 3, a thorough description of the APS problem under investigation is presented. Then, a Mixed Integer Programming (MIP) model, which succeeds in a system integration of the production planning and shop floor scheduling, is formulated. The integrated model is verified with a commercial software package, CPLEX. Thereafter, the complexity of the APS problem is analyzed.

In Chapter 4, a genetic algorithm for solving the APS problem is proposed. The primary procedure and key issues in the established GA method are elaborately introduced. The performance of the GA-based approach is examined and compared with the optimal solutions gained from Chapter 3.

In Chapter 5, for the Dynamic Advanced Planning and Scheduling (DAPS) problem, both the MIP in Chapter 3 and the GA in Chapter 4 are further extended by incorporating a periodic policy with a frozen interval. The effectiveness of the mechanism in the dynamic environment is tested.

In Chapter 6, a prototype of the Advanced Planning and Scheduling Decision Support System (APSDSS) is constructed. The infrastructural framework and the functional modules included in the system are discussed. An example is illustrated to validate the applicability of the proposed system.

In Chapter 7, a case study for the Advanced Planning and Scheduling (APS) problem in a light source manufacturer is reported. The case problem and the computational results obtained from the developed system are described in detail.

In Chapter 8, the distinctive achievements of this project are provided. Both the academic and industrial contributions of this research are concluded. Finally, some recommendations for future work are suggested.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter is organized as follows. Section 2.2 highlights the evolution of Material Requirements Planning (MRP). In Section 2.3, the basic concepts and pertinent problems in scheduling are discussed with the emphasis on scheduling with earliness and tardiness penalties, job shop scheduling and dynamic scheduling. Afterwards, the studies on the problem of Advanced Planning and Scheduling (APS) as well as some APS systems developed by both academia and commercial companies are elaborately surveyed in Section 2.4. Generally, there are two classes of mathematical techniques to the optimization problems, that is, exact methods and heuristic methods. Section 2.5 provides an overview on the commonly used approaches in these two classes. Finally, some remarks concerning the reviews are summarized in Section 2.6.

2.2 MATERIAL REQUIREMENTS PLANNING (MRP)

In 1969, Orlicky [Orl69] first systematically proposed the Material Requirements Planning (MRP) associated with concepts and methods. Then MRP is widely believed to be a tremendous improvement over older production management systems that were just useful in the make-to-stock environment. Shortly after its development, MRP grew into a closed loop production planning system integrating MPS, MRP and capacity requirement planning (CRP). In such a closed loop system, the capacity check is followed by adjustments to MPS before another execution of MRP, which essentially coordinated a company in terms of its manufacturing planning and control infrastructure. About a decade later, MRP became popular in general manufacturing planning and control strategies.

The successful implementation of MRP systems initiated the development of Manufacturing Resource Planning (MRP-II) during the 1970's. Wight [Wig74] who is widely believed to have invented MRP-II helped make MRP-II logic correct and lead MRP-II to successes. MRP-II was an extended planning system to support cost based functions through inclusion of accounting, finance and other important segments. During the 1980's, MRP-II was further extended by incorporating the sales and marketing planning, which required the involvement of sales and marketing departments in the operation of the system [Sil98]. According to Plossl [Plo94], MRP-II has been the most widely used planning and control system in the manufacturing organizations.

With the rapid advances in information technology, it is crucial for every manufacturing enterprise to have a well designed decision support system. This background gives rise to the use of Enterprise Resource Planning (ERP) on a universal basis. An ERP system is characterized by computerizing an entire business with all functional activities in an enterprise involved. In the past several years, more and more attentions have been directed to ERP systems. Many companies, from small and mid-size to large corporations, have been or are working hard to implement such a system. The market for ERP is growing at a high speed [Ole00, Gar03].

Despite the developments that the manufacturing production planning and control systems have gone through, from MRP to MRP-II, and to ERP, one thing remains constant, that is, MRP is always the backbone of all manufacturing applications. For a broader scope of review on MRP, a number of writings can be taken as excellent references [New74, Cha85, Lan89, Mcc92, Gra93, Sil98]. A classic text by Vollmann et al. [Vol88] placed MRP and the associated planning tools right within the whole manufacturing planning and control (MPC) picture, which was generally considered to be an advanced concept. Meanwhile, numerous books and papers offer practical guidelines for carrying out MRP, MRP-II and ERP, such as White et al. [Whi82], Callerman and Heyl [Cal86], Cerveny and Scott [Cer89], Wallace [Wal90], Luscombe [Lus93] and Alberto [Alb02]. Moreover, many software packages are commercially available, on which Bourke [Bou80] and Schubert [Sch00] provided an overview.

At about the same time, some of the drawbacks inherent in the overall principle of MRP were beginning to be identified and discussed [Ste80, Kru84, Kan88, Mcc90, Spe90, Bak93, Spe98]. In order to avoid only verbally describing the philosophy of MRP, tasks on developing formal mathematical models by means of objective functions and constraints have been undertaken in [Vos03].

2.3 SCHEDULING

Scheduling has been a subject of a significant amount of literature in the operations research field since the early 1950's. In this section, the basic concepts and related topics of scheduling are reviewed.

2.3.1 Overview

Scheduling is a relatively mature research field with numerous research findings, such as White [Whi90], MacCarthy and Liu [Mac93], Blazewicz [Bla96], Pinedo [Pin02], Brucker [Bru04]. The problem of scheduling involves allocating various machines to a number of jobs over periods of time, with the objective of optimizing one or more performance measures. In scheduling theory, characters of the machines can be classified as single machine, parallel machines (possessing the same functions), and dedicated machines (specialized for the completion of certain jobs or operations) [Bla88].

The single machine environment is a simplified and special one of all other more complex machine environments. In single-machine problems, there is only one machine and all jobs must be fulfilled on it. The machine can execute at most one job at any time. Once a job has been processed by the machine, it is completed.

The parallel machines can be divided into three types according to their speeds. Machines are called identical, when all of them have equal job-processing speeds. If the machines are with different speeds, but the speed of each machine is constant and is not job-dependent, then the machines are referred to as uniform. Finally, if the speeds of the machines depend on the particular job that is processed, then they belong to the unrelated ones.

In the case of dedicated machines, there are three modes of processing: flow shop, open shop and job shop, distinguished by how jobs go through machines. In the flow shop environment, there are a defined number of machines in series, and each job has to be processed on each machine while all jobs follow the same route. An open shop is an environment where each job has to visit each of the machines; however, jobs may have different routes among the machines. In the classical job shop, each job has its own pre-defined route to follow, while all machines may not be required by all jobs. Moreover, a special case can exist where a particular job may visit a particular machine more than once in its route.

An example of three jobs processed in a four-machine job shop is given in Figure 2.1. Job 1 consists of three operations, the first on machine B, the second on machine A, and the final operation on machine C. These operations must comply with the order specified by technological requirements. For example, drilling a hole (operation *j*) must precedes tapping it (operation *j* + 1). Job 2 is processed in the order A-B-D, while job 3 follows the route A-C-B-D. The data for this example including the technological sequence of machines for each job with the processing time are listed in Table 2.1.

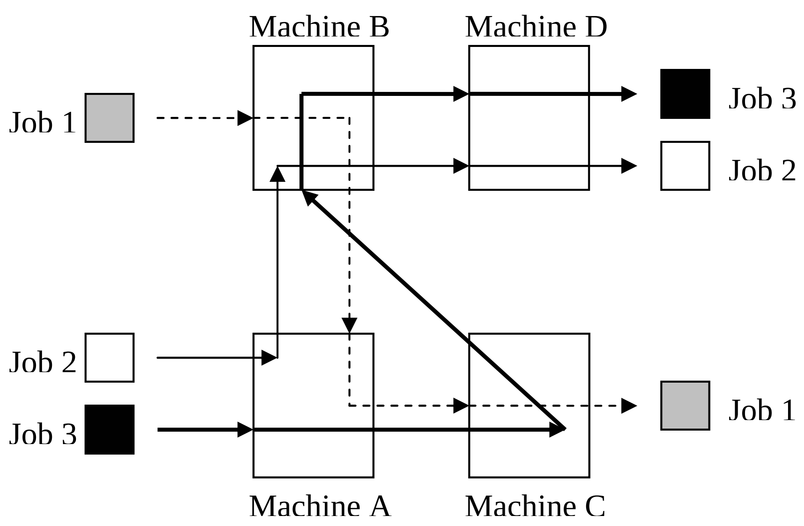


Figure 2.1 A typical job shop

Table 2.1 Data for a job shop example

Job	Machine (Processing Time)			
1	B(1)	A(2)	C(3)	
2	A(4)	B(3)	D(2)	
3	A(2)	C(2)	B(2)	D(2)

A Gantt chart is a pictorial representation of a schedule. It shows time units at the abscissa and machine numbers at the axis of the ordinate. One of the feasible schedules for the above example is represented in the Gantt chart as shown in Figure 2.2. In the figure, each square box illustrates an operation with its left edge placed at its starting time and with its horizontal length indicating the processing time. The makespan of this schedule is 15 time units.

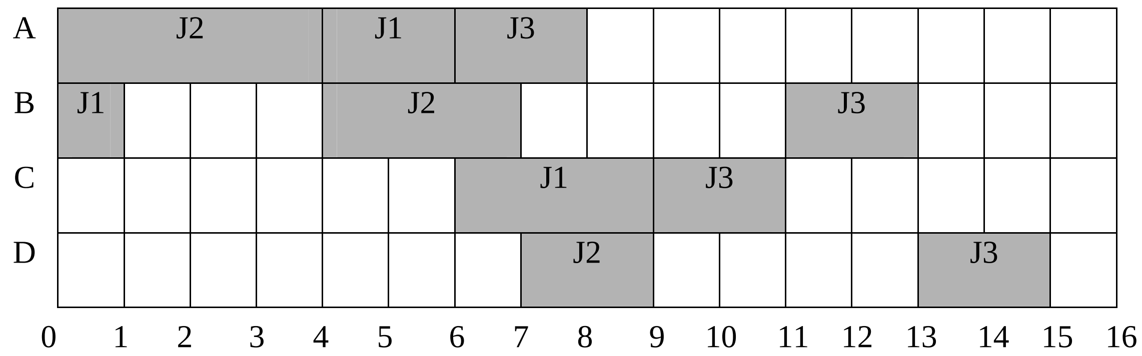


Figure 2.2 The Gantt chart of a job shop example

The main assumptions of job shop scheduling are as follows [Rin76]:

- Machines are always available and never break down.
- Each job is processed by one machine at any time.
- A machine can perform only one operation at any time.
- All operations are not preemptive.
- The processing times are fixed and sequence-independent.
- The processing order of each job is predetermined.

Job shop scheduling has been proven to be one of the most complicated combinatorial problems [Gar76]. For instance, when sequencing 10 jobs (each including 10 operations) on 10 machines in the general job shop, there are $(10!)^{10}$ or more than 10^{65} possible schedules. Although many researchers have invested a great deal of efforts in attacking job shop scheduling, a method of finding an optimal solution effectively and efficiently has not been yielded yet.

2.3.2 Earliness and tardiness penalty

Traditionally, scheduling researches focused on regular measures, which are non-decreasing in job completion times. Most of the studies deal with such performance measures as makespan, maximum lateness, and weighted number of tardy jobs. In line with the trends towards Just-In-Time (JIT) manufacturing strategies, where jobs are encouraged to be completed neither too late nor too early, non-regular scheduling objectives related to earliness and tardiness penalties become more and more popular. This stems from the fact that every job has its due date. If a job is finished before its due date, it has to be held in inventory until its due date and hence incurs an earliness penalty. On the other hand, if a job is finished after the due date, it incurs a tardiness penalty due to customer dissatisfaction, contract penalty, or potential loss of reputation. For a comprehensive review of researches on earliness and tardiness objectives, a number of survey papers have been presented by Sen and Gupta [Sen84], Cheng and Gupta [Che89], Baker and Scudder [Bak90], Gordon et al. [Gor02].

For the single machine case, the literature can be classified into two categories. One category involves a common due date for all jobs, while the other one allows due dates to be different. With respect to the common due date, it is useful to understand that there is a characteristic difference between the solutions when the due date is unrestricted and when it is restricted. The case of an unrestricted common due date for jobs to be scheduled on a single machine is treated by Kanet [Kan81], Hall and Posner [Hal91a]. Algorithms to determine optimal schedules under restricted assumptions about the common due date have been offered by Szwarc [Szw89], Hall et al. [Hal91b], Hoogeveen and van de Velde [Hoo91]. Among others, Bagchi et al. [Bag86, Bag87], De et al. [Dep91, Dep93] have studied the earliness and tardiness scheduling problem on a single machine when the common due date is arbitrary. Raghavachari [Rag86] concluded that for any common due date, the optimal schedule is V-shaped with no inserted idle time between the jobs. One job completes exactly at the due date and one starts at the due date. Jobs that complete before the due date are scheduled according to LPT (Longest Processing Time first rule), and jobs that complete after the due date are in SPT (Shortest Processing Time first rule) sequence. In the second important category, where jobs have distinct due dates, the problems become more complicated, since the optimal schedule may contain idle times between the processing of consecutive jobs. Garey et al. [Gar88] proved that even with symmetric earliness and tardiness penalties, the single machine scheduling problem is NP-hard. Solutions to scheduling with distinct due dates for each job have been proposed by Abdul-Razaq and Potts [Abd88], Peng and Morton [Pen89], Yano and Kim [Yan91], Nandkeolyar et al. [Nan93], Sridharan and Zhou [Sri96].

For parallel machines, Sundararaghavan and Ahmed [Sun84], Hall [Hal86], Li and Cheng [Lic94], Webster [Web97], Mosheiov and Shadmon [Mos01] have investigated the earliness and tardiness performance in the identical setting. Emmons [Emm87], Sivrikaya-Serifoglu and Ulusoy [Siv99] provided efficient approaches to minimize the total weighted earliness and tardiness on both identical and uniform parallel machines. When parallel machines are unrelated, Kubiak et al. [Kub90] proposed an algorithm to reduce the weighted sum of absolute deviation problem to a corresponding transportation problem.

There are also some results on open, flow and job shop scheduling with earliness and tardiness penalties, such as Sarper [Sar95], Sung and Min [Sun01b], Mosheiov [Mos03].

2.3.3 Job shop scheduling

The specific area to be reviewed in this section is the job shop scheduling. Although it is simple to state, the job shop scheduling problem is one of the hardest combinatorial optimization problems. Even among the NP-hard problems, it appears to be the most difficult one [Law82]. This is one of the reasons why the problem has been so broadly explored.

The job shop scheduling problem was initially tackled by exact methods, such as integer programming, dynamic programming, etc. All of these methods require at least partial enumeration of possible solutions. Due to the fact that the number of possible solutions grows exponentially as the problem size increases slightly, these exact methods become very computationally intensive for even easy job shop scheduling problems, and in most cases they have not facilitated solutions for hard problems. However, in practice, it is critical to find acceptable solutions within a limited amount of time, especially for large scale problems. Heuristics, including rule-based heuristics and meta-heuristics, are such approximation methods for overcoming this problem. Heuristic methods usually generate satisfactory schedules in a reasonable computation time, but it is generally very difficult to evaluate the optimality of the solutions. The above two categories of solution methods for job shop scheduling are shown in Figure 2.3.

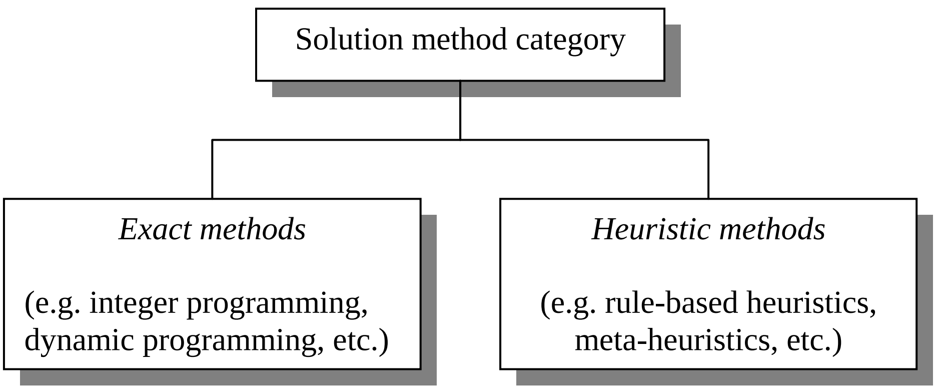


Figure 2.3 Two categories of solution methods for job shop scheduling

2.3.3.1 Exact methods

Bowman [Bow59] was one of the pioneers to address job shop scheduling problems in the integer programming form, and found that a relatively simple problem involving only three jobs and four machines would need up to 600 variables and many more constraints. Manne [Man60] extended this earlier work with the use of a mixed integer programming formulation and established a more compact mathematical model. Balas [Bal67, Bal69] proposed integer programming methods for job shop scheduling based on the strategy of finding a mini-maximal path in a disjunctive graph, and exploited the growing power of the computer to store the pertinent data of nodes, which theoretically allows job shop scheduling problems to be solved optimally. A large number of integer programming formulations and procedures for job shop scheduling then followed [Fis73, Lag77, Fis83, Nem88, Rog91].

There are also numerous researchers concentrating on only adopting the branch and bound methods to attack job shop scheduling problems without presenting the detailed mathematical formulations. In such branch and bound techniques, schedules are derived through direct enumeration. Rather than enumerating all possible solutions, the procedure only detects the branch of the enumeration tree that has attributes associated with the optimal solution. On the basis of the gradually refined bounds, branches of the tree are examined and some are eliminated from further consideration. One of the first major published studies of employing branch and bound techniques for job shop scheduling problems was carried out by Brooks and White [Bro65]. Later, Florian et al. [Flo71] developed an algorithm based on graph-theoretical representation of the job shop scheduling problem, where all operations with an unscheduled predecessor were candidates for branching. In 1989, Carlier and Pinson [Car89] constructed a branch and bound methodology to cope with the famous 10×10 instance formulated by Muth and Thompson [Mut63]. Applegate and Cook [App91], Brucker and Jurisch [Bru93] and Brucker et al. [Bru94] also concentrated on establishing lower bounds and using branch and bound approaches to reduce the number of schedule enumeration in the job shop environment. Although a considerable amount of interests and researches have focused on taking advantage of branch and bound algorithms for settling the issue of job shop scheduling, these algorithms are still impractical for larger size problems due to the computational complexity.

In the 1960's and 1970's, Szwarc [Szw60] and others implemented dynamic programming for the job shop scheduling problems; however, such methods have been competitive with the integer programming as well as branch and bound methods mostly for a restricted class of problems.

2.3.3.2 Heuristic methods

There are two kinds of heuristic methods that have been comprehensively applied to the job shop scheduling problems: rule-based heuristics and meta-heuristics.

As early as 1960, Giffler and Thompson [Gif60] presented a number of heuristic rules. Then Panwalker and Iskander [Pan77] listed more than 100 dispatching rules and offered a classification scheme. In 1996, Chang et al. [Cha96] proposed 3750 test problems and utilized them to rank 42 dispatching rules according to the individual performance and the number of best solutions. In general, dispatching rules are myopic when creating production schedules, and as a result do not have satisfactory performance [Bla82, Gup89, Hau89]. In order to improve the inadequate performance of dispatching rules, efforts have been made to overcome the drawbacks by developing composite rules. Anderson and Nyirenda [And90] came up with two new combination rules, in which priorities of jobs are determined according to both process times and due dates. Raghu and Rajendran [Rag93] proposed efficient approaches to incorporate existing heuristic rules for scheduling in the job shop environment. Ramaswamy et al. [Ram94] also established a response surface modeling methodology for the integration of dispatching rules based on shop conditions. While combinations of priority rules perform better than the individual ones, their myopic natures have not been essentially changed.

In comparison with rule-based heuristics, meta-heuristics including simulated annealing (SA), tabu search (TS), genetic algorithms (GAs) have been quite successfully applied to job shop scheduling.

Matsuó et al. [Mat88], and Van Laarhoven et al. [Van92] provided simulated annealing (SA) based heuristics to solve the job shop scheduling problems. Computational experiments showed that compared with dispatching rules, SA-based methods always produced better production performances.

Widmer [Wid91], Dell'Amico and Trubian [Del93], Taillard [Tai94], Barnes and Chambers [Bar95], and Nowicki and Smutmicki [Now96] proposed tabu search techniques to attack job shop problems, and concluded that tabu search worked well in both solution quality and computational time according to their numerical results.

Early in 1985, Davis [Dav85] first offered a genetic algorithm (GA) based technique to address job shop scheduling. Later, Nakano and Yamada [Nak91] established a conventional genetic algorithm using a binary genotype to represent each solution, and applied this approach to three job shop benchmarks. Meantime, Falkenauer and Bouffouix [Fal91] designed a GA method to optimize the job shop problem with release times, due dates and a specially defined cost function. To improve the previous results, DellaCroce et al. [Del95] introduced preference rules into the genetic algorithm, which was shown to be competitive with simulated annealing [Van92] and tabu search [Del93]. While most researchers proposed the literal permutation ordering encoding mechanisms for job shop problems [Bag91, Nak91, Par92], Bean [Bea94] presented a general genetic algorithm, based on random keys representation technique, to explore a wide variety of optimization problems including job shop scheduling. The main advantage of the random keys encoding is that it guarantees feasibility of all offspring. Recent surveys of Cheng et al. [Che96, Che99], Proudlove et al. [Pro98], Ponnambalam et al. [Pon01] and Aytug et al. [Ayt03] contain much more extensive and thorough discussions of GAs for job shop scheduling problems.

2.3.4 Dynamic scheduling

Much of the research in scheduling is based on the assumption that the manufacturing environment is static, which rarely holds in real situations. In practice, some unexpected events, such as the arrival of new orders, machine breakdowns, etc., may arise and disrupt the manufacturing system. This leads to the study of dynamic scheduling [Mat93, Suh98, Cow02].

Currently, more studies have considered scheduling problems in the dynamic condition. In 1991, Bean et al. [Bea91] provided a heuristic method by reconstructing part of the schedule, when a disruption occurs, to match up with the pre-schedule at some future time. Also, match-up approaches with minimum schedule changes were adopted for responding to disturbances [Sun01a]. Li et al. [Lir93] constructed an iterative two-step procedure to dynamically create product schedules in response to unexpected events that take place on the shop floor. For dynamic scheduling in flexible flow shops, Chang and Liao [Cha94] developed efficient algorithms based on Lagrangian relaxation to cope with changes in production environment. Jain and Elmaraghy [Jai97] built genetic algorithms to deal with different types of disruptions in the flexible manufacturing systems.

Dynamic scheduling with only taking into account the arrival of new orders has been attempted by some researchers. Church and Uzsoy [Chu92] addressed the problem of production systems in the presence of dynamic job arrivals, and compared the performances of periodic and event-driven rescheduling policies. Unal et al. [Una97] modeled a single machine in the face of newly arrived jobs with part-type dependent setup times, and provided efficient algorithms to insert the new jobs into the existing schedule so as to minimize the disruption of the jobs in the system. Bierwirth and Mattfeld [Bie99] described genetic algorithms for job shop scheduling and rescheduling, and demonstrated that their approaches produced far better results than priority-rule based methods. Hall and Potts [Hal04] suggested efficient solution procedures to insert the new jobs into the existing schedule under the single machine condition, where the disruption was modeled either as a constraint or as a component in the objective. When jobs arrive at the job shop on a continuous basis, Rangsaritratsamee et al. [Ran04] proposed a genetic local search methodology that simultaneously considers efficiency and stability through a multi-objective measure.

There are also approaches to handle machine breakdowns on the shop floor. Wu et al. [Wus93] presented heuristics for solving the one-machine dynamic problem subject to a machine breakdown, and the solutions showed to effectively increase the schedule stability with little sacrifice in efficiency. To settle machine breakdowns in job shops, Leon [Leo94] took a game-theoretic view, and came up with a heuristic search methodology. Jensen [Jen03a] proposed a genetic algorithm to find robust and flexible job shop schedules, and demonstrated that these schedules performed significantly better than ordinary ones after a machine breakdown.

General references on dynamic scheduling include [Nof91, Sur93, Sab00, Mar01, Vie03]. One of the influential reviews by Vieira et al. [Vie03] presented standard definitions of rescheduling strategies, policies as well as methods, and also described a framework for better understanding rescheduling research.

2.4 ADVANCED PLANNING AND SCHEDULING (APS)

Advanced Planning and Scheduling (APS) aims at integrating production planning and shop floor scheduling, and deals with effectively allocating production resources to complete the multi-level products so that production constraints are satisfied and production objectives are met. The Advanced Planning and Scheduling (APS) problem has received tremendous attentions in recent years [Lee02, Rom02, Moo04, Zen05].

2.4.1 Approaches for APS

Studies on Advanced Planning and Scheduling (APS) have focused primarily on the development of heuristic approaches.

An early paper by Hastings et al. [Has82] used a form of forward loading to plan and schedule jobs on the available capacity. Bahl and Ritzman [Bah84] provided an integrated model and a heuristic solution procedure which decomposes the overall problem into smaller sub-problems and solves them in an iterative fashion. Faaland and Schmitt [Faa87] devised a two-phase heuristic technique to generate feasible production schedules by solving a sequence of maximum flow problems. Sum and Hill [Sum93] proposed a new framework for manufacturing planning and scheduling systems. The framework formulates an iterative process between the order network and the operation network to determine order sizes and operation schedules. Agrawal et al. [Agr96] exploited a precedence network to represent the precedence relationships among items and then developed a heuristic approach to generate near-optimal schedules, employing critical path concept. Taal and Wortmann [Taa97] described an intuitive planning method that integrates MRP with several finite capacity planning, based on scheduling techniques. Reeja and Rajendran [Ree00a, Ree00b] developed new dispatching rules and compared with the best existing rules based on various measures of performance related to flow time and tardiness through an exhaustive simulation. In order to provide a broad perspective to the planning and scheduling of the multi-level jobs (customer orders) on the shop floor, Yeh [Yeh00] presented a job-oriented finite capacity scheduling system, which has a basic similarity to the manual loading method of the Gantt chart. Riane et al. [Ria01] adopted a hierarchical approach with an iterative link between the planning module and the scheduling module for designing an integrated planning and control system.

There appears to be scant research on presenting exact mathematical formulations and methods to settle the issue.

Lasserre [Las92] proposed a decomposition approach to solve the APS problem. His approach alternated between solving a planning problem with a fixed sequence of products on the machines and a job shop scheduling problem for a fixed choice of the production plan. Dillenberger et al. [Dil94] established a Mixed Integer Programming (MIP) model for resource allocation and multi-period production planning and scheduling. However, these authors did not take into account the precedence relationships among the items. Kolish [Kol00] introduced an MIP model to address the APS problem in which different customer orders need the same part types, and proposed a two-level, backward oriented, top-down approach to solve it. A major limitation of this work is that it considered only a product with two levels, while a real product usually has many levels. Rom et al. [Rom02] exploited a resource constrained project scheduling model to augment MRP by incorporating precedence constraints as well as capacity constraints and utilizing variable lead time lengths. The efficacy of this approach was tested against the traditional MRP, while assuming that MRP provides feasible production plans. More recently, Moon et al. [Moo04] suggested an advanced planning and scheduling model which integrates capacity constraints and precedence constraints to minimize the makespan only.

In the general planning and scheduling problem, the most common objective is the minimization of the makespan. Due to the growing interests in JIT production strategy in industry, planning and scheduling with earliness-tardiness penalties has received attentions increasingly [Che89, Bak90, Gor02].

In 1994, Czerwinski and Luh [Cze94] chose an improved Lagrangian relaxation technique to address the APS problem with the objective function containing quadratic earliness and tardiness penalties, but the solution oscillation has not been completely eliminated, which slows convergence of the algorithm. Wang [Wan95] presented a mathematical description of earliness-tardiness production systems with capacity constraints, and developed two approaches to solve it. One was to translate the problem into a linear programming model by means of mathematic deduction and solve it by a relaxation procedure. The other was to develop a heuristic algorithm and combine it with the branch-and-bound method to quicken the optimization process. However, only mass and one-of-a-kind product manufacturing systems, usually single-machine and flow shop environment, were studied. Although Kim and Kim [Kim96] explored a short-term production problem for products with multi-level structures, with the objective of minimizing the weighted sum of tardiness and earliness of the items, their research was based on group technology (GT) assumption which simplified the integrated problem. For the planning and scheduling of complex products with multiple resource constraints and deep product structure, Pongcharoen et al. [Pon02, Pon04] developed a genetic algorithm-based tool which includes a repair process to rectify infeasible schedules. The tool takes account of the requirements to minimize the penalties caused by both early supply and late delivery of the products. Unfortunately, no comparisons of the results obtained from the algorithm with the optimal solutions have been offered.

2.4.2 APS systems

The past decade has seen a great revolution in computer and information technology. Nowadays, the powerful hardware and the advanced computer programming languages support to embody optimization models and methods into Advanced Planning and Scheduling (APS) systems. This has attracted considerable attentions from both academia and commercial companies.

Taal and Wortmann [Taa97] proposed a consistent production system that incorporates several different planning and scheduling techniques. McKay and Wiers [Mck03] presented a design of an integrated decision support system for planning and control tasks in a focused factory. More recently, a software system architecture, referred to as collaborative agents for production activities, is constructed by Nishioka [Nis04].

Meantime, many software vendors provide a broad range of APS software systems. A brief description of some of these systems is given below.

SAP, a German company, is always active in the APS market. The Advanced Planner and Optimizer (APO) was originally sold as an independent APS software, and now is a part of the “mySAP Supply Chain Management”. APO is a fast and efficient decision support tool, and involves a variety of optimization methods. Production plans and schedules are created using intelligent heuristic algorithms in combination with mathematical computation.

i2 Technologies, established in 1988 and based in Dallas, releases an APS package called Factory Planner. Factory Planner offers detailed visibility over the production plans and schedules, and its graphical interface is powerful and intuitive. It generates plans and schedules in a heuristically forward and backward way while considering material and capacity constraints simultaneously. Furthermore, it provides what-is and what-if functions, and adds dynamic and interactive simulation and impact analyses on material planning, capacity planning and scheduling.

Preactor International, headquartered in the UK, is one of the companies specializing in APS software. Its flagship product, Preactor APS, includes the features of materials and capacity synchronization, real-time order promising, and multi-site planning and scheduling. Preactor APS is supplied with a simulation based APS engine with built in standard optimization rules, such as Forward, Backward, "Theory of Constraints" type rules, or any combination.

Elliott [Ell00] and Stadtler and Kilger [Sta05] provided reviews on APS systems.

2.5 OPTIMIZATION METHODOLOGY

Generally, there are two classes of mathematical approaches to combinatorial optimization problems, that is, exact methods and heuristic methods. The approaches in the first class yield the optimal solutions, but the computational requirements grow exponentially as the problem size increases. The approaches in the second class require modest computation, but do not guarantee that the solutions are optimal. In this part, the commonly used approaches in these two classes are presented.

2.5.1 Exact methods

In this section, attention is confined to linear programming and integer programming, since the Advanced Planning and Scheduling (APS) problem to be investigated in this project will be formulated on the basis of these two types of methods.

2.5.1.1 Linear programming

Linear programming (LP) is a basic mathematical modeling technique designed to optimize the usage of the limited resources. In LP models, both the objective function and the constraints only involve linear expressions, and the decision variables are continuous. LP is the basis for the development of other types of operations research models, including integer programming, nonlinear programming, and so on.

The simplex algorithm, introduced by G. B. Dantzig, is the general method for solving linear programming models. Because of its high computational efficiency, LP models are given much attention in practice. The simplex algorithm is developed from the idea that if the optimal solution of a linear programming model exists, it is attained at a basic feasible solution (a corner point of the solution space). A sequence of basic feasible solutions is generated by the simplex algorithm to improve the objective value sequentially. To achieve this, the elemental pivoting operation is iteratively applied by exchanging basic variables and non-basic ones. The process terminates when the optimal solution is obtained, no feasible solution is detected, or unbounded solution is found to exist [Tah03].

Another technique for solving linear programs is the interior point algorithm, which was developed by N. Karmarkar [Kar84]. While the simplex algorithm searches for the optimal solution by traversing the extreme points of the feasible region, the interior point algorithm approaches the optimum from the strict interior of the feasible space. The interior point algorithm has a theoretical importance because it provides a polynomial bound on the computational efforts to solve a problem, and it also has a practical significance because it produces solutions to many industrial problems that previously were intractable [Car01, Jen03b].

2.5.1.2 Integer programming

In the environment of job-shop, many decision variables actually make sense only if they have integer values. If requiring integer values is the only way in which a problem deviates from a linear programming formulation, then it is an integer programming (IP). The mathematical model for IP is the linear programming model with one additional restriction that the variables must have integer values. Generally, there are three types of integer programming. Firstly, if all of the variables are required to have integer values, the model is referred to as pure integer programming. Secondly, if only some of the variables are required to have integer values, the model is referred to as mixed integer programming. Thirdly, if all the variables are required to be either 0 or 1, the model is referred to as binary integer programming.

There are wide practical applications for integer programming. However, a good algorithm for solving IP problems has not been developed [Wil99]. Generally, the IP algorithms are based on exploiting the tremendous computational success of LP. The strategy of these algorithms involves three steps:

Step 1: Relax the solution space of the IP by replacing any integer variable with the continuous value, and deleting the integer restrictions on all the integer variables. The result of the relaxation is a regular LP.

Step 2: Solve the LP and identify its continuous optimum.

Step 3: Starting from the continuous optimum point, add special constraints that iteratively modify the LP solution space in a manner that will eventually render an optimum extreme point that satisfies the integer requirements.

Two general methods have been developed for generating the special constraints referred to step 3, that is, the branch-and-bound (B & B) method and the cutting plane method.

2.5.1.2.1 The branch-and-bound method

The most popular method for IP algorithms is the branch-and-bound (B & B) technique and the related ideas to implicitly enumerate the feasible integer solutions. The basic concept underlying the branch-and-bound technique is to divide and conquer. Since the original "large" problem is too difficult to be solved directly, it is divided into smaller and smaller sub-problems until these sub-problems can be conquered. The dividing (branching) is done by partitioning the entire set of feasible solutions into smaller and smaller subsets. The conquering is done partially by bounding how good the best solution in the subset can be and then discarding the subset if its bound indicates that it cannot contain an optimal solution for the original problem. The process terminates when an integer solution is found or the original model is shown to be infeasible [Cas02].

2.5.1.2.2 The cutting plane method

The other methodology for solving IP problems is the cutting plane method. Firstly, the algorithm finds the optimal tableau for the IP's linear programming relaxation. If all variables in the optimal solution assume integer values, an optimal solution to the IP has been found. Otherwise, a constraint in the LP relaxation optimal tableau whose right-hand side has the fractional part closest to 1/2. This constraint will be used to generate a cut, which is added to the tableau. The process is continued until a solution in which all variables are integers is obtained. This will be an optimal solution to the IP [Wil99, Jen03b].

2.5.2 Heuristic methods

There are some heuristic procedures that have been termed meta-heuristics, including simulated annealing (SA), tabu search (TS), genetic algorithms (GAs). The word "meta" comes from the fact that these heuristics work in an iterative master process that guides and modifies the operations of subordinate heuristics by combining intelligence, biological evolution, neural systems, and statistical mechanics. Although they do not guarantee optimal solutions to optimization problems, they are able to find near-optimal solutions efficiently by exploring and exploiting the search spaces using learning strategies. In this section, three meta-heuristics, namely simulated annealing (SA), tabu search (TS), genetic algorithms (GAs), will be introduced briefly, since these three approaches are very general and have been applied to a wide variety of optimization problems with great successes.

2.5.2.1 Simulated annealing

Simulated annealing (SA), first proposed by Kirkpatrick et al. in 1983 [Kir83], is an optimization technique analogizing the thermodynamics process of annealing in physics. SA starts with an initial solution and repeatedly generates a new solution from the neighborhood. If the new solution is better, it is accepted as the current solution. If it is worse, the new solution may be accepted and the acceptance depends on the acceptance function, the temperature parameter, and the difference in the objective values of the two solutions. Initially, the temperature parameter is large, and the new solution is accepted quite frequently. As the algorithm progresses, the temperature is slowly reduced, lowering the probability that the acceptance function will accept a worse solution. Figure 2.4 summarizes the general SA procedure [Pha00].

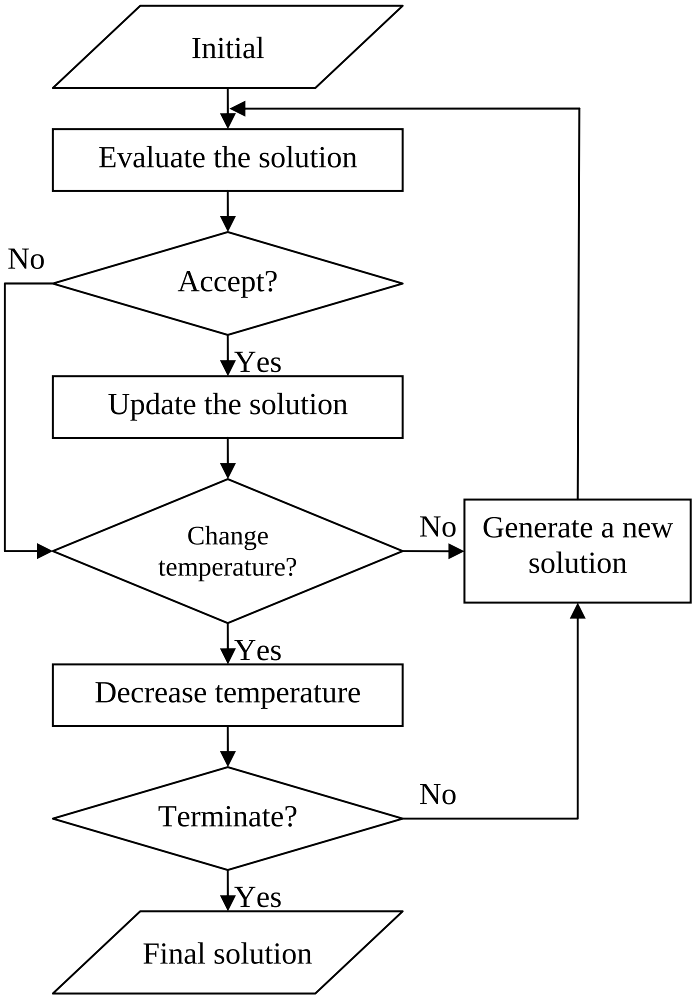


Figure 2.4 Flowchart of a standard simulated annealing method

2.5.2.2 Tabu search

Tabu search (TS), primarily suggested by Glover and Hansen in 1986, is a strategy for solving combinatorial optimization problems by using especially designed memory structures to escape from the local optima. Like all other neighborhood search techniques, TS starts with an initial solution and evaluates all its neighborhood solutions. The best solution in the neighborhood will be selected to replace the current solution even though it may not be better than the current one. In some occasions, the best neighborhood solution may not be selected if the solution is in the tabu memory lists. There are two classes of memory lists: recency (short term) memory and frequency (long term) memory. Both memories are responsible for recording the history of the search and storing the forbidden moves (attributes). This mechanism attempts to prevent cycling behavior and to force the search to new solution regions. If the selection is forbidden (tabu), the second best neighborhood solution will be chosen as the candidate to update the current one. Also there is such a case that a tabu move may be accepted if certain criteria, called aspiration criteria, are met, such as the solution obtained by the application of the move being better than the best solution found so far. Then the newly updated solution is set as the primal for the next iteration. The search process continues until the stopping rule is satisfied. The flowchart of a standard TS method is illustrated in Figure 2.5 [Glo93, Glo97].

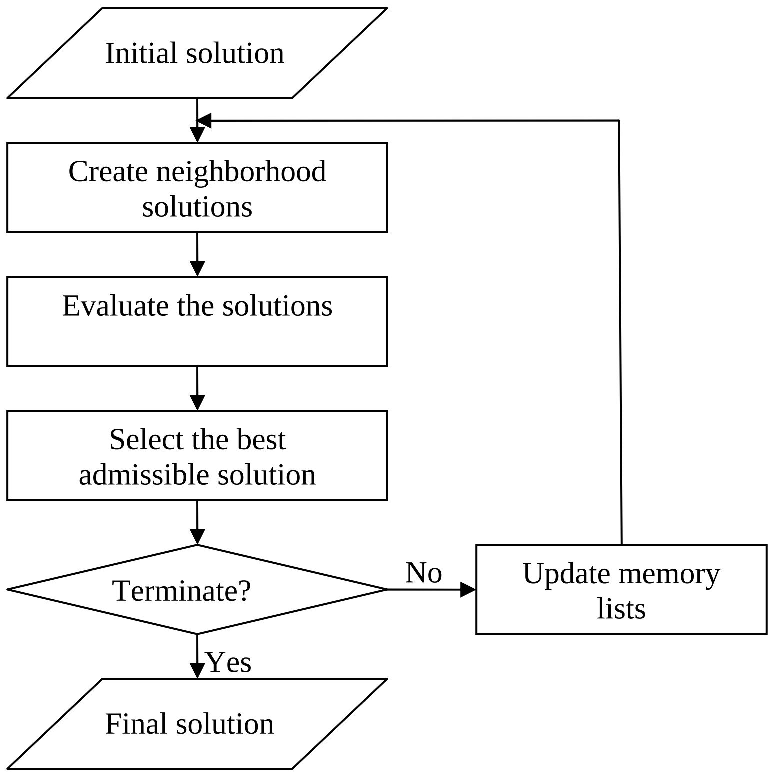


Figure 2.5 Flowchart of a standard tabu search method

2.5.2.3 Genetic algorithms

Genetic algorithms (GAs) were originally developed by Holland and his associates in the 1960's. Essentially, the search methods a GA employs are inspired by the natural evolution process. Different from SA and TS, GA starts with an initial set of random solutions called population. Each potential solution in the search space is represented by the form of a chromosome. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated using some measures of fitness. The fitter the chromosomes, the higher the probabilities of being selected to perform genetic operations. There are two important genetic operators: crossover and mutation. The crossover operator serves to generate new offspring by combining existing parents (two chromosomes in the population). The mutation operator is used to randomly modify the chromosomes. Naturally, the crossover operator speeds up the process to reach better solutions, while the mutation operator explores a wider search space to avoid being trapped in local optima. Then a new generation is formed by selecting some of the parents and offspring according to their fitness and rejecting the others to keep the population size constant. When some termination condition is met, the algorithm converges to the best chromosome. The whole procedure is shown in Figure 2.6 [Dav91, Mit96, Cha99, Gen00].

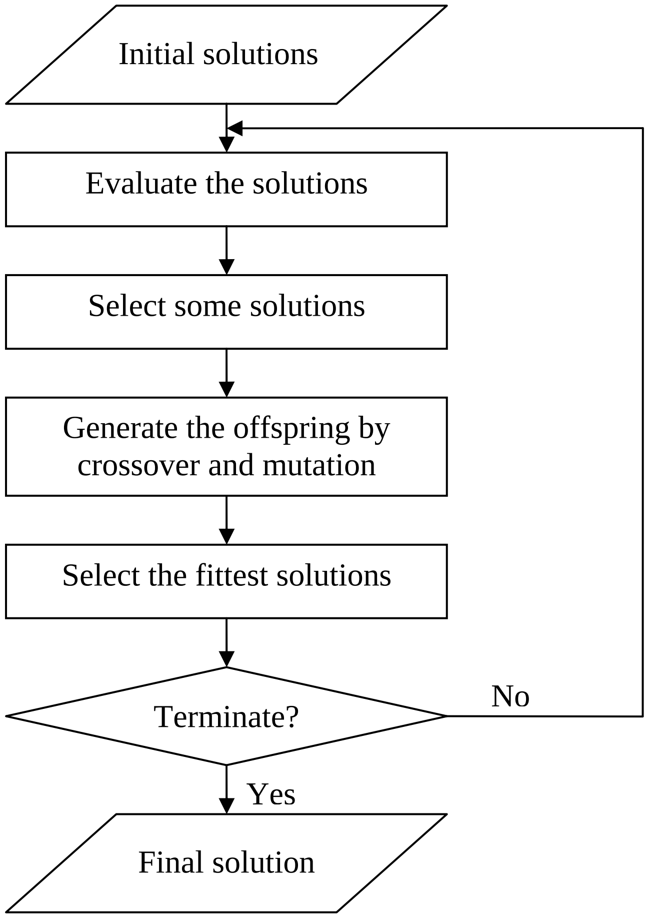


Figure 2.6 Flowchart of standard genetic algorithms

2.6 SUMMARY

In this chapter, an extensive literature review on MRP, scheduling as well as APS has been conducted. Since mathematical programming and genetic algorithm are to be adopted for solving the APS problem in this project, the fundamental concepts and procedures of the pertinent exact algorithms and heuristic methods have been also surveyed. Some remarks concerning the reviews can be itemized as follows.

1. Conventional planning and scheduling are considered hierarchically and separately, which creates many problems on the shop floor for production [Bil83, Har85, Sum93, Taa97, Rom02, Vos03]. Unquestionably, production planning and scheduling are closely interrelated, and they should be integrated together to generate realistic production schedules for the shop, which leads to the problem of Advanced Planning and Scheduling (APS). In recent years, the APS problem has received tremendous attentions, and many achievements have been obtained [Yeh00, Lee02, Moo04, Zen05]. However, the studies on APS have focused primarily on the development of heuristic approaches. There appears to be scant research on presenting exact mathematical formulations and methods to settle the issue, and besides, most of them [Las92, Dil94, Kol00, Moo04] are based on simplified assumptions. Hence, it is necessary to establish a complete mathematical model for the APS problem.
2. The Advanced Planning and Scheduling (APS) problem has been proved to be NP-hard [Faa87, Moo03]. Any exact optimization approach is highly impossible to solve this kind of problems efficiently, and heuristic methods are often adopted to tackle this issue. Currently, due to their simplicity and flexibility, genetic algorithms (GAs) have been widely applied to find good solutions to the APS problems [Kim96, Lee02, Pon04, Cha05]. In this project, the GA technique will also be selected as a tool to attack the Advanced Planning and Scheduling (APS) problem.

3. Referring to Section 2.3.4, nowadays, more studies have considered planning and scheduling problems in the dynamic environment [Sur93, Sab00, Vie03]. Nevertheless, most of the research efforts have concentrated on one machine, flow shop, and job shop situations, assuming operations are performed in series. In summary, there is clearly a need for introducing dynamic mechanism into Advanced Planning and Scheduling (APS).
4. Although both academia and commercial companies have invested great efforts in developing decision support for Advanced Planning and Scheduling (APS), most of the researches have restricted themselves to embed trial-and-error methods in their computer-based systems. Better production plans and schedules can be generated by decision support tools with the employment of intelligent heuristic approaches, such as genetic algorithms (GAs).
5. While the past decade has seen a substantial literature on Advanced Planning and Scheduling (APS), very few results are available on real world cases. Czerwinski and Luh [Cze94] explored the APS problem in Pratt & Whitney, a manufacturer of turbine engines, and proposed an improved Lagrangian relaxation technique for solving it. A major limitation of the developed method is that solution oscillation was not completely eliminated. For the planning and scheduling of complex products with multiple resource constraints and deep product structure in a company that produces capital goods, Pongcharoen et al. [Pon04] developed a genetic algorithm-based tool which includes a repair process to rectify infeasible schedules. Unfortunately, the tested problem and the results obtained from the algorithm have not been

described in detail. Therefore, it is concluded that the APS problem originating from the real industries has not been adequately studied and analyzed.

The next chapter will present a thorough description of the APS problem under investigation. Then, a mathematical programming model, which succeeds in a system integration of the production planning and shop floor scheduling, will be formulated. The integrated model will be verified and illustrated with two examples. Thereafter, the complexity analysis of the APS problem will be made.

CHAPTER 3

A MATHEMATICAL PROGRAMMING MODEL FOR ADVANCED PLANNING AND SCHEDULING (APS)

3.1 INTRODUCTION

Mathematical modeling is to describe a problem in a mathematical way, and is a significant activity for better understanding and analyzing the problem. In such a mathematical way, much of the ambiguity and imprecision verbal communication can be overcome. Meanwhile, an effective mathematical model can help to capture the essential features of the problem and provide considerable insights into the problem. Furthermore, by solving the model, the best solution to the problem can be obtained, which is the exact optimization approach. Although heuristic methods and simulation are alternative techniques for optimization, nobody can tell the quality of the solutions yielded by these techniques without comparing with the optimal solutions.

In this chapter, a mathematical programming model is formulated for Advanced Planning and Scheduling (APS), which succeeds in a system integration of production planning and shop floor scheduling. The proposed model explicitly considers capacity constraints, operation sequences, lead times and due dates in a multi-order environment. The objective of the model is to seek the minimum cost of both production idle time and tardiness or earliness penalty of an order. The output of the model is production schedules with starting time and finish time for each item of an order. The integrated model is verified with a commercial software package and illustrated with two examples.

This chapter is organized as follows: Section 3.2 introduces the problem under investigation. In Section 3.3, a Mixed Integer Programming (MIP) model for Advanced Planning and Scheduling (APS) is developed. To verify the model, two examples, a simple one and a representative one modified from the literature, are illustrated in Section 3.4. Section 3.5 analyzes the complexity of the APS problem. Finally, Section 3.6 concludes the chapter with a summary.

3.2 PROBLEM DESCRIPTION

We consider a production planning and scheduling problem for products with multi-level structures. A simple example of the product structure is shown in Figure 3.1 (a). The root node represents the final product (F1) which is composed of one subassembly (S1) and two components (C1s). Meanwhile, the subassembly (S1) is made up of components C2 and C3. This multi-level product structure is typical in industry and is often more complex than this example. In such a structure, items (final products, subassemblies and components) have precedence constraints among them, that is, before processing parent items, their child items should be completed first. Here, a child item represents a lower level item that belongs to a parent item. Each of these items requires various operations on eligible machines which are continuously available for production. Therefore, constraints on the time capacity should be considered. Several additional assumptions are made here. The product structure, orders of items and their due dates are known in advance and similarly for processing time of operations. A lot-for-lot strategy is adopted for making items, while the setup times (including the transfer times between operations) are negligible or are included in the processing times. If there are several operations needed for an item, this item is divided further into several items to reflect the operations. For example, if the item C1 has three operations (OP1, OP2, OP3) to process, C1 can be further divided into three child items: C1OP1, C1OP2, C1OP3. In this case, C1OP3 is a child item of C1OP2, C1OP2 is a child item of C1OP1, and C1OP1 is a child item of F1, as depicted in Figure 3.1 (b). Without loss of generality, we can only employ final products, subassemblies, and components in the problem. Furthermore, each operation can be processed on at most one machine at a given time and is non-preemptive. A machine can perform one operation at a time and only works for eight hours a day.

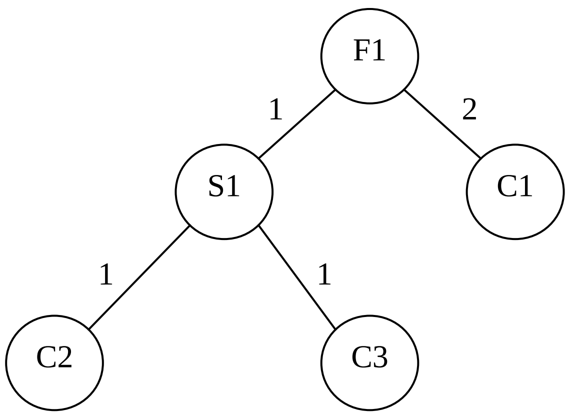


Figure 3.1 (a) A simple example of a product structure

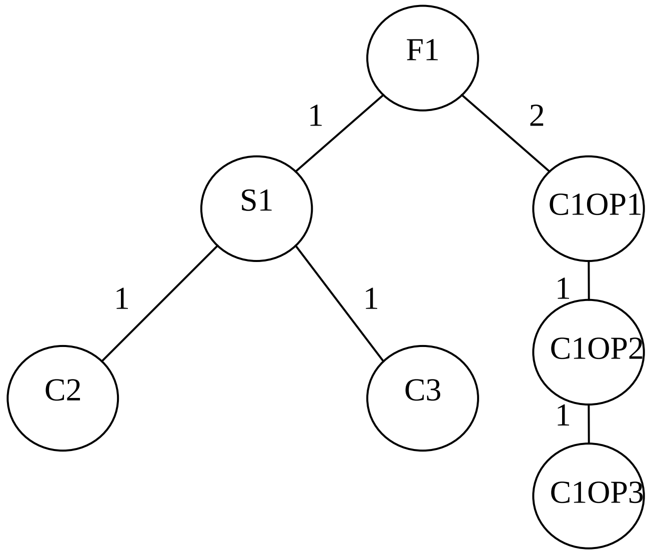


Figure 3.1 (b) A simple example of a product structure

Our problem is to find an optimal schedule for the orders such that both production idle time and penalties on tardiness and earliness are minimized. Minimizing production idle time is equivalent to minimizing flow time or maximizing machine utilization. In addition, production idle time is chosen as the objective to be reduced because it is able to reflect two focuses in shops: manufacturing lead time and WIP (work-in-process) inventory level. Another objective of the problem is to find a schedule with all jobs completed as close to their due dates as possible, which is predicated on the fact that either early or late delivery of an order results in an increase in the costs. If an order is finished before its due date, it has to be held in inventory until its due date and hence incurs an earliness penalty. On the other hand, if an order is finished after the due date, it incurs a tardiness penalty due to customer dissatisfaction, contract penalty, or potential loss of reputation.

A schematic diagram of the Advanced Planning and Scheduling (APS) problem is depicted in Figure 3.2. Hence, the APS problem is characterized by satisfying customer requests and reducing WIP inventory, subject to multiple resources capacity constraints and complex precedence constraints among operations.

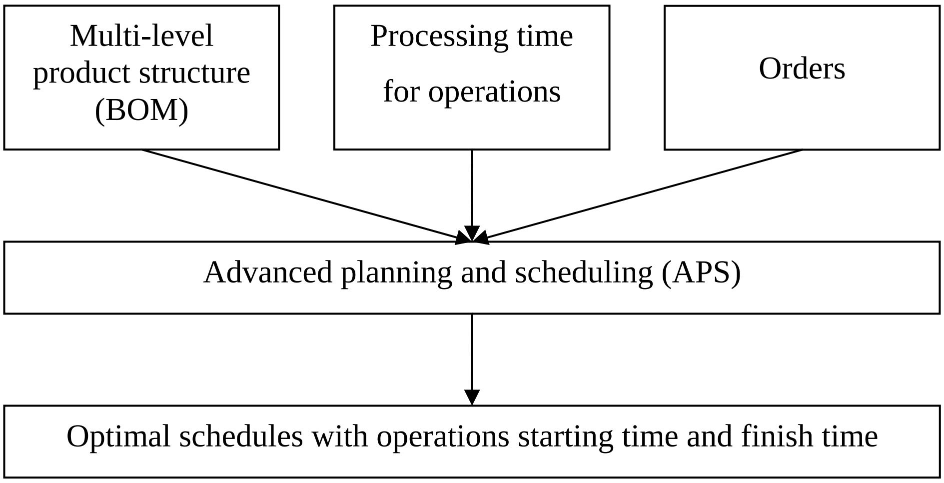


Figure 3.2 A schematic diagram of the APS problem

3.3 A MODEL FORMULATION FOR THE APS PROBLEM

The proposed APS model is primarily based on the production strategy, and explicitly takes into account capacity constraints of the manufacturing system, operation sequence among items, as well as lead times and due dates of products in a multi-order environment. The objective of the model is to seek the minimum cost of both production idle time and tardiness or earliness penalty of an order. The output of the model is production schedules with starting time and finish time for each item of an order.

3.3.1 Notation

In order to formulate the APS model, the following parameters and variables are introduced:

Parameters:

n	Number of orders ($i, j = 1, \dots, n$)
m	Number of machines ($k, l = 1, \dots, m$)
O_i	Order i
M_k	Machine k
P_i	Final product of order O_i ($p, q = 1, \dots, P_i$)
Q_i	Quantity of order O_i
N_{ip}	Number of item p needed for one final product P_i
t_{ipk}	Processing time required by item p of order O_i on machine M_k
r_k	Ready time of machine M_k
DD_i	Due date of order O_i
I	Cost of idle time per hour
TC	Cost of tardy orders per day per job
EC	Cost of early orders per day per job
α	A large positive number
$A(p)$	Set of child items of item p

B	Set of item p, where $A(p) = \varnothing$

Variables:

C_{max}	Production makespan
S_{ipk}	Production start time of item p of order O_i on machine M_k
C_i	Production completion time of order O_i
L_i	Number of tardy days (real number) for order O_i
E_i	Number of early days (real number) for order O_i
L_i^I	Number of tardy days (integer) for order O_i
E_i^I	Number of early days (integer) for order O_i
X_{ipjqk}	1 if item p of order O_i precedes item q of order O_j on machine M_k; and 0 otherwise

3.3.2 The model

Now, we have the following Mixed Integer Programming (MIP) model for the Advanced Planning and Scheduling (APS) problem.

$$\text{MIN}\left\{ I(mC_{\max} - \sum_{i=1}^{n}\sum_{p=1}^{P_i} t_{ipk}\cdot N_{ip}\cdot Q_i - \sum_{k=1}^{m} r_k) + \sum_{i=1}^{n}\left[TC\times L_i^I + EC\times E_i^I\right]\right\} \qquad (3.1)$$

subject to:

$$C_i \le C_{\max} \qquad \forall i. \qquad (3.2)$$

$$S_{ipk} \ge r_k \qquad p \in B,\ \forall i, k. \qquad (3.3)$$

$$S_{ipk} - S_{iql} \ge t_{iql} \cdot N_{iq} \cdot Q_i \qquad q \in A(p),\ \forall i,\ k,\ l. \qquad (3.4)$$

$$S_{iP_ik} + t_{iP_ik} \cdot Q_i = C_i \qquad \forall i, k. \qquad (3.5)$$

$$S_{ipk} \ge S_{jqk} + t_{jqk} \cdot N_{jq} \cdot Q_j - \alpha (X_{ipjqk}) \qquad \forall i, j, k, p, q. \qquad (3.6)$$

$$X_{ipjqk} + X_{jqipk} = 1 \qquad \forall i, j, k, p, q. \qquad (3.7)$$

$$\frac{C_i}{8} - DD_i \le L_i \qquad \forall i \qquad (3.8)$$

$$DD_i - \frac{C_i}{8} \le E_i \qquad \forall i \qquad (3.9)$$

$$L_i^I \ge L_i \qquad \forall i \qquad (3.10)$$

$$E_i^I \ge E_i - 0.99 \qquad \forall i \qquad (3.11)$$

$$C_{\max} \ge 0 \qquad (3.12)$$

$$S_{ipk} \ge 0 \qquad \forall i, p, k. \qquad (3.13)$$

$$C_i, L_i, E_i \ge 0 \qquad \forall i \qquad (3.14)$$

$$L_i^I, E_i^I \ge 0 \text{ and integer} \qquad \forall i \qquad (3.15)$$

$$X_{ipjqk} \in \{0,1\} \qquad \forall i, p, j, q, k. \qquad (3.16)$$

Since there are real variables, binary variables as well as integer variables in the objective function and the constraints, the formulation is a mixed integer linear programming model.

The objective (3.1) which contains two parts is to minimize the production idle time, order tardiness and earliness in order to minimize the machine idle costs and the penalty costs. In the first part, it is assumed that no other employment than orders O_1, … , O_n are available for the machines from the release time until C_{max}, then idle time costs are expressed as $I(mC_{\max} - \sum_{i=1}^{n}\sum_{p=1}^{P_i} t_{ipk} \cdot N_{ip} \cdot Q_i - \sum_{k=1}^{m} r_k)$. Since the sum of the processing times ($\sum_{i=1}^{n}\sum_{p=1}^{P_i} t_{ipk} \cdot N_{ip} \cdot Q_i$) and the sum of the ready times ($\sum_{k=1}^{m} r_k$) are constant, minimizing production idle time ($mC_{\max} - \sum_{i=1}^{n}\sum_{p=1}^{P_i} t_{ipk} \cdot N_{ip} \cdot Q_i - \sum_{k=1}^{m} r_k$) is equivalent to minimizing flow time or maximizing machine utilization. The second part represents the penalty costs for all orders, including the tardiness costs ($\sum_{i=1}^{n} TC \times L_i^I$) of those orders that are completed after their due dates, and the earliness costs ($\sum_{i=1}^{n} EC \times E_i^I$) of the orders that are fulfilled before their due dates. The objective function is to minimize the total excessive costs involved in the problem.

Constraints (3.2) show that the completion time of any order (C_i) has to be less than or equal to production makespan (C_{max}).

Constraints (3.3) ensure that the start time of the components (S_{ipk}) should be greater than or equal to the machine ready time (r_k).

Precedence constraints among the items are satisfied in constraints (3.4). If item q is a child item of item p, the start time of the parent item p (S_{ipk}) minus the start time of the child item q (S_{iql}) should be larger than or equal to the processing time of the child item q ($t_{iql} \cdot N_{iq} \cdot Q_i$).

The completion time of any order (C_i) is given by the start time of the final product (S_{iP_ik}) plus the processing time of the final product ($t_{iP_ik} \cdot Q_i$). The expression is specified in constraints (3.5).

Constraints (3.6) and (3.7) require that no two operations can be processed simultaneously on the same machine. The formulation of this expression uses the type of 0–1 variables, which are the pairwise precedence variables X_{ipjqk}, referred to as the "disjunctive graph" formulation. In this formulation, the start time of an operation on a machine must follow the completion time of any other operation that is picked to precede it. For example, assume that both item p of order i and item q of order j should be processed on machine k. When $X_{ipjqk} = 0$ and $X_{jqipk} = 1$, constraint (3.6) requires that the start time of item p of order i (S_{ipk}) is larger than or equal to the start time of item q of order j (S_{jqk}) plus the total processing time of item q of order j ($t_{jqk} \cdot N_{jq} \cdot Q_j$), that is, item q of order j precedes item p of order i on machine k. On the other hand, when X_{ipjqk} equals to 1, constraint (3.6) could always be ensured, since α represents a large positive number.

Constraints (3.8) and (3.9) define tardiness and earliness of orders, respectively. Meanwhile, a unit conversion is conducted in order to transform all the hours into days. To do this, the completion time (C_i) is just divided by 8 in the equations because it is assumed that there are eight hours per day.

Expressions (3.10) and (3.11) are to convert the value of tardiness and earliness to an integer when the penalty costs are in the unit of days. That is to say, if an order is finished 4 hours or 0.5 day before its due date, we believe there is no earliness under such a situation. On the other hand, if an order is finished 4 hours or 0.5 day after its due date, one day tardiness occurs. In view of calculation precision, the number 0.99 is introduced to help conduct the conversion.

Finally, constraints (3.12)-(3.16) define the non-negative variables, the non-negative integer variables, and the binary variables, respectively.

3.4 NUMERICAL RESULTS

In order to demonstrate how the APS problem is formulated using the proposed model and verify the model, two examples are illustrated and solved adopting the software CPLEX on a personal computer. One example is a simple one and consists of two orders, two machines as well as a four-level product structure, which is then denoted example $2 \times 2 \times 4$. The other example, which is modified from the literature, deals with five orders, six machines and three different products among which the most complex one has a five-level structure; accordingly, it is called example $5 \times 6 \times 5$. By solving the established Mixed Integer Programming (MIP) model, the optimal production schedules to these problems can be found.

3.4.1 A simple example

For an illustrative example, consider the four-level product structure shown in Figure 3.1 (b). A customer may order the final product F1, as well as some major components, like S1 and C1. Meantime, component C1 has three operations (OP1, OP2, OP3) to process, and then C1 is further divided into three child items: C1OP1, C1OP2 and C1OP3. Two machines, with 8 hours available per day, are eligible to process the items (Table 3.1). The ready times of M1 and M2 are Hour 5 and Hour 2.5, respectively. There are two orders, one requiring 10 Product F1s with due date Day 4 and the other requiring 15 Product S1s with due date Day 3. The following data are useful for calculating the costs.

- Cost of idle time at $50 per hour.
- Cost of tardiness at $250 per day per order.
- Cost of earliness at $50 per day per order.

Table 3.1 Machine processing time for the items in the simple example (2 × 2 × 4)

Items	Machine Number	Processing Time (hours)
F1	M1	0.7
S1	M1	0.5
C1OP1	M2	0.2
C1OP2	M2	0.2
C1OP3	M2	0.1
C2	M2	0.1
C3	M2	0.2

Based on the above data, the whole MIP model for the simple example (2 × 2 × 4) can be formulated as follows.

Minimize

100Cmax - 2225

+250LI1+250LI2

+50EI1+50EI2

Subject to

(3.2):

cons1: C1-Cmax<=0

cons2: C2-Cmax<=0

(3.3):

cons3: S111>=5

cons4: S121>=5

cons5: S211>=5

cons6: S132>=2.5

cons7: S142>=2.5

cons8: S152>=2.5

cons9: S162>=2.5

cons10: S172>=2.5

cons11: S222>=2.5

cons12: S232>=2.5

(3.4):

cons13: S111-S121>=5.000000

cons14: S111-S132>=4.000000

cons15: S121-S162>=1.000000

cons16: S121-S172>=2.000000

cons17: S132-S142>=4.000000

cons18: S142-S152>=2.000000

cons19: S211-S222>=1.500000

cons20: S211-S232>=3.000000

(3.5):

cons21: C1-S111=7

cons22: C2-S211=7.5

(3.6) and (3.7):

cons23: S111-S121+999X11121>=5.000000

cons24: S121-S111+999X12111>=7.000000

cons25: X11121+X12111=1

cons26: S111-S211+999X11211>=7.500000

cons27: S211-S111+999X21111>=7.000000

cons28: X11211+X21111=1

cons29: S121-S211+999X12211>=7.500000

cons30: S211-S121+999X21121>=5.000000

cons31: X12211+X21121=1

cons32: S132-S142+999X13142>=4.000000

cons33: S142-S132+999X14132>=4.000000

cons34: X13142+X14132=1

cons35: S132-S152+999X13152>=2.000000

cons36: S152-S132+999X15132>=4.000000

cons37: X13152+X15132=1

cons38: S132-S162+999X13162>=1.000000

cons39: S162-S132+999X16132>=4.000000

cons40: X13162+X16132=1

cons41: S132-S172+999X13172>=2.000000

cons42: S172-S132+999X17132>=4.000000

cons43: X13172+X17132=1

cons44: S132-S222+999X13222>=1.500000

cons45: S222-S132+999X22132>=4.000000

cons46: X13222+X22132=1

cons47: S132-S232+999X13232>=3.000000

cons48: S232-S132+999X23132>=4.000000

cons49: X13232+X23132=1

cons50: S142-S152+999X14152>=2.000000

cons51: S152-S142+999X15142>=4.000000

cons52: X14152+X15142=1

cons53: S142-S162+999X14162>=1.000000

cons54: S162-S142+999X16142>=4.000000

cons55: X14162+X16142=1

cons56: S142-S172+999X14172>=2.000000

cons57: S172-S142+999X17142>=4.000000

cons58: X14172+X17142=1

cons59: S142-S222+999X14222>=1.500000

cons60: S222-S142+999X22142>=4.000000

cons61: X14222+X22142=1

cons62: S142-S232+999X14232>=3.000000

cons63: S232-S142+999X23142>=4.000000

cons64: X14232+X23142=1

cons65: S152-S162+999X15162>=1.000000

cons66: S162-S152+999X16152>=2.000000

cons67: X15162+X16152=1

cons68: S152-S172+999X15172>=2.000000

cons69: S172-S152+999X17152>=2.000000

cons70: X15172+X17152=1

cons71: S152-S222+999X15222>=1.500000

cons72: S222-S152+999X22152>=2.000000

cons73: X15222+X22152=1

cons74: S152-S232+999X15232>=3.000000

cons75: S232-S152+999X23152>=2.000000

cons76: X15232+X23152=1

cons77: S162-S172+999X16172>=2.000000

cons78: S172-S162+999X17162>=1.000000

cons79: X16172+X17162=1

cons80: S162-S222+999X16222>=1.500000

cons81: S222-S162+999X22162>=1.000000

cons82: X16222+X22162=1

cons83: S162-S232+999X16232>=3.000000

cons84: S232-S162+999X23162>=1.000000

cons85: X16232+X23162=1

cons86: S172-S222+999X17222>=1.500000

cons87: S222-S172+999X22172>=2.000000

cons88: X17222+X22172=1

cons89: S172-S232+999X17232>=3.000000

cons90: S232-S172+999X23172>=2.000000

cons91: X17232+X23172=1

cons92: S222-S232+999X22232>=3.000000

cons93: S232-S222+999X23222>=1.500000

cons94: X22232+X23222=1

(3.8) and (3.9):

cons95: 0.125C1-L1<=4

cons96: E1+0.125C1>=4

cons97: 0.125C2-L2<=3

cons98: E2+0.125C2>=3

(3.10) and (3.11):

cons99: L1-LI1<=0

cons100: E1-EI1<=0.99

cons101: L2-LI2<=0

cons102: E2-EI2<=0.99

Bounds

LI1 free LI2 free EI1 free EI2 free

Integers

EI1 LI1 EI2 LI2

X11121 X12111
X11211 X21111
X12211 X21121
X13142 X14132
X13152 X15132
X13162 X16132
X13172 X17132
X13222 X22132

X13232 X23132
X14152 X15142
X14162 X16142
X14172 X17142
X14222 X22142
X14232 X23142
X15162 X16152
X15172 X17152

X15222 X22152
X15232 X23152
X16172 X17162
X16222 X22162
X16232 X23162
X17222 X22172
X17232 X23172
X22232 X23222

End

3.4.2 The optimal solution to the simple example

To solve the simple APS example, the mixed integer programming formulation can be input into CPLEX, a commercial package. Meanwhile, it should be noticed that 999 is taken as the large positive number α for the convenience in the CPLEX input process.

The detailed results generated from the software CPLEX 9.1 on a personal computer with Pentium 2.66 GHz CPU and 512 MB RAM are listed in the following. The important data extracted from the optimal results are summarized in Table 3.2. For this simple example, the developed MIP model requires 102 constraints and 69 variables, where 52 are integers, and CPLEX takes only 0.06 second to reach the optimal solution with the total costs of 475.

Integer optimal
Objective = 4.750000000e+002
Solution time = 0.06 sec.
Iterations = 535
Nodes = 130

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	27.000000	S152	7.000000
C1	27.000000	S162	2.500000
C2	19.500000	S172	5.000000
S111	20.000000	S222	3.500000
S121	7.000000	S232	9.000000
S211	12.000000	X12111	1.000000
S132	16.000000	X21111	1.000000
S142	12.000000	X12211	1.000000

X14132	1.000000	X17152	1.000000
X15132	1.000000	X22152	1.000000
X16132	1.000000	X15232	1.000000
X17132	1.000000	X16172	1.000000
X22132	1.000000	X16222	1.000000
X23132	1.000000	X16232	1.000000
X15142	1.000000	X22172	1.000000
X16142	1.000000	X17232	1.000000
X17142	1.000000	X22232	1.000000
X22142	1.000000	E1	0.990000
X23142	1.000000	E2	0.990000
X16152	1.000000		

All other variables are zero.

Table 3.2 Optimal results of the simple example (2 × 2 × 4)

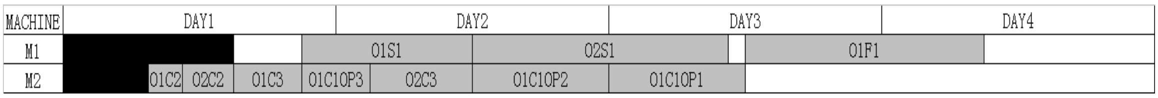
Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
102	69	52	27	0.06

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
0	0	0	0	475

The results are graphically represented in the Gantt chart as shown in Figure 3.3. For convenience, item *p* of order O_i is denoted *Oip*. From the Gantt chart, it is easy to recognize that the optimal makespan for the illustrative example is 27 hours. Both Order 1 and 2 are fulfilled on time.

The results show that the developed model can generate the optimal schedule with operation starting time and finish time, which is more realistic for the shop floor.

Figure 3.3 Optimal results of the example (2 × 2 × 4) in the form of Gantt chart



3.4.3 A representative example and its optimal solution

In this example, three typical product structures (Figure 3.4) were chosen: flat, tall and complex, which were defined in [Fry89]. A wide variety of products could be characterized by these three structures. Besides, it should be noted that S7, C2, and C3 are common items. Subassembly S7 is common to subassemblies S3 and S4. Component C2 is a common child of final products F1, F2 and F3. Component C3 is shared by final product F1 and subassembly S2. Meantime, component C13 has two operations (OP1, OP2) to process, and then C13 is further divided into two child items: C13OP1 and C13OP2. A customer may order the final products F1, F2 and F3, and also some major components, like S2 and C3. It was observed that the number of machines was not a significant factor in the shop and the consideration of shop size with more than six machines would suffice [Ree00a]. In this example, six machines, with 8 hours available per day, are eligible to process the items (Table 3.3).

Moreover, M5 and M6 are responsible for assembling, while the other machines deal with the components. The ready times of these machines are Hour 1, Hour 2, Hour 3, Hour 3, Hour 2 and Hour 1, respectively. There are five orders: 5 Product F1s with due date Day 6, 5 Product F2s with due date Day 7, 10 Product F3s with due date Day 14, 10 Product S2s with due date Day 3, and 30 Product C3s with due date Day 1. The penalty rates are as follows: cost of idle time at \$50 per hour, cost of tardiness at \$250 per day per order, and cost of earliness at \$50 per day per order.

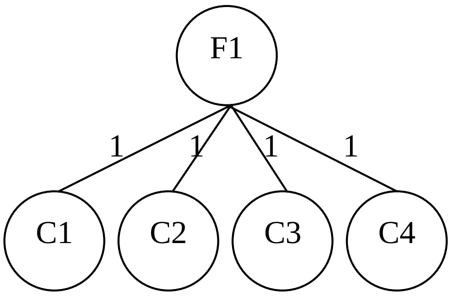


Figure 3.4 (a) The product structure of F1 in example 5 × 6 × 5

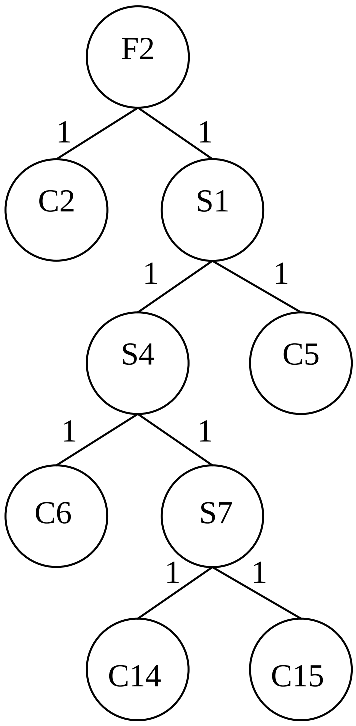


Figure 3.4 (b) The product structure of F2 in example 5 × 6 × 5

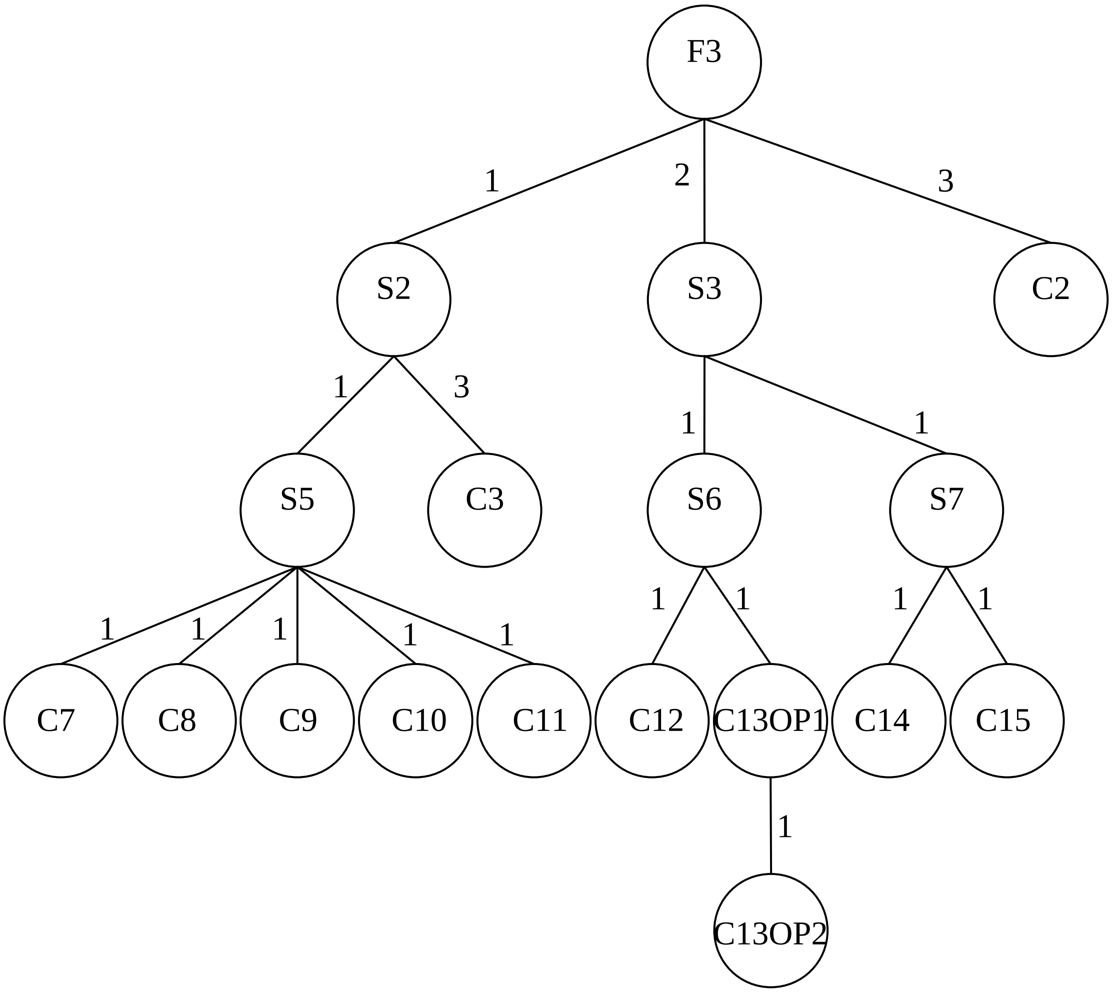


Figure 3.4 (c) The product structure of F3 in example 5 × 6 × 5

Table 3.3 Machine processing time for the items in the example (5 × 6 × 5)

Items	Machine number	Processing time (hours)
F1	M6	0.7
F2	M6	0.6
F3	M6	0.7
S1	M6	0.5
S2	M6	0.6
S3	M5	0.5
S4	M5	0.5
S5	M6	0.6
S6	M5	0.3
S7	M5	0.3
C1	M3	0.2
C2	M1	0.1
C3	M1	0.1
C4	M2	0.2
C5	M2	0.2
C6	M3	0.4
C7	M3	0.2
C8	M3	0.2
C9	M3	0.1
C10	M2	0.2
C11	M2	0.1
C12	M2	0.3
C13OP1	M4	0.2
C13OP2	M4	0.2
C14	M4	0.1
C15	M4	0.1

The problem formulation for the representative example (5 × 6 × 5) is listed in Appendix I, and the corresponding optimal solution produced from the CPLEX 9.1 on a personal computer with Pentium 2.66 GHz CPU and 512 MB RAM is attached in Appendix II. The large positive number α also takes the value 999 in the CPLEX input process. The important data extracted from the optimal results are summarized in Table 3.4. For the illustrative example, the developed MIP model requires 463 constraints and 313 variables, where 256 are integers, and CPLEX takes about 20 hours to reach the optimal solution with the total costs of 7575.

Table 3.4 Optimal results of the example (5 × 6 × 5)

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
463	313	256	45	69814.53

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
0	13	0	650	7575

The optimal results in the form of Gantt chart are illustrated in Figure 3.5. For convenience, item *p* of order O_i is denoted *Oip*. The Gantt chart clearly indicates that the optimal makespan for the illustrative example is 45 hours. Order 1, 2 and 3 are completed before their due dates, while the other two orders are satisfied on time.

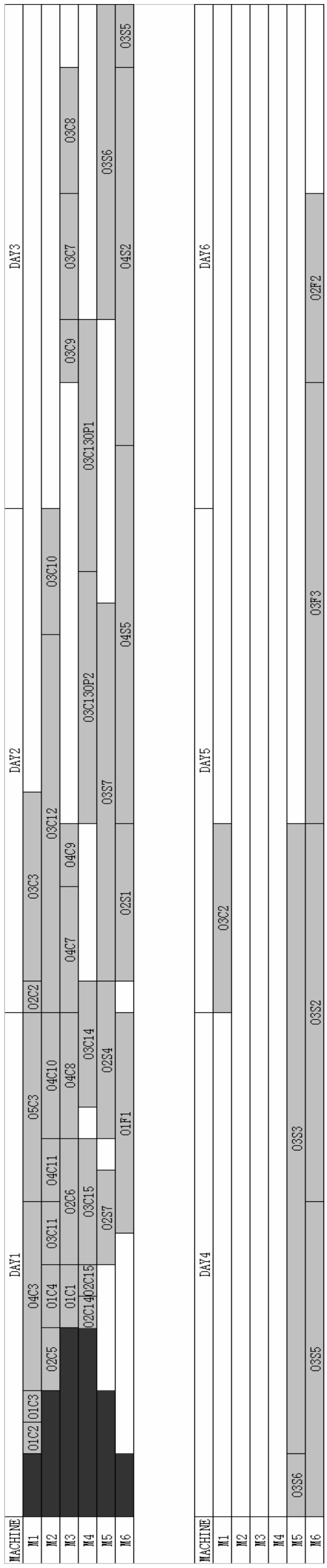


Figure 3.5 Optimal results of the example (5 × 6 × 5) in the form of Gantt chart

The results further confirm that the formulated model can generate the optimal schedule with operation starting time and finish time, which overcomes the principal difficulty inherent in MRP and is more realistic for the shop floor.

3.5 COMPLEXITY ANALYSIS

Early in 1976, Garey and Johnson [Gar76] have proved that the general job shop scheduling problem is NP-hard. On the basis of group technology (GT) assumption, Kim and Kim [Kim96] aggregated operations into two basic ones, machining and assembly, and assumed that the manufacturing system is composed of a machining shop and an assembly shop in an APS problem. Such an APS problem, with the objective of minimizing the weighted sum of tardiness and earliness of the items, has been shown to be a hard combinatorial optimization problem. More recently, Moon et al. [Moo04] suggested an advanced planning and scheduling model to minimize the makespan only, and concluded that it is among the class of NP-hard problems. The APS problem addressed in this project is much more complicated than these. It is reasonable to believe that the APS problem is strongly NP-hard.

Besides, for the simple APS example ($2 \times 2 \times 4$), the developed MIP model requires 102 constraints and 69 variables, where 52 are integers, and it takes only 0.06 second to reach the optimal solution. By contrast, in the representative APS example ($5 \times 6 \times 5$), there are 463 constraints and 313 variables including 256 integer ones, and the computational time is about 20 hours. The computing complexity also demonstrates that the APS problem is among the NP-hard class and the solution time will grow exponentially as the problem size increases.

3.6 SUMMARY

In this chapter, a complete mathematical programming model for the Advanced Planning and Scheduling (APS) problem has been formulated. The insights gained from the model can be concluded in the following:

1. The Advanced Planning and Scheduling (APS) problem aims to synthesize production planning and shop floor scheduling, and is characterized by satisfying customer requests and reducing WIP (work-in-process) inventory, subject to multiple resources capacity constraints and complex precedence constraints among operations.
2. The objective of the APS problem is to find an optimal schedule for the orders such that both production idle time and penalties on tardiness and earliness are minimized. Minimizing production idle time is equivalent to minimizing flow time or maximizing machine utilization. Another objective of the APS problem is to derive a schedule with all orders completed as close to their due date as possible, which fits to the JIT production control policy where either early or late delivery of an order results in an increase in the production costs.
3. A Mixed Integer Programming (MIP) model has been developed for the APS problem, which succeeds in a system integration of production planning and shop floor scheduling.

4. The proposed model explicitly considers capacity constraints, operation sequences, lead times and due dates in a multi-order environment and generates useful operation schedules for the shop floor, which overcomes the principal difficulty inherent in the existing MRP procedures.

5. Two examples, a simple one and a representative one modified from the literature, are elaborately illustrated to verify the model and solved adopting the software CPLEX on a personal computer. The numerical results have demonstrated the optimality and effectiveness of the established model.

6. For the simple example (2 × 2 × 4), the developed MIP model requires 102 constraints and 69 variables, where 52 are integers, and it takes only 0.06 second to reach the optimal solution. By contrast, in the representative example (5 × 6 ×5), there are 463 constraints and 313 variables including 256 integer ones, and the computational time is about 20 hours. It is reasonable to believe that the APS problem is strongly NP-hard and the solution time will grow exponentially as the problem size increases.

We have investigated the APS problem in this chapter and built an MIP model for the problem. Since the APS problem is NP-hard, a heuristic should be exploited to solve the problem in a reasonable time. In the next chapter, a genetic algorithm (GA) for efficiently settling the Advanced Planning and Scheduling (APS) problem will be proposed. The primary procedure and key issues in the established GA method will be introduced in detail. The performance of the GA-based approach will be examined and compared with the optimal solutions gained from the mathematical model.

CHAPTER 4

A GENETIC ALGORITHM FOR ADVANCED PLANNING AND SCHEDULING (APS)

4.1 INTRODUCTION

In Chapter 3, a mathematical programming model for Advanced Planning and Scheduling (APS) has been built. Although the optimal production plan and schedule for the APS problem can be yielded by the established Mixed Integer Programming (MIP) model, the APS problem is among the class of theoretically difficult problems (NP-hard), and the computational efforts will become extremely intensive when finding the global optimum to a large problem. Thus, meta-heuristics should be used to solve the problem more efficiently.

Genetic algorithms (GAs), invented by Holland and his associates in the 1960's, are random search techniques for seeking "optimal" or "near-optimal" solutions within complex search spaces. Essentially, the search methods a GA employs are inspired by the natural evolution process, and implement a "survival-of-the-fittest" strategy. GAs differ from conventional optimization approaches and have many attractive features. They conduct a multidirectional search using a population of solutions rather than a single solution. Moreover, no information on differentiability, convexity or other mathematical properties is required by GAs [Dav91, Mit96, Cha99, Gen00]. Due to their simplicity and flexibility, genetic algorithms (GAs) have been successfully applied to a wide variety of optimization problems, including the Advanced Planning and Scheduling (APS) problems [Kim96, Lee02, Pon04]. Also, GAs have demonstrated to perform better in production planning and scheduling problems than other heuristic methods, such as simulated annealing (SA), tabu search (TS) and the shifting bottleneck procedure [Del95, Ree95]. In this chapter, a GA-based method for APS, with the objective of minimizing cost of both production idle time and tardiness or earliness penalty of an order, is to be developed.

This chapter is organized as follows: Section 4.2 provides a GA-based approach for efficiently solving the Advanced Planning and Scheduling (APS) problem. In Section 4.3, the same representative example as in Chapter 3 (refer to Section 3.4.3) is used to evaluate the algorithm. In Section 4.4, the results of the GA are compared with the optimal solutions gained from the mathematical model. Finally, Section 4.5 concludes the chapter.

4.2 A GENETIC ALGORITHM FOR THE APS PROBLEM

The key issues in developing a GA-based approach are the encoding scheme of the solution, the initialization of the population, the evaluation measurement, reproduction, crossover, mutation, and selection strategy. In this section, these issues are described in detail to present a GA-based approach for the APS problem.

4.2.1 Encoding

Our encoding scheme is based on the concept of random keys suggested by Bean [Bea94]. This scheme encodes a solution with a string of random numbers. Each item in the product structure has one random number generated from the range

[0, 1]. These random numbers denote the priorities of the items, while a smaller value represents the higher priority. Table 4.1 shows an example of the encoding scheme for the product whose structure is given in Figure 3.1 (a) and replicated in Figure 4.1. The random key encoding has the advantage that it eliminates the offspring feasibility problem and is robust to problem structures [Bea94].

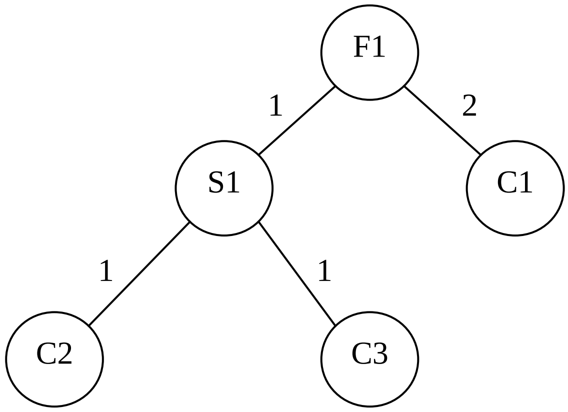


Figure 4.1 A simple example of a product structure

Table 4.1 A simple example of the encoding scheme

Items	Random number	Machine number	Processing time (hours)
F1	0.14	M1	7
S1	0.27	M1	8
C1	0.65	M2	5
C2	0.31	M2	2
C3	0.79	M2	4

A feasible schedule can be derived from the product structure and the string of random numbers. From the random numbers, the priorities of items are obtained first. An item with the highest priority is selected among the available items which have no child items or whose child items have all been scheduled. The starting time of the selected item is determined considering both the finish time of its child item and the available time of its processing machine. This procedure continues until all items are allocated. For instance, in Figure 4.1 and Table 4.1, when C1, C2 and C3 have no child simultaneously, C2 is selected for the operation sequence because its priority number is 0.31, higher than those of C1 and C3. After selecting C2, only C1 and C3 are available, and C1 is chosen for the next operation because it has a higher priority than C3. In the same manner, the operation sequence C2-C1-C3-S1-F1 is determined. Suppose that the corresponding machines and processing times are shown in Table 4.1. A feasible schedule is then obtained in Figure 4.2(a). Finally, since our objective is to minimize both production flow time and penalty costs including earliness and tardiness, unforced idle time may be introduced into the schedule. To achieve this, for the early orders in the obtained schedule, the final products are moved to their latest possible time, based on the fact that only the completion time of the final product affects the earliness and tardiness of an order. If the overall objective improves, the new schedule updates the former one; otherwise, the former one remains. For the above simple example, it is assumed that the order of F1 has the due date Day 5, and then the schedule in Figure 4.2(a) is early. According to the heuristic rules, only F1 is moved to its latest possible time, that is, F1 is completed at Hour $(DD_{F1} - 1) \times 8 + 1$. Here, DD_{F1} represents the due date of F1.

Since the unit of production idle time is in hours and earliness and tardiness are measured in days, a new schedule with no earliness and tardiness could be obtained in Figure 4.2(b). These two schedules in Figure 4.2(a) and 4.2(b) are assessed on the basis of the objective function, and the one with fewer costs would be chosen. Meanwhile, if operation of F1 is not the last one on its processing machine M1, there are two cases: either small idle time or large idle time between F1 and the next operation, as depicted in Figure 4.2(c) and 4.2(e). In both cases, F1 is changed to its latest possible time, which are shown in Figure 4.2(d) and 4.2(f). The schedules in Figure 4.2(c) and 4.2(d) or Figure 4.2(e) and 4.2(f) are compared to keep the better one.

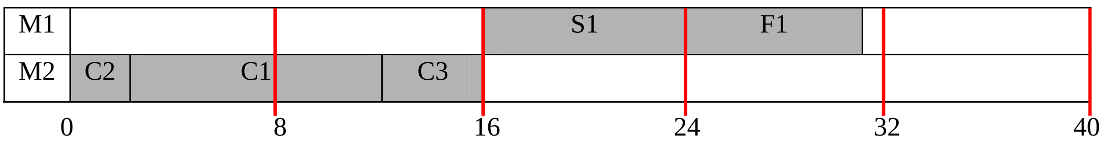


Figure 4.2 (a) The possible schedules for the simple example

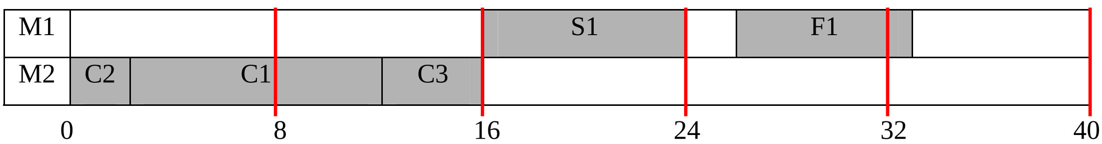


Figure 4.2 (b) The possible schedules for the simple example

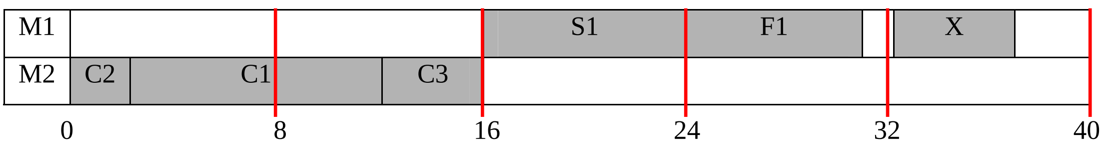


Figure 4.2 (c) The possible schedules for the simple example

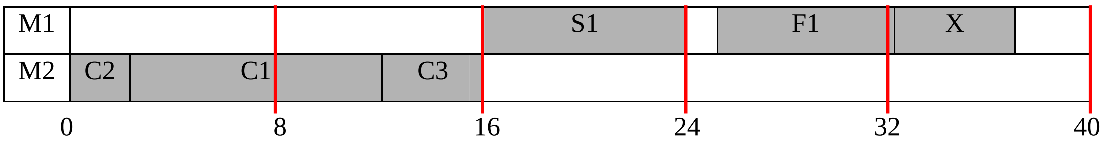


Figure 4.2 (d) The possible schedules for the simple example

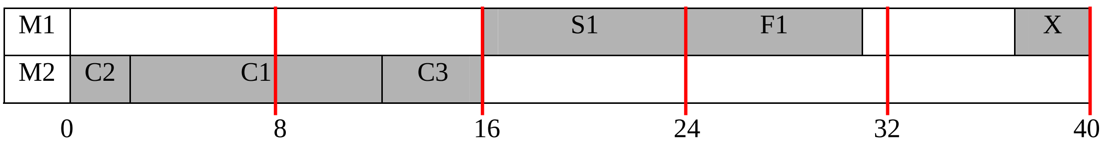


Figure 4.2 (e) The possible schedules for the simple example

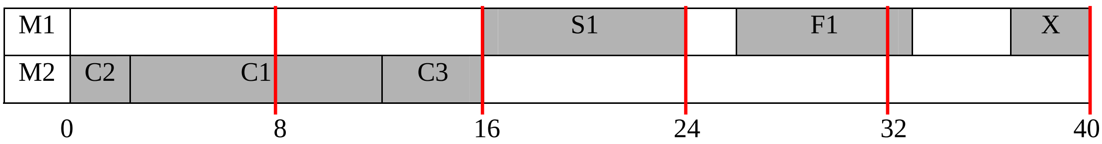


Figure 4.2 (f) The possible schedules for the simple example

4.2.2 Initialization

The initialization of the population of chromosomes can be done by generating the chromosomes randomly as much as the desired population size. Each chromosome contains a string of random numbers that represent the priorities of the genes. The genetic operations are then performed on the chromosomes, that is, the random keys, not on the schedule, which always leads to a feasible solution.

4.2.3 Evaluation

In GAs, the chromosomes contain much information, and each one should be evaluated by some measures of fitness. The fitness values indicate relative superiority of the chromosomes, which is necessary for the subsequent procedures including the selection operation and the reproduction operation. For the APS problem, the schedule represented by each chromosome is evaluated using the fitness function given in the following equation (4.1), which aggregates production idle time, earliness and tardiness penalty. The objective is to find a chromosome with the optimal schedule, minimizing the total costs. Let $eval(X_h)$ be the fitness function for chromosome X_h in the scheduling problem, then

$$eval(X_h) = \left\{ I(mC_{\max} - \sum_{i=1}^{n}\sum_{p=1}^{P_i} t_{ipk} \cdot N_{ip} \cdot Q_i - \sum_{k=1}^{m} r_k) + \sum_{i=1}^{n}\left[TC \times L_i^I + EC \times E_i^I\right] \right\} \quad (4.1)$$

where

n	Number of orders (i = 1, … , n)
m	Number of machines (k = 1, … , m)
P_i	Final product of order O_i (p = 1, … , P_i)
Q_i	Quantity of order O_i
N_{ip}	Number of item p needed for one final product P_i
t_{ipk}	Processing time required by item p of order O_i on machine M_k
r_k	Ready time of machine M_k
I	Cost of idle time per hour
TC	Cost of tardy orders per day per job
EC	Cost of early orders per day per job
C_{max}	Production makespan
L_i^I	Number of tardy days (integer) for order O_i
E_i^I	Number of early days (integer) for order O_i.

For instance, if I = \$ 50 / hour, TC = \$ 250 / day / order, and EC = \$ 50 / day / order, the fitness function for the schedule in Figure 4.2 (a) is:

$eval(X_h)$ = 50 * [2 * 31 – (7 + 8 + 10 + 2 + 4)] + 50 * 1 = 1600.

4.2.4 Selection

The well-known roulette wheel approach [Gol89, Gen00] is employed for selecting some chromosomes to conduct genetic operations. Based on this approach, the probability of selecting a chromosome is determined by its fitness. Chromosomes having larger fitness values are more likely being selected. Although the roulette wheel selection mechanism chooses chromosomes probabilistically, not deterministically, it is certain that on average a chromosome will be selected with the probability proportional to its fitness. Suppose that the population size is *psize*, then the selection procedure is as follows:

Step 1: Calcualte the total fitness of the population:

$$F = \sum_{h=1}^{psize} eval(X_h) .$$

Step 2: Calculate the selection probability p_h for each chromosome X_h:

$$p_h = \frac{F - eval(X_h)}{F \times (psize - 1)}, \quad h = 1, 2, \ldots, psize.$$

Step 3: Calculate the cumulative probability q_h for each chromosome X_h:

$$q_h = \sum_{j=1}^{h} p_j , \qquad h = 1, 2, \ldots, psize.$$

Step 4: Generate a random number r in the range (0, 1].

Step 5: If $q_{h-1} < r \le q_h$, then chromosome X_h is selected.

4.2.5 Genetic operations

Genetic operations such as reproduction, crossover and mutation are executed to produce a new set of chromosomes called offspring. There are many variations of genetic operations that could be used in GAs. Since random key encoding preserves to generate feasible solutions, there is no need to design specialized operations. The genetic operations employed here are elitist reproduction, parameterized uniform crossover and immigration, which have been proved very robust in computational tests [Bea94]. Meanwhile, the number of chromosomes selected for carrying out reproduction, crossover and mutation are set by the GA user and denoted as *Nr*, *Nc* and *Nm*. Definitely, the total number of chromosomes for reproduction, crossover and mutation equals to the population size, that is, $Nr + Nc + Nm = psize$.

Elitist reproduction is performed by directly copying the best *Nr* chromosomes from the current generation to the next. The advantage of the elitist strategy is that the best chromosomes associated with schedules are monotonically improving from one generation to another.

Parameterized uniform crossover introduced by [Bea94, Had97] could be detailed as: first, choose two chromosomes as parents from the current generation according to the selection mechanism stated above. Let $X = (x_1, x_2, \dots, x_k)$ and $Y = (y_1, y_2, \dots, y_k)$ be the k random key alleles in these two chromosomes (parents), respectively. Next, k independent random numbers could be uniformly generated from [0, 1], and denoted as $Z = (z_1, z_2, \dots, z_k)$. Then, let $U = (u_1, u_2, \dots, u_k)$ and $V = (v_1, v_2, \dots, v_k)$ be the two offspring that will result from the crossover of the two parents, and *Pc* be the probability of crossover for each gene. The two offspring *U* and *V* could be determined as

$$\begin{cases} u_i = x_i \text{ and } v_i = y_i \text{ if } z_i < P_c, \\ u_i = y_i \text{ and } v_i = x_i \text{ if } z_i \geq P_c. \end{cases}$$

All these chromosomes *X*, *Y*, *U*, and *V* are evaluated, and only those two with better fitness are permitted to enter into the next generation. This parameterized uniform crossover operation has shown to be computationally better than the one-point or two-point crossover [Had97].

Mutation is implemented by randomly generating one or more entirely new chromosomes from the same distribution as the original generation and including them in the next generation, which is referred as “immigration” in [Bea94, Had97]. Such an immigration operation plays an important role in preventing premature convergence of the population [Bea94, Had97].

4.2.6 The algorithm

On the whole, the genetic algorithm shown in Figure 4.3 is described as follows. After the parameters are set, including the maximum generation, the population size, the number of reproduction, the number of crossover, the crossover probability, and the number of mutation, the GA creates an initial set of random solutions. Each potential solution in the search space is represented by the form of a chromosome, a string of random numbers. All the obtained chromosomes are evaluated using the measure of fitness. On the basis of their fitness values, three genetic operations, elitist reproduction, parameterized uniform crossover and immigration, are executed to produce a new set of chromosomes, the offspring. These steps form an iteration, and then the evaluation is performed again to start the next iteration. When the maximum generation is reached, the algorithm converges to the best solution.

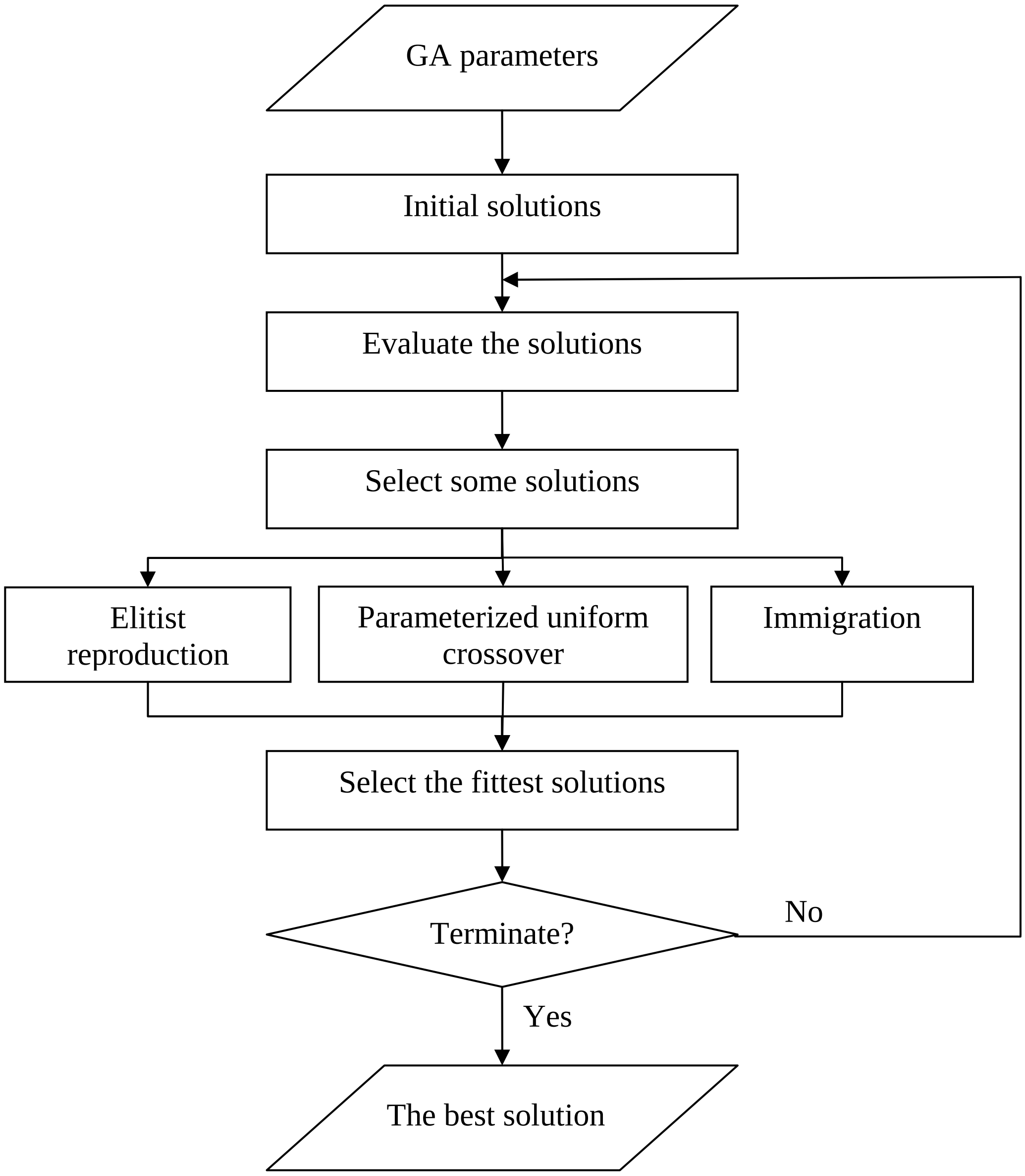


Figure 4.3 The overall structure of the genetic algorithm

The overall procedure of the proposed GA approach for the APS problem is listed in the following.

Step 1: Set the GA parameters, including the maximum generation (*genno*), the population size (*psize*), the number of reproduction (*Nr*), the number of crossover (*Nc*), the crossover probability (*Pc*), and the number of mutation (*Nm*).

Step 2: Generate initial *psize* chromosomes according to the encoding strategy in Section 4.2.1.

Step 3: Evaluate the fitness value $eval(X_h)$ for all chromosomes in the population according to the evaluation strategy in Section 4.2.3.

Step 4: Perform the elitist reproduction in Section 4.2.5.

Step 5: Perform the parameterized uniform crossover in Section 4.2.5.

Step 6: Perform the immigration in Section 4.2.5.

Step 7: Repeat Steps 3-6 until the maximum generation is reached.

4.3 NUMERICAL RESULTS

With respect to the simple example ($2 \times 2 \times 4$) in Section 3.4.1, the established genetic algorithm can easily find the optimal solution. In this section, the performance of the proposed GA method is investigated by use of the same representative example ($5 \times 6 \times 5$) as in Section 3.4.3. The developed GA-based approach was coded in the C language, as in the enclosed CD-ROM, and run on a personal computer with a Pentium 2.66 GHz CPU and 512 MB RAM.

4.3.1 The first iteration

In order to illustrate how the GA works, the procedure of seeking the best operation schedule for the APS example is elaborately described as follows.

Step 1: Set the GA parameters, including the maximum generation (*genno*), the population size (*psize*), the number of reproduction (*Nr*), the number of crossover (*Nc*), the crossover probability (*Pc*), and the number of mutation (*Nm*). In this case, *genno* = 20, *psize* = 8, *Nr* = 2, *Nc* = 4, *Pc* = 0.7, and *Nm* = 2. That is to say, the best 2 chromosomes from the current generation will be directly copied to the next, while the number of chromosomes for undergoing the crossover operation is 4. Meantime, at each iteration, 2 entirely new chromosomes will be created and included in the next generation.

Step 2: Generate the population size (*psize* = 8) initial chromosomes according to the encoding strategy in Section 4.2.1. The randomly created chromosomes are listed in Table 4.2. Each chromosome (X_h) contains a string of random numbers that represent the priorities of the items.

Table 4.2 The initial chromosomes obtained by the GA

Items	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
O1F1	0.98	0.01	0.28	0.74	0.55	0.23	0.35	0.53
O1C1	0.20	0.45	0.34	0.26	0.32	0.01	0.88	0.17
O1C2	0.68	0.67	0.71	0.71	0.33	0.32	0.40	0.68
O1C3	0.02	0.73	0.46	0.16	0.53	0.19	0.30	0.88
O1C4	0.75	0.38	0.89	0.47	0.01	0.58	0.89	0.80
O2F2	0.51	0.20	0.47	0.77	0.27	0.22	0.32	0.01
O2C2	0.01	0.18	0.24	0.17	0.19	0.35	0.44	0.46
O2S1	0.02	0.22	0.83	0.03	0.69	0.42	0.98	0.51
O2S4	0.67	0.14	0.26	0.28	0.44	0.43	0.65	0.57
O2C5	0.68	0.15	0.31	0.33	0.84	0.38	0.29	0.55
O2C6	0.53	0.83	0.14	0.10	0.61	0.27	0.03	0.45
O2S7	0.78	0.65	0.53	0.06	0.22	0.56	0.55	0.41
O2C14	0.15	0.95	0.47	0.10	0.78	0.09	0.41	0.58
O2C15	0.68	0.18	0.23	0.62	0.52	0.62	0.29	0.95
O3F3	0.78	0.05	0.05	0.29	0.90	0.72	0.36	0.15
O3S2	0.64	0.09	0.25	0.89	0.87	0.10	0.79	0.24
O3S3	0.43	0.07	0.81	0.46	0.21	0.68	0.70	0.01
O3C2	0.97	0.02	0.77	0.35	0.18	0.10	0.26	0.19
O3S5	0.39	0.03	0.30	0.42	0.64	0.29	0.64	0.90
O3C3	0.45	0.25	0.31	0.10	0.57	0.17	0.20	0.76
O3S6	0.62	0.98	0.59	0.01	0.94	0.28	0.69	0.29
O3S7	0.81	0.29	0.57	0.71	0.02	0.59	0.96	0.89
O3C7	0.39	0.66	0.61	0.70	0.50	0.50	0.51	0.91
O3C8	0.98	0.04	0.89	0.82	0.77	0.61	0.06	0.71
O3C9	0.98	0.59	0.44	0.04	0.94	0.14	0.21	0.08
O3C10	0.90	0.34	0.01	0.08	0.04	0.28	0.62	0.76
O3C11	0.80	0.78	0.58	0.61	0.60	0.57	0.58	0.89
O3C12	0.41	0.40	0.49	0.48	0.97	1.00	0.32	0.36
O3C13OP1	0.05	0.11	0.81	0.89	0.02	0.33	0.50	0.27
O3C13OP2	0.12	0.01	0.74	0.64	0.15	0.40	0.82	0.05
O3C14	0.56	0.20	0.42	0.04	0.87	0.20	0.91	0.38
O3C15	0.88	0.32	0.87	0.98	0.82	0.40	1.00	0.77
O4S2	0.65	0.68	0.08	0.86	0.33	0.06	0.49	0.60
O4S5	0.07	0.15	0.09	0.28	0.62	0.02	0.97	0.13
O4C3	0.61	0.78	0.80	0.04	0.87	0.32	0.30	0.27
O4C7	0.29	0.26	0.89	0.90	0.46	0.08	0.66	0.42
O4C8	0.03	0.59	0.02	0.96	0.47	0.24	0.06	0.46
O4C9	0.42	0.11	0.77	0.94	0.18	0.05	0.18	0.35
O4C10	0.02	0.29	0.35	0.02	0.50	0.20	0.67	0.05
O4C11	0.63	0.09	0.31	0.33	0.30	0.30	0.01	0.43
O5C3	0.39	0.73	0.50	0.95	0.50	0.01	0.04	0.16

Step 3: Evaluate the fitness value $eval(X_h)$ for all chromosomes in the population ($psize$ = 8) according to the evaluation strategy in Section 4.2.3. Table 4.3 below summarizes the fitness values of all 8 chromosomes.

Table 4.3 The fitness values of the initial chromosomes

Chromosome	Fitness value	Chromosome	Fitness value
X_1	9325	X_5	8975
X_2	9775	X_6	8925
X_3	9375	X_7	10575
X_4	10175	X_8	9525

Step 4: Perform the elitist reproduction in Section 4.2.5. In this initial population, chromosomes X_5 and X_6 have the lowest fitness values, 8975 and 8925, and then they are the elitist ones. Thus, these two chromosomes are straightforwardly reproduced to the next generation.

Step 5: Perform the parameterized uniform crossover in Section 4.2.5.

Firstly, two chromosomes should be selected as parents according to the selection mechanism stated in 4.2.4. The procedure of the roulette wheel approach is depicted in the following.

Step 5.1: Calcualte the total fitness of the population:

$$F = \sum_{h=1}^{psize} eval(X_h) = 76{,}650.$$

Step 5.2: Calculate the selection probability p_h for each chromosome X_h:

$$p_h = \frac{F - eval(X_h)}{F \times (psize - 1)}, \quad h=1, 2, \dots, 8.$$

The selection probability p_h for each chromosome X_h is listed in Table 4.4.

Table 4.4 The selection probabilities of the initial chromosomes

Chromosome	Selection probability	Chromosome	Selection probability
X_1	0.125478	X_5	0.126130
X_2	0.124639	X_6	0.126223
X_3	0.125384	X_7	0.123148
X_4	0.123893	X_8	0.125105

Step 5.3: Calculate the cumulative probability q_h for each chromosome X_h, as given in Table 4.5:

$$q_h = \sum_{j=1}^{h} p_j, \qquad h=1, 2, \dots, 8.$$

Table 4.5 The cumulative probabilities of the initial chromosomes

Chromosome	Cumulative probability	Chromosome	Cumulative probability
X_1	0.125478	X_5	0.625524
X_2	0.250116	X_6	0.751747
X_3	0.375501	X_7	0.874895
X_4	0.499394	X_8	1.000000

Step 5.4: Generate two random numbers r_1 and r_2 in the range (0, 1]. It is supposed that the two random numbers are 0.114875 and 0.200223.

Step 5.5: Since $r_1 = 0.114875 < q_1 = 0.125478$ and $q_1 < r_2 = 0.200223 < q_2 = 0.250116$, chromosomes X_1 and X_2 are selected as parents.

Secondly, since there are 41 alleles in each chromosome, another 41 independent random numbers should be uniformly generated from [0, 1] to execute the crossover, and denoted as:

Z = (0.80, 0.99, 0.94, 0.71, 0.57, 0.81, 0.09, 0.17, 0.67, 0.40,
0.84, 0.35, 0.42, 0.74, 0.46, 0.06, 0.17, 0.70, 0.11, 0.89,
0.26, 0.32, 0.16, 0.05, 0.18, 0.68, 0.29, 0.62, 0.24, 0.42,
0.28, 0.46, 0.31, 0.01, 0.54, 0.51, 0.25, 0.44, 0.11, 0.37, 0.13).

Then, we have the two offspring, X_9 and X_{10}, that result from the crossover of the two parents, X_1 and X_2, as shown in Table 4.6. For instance, since $z_1 = 0.80 > Pc = 0.7$, the first alleles in the offspring $x_{9,1} = x_{2,1} = 0.01$ and $x_{10,1} = x_{1,1} = 0.98$.

Table 4.6 Two illustrative offspring obtained by a crossover

Items	X_9	X_{10}
O1F1	0.01	0.98
O1C1	0.45	0.20
O1C2	0.67	0.68
O1C3	0.73	0.02
O1C4	0.75	0.38
O2F2	0.20	0.51
O2C2	0.01	0.18
O2S1	0.02	0.22
O2S4	0.67	0.14
O2C5	0.68	0.15
O2C6	0.83	0.53
O2S7	0.78	0.65
O2C14	0.15	0.95
O2C15	0.18	0.68
O3F3	0.78	0.05
O3S2	0.64	0.09
O3S3	0.43	0.07
O3C2	0.02	0.97
O3S5	0.39	0.03
O3C3	0.25	0.45
O3S6	0.62	0.98
O3S7	0.81	0.29
O3C7	0.39	0.66
O3C8	0.98	0.04
O3C9	0.98	0.59
O3C10	0.90	0.34
O3C11	0.80	0.78
O3C12	0.41	0.40
O3C13OP1	0.05	0.11
O3C13OP2	0.12	0.01
O3C14	0.56	0.20
O3C15	0.88	0.32
O4S2	0.65	0.68
O4S5	0.07	0.15
O4C3	0.61	0.78
O4C7	0.29	0.26
O4C8	0.03	0.59
O4C9	0.42	0.11
O4C10	0.02	0.29
O4C11	0.63	0.09
O5C3	0.39	0.73

The fitness values of X_9 and X_{10} are 9575 and 10775, respectively. All these chromosomes X_1, X_2, X_9 and X_{10} are compared by their fitness values, and X_1 and X_9 with better fitness are permitted to enter into the next generation. Similarly, the remaining 2 (= Nc - 2) chromosomes can be produced.

Step 6: Perform the immigration in Section 4.2.5. Referring to Table 4.7, two (Nm = 2) completely new chromosomes, X_{11} and X_{12}, are randomly generated and included in the next generation.

Table 4.7 Two new chromosomes obtained by a mutation

Items	X_{11}	X_{12}	Items	X_{11}	X_{12}
O1F1	0.49	0.49	O3S7	0.79	0.83
O1C1	0.10	0.55	O3C7	0.97	0.59
O1C2	0.43	0.60	O3C8	0.89	0.40
O1C3	0.55	0.08	O3C9	0.99	0.52
O1C4	0.19	0.54	O3C10	0.51	0.87
O2F2	0.13	0.19	O3C11	0.18	0.49
O2C2	0.79	0.01	O3C12	0.47	0.77
O2S1	0.98	0.19	O3C13OP1	0.98	0.82
O2S4	0.38	0.15	O3C13OP2	0.70	0.78
O2C5	0.68	0.83	O3C14	0.58	0.71
O2C6	0.59	0.85	O3C15	0.25	0.24
O2S7	0.47	1.00	O4S2	0.54	0.33
O2C14	0.36	0.96	O4S5	0.69	0.46
O2C15	0.31	0.57	O4C3	0.66	0.42
O3F3	0.95	0.85	O4C7	0.41	0.31
O3S2	0.17	0.39	O4C8	0.33	0.42
O3S3	0.19	0.37	O4C9	0.35	0.22
O3C2	0.70	0.45	O4C10	0.08	0.42
O3S5	0.57	0.06	O4C11	0.42	0.42
O3C3	0.66	0.23	O5C3	0.22	0.57
O3S6	0.60	0.05			

Step 7: Repeat Steps 3-6 until the maximum generation (*genno* = 20) is reached.

4.3.2 Identifying efficient genetic parameters

It is well known that the performance of a genetic algorithm is influenced by various genetic parameters. When solving the Advanced Planning and Scheduling (APS) problem, the population size and the number of generations are the main factors [Moo04, Pon04]. These two parameters determine the total number of chromosomes generated which further affect the amount of search, the chance of finding an optimal solution, the execution time and the computer storage space needed. In order to explore the effect of the population size and the number of generations, the GA program is tested with different levels of parameters and replicated five times at each level. The levels of parameters in Table 4.8 were chosen on the basis of the results of [Moo04, Pon04]. The other parameters of the GA for the problem are preset as: reproduction rate = 0.1, crossover rate = 0.8, crossover probability = 0.7, mutation rate = 0.1. The tests were run on a personal computer with a Pentium 2.66 GHz CPU and 512 MB RAM.

Table 4.8 Experimental parameters

Parameters	Levels
Population size (*psize*)	20, 100, 200
Number of generation (*genno*)	20, 100, 200

Figure 4.4 provides a scatter plot that illustrates the final total penalty costs arising from the schedules produced with different parameters. It could be seen from the scatter plot that the runs that employ a population of 200 with 200 generations obtain the lowest costs with a small spread. This may be due to the fact that more offspring are generated at each iteration and more generations are produced to explore the search space. Although the GA with *psize* = 200 and *genno* = 200 has better performance, it also requires longer computation time, about 90 times compared with *psize* = 20 and *genno* = 20. The comparison among the different GA parameter values is summarized in Table 4.9.

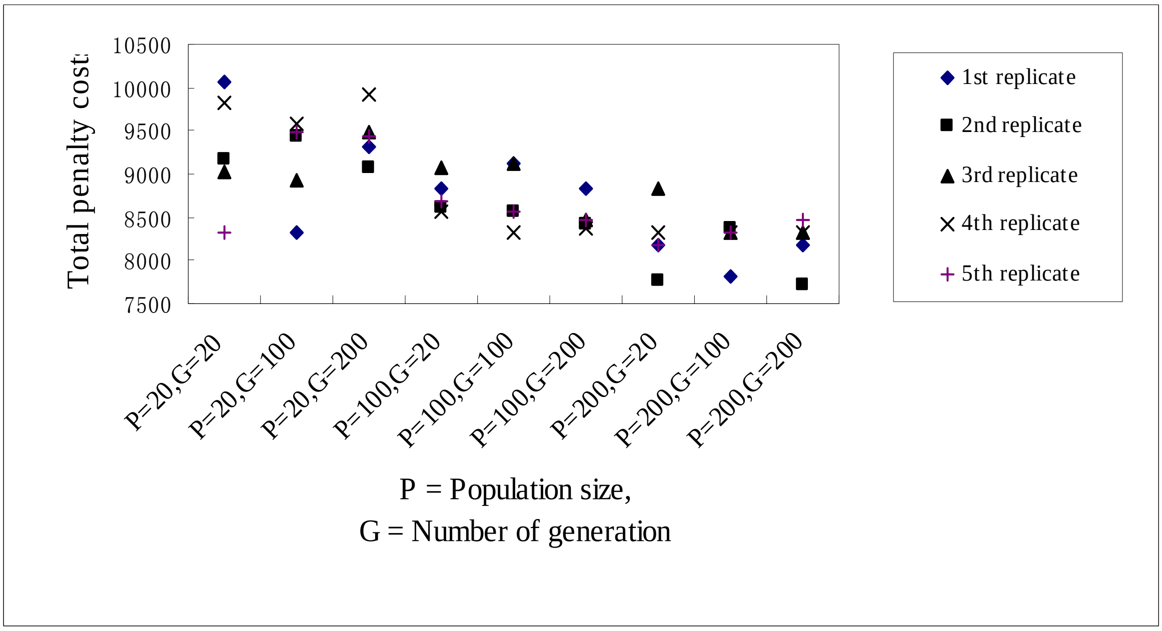


Figure 4.4 Scatter plot of results from five replications

Table 4.9 A comparison of different GA parameter values

Parameters values	Mean of best one in the initial population(s)	Mean of final best solution(s)	Mean of computation time (sec.)
psize = 20, *genno* = 20	9935	9285	0.063
psize = 20, *genno* = 100	9525	9145	0.266
psize = 20, *genno* = 200	9535	9445	0.547
psize = 100, *genno* = 20	9105	8755	0.262
psize = 100, *genno* = 100	9205	8745	1.301
psize = 100, *genno* = 200	9125	8515	2.675
psize = 200, *genno* = 20	9085	8255	0.531
psize = 200, *genno* = 100	9085	8235	2.703
psize = 200, *genno* = 200	9085	8205	5.403

4.3.3 Results analysis

When *psize* = 200 and *genno* = 200, the best solution associated with the chromosome obtained by the GA method is illustrated in Table 4.10 and 4.11, whereas the lowest cost in each generation is depicted in Figure 4.5. It can be seen in Figure 4.5 that the total penalty cost decreases and converges as the number of generation increases. In this particular case, the best production schedule with the total costs of 7725 was produced after 36 generations. Meanwhile, the best operation sequences are graphically represented in the Gantt chart as shown in Figure 4.6. For convenience, item *p* of order O_i is denoted *Oip*. From the Gantt chart, it is easy to find that the best makespan generated by the GA-based approach for the illustrative example is 45.5 hours. Orders 1, 2 and 3 are finished before their due dates.

Table 4.10 (a) The best schedule with the chromosome obtained by the GA for the example (5 × 6 × 5)

Items	Starting time (hour)	Finish time (hour)	Random number
O1F1	5.5	9.0	0.09
O1C1	4.0	5.0	0.23
O1C2	5.0	5.5	0.64
O1C3	1.5	2.0	0.37
O1C4	3.0	4.0	0.05
O2F2	42.5	45.5	0.99
O2C2	1.0	1.5	0.21
O2S1	21.0	23.5	0.29
O2S4	15.5	18.0	0.13
O2C5	2.0	3.0	0.01
O2C6	12.0	14.0	0.86
O2S7	14.0	15.5	0.72
O2C14	3.0	3.5	0.14
O2C15	3.5	4.0	0.25
O3F3	35.5	42.5	0.24
O3S2	29.5	35.5	0.06
O3S3	24.0	34.0	0.18
O3C2	5.5	8.5	0.65
O3S5	23.5	29.5	0.85
O3C3	11.5	14.5	0.87

Table 4.10 (b) The best schedule with the chromosome obtained by the GA for the example (5 × 6 × 5)

Items	Starting time (hour)	Finish time (hour)	Random number
O3S6	18.0	24.0	0.10
O3S7	8.0	14.0	0.55
O3C7	14.0	16.0	0.87
O3C8	10.0	12.0	0.80
O3C9	9.0	10.0	0.72
O3C10	14.0	16.0	0.63
O3C11	6.0	7.0	0.28
O3C12	8.0	14.0	0.43
O3C13OP1	12.0	16.0	0.78
O3C13OP2	8.0	12.0	0.94
O3C14	6.0	8.0	0.61
O3C15	4.0	6.0	0.53
O4S2	15.0	21.0	0.19
O4S5	9.0	15.0	0.69
O4C3	8.5	11.5	0.85
O4C7	5.0	7.0	0.27
O4C8	7.0	9.0	0.61
O4C9	3.0	4.0	0.17
O4C10	4.0	6.0	0.14
O4C11	7.0	8.0	0.34
O5C3	2.0	5.0	0.54

Table 4.11 The best solution obtained by the GA for the example (5 × 6 × 5)

Makespan (hour)	Number of tardiness	Number of earliness	Total cost	CPU time (sec.)
45.5	0	13	7725	5.393

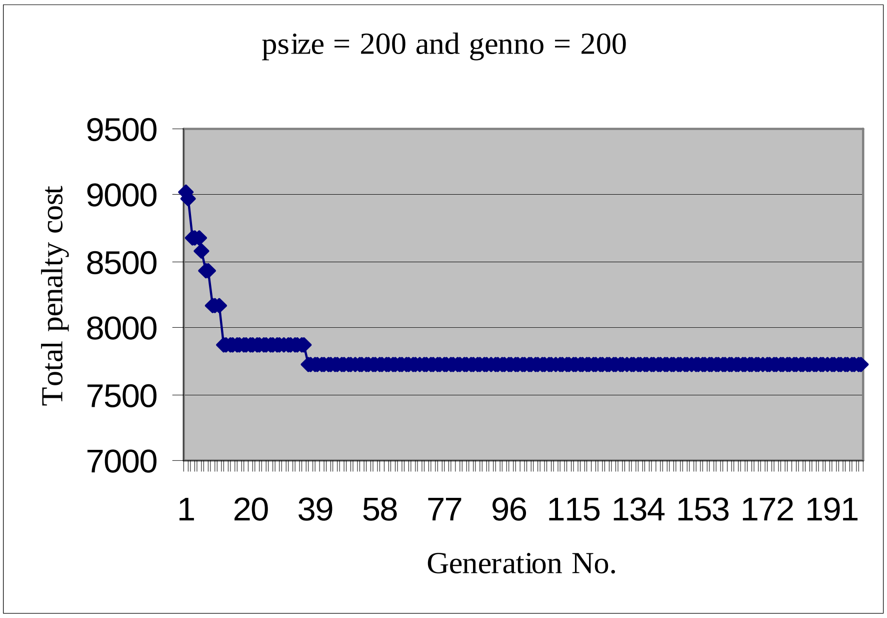


Figure 4.5 The lowest cost in each generation

The optimal solution obtained from the developed Mixed Integer Programming (MIP) model is with the total costs of 7575. The gap between the optimal costs and the best costs of our GA method is 1.98% ((7725-7575)/7575*100%), which is relatively small. However, it took the MIP model about 20 hours to reach the optimal solution. Obviously, our GA method spends much less time, which only needs 5.393 seconds.

The results also indicate that the presented genetic algorithm can generate realistic schedules with operation starting time and finish time for the shop floor.

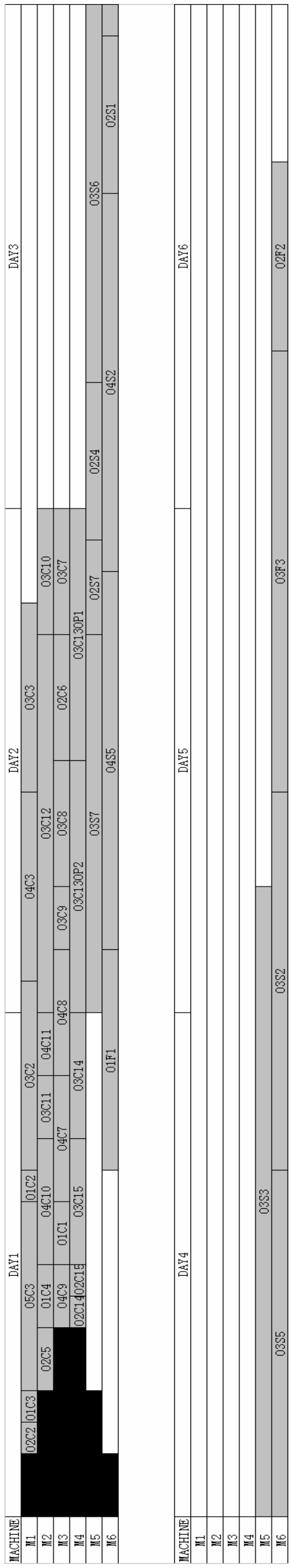


Figure 4.6 The GA results of the example (5 × 6 × 5) in the form of Gantt chart

4.4 COMPARISON TO OPTIMAL SOLUTIONS

A series of experiments using randomly generated problems were conducted to test the developed Mixed Integer Programming (MIP) and genetic algorithm (GA). The test problems are synthesized from the work of [Fry89, Kim96, Lee02, Pon02, Ran04]. The structure of each product is extracted or modified from [Fry89, Lee02, Pon02], and the number of levels in the structure ranges from 3 to 5. In any product structure, the number of subassemblies is between 5 and 15, whereas components are in [5, 30]. It should be noted that there are common items in the product structures, and all items in the product structure, including final products, subassemblies and components, could be ordered by the customers. Number of machines is drawn from a uniform distribution between 4 and 6. The time capacity of each machine is 8 hours a day. The machines, with varying ready time between 0 and 4 hours, are randomly assigned to process the items in the product structure. The processing time of components range from 0.1 to 0.4 hour in steps of 0.1, and those of subassemblies and final products are from 0.3 to 0.7. At the beginning of the planning horizon, between 1 and 5 orders arrive in the production system. While the due dates of orders in days are uniformly generated from [1, 15], the order quantities are from [5, 30] in steps of 5. The penalty rates, including cost of idle time and cost of earliness, are selected from the range between 50 and 100 in steps of 10, and then the tardiness penalty is set 5 times the earliness penalty. The tests were run on a personal computer with a Pentium 2.66 GHz CPU and 512 MB RAM.

The results of four typical examples are given in Table 4.12, which includes the optimal solution and the CPU time by CPLEX, the best solution and the CPU time by the GA with the efficient parameter settings identified in Section 4.3.2, and the gap between the optimal costs and the best costs of the GA for each problem size.

Table 4.12 Comparisons between the MIP and the GA

Numbers of orders, machines and levels	Optimal solution by CPLEX (obj)	CPU time by CPLEX (hh:mm:ss)	Best solution by GA (obj’)	CPU time by GA (sec.)	The gap ((obj’-obj)/obj *100%)
$3 \times 4 \times 5$	1810	00:00:03	1810	2.667	0
$4 \times 5 \times 5$	5550	00:02:15	5550	5.328	0
$5 \times 5 \times 4$	6550	11:52:07	6750	5.236	3.05
$5 \times 6 \times 5$	7575	19:23:34	7725	5.393	1.98

By solving the established MIP model, the optimal production schedules can be obtained for all test problems. For the problems, the optimal costs are 1810, 5550, 6550, and 7575, respectively. Although the global optimum could be found, the MIP is inefficient in doing so. From Table 4.12, it can be seen that the needed CPU time are 2.68 seconds, 2.25 minutes, 11.87 hours, and 19.39 hours, respectively. Obviously, the computational time grows exponentially with the problem size. In Appendices I and II, the CPLEX formulation and optimal results of the APS problem with five orders, six machines and five-level product structures are illustrated in detail. Moreover, Appendices III, IV and V list the data of example $3 \times 4 \times 5$, $4 \times 5 \times 5$, and $5 \times 5 \times 4$, as well as their CPLEX solutions.

On the contrary, the best solutions generated by the GA are 1810, 5550, 6750, and 7725, respectively. In other words, the GA, as a heuristic method, can reach the global optima for the small size problems, and only achieve the near-optimal solutions as the problem size increases. For the examples of 5 × 5 × 4 and 5 × 6 × 5, the gaps between the optimal costs and the best costs of the GA are 3.05% and 1.98%, both of which are relatively small. Furthermore, to compensate this, the GA requires much shorter computation time. As shown in Table 4.12, the longest time spent is only 5.393 seconds for the example 5 × 6 × 5. The results demonstrate that the suggested GA-based approach can efficiently find effective schedules for the APS problems with operation starting time and finish time in a reasonable computational time.

4.5 SUMMARY

In this chapter, a genetic algorithm (GA) is developed for the Advanced Planning and Scheduling (APS) problem. Some remarks can be summarized as follows.

1. The proposed GA-based approach explicitly takes into account due dates of products, operation sequences among items, and capacity constraints of the manufacturing system. The objective of the approach is to seek the minimum cost of both production idle time and tardiness or earliness penalty of an order.
2. The performance of the established genetic algorithm is investigated by the use of the same representative example (5 × 6 × 5) as in Chapter 3. The results indicate that the presented methodology can efficiently generate

realistic operation schedules with operation starting time and finish time for the shop floor and perform well on the tested problem. It is also found that the better results could be produced with higher levels of population size and number of generations. These two factors together determine the amount of search and the algorithm execution time.

3. The developed genetic algorithm constitutes a general approach that can be easily modified to adapt to a variety of APS problems, such as dynamic situation.

4. A series of computational experiments using randomly generated problems were conducted to compare the developed Mixed Integer Programming (MIP) and genetic algorithm (GA). By solving the established MIP, the optimal production schedules can be obtained for all test problems. However, the computational time grows exponentially with the problem size. On the contrary, the GA, as a heuristic method, can reach the global optima for the small size problems, and only achieve the near-optimal solutions for large problems, but it requires much less computation time.

For the APS problem, we have constructed an MIP model in Chapter 3 and a GA method in Chapter 4. An assumption made so far is that the APS problem is under a static environment. However, in a practical situation, the assumption is unrealistic. Thus, the next chapter will cope with the Dynamic Advanced Planning and Scheduling (DAPS) problem where new orders arrive on a continuous basis. Both the MIP in Chapter 3 and the GA in Chapter 4 will be further extended by incorporating a periodic policy with a frozen interval. The objective is to obtain a good schedule such that both production idle time and penalties on tardiness and earliness of both original orders and new orders are minimized at each rescheduling point. The effectiveness of the mechanism in the dynamic environment will be tested.

CHAPTER 5

DYNAMIC APS AND ITS SOLUTIONS

5.1 INTRODUCTION

Much of the research in the Advanced Planning and Scheduling (APS) is based on the assumption that the manufacturing environment is static, which rarely holds in real situations. In practice, some unexpected events, such as the arrival of new orders, may arise and disrupt the manufacturing system. This leads to the study of Dynamic Advanced Planning and Scheduling (DAPS).

In recent years, more studies have considered planning and scheduling problems in the dynamic environment, as reviewed in Chapter 2 (refer to Section 2.3.4). Unfortunately, most of the research efforts have been concentrated on one machine, flow shop, and job shop situations, assuming operations are performed in series. There appears to be scant research on introducing dynamic mechanism into Advanced Planning and Scheduling (APS).

For the dynamic production problems, traditional methods only consider the capability of generating new plans and schedules to optimize the efficiency measure like mean flow time, earliness and tardiness, etc. These strategies often make the plans and schedules experience large changes when new conditions occur, which are simply unacceptable in practice. For example, if the starting time of a job is delayed, excess inventory will be held to support the new schedule. On the other hand, when materials and tools are required to be delivered earlier than originally planned, rush costs will surely be added. Clearly, the trade-off between efficiency and stability should be addressed in the dynamic systems [Wus93, Ran04].

This chapter investigates a Dynamic Advanced Planning and Scheduling (DAPS) problem where new orders arrive on a continuous basis. A periodic policy with a frozen interval is adopted to increase stability on the shop floor. Both the MIP model in Chapter 3 and the GA method in Chapter 4 are further extended by incorporating the dynamic policy to find a schedule such that both production idle time and penalties on tardiness and earliness of both original orders and new orders are minimized at each rescheduling point. The two examples in Chapter 3 with the arrival of new orders are illustrated to indicate that the suggested approach can improve the schedule stability while retaining efficiency.

This chapter is organized as follows. In Section 5.2, the proposed methodology for the Dynamic Advanced Planning and Scheduling (DAPS) is presented. The numerical examples to illustrate the methodology are shown in Section 5.3. Finally, Section 5.4 concludes this chapter.

5.2 THE PROPOSED METHODOLOGY

Much of the previous research on dynamic problems only takes into account efficiency performance to minimize the cost objectives like mean flow time, earliness and tardiness, etc. Usually, doing so will greatly change the production schedule when new conditions occur and induce instability, which is highly undesirable in the practical shop. This section is to investigate a Dynamic Advanced Planning and Scheduling (DAPS) problem where new orders arrive on a continuous basis. Both the MIP model in Chapter 3 and the GA method in Chapter 4 are further extended by incorporating a periodic policy with a frozen interval to increase stability on the shop floor.

5.2.1 Policy

This research addresses a Dynamic Advanced Planning and Scheduling (DAPS) problem where new orders arrive on a continuous basis. In such a situation, if we construct a new schedule every time when a new order arrives, the system may be in a permanent state of replanning and rescheduling, and instability will be induced on the shop floor. Meanwhile, it is observed that the stability of the production system will decrease more when changes are made closer to the current period [Lin94, Ran04]. Therefore, this study adopts a periodic policy with a frozen interval. In other words, the schedule is revised periodically at the rescheduling point but not every time a new order arrives. Moreover, operations near the current time and within the frozen interval are fixed. Those operations outside the frozen interval and the newly arrived orders are available for building a new schedule. This policy provides a framework for balancing efficiency and stability.

The dynamic policy can be introduced into the developed MIP model. At each rescheduling point, the original orders and the new orders are combined to form a new APS problem. To fix the operations within the frozen interval, their start times (S_{ipk}) are given the optimal values (S_{ipk}^{*}) in the original problem. In other words, the following constraints are added to the MIP model.

$$S_{ipk} = S^{*}_{ipk} \qquad p \in C, \ \forall i, k. \qquad (5.1)$$

where

S_{ipk}	Production start time of item p of order O_i on machine M_k
S^{*}_{ipk}	The optimal start time of item p of order O_i on machine M_k in the original problem
C	Set of fixed items

The established GA method can also be extended by incorporating the periodic policy with a frozen interval. Our genetic algorithm is based on the random keys encoding of Bean [Bea94]. The idea of random keys encoding is to represent a solution with a string of random numbers. At each rescheduling point, the length of the random number string equals to the number of unfrozen items in both original orders and new orders, that is, each unfrozen item takes a random value in the range [0, 1]. These random numbers act as sort keys to decode the string into a feasible schedule, while the smaller value means the higher priority of the corresponding item.

5.2.2 The objective function

At each rescheduling point, our problem is to find a schedule for all the orders including original orders and new orders such that both production idle time and penalties on tardiness and earliness are minimized. Minimizing production idle time is equivalent to minimizing flow time or maximizing machine utilization. In addition, production idle time is chosen as the objective to be reduced because it is able to reflect two focuses in shops: manufacturing lead time and WIP (work-in-process) inventory level. Another objective of the problem is to find a schedule with all orders completed as close to their due dates as possible, which is on the fact that either early or late delivery of an order results in an increase in the costs.

5.2.3 Additional assumptions

To implement the methodology, some additional assumptions are made. In the make-to-assemble manufacturing system, there are multiple eligible machines with varying ready times. A machine can perform one operation at a time and only works for eight hours a day. Each operation can be processed on at most one machine at a given time and is non-preemptive. A lot-for-lot strategy is employed for making items, while the setup times (including the transfer times between operations) are negligible or are included in the processing times. Finally, it is assumed that new orders are continuously introduced into the production system on the infinite time horizon, and at the beginning, a preschedule has been generated that is optimal or near optimal with respect to the above objective function. If no new order occurs at some rescheduling point, the entire original schedule will be followed.

5.3 NUMERICAL RESULTS

To examine the effectiveness of the proposed mechanism in the dynamic environment, the two examples in Chapter 3 with the arrival of new orders are illustrated in the following. With respect to the simple example, there are two orders at the beginning of the planning horizon, and then one more order arrives at the first rescheduling point. In terms of the representative example, there are five orders at the beginning, and afterwards two new orders are received. The results indicate that the suggested approach can improve the schedule stability while retaining efficiency.

5.3.1 The simple DAPS example and its optimal solutions

With respect to the simple example in Section 3.4.1, there are two orders at the beginning of the planning horizon (t = 0): one requiring 10 Product F1s with due date Day 4 and the other requiring 15 Product S1s with due date Day 3. The rescheduling interval is determined as 1 day, that is, 8 hours. Then, at the first rescheduling point (t = Day 1), one more order arrives: 20 Product C1s with due date Day 4.

5.3.1.1 The MIP with a frozen interval

For the original problem, the optimal preschedule has been obtained in Section 3.4.2 by solving the developed Mixed Integer Programming (MIP). In the optimal preschedule, both Order 1 and 2 are fulfilled on time.

On Day 1 when the new order arrives, frozen interval = 1 day (8 hours) was used. The Gantt chart in Figure 3.3 clearly indicates that items O1S1, O1C1OP2, O1C1OP3, O1C2, O1C3, O2C2 and O2C3 have been completed when t = Hour 16 (= 8 + 8), while item O2S1 has begun its processing but has not finished yet. Hence, these items are fixed in the production schedule. The other items, including O1C1OP1 and O1F1, are outside the frozen interval, and they, together with the newly arrived order, are required to build a new schedule.

The MIP model with the dynamic policy is applied to this new situation as follows.

Minimize

100Cmax-2725

+250LI1+250LI2+250LI3

+50EI1+50EI2+50EI3

Subject to

(3.2):

cons1: C1-Cmax<=0

cons2: C2-Cmax<=0

cons3: C3-Cmax<=0

(3.3):

cons4: S111>=19.5

cons5: S132>=16

cons6: S312>=16

cons7: S322>=16

cons8: S332>=16

(3.4):

cons9: S111-S121>=5.000000

cons10: S111-S132>=4.000000

cons11: S121-S162>=1.000000

cons12: S121-S172>=2.000000

cons13: S132-S142>=4.000000

cons14: S142-S152>=2.000000

cons15: S211-S222>=1.500000

cons16: S211-S232>=3.000000

cons17: S312-S322>=4.000000

cons18: S322-S332>=2.000000

(3.5):

cons19: C1-S111=7

cons20: C2-S211=7.5

cons21: C3-S312=4

(3.6) and (3.7):

cons22: S111-S121+999X11121>=5.000000

cons23: S121-S111+999X12111>=7.000000

cons24: X11121+X12111=1

cons25: S111-S211+999X11211>=7.500000

cons26: S211-S111+999X21111>=7.000000

cons27: X11211+X21111=1

cons28: S121-S211+999X12211>=7.500000

cons29: S211-S121+999X21121>=5.000000

cons30: X12211+X21121=1

cons31: S132-S142+999X13142>=4.000000

cons32: S142-S132+999X14132>=4.000000

cons33: X13142+X14132=1

cons34: S132-S152+999X13152>=2.000000

cons35: S152-S132+999X15132>=4.000000

cons36: X13152+X15132=1

cons37: S132-S162+999X13162>=1.000000

cons38: S162-S132+999X16132>=4.000000

cons39: X13162+X16132=1

cons40: S132-S172+999X13172>=2.000000

cons41: S172-S132+999X17132>=4.000000

cons42: X13172+X17132=1

cons43: S132-S222+999X13222>=1.500000

cons44: S222-S132+999X22132>=4.000000

cons45: X13222+X22132=1

cons46: S132-S232+999X13232>=3.000000

cons47: S232-S132+999X23132>=4.000000

cons48: X13232+X23132=1

cons49: S132-S312+999X13312>=4.000000

cons50: S312-S132+999X31132>=4.000000

cons51: X13312+X31132=1

cons52: S132-S322+999X13322>=4.000000

cons53: S322-S132+999X32132>=4.000000

cons54: X13322+X32132=1

cons55: S132-S332+999X13332>=2.000000

cons56: S332-S132+999X33132>=4.000000

cons57: X13332+X33132=1

cons58: S142-S152+999X14152>=2.000000

cons59: S152-S142+999X15142>=4.000000

cons60: X14152+X15142=1

cons61: S142-S162+999X14162>=1.000000

cons62: S162-S142+999X16142>=4.000000

cons63: X14162+X16142=1

cons64: S142-S172+999X14172>=2.000000

cons65: S172-S142+999X17142>=4.000000

cons66: X14172+X17142=1

cons67: S142-S222+999X14222>=1.500000

cons68: S222-S142+999X22142>=4.000000

cons69: X14222+X22142=1

cons70: S142-S232+999X14232>=3.000000

cons71: S232-S142+999X23142>=4.000000

cons72: X14232+X23142=1

cons73: S142-S312+999X14312>=4.000000

cons74: S312-S142+999X31142>=4.000000

cons75: X14312+X31142=1

cons76: S142-S322+999X14322>=4.000000

cons77: S322-S142+999X32142>=4.000000

cons78: X14322+X32142=1

cons79: S142-S332+999X14332>=2.000000

cons80: S332-S142+999X33142>=4.000000

cons81: X14332+X33142=1

cons82: S152-S162+999X15162>=1.000000

cons83: S162-S152+999X16152>=2.000000

cons84: X15162+X16152=1

cons85: S152-S172+999X15172>=2.000000

cons86: S172-S152+999X17152>=2.000000

cons87: X15172+X17152=1

cons88: S152-S222+999X15222>=1.500000

cons89: S222-S152+999X22152>=2.000000

cons90: X15222+X22152=1

cons91: S152-S232+999X15232>=3.000000

cons92: S232-S152+999X23152>=2.000000

cons93: X15232+X23152=1

cons94: S152-S312+999X15312>=4.000000

cons95: S312-S152+999X31152>=2.000000

cons96: X15312+X31152=1

cons97: S152-S322+999X15322>=4.000000

cons98: S322-S152+999X32152>=2.000000

cons99: X15322+X32152=1

cons100: S152-S332+999X15332>=2.000000

cons101: S332-S152+999X33152>=2.000000

cons102: X15332+X33152=1

cons103: S162-S172+999X16172>=2.000000

cons104: S172-S162+999X17162>=1.000000

cons105: X16172+X17162=1

cons106: S162-S222+999X16222>=1.500000

cons107: S222-S162+999X22162>=1.000000

cons108: X16222+X22162=1

cons109: S162-S232+999X16232>=3.000000

cons110: S232-S162+999X23162>=1.000000

cons111: X16232+X23162=1

cons112: S162-S312+999X16312>=4.000000

cons113: S312-S162+999X31162>=1.000000

cons114: X16312+X31162=1

cons115: S162-S322+999X16322>=4.000000

cons116: S322-S162+999X32162>=1.000000

cons117: X16322+X32162=1

cons118: S162-S332+999X16332>=2.000000

cons119: S332-S162+999X33162>=1.000000

cons120: X16332+X33162=1

cons121: S172-S222+999X17222>=1.500000

cons122: S222-S172+999X22172>=2.000000

cons123: X17222+X22172=1

cons124: S172-S232+999X17232>=3.000000

cons125: S232-S172+999X23172>=2.000000

cons126: X17232+X23172=1

cons127: S172-S312+999X17312>=4.000000

cons128: S312-S172+999X31172>=2.000000

cons129: X17312+X31172=1

cons130: S172-S322+999X17322>=4.000000

cons131: S322-S172+999X32172>=2.000000

cons132: X17322+X32172=1

cons133: S172-S332+999X17332>=2.000000

cons134: S332-S172+999X33172>=2.000000

cons135: X17332+X33172=1

cons136: S222-S232+999X22232>=3.000000

cons137: S232-S222+999X23222>=1.500000

cons138: X22232+X23222=1

cons139: S222-S312+999X22312>=4.000000

cons140: S312-S222+999X31222>=1.500000

cons141: X22312+X31222=1

cons142: S222-S322+999X22322>=4.000000

cons143: S322-S222+999X32222>=1.500000

cons144: X22322+X32222=1

cons145: S222-S332+999X22332>=2.000000

cons146: S332-S222+999X33222>=1.500000

cons147: X22332+X33222=1

cons148: S232-S312+999X23312>=4.000000

cons149: S312-S232+999X31232>=3.000000

cons150: X23312+X31232=1

cons151: S232-S322+999X23322>=4.000000

cons152: S322-S232+999X32232>=3.000000

cons153: X23322+X32232=1

cons154: S232-S332+999X23332>=2.000000

cons155: S332-S232+999X33232>=3.000000

cons156: X23332+X33232=1

cons157: S312-S322+999X31322>=4.000000

cons158: S322-S312+999X32312>=4.000000

cons159: X31322+X32312=1

cons160: S312-S332+999X31332>=2.000000

cons161: S332-S312+999X33312>=4.000000

cons162: X31332+X33312=1

cons163: S322-S332+999X32332>=2.000000

cons164: S332-S322+999X33322>=4.000000

cons165: X32332+X33322=1

(3.8) and (3.9):

cons166: 0.125C1-L1<=4

cons167: E1+0.125C1>=4

cons168: 0.125C2-L2<=3

cons169: E2+0.125C2>=3

cons170: 0.125C3-L3<=4

cons171: E3+0.125C3>=4

(3.10) and (3.11):

cons172: L1-LI1<=0

cons173: E1-EI1<=0.99

cons174: L2-LI2<=0

cons175: E2-EI2<=0.99

cons176: L3-LI3<=0

cons177: E3-EI3<=0.99

(5.1):

cons178: S121=7

cons179: S152 =7.0

cons180: S162 = 2.5

cons181: S172= 5.0

cons182: S222=3.5

cons183: S232=9

cons184: S211=12

cons185: S142=12

Bounds

LI1 free	LI3 free	EI2 free
LI2 free	EI1 free	EI3 free

Integers

EI1 LI1	EI2 LI2	EI3 LI3
X11121 X12111	X14232 X23142	X16332 X33162
X11211 X21111	X14312 X31142	X17222 X22172
X12211 X21121	X14322 X32142	X17232 X23172
X13142 X14132	X14332 X33142	X17312 X31172
X13152 X15132	X15162 X16152	X17322 X32172
X13162 X16132	X15172 X17152	X17332 X33172
X13172 X17132	X15222 X22152	X22232 X23222
X13222 X22132	X15232 X23152	X22312 X31222
X13232 X23132	X15312 X31152	X22322 X32222
X13312 X31132	X15322 X32152	X22332 X33222
X13322 X32132	X15332 X33152	X23312 X31232
X13332 X33132	X16172 X17162	X23322 X32232
X14152 X15142	X16222 X22162	X23332 X33232
X14162 X16142	X16232 X23162	X31322 X32312
X14172 X17142	X16312 X31162	X31332 X33312
X14222 X22142	X16322 X32162	X32332 X33322

End

The simple DAPS example was solved using CPLEX 9.1 on a personal computer with Pentium 2.66 GHz CPU and 512 MB RAM. It should be noticed that 999 is taken as the large positive number α for the convenience in the CPLEX input process. The optimal solution generated by CPLEX is intensively shown in the following. The important data derived from the optimal results are summarized in Table 5.1, while Figure 5.1 displays the optimal solution in the form of Gantt chart. For convenience, item p of order O_i is denoted Oip. In this simple DAPS example, all of the orders, including the original ones and the new one, are exactly completed without earliness or tardiness.

Integer optimal
Objective = 2.750000000e+002
Solution time = 0.02 sec.
Iterations = 15
Nodes = 0

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	30.000000	S162	2.500000
C1	27.000000	S172	5.000000
C2	19.500000	S222	3.500000
C3	30.000000	S232	9.000000
S111	20.000000	S312	26.000000
S121	7.000000	S322	22.000000
S211	12.000000	S332	20.000000
S132	16.000000	X12111	1.000000
S142	12.000000	X21111	1.000000
S152	7.000000	X12211	1.000000

X14132	1.000000	X16172	1.000000
X15132	1.000000	X16222	1.000000
X16132	1.000000	X16232	1.000000
X17132	1.000000	X16312	1.000000
X22132	1.000000	X16322	1.000000
X23132	1.000000	X16332	1.000000
X13312	1.000000	X22172	1.000000
X13322	1.000000	X17232	1.000000
X13332	1.000000	X17312	1.000000
X15142	1.000000	X17322	1.000000
X16142	1.000000	X17332	1.000000
X17142	1.000000	X22232	1.000000
X22142	1.000000	X22312	1.000000
X23142	1.000000	X22322	1.000000
X14312	1.000000	X22332	1.000000
X14322	1.000000	X23312	1.000000
X14322	1.000000	X23322	1.000000
X14332	1.000000	X23332	1.000000
X16152	1.000000	X32312	1.000000
X17152	1.000000	X33312	1.000000
X22152	1.000000	X33322	1.000000
X15232	1.000000	E1	0.990000
X15312	1.000000	E2	0.990000
X15322	1.000000	E3	0.990000
X15332	1.000000		

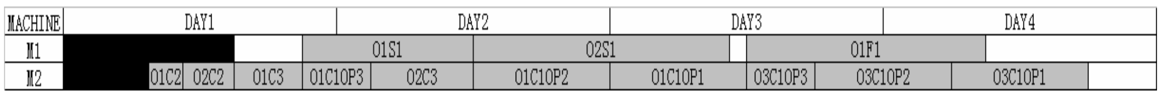
All other variables are zero.

Table 5.1 Optimal results of the simple DAPS example when frozen interval = 1 day

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
185	125	102	30	0.02

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
0	0	0	0	275

Figure 5.1 Optimal results of the simple DAPS example in the form of Gantt chart when frozen interval = 1 day



5.3.1.2 The MIP without a frozen interval

In order to examine the effect of the frozen interval, we also adopted the MIP model without a frozen interval to tackle the simple dynamic problem. In this case, items O1C1OP2, O2C3 as well as O2S1 are not fixed and also free for rescheduling. That is, at the rescheduling point (t = Day 1), items O1C1OP1, O1C1OP2, O2C3, O2S1 and O1F1, together with the newly arrived order, are needed to construct a new schedule. The MIP model in the following is the mathematical formulation for this situation.

Minimize

100Cmax-2725

+250LI1+250LI2+250LI3

+50EI1+50EI2+50EI3

Subject to

(3.2):

cons1: C1-Cmax<=0

cons2: C2-Cmax<=0

cons3: C3-Cmax<=0

(3.3):

cons4: S111>=12

cons5: S211>=12

cons6: S132>=9

cons7: S142>=9

cons8: S232>=9

cons9: S312>=9

cons10: S322>=9

cons11: S332>=9

(3.4):

cons12: S111-S121>=5.000000

cons13: S111-S132>=4.000000

cons14: S121-S162>=1.000000

cons15: S121-S172>=2.000000

cons16: S132-S142>=4.000000

cons17: S142-S152>=2.000000

cons18: S211-S222>=1.500000

cons19: S211-S232>=3.000000

cons20: S312-S322>=4.000000

cons21: S322-S332>=2.000000

(3.5):

cons22: C1-S111=7

cons23: C2-S211=7.5

cons24: C3-S312=4

(3.6) and (3.7):

cons25: S111-S121+999X11121>=5.000000

cons26: S121-S111+999X12111>=7.000000

cons27: X11121+X12111=1

cons28: S111-S211+999X11211>=7.500000

cons29: S211-S111+999X21111>=7.000000

cons30: X11211+X21111=1

cons31: S121-S211+999X12211>=7.500000

cons32: S211-S121+999X21121>=5.000000

cons33: X12211+X21121=1

cons34: S132-S142+999X13142>=4.000000

cons35: S142-S132+999X14132>=4.000000

cons36: X13142+X14132=1

cons37: S132-S152+999X13152>=2.000000

cons38: S152-S132+999X15132>=4.000000

cons39: X13152+X15132=1

cons40: S132-S162+999X13162>=1.000000

cons41: S162-S132+999X16132>=4.000000

cons42: X13162+X16132=1

cons43: S132-S172+999X13172>=2.000000

cons44: S172-S132+999X17132>=4.000000

cons45: X13172+X17132=1

cons46: S132-S222+999X13222>=1.500000

cons47: S222-S132+999X22132>=4.000000

cons48: X13222+X22132=1

cons49: S132-S232+999X13232>=3.000000

cons50: S232-S132+999X23132>=4.000000

cons51: X13232+X23132=1

cons52: S132-S312+999X13312>=4.000000

cons53: S312-S132+999X31132>=4.000000

cons54: X13312+X31132=1

cons55: S132-S322+999X13322>=4.000000

cons56: S322-S132+999X32132>=4.000000

cons57: X13322+X32132=1

cons58: S132-S332+999X13332>=2.000000

cons59: S332-S132+999X33132>=4.000000

cons60: X13332+X33132=1

cons61: S142-S152+999X14152>=2.000000

cons62: S152-S142+999X15142>=4.000000

cons63: X14152+X15142=1

cons64: S142-S162+999X14162>=1.000000

cons65: S162-S142+999X16142>=4.000000

cons66: X14162+X16142=1

cons67: S142-S172+999X14172>=2.000000

cons68: S172-S142+999X17142>=4.000000

cons69: X14172+X17142=1

cons70: S142-S222+999X14222>=1.500000

cons71: S222-S142+999X22142>=4.000000

cons72: X14222+X22142=1

cons73: S142-S232+999X14232>=3.000000

cons74: S232-S142+999X23142>=4.000000

cons75: X14232+X23142=1

cons76: S142-S312+999X14312>=4.000000

cons77: S312-S142+999X31142>=4.000000

cons78: X14312+X31142=1

cons79: S142-S322+999X14322>=4.000000

cons80: S322-S142+999X32142>=4.000000

cons81: X14322+X32142=1

cons82: S142-S332+999X14332>=2.000000

cons83: S332-S142+999X33142>=4.000000

cons84: X14332+X33142=1

cons85: S152-S162+999X15162>=1.000000

cons86: S162-S152+999X16152>=2.000000

cons87: X15162+X16152=1

cons88: S152-S172+999X15172>=2.000000

cons89: S172-S152+999X17152>=2.000000

cons90: X15172+X17152=1

cons91: S152-S222+999X15222>=1.500000

cons92: S222-S152+999X22152>=2.000000

cons93: X15222+X22152=1

cons94: S152-S232+999X15232>=3.000000

cons95: S232-S152+999X23152>=2.000000

cons96: X15232+X23152=1

cons97: S152-S312+999X15312>=4.000000

cons98: S312-S152+999X31152>=2.000000

cons99: X15312+X31152=1

cons100: S152-S322+999X15322>=4.000000

cons101: S322-S152+999X32152>=2.000000

cons102: X15322+X32152=1

cons103: S152-S332+999X15332>=2.000000

cons104: S332-S152+999X33152>=2.000000

cons105: X15332+X33152=1

cons106: S162-S172+999X16172>=2.000000

cons107: S172-S162+999X17162>=1.000000

cons108: X16172+X17162=1

cons109: S162-S222+999X16222>=1.500000

cons110: S222-S162+999X22162>=1.000000

cons111: X16222+X22162=1

cons112: S162-S232+999X16232>=3.000000

cons113: S232-S162+999X23162>=1.000000

cons114: X16232+X23162=1

cons115: S162-S312+999X16312>=4.000000

cons116: S312-S162+999X31162>=1.000000

cons117: X16312+X31162=1

cons118: S162-S322+999X16322>=4.000000

cons119: S322-S162+999X32162>=1.000000

cons120: X16322+X32162=1

cons121: S162-S332+999X16332>=2.000000

cons122: S332-S162+999X33162>=1.000000

cons123: X16332+X33162=1

cons124: S172-S222+999X17222>=1.500000

cons125: S222-S172+999X22172>=2.000000

cons126: X17222+X22172=1

cons127: S172-S232+999X17232>=3.000000

cons128: S232-S172+999X23172>=2.000000

cons129: X17232+X23172=1

cons130: S172-S312+999X17312>=4.000000

cons131: S312-S172+999X31172>=2.000000

cons132: X17312+X31172=1

cons133: S172-S322+999X17322>=4.000000

cons134: S322-S172+999X32172>=2.000000

cons135: X17322+X32172=1

cons136: S172-S332+999X17332>=2.000000

cons137: S332-S172+999X33172>=2.000000

cons138: X17332+X33172=1

cons139: S222-S232+999X22232>=3.000000

cons140: S232-S222+999X23222>=1.500000

cons141: X22232+X23222=1

cons142: S222-S312+999X22312>=4.000000

cons143: S312-S222+999X31222>=1.500000

cons144: X22312+X31222=1

cons145: S222-S322+999X22322>=4.000000

cons146: S322-S222+999X32222>=1.500000

cons147: X22322+X32222=1

cons148: S222-S332+999X22332>=2.000000

cons149: S332-S222+999X33222>=1.500000

cons150: X22332+X33222=1

cons151: S232-S312+999X23312>=4.000000

cons152: S312-S232+999X31232>=3.000000

cons153: X23312+X31232=1

cons154: S232-S322+999X23322>=4.000000

cons155: S322-S232+999X32232>=3.000000

cons156: X23322+X32232=1

cons157: S232-S332+999X23332>=2.000000

cons158: S332-S232+999X33232>=3.000000

cons159: X23332+X33232=1

cons160: S312-S322+999X31322>=4.000000

cons161: S322-S312+999X32312>=4.000000

cons162: X31322+X32312=1

cons163: S312-S332+999X31332>=2.000000

cons164: S332-S312+999X33312>=4.000000

cons165: X31332+X33312=1

cons166: S322-S332+999X32332>=2.000000

cons167: S332-S322+999X33322>=4.000000

cons168: X32332+X33322=1

(3.8) and (3.9):

cons169: 0.125C1-L1<=4

cons170: E1+0.125C1>=4

cons171: 0.125C2-L2<=3

cons172: E2+0.125C2>=3

cons173: 0.125C3-L3<=4

cons174: E3+0.125C3>=4

(3.10) and (3.11):

cons175: L1-LI1<=0

cons176: E1-EI1<=0.99

cons177: L2-LI2<=0

cons178: E2-EI2<=0.99

cons179: L3-LI3<=0

cons180: E3-EI3<=0.99

(5.1):

cons181: S121 = 7

cons182: S152 = 7.0

cons183: S162 = 2.5

cons184: S172 = 5.0

cons185: S222 = 3.5

Bounds

LI1 free	LI3 free	EI2 free
LI2 free	EI1 free	EI3 free

Integers

EI1 LI1	EI2 LI2	EI3 LI3
X11121 X12111	X14232 X23142	X16332 X33162
X11211 X21111	X14312 X31142	X17222 X22172
X12211 X21121	X14322 X32142	X17232 X23172
X13142 X14132	X14332 X33142	X17312 X31172
X13152 X15132	X15162 X16152	X17322 X32172
X13162 X16132	X15172 X17152	X17332 X33172
X13172 X17132	X15222 X22152	X22232 X23222
X13222 X22132	X15232 X23152	X22312 X31222
X13232 X23132	X15312 X31152	X22322 X32222
X13312 X31132	X15322 X32152	X22332 X33222
X13322 X32132	X15332 X33152	X23312 X31232
X13332 X33132	X16172 X17162	X23322 X32232
X14152 X15142	X16222 X22162	X23332 X33232
X14162 X16142	X16232 X23162	X31322 X32312
X14172 X17142	X16312 X31162	X31332 X33312
X14222 X22142	X16322 X32162	X32332 X33322

End

The optimal schedule obtained from CPLEX is given as follows and summarized in Table 5.2, whereas the Gantt chart is portrayed in Figure 5.2.

Integer optimal
Objective = 2.750000000e+002
Solution time = 0.02 sec.
Iterations = 72
Nodes = 9

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	30.000000	X14132	1.000000
C1	29.000000	X15132	1.000000
C2	19.500000	X16132	1.000000
C3	30.000000	X17132	1.000000
S111	22.000000	X22132	1.000000
S121	7.000000	X23132	1.000000
S211	12.000000	X13312	1.000000
S132	18.000000	X13322	1.000000
S142	12.000000	X33132	1.000000
S152	7.000000	X15142	1.000000
S162	2.500000	X16142	1.000000
S172	5.000000	X17142	1.000000
S222	3.500000	X22142	1.000000
S232	9.000000	X23142	1.000000
S312	26.000000	X14312	1.000000
S322	22.000000	X14322	1.000000
S332	16.000000	X14332	1.000000
X12111	1.000000	X16152	1.000000
X21111	1.000000	X17152	1.000000
X12211	1.000000	X22152	1.000000

X15232	1.000000	X17332	1.000000
X15312	1.000000	X22232	1.000000
X15322	1.000000	X22312	1.000000
X15332	1.000000	X22322	1.000000
X16172	1.000000	X22332	1.000000
X16222	1.000000	X23312	1.000000
X16232	1.000000	X23322	1.000000
X16312	1.000000	X23332	1.000000
X16322	1.000000	X32312	1.000000
X16332	1.000000	X33312	1.000000
X22172	1.000000	X33322	1.000000
X17232	1.000000	E1	0.990000
X17312	1.000000	E2	0.990000
X17322	1.000000	E3	0.990000

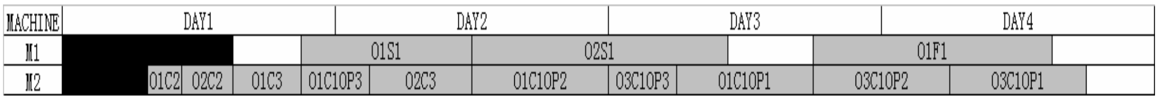
All other variables are zero.

Table 5.2 Optimal results of the simple DAPS example without a frozen interval

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
185	125	102	30	0.02

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
0	0	0	0	275

Figure 5.2 Optimal results of the simple DAPS example without a frozen interval in the form of Gantt chart



5.3.1.3 Comparison of performance

In this section, the effectiveness of the frozen interval, as measured by efficiency and stability, will be investigated.

The measure of stability defined by Rangsaritratsamee et al. [Ran04] includes two components. One is the starting time deviation for all operations between the new schedule and the original one, and the other is the penalty associated with total deviation from the current time. If the current time is t, the operation starting time in the original schedule is t_i, and in the new schedule is t_i', the stability is defined as $\sum_i |t_i - t_i'| + \sum_i PF(t_i - t + t_i' - t)$, where $PF(x)$ is the penalty function. In particular, $PF(x) = 10/(x^{0.5})$ is used, and moreover, when the total derivation from the current time is zero, the penalty is assumed to be zero.

Table 5.3 shows the deduction of the stability measure when the frozen interval is 1 day, while Table 5.4 lists the calculation of the stability when the MIP without a frozen interval. Table 5.5 summarizes the testing results from different methodologies. It can be seen that the suggested approach is capable of improving the schedule stability while retaining efficiency, which is particularly significant in the DAPS problem.

Table 5.3 The stability of the simple DAPS example when frozen interval = 1 day

Items	Current time t (hour)	Original starting time t_i (hour)	New starting time t_i' (hour)	$\lvert t_i' - t_i \rvert$	$PF(t_i - t + t_i' - t)$	Stability
O1F1	8	20	20	0	2.041	2.041
O1C1OP1	8	16	16	0	2.500	2.500
O1C1OP2	8	12	12	0	3.536	3.536
O2S1	8	12	12	0	3.536	3.536
O2C3	8	9	9	0	7.071	7.071
Total				0	18.684	18.684

Table 5.4 The stability of the simple DAPS example without a frozen interval

Items	Current time t (hour)	Original starting time t_i (hour)	New starting time t_i' (hour)	$\lvert t_i' - t_i \rvert$	$PF(t_i - t + t_i' - t)$	Stability
O1F1	8	20	22	2	1.961	3.961
O1C1OP1	8	16	12	4	2.887	6.887
O1C1OP2	8	12	18	6	2.673	8.673
O2S1	8	12	12	0	3.536	3.536
O2C3	8	9	9	0	7.071	7.071
Total				12	18.128	30.128

Table 5.5 The optimal results from different methodologies in the simple DAPS example

Methodology	Objective function	Stability
MIP with a frozen interval	275	18.684
MIP without a frozen interval	275	30.128

5.3.2 The representative DAPS example and its optimal solutions

In terms of the representative example in Section 3.4.3, the rescheduling interval is also chosen as 1 day, that is, 8 hours. Originally (t = 0), there are five orders in the production system: 5 Product F1s with due date Day 6, 5 Product F2s with due date Day 7, 10 Product F3s with due date Day 14, 10 Product S2s with due date Day 3, and 30 Product C3s with due date Day 1. Afterwards, at the first rescheduling point (t = Day 1), two new orders are received: 10 Product S1s with due date Day 5 and 5 Product S5s with due date Day 3.

5.3.2.1 The MIP with a frozen interval

The optimal solution of the original problem has been generated in Section 3.4.3. In the optimal solution, Orders 1, 2 and 3 are completed before their due dates, while the other two orders are satisfied on time.

When two new orders arrive at the first rescheduling point (t = Day 1), frozen interval = 1 day (8 hours) was employed. On the basis of the Gantt chart in Figure 3.5, items O1F1, O1C1, O1C2, O1C3, O1C4, O2S1, O2S4, O2S7, O2C2, O2C5, O2C6, O2C14, O2C15, O3S7, O3C3, O3C10, O3C11, O3C12, O3C13OP1,

O3C13OP2, O3C14, O3C15, O4S5, O4C3, O4C7, O4C8, O4C9, O4C10, O4C11 and O5C3 are fixed in the schedule, whereas items O2F2, O3F3, O3S2, O3S3, O3S5, O3S6, O3C2, O3C7, O3C8, O3C9 and O4S2 together with the two newly arrived orders are needed to derive a new production schedule.

To settle the dynamic issue, the MIP model with a frozen interval is formulated as in Appendix VI. The optimal solution generated by CPLEX 9.1 on a personal computer with Pentium 2.66 GHz CPU and 512 MB RAM is attached in Appendix VII and summarized in Table 5.6, while the optimal operation sequences in the form of Gantt chart are displayed in Figure 5.3. For convenience, item *p* of order O_i is denoted *Oip*. In this DAPS example, Order 7 can not be fulfilled on time with one day delay, while Orders 1, 2 and 3 are early.

Table 5.6 Optimal results of the representative DAPS example when frozen interval = 1 day

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
821	543	466	53	0.45

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
1	12	250	600	8775

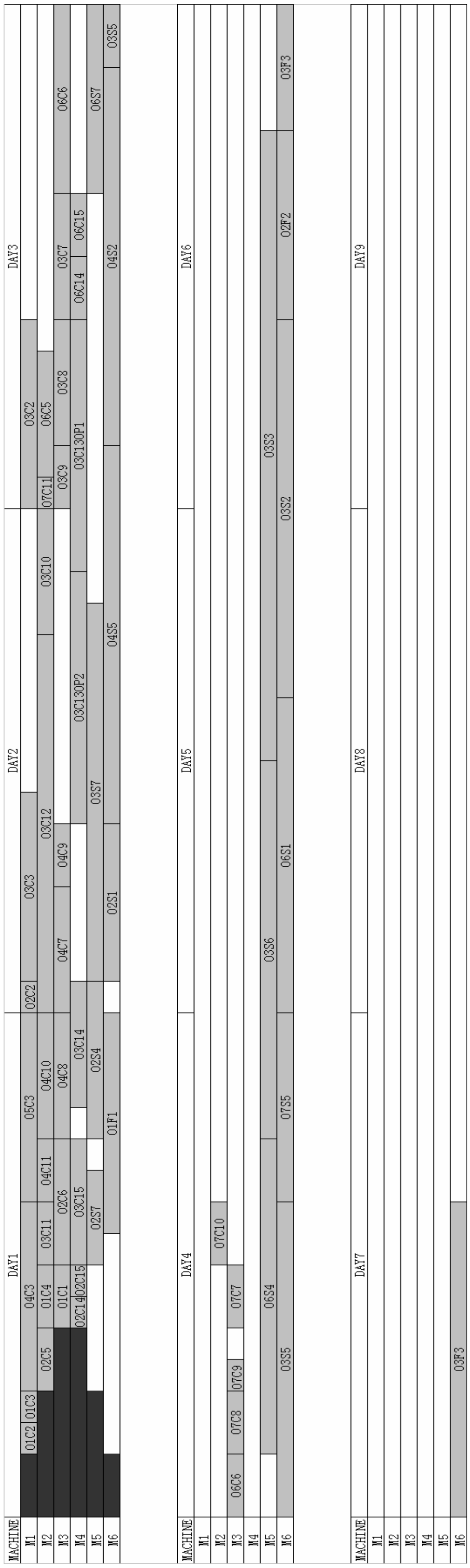


Figure 5.3 Optimal results of the representative DAPS example in the form of Gantt chart when frozen interval = 1 day

5.3.2.2 The MIP without a frozen interval

When the MIP without a frozen interval is applied to this representative DAPS problem, not only items O2F2, O3F3, O3S2, O3S3, O3S5, O3S6, O3C2, O3C7, O3C8, O3C9, O4S2 but also items O2S1, O2C2, O3S7, O3C3, O3C10, O3C12, O3C13OP1, O3C13OP2, O4S5, O4C7, O4C9, combined with the new orders, are required to construct a new schedule at the rescheduling point (*t* = Day 1). The MIP model in Appendix VIII is the mathematical formulation for this situation, and the corresponding optimal solution obtained from CPLEX is listed in Appendix IX. The summary of the optimal solution is shown in Table 5.7. The optimal production schedule is graphically represented in the Gantt chart as illustrated in Figure 5.4.

Table 5.7 Optimal results of the representative DAPS example without a frozen interval

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
821	543	466	53	426.41

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
1	11	250	550	8725

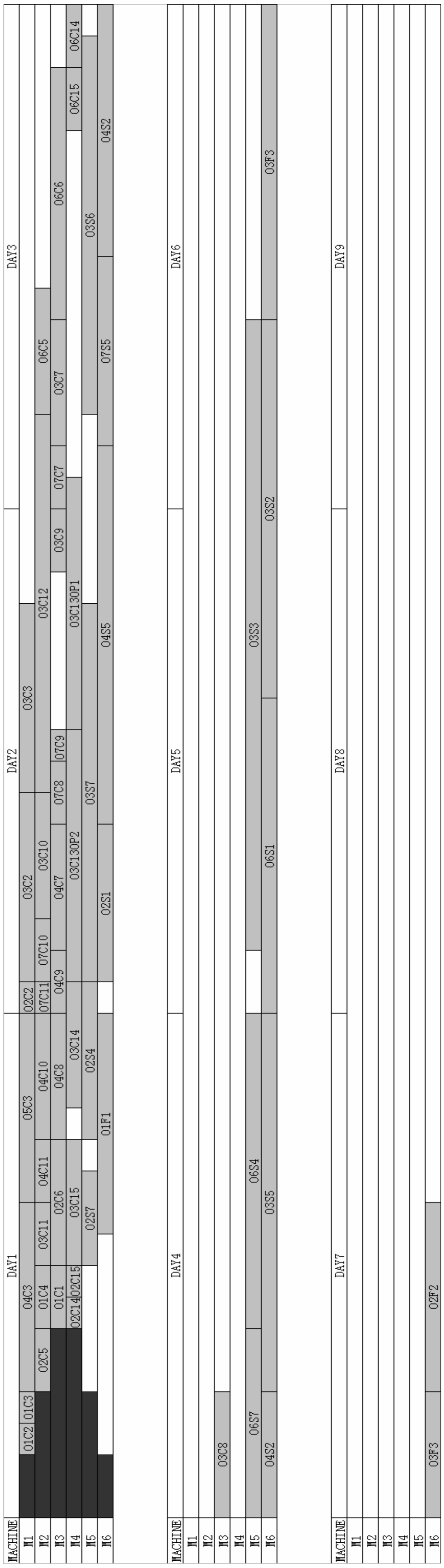


Figure 5.4 Optimal results of the representative DAPS example without a frozen interval in the form of Gantt chart

5.3.2.3 Comparison of performance

Table 5.8 lists the optimal results from different methodologies in the representative DAPS example. The tests reveal that when the frozen interval is adopted, the stability improves while efficiency worsens. Specifically, the improvement in the stability is as much as 26.79% ((172.342-126.163)/172.342*100%), while the sacrifice in the objective function is only 0.57% ((8775-8725)/8775*100%). The results are considerably exciting, since the stability improves much more than efficiency degrades. Moreover, with regard to the representative DAPS example, the MIP with a frozen interval can find its optimal solution in much less computation time.

Table 5.8 The optimal results from different methodologies in the representative DAPS example

Methodology	Objective function	Stability	CPU time (sec.)
MIP with a frozen interval	8775	126.163	0.45
MIP without a frozen interval	8725	172.342	426.41

5.3.3 GA solutions to the representative DAPS example

The GA with a frozen interval as well as without a frozen interval is utilized to solve the representative DAPS example. The results confirm that the dynamic policy improves the stability much more than it degrades efficiency.

5.3.3.1 The GA with a frozen interval

The GA method has been used to solve the original problem in Section 4.3. The best production schedule obtained by the GA shows that Orders 1, 2 and 3 are early relative to their due dates.

At the first rescheduling point (*t* = Day 1), frozen interval = 1 day (8 hours) was adopted. Thus, the fixed items are O1F1, O1C1, O1C2, O1C3, O1C4, O2S4, O2S7, O2C2, O2C5, O2C6, O2C14, O2C15, O3S7, O3C2, O3C3, O3C7, O3C8, O3C9, O3C10, O3C11, O3C12, O3C13OP1, O3C13OP2, O3C14, O3C15, O4S2, O4S5, O4C3, O4C7, O4C8, O4C9, O4C10, O4C11 and O5C3. The other items, including O2F2, O2S1, O3F3, O3S2, O3S3, O3S5 and O3S6, are outside the frozen interval, and they, together with the two recently arrived orders, are required to build a new schedule.

We apply our established GA approach to this new situation. Our methodology was coded in the C Language, as in the enclosed CD-ROM, and run on a personal computer with a Pentium 2.66 GHz CPU and 512 MB RAM. The genetic parameters were set to maximum generation = 200, population size = 100, number of reproduction = 10, number of crossover = 80, crossover probability = 0.7, number of mutation = 10. The GA-based program was replicated 5 times, and the same best operation sequences were achieved. The best solution with the total costs of 8875, as displayed in Figure 5.5, was reached in 0.968 second on average, whereas Table 5.9 lists the best schedule with the chromosome created by the genetic algorithm. Obviously, only Order 7 could not be fulfilled on time with one day delay, while Orders 1 and 3 are early. The summary of the results is illustrated in Table 5.10.

Table 5.9 The best schedule with the chromosome obtained by the GA when frozen interval = 1 day

Items	Starting time (hour)	Finish time (hour)	Random number
O2F2	50.5	53.5	0.96
O2S1	30.0	32.5	0.79
O3F3	43.5	50.5	0.95
O3S2	37.5	43.5	0.95
O3S3	32.0	42.0	0.83
O3S5	21.0	27.0	0.09
O3S6	18.0	24.0	0.03
O6S1	32.5	37.5	0.44
O6S4	27.0	32.0	0.49
O6C5	17.5	19.5	0.57
O6C6	16.5	20.5	0.24
O6S7	24.0	27.0	0.01
O6C14	16.0	17.0	0.09
O6C15	17.0	18.0	0.83
O7S5	27.0	30.0	0.71
O7C7	21.5	22.5	0.42
O7C8	20.5	21.5	0.24
O7C9	16.0	16.5	0.08
O7C10	16.0	17.0	0.34
O7C11	17.0	17.5	0.49

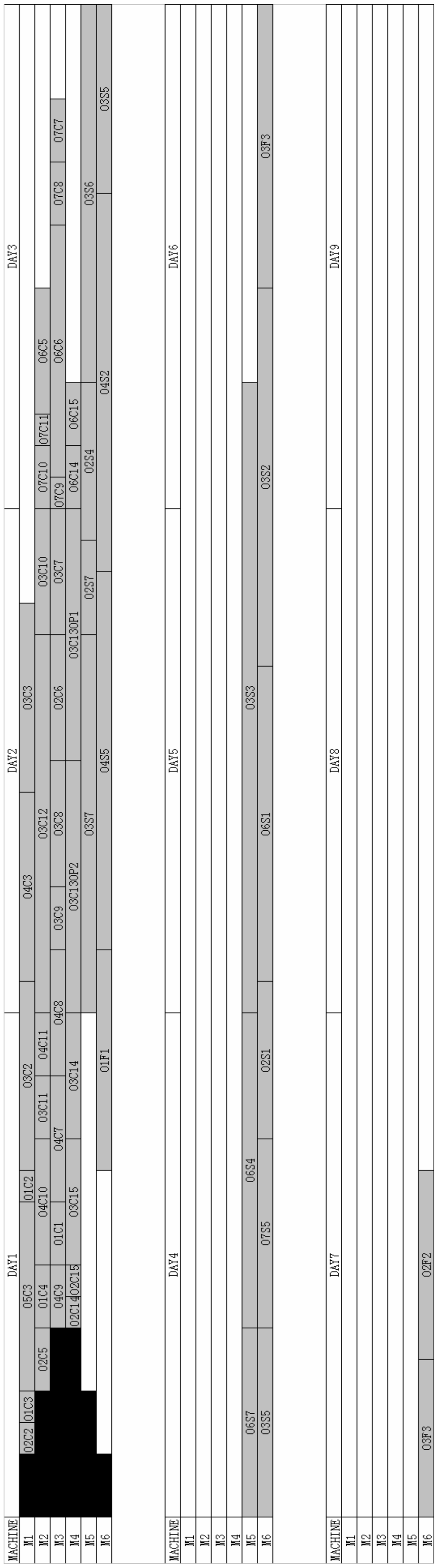


Figure 5.5 The GA results of the representative DAPS example in the form of Gantt chart when frozen interval = 1 day

Table 5.10 The best solution obtained by the GA for the representative DAPS example when frozen interval = 1 day

Makespan (hour)	Number of tardiness	Number of earliness	Total cost	CPU time (sec.)
53.5	1	11	8875	0.968

5.3.3.2 The GA without a frozen interval

The GA without a frozen interval is also implemented in the representative DAPS example to compare the results. Under such a condition, on Day 1, items O2F2, O2S1, O2S4, O2S7, O2C6, O3F3, O3S2, O3S3, O3S5, O3S6, O3S7, O3C3, O3C7, O3C8, O3C9, O3C10, O3C12, O3C13OP1, O3C13OP2, O4S2, O4S5 and O4C3, together with the newly arrived orders, need rescheduling. The problem was solved using the proposed GA on a personal computer with Pentium 2.66 GHz CPU and 512 MB RAM. The same genetic parameters as in Section 5.3.3.1 were adopted, and the GA was also replicated 5 times. The best production schedule generated by the GA from the 5 replications is summarized in Table 5.11 and graphically represented in Figure 5.6.

Table 5.11 The best solution obtained by the GA for the representative DAPS example without a frozen interval

Makespan (hour)	Number of tardiness	Number of earliness	Total cost	CPU time (sec.)
53.5	0	14	8775	2.231

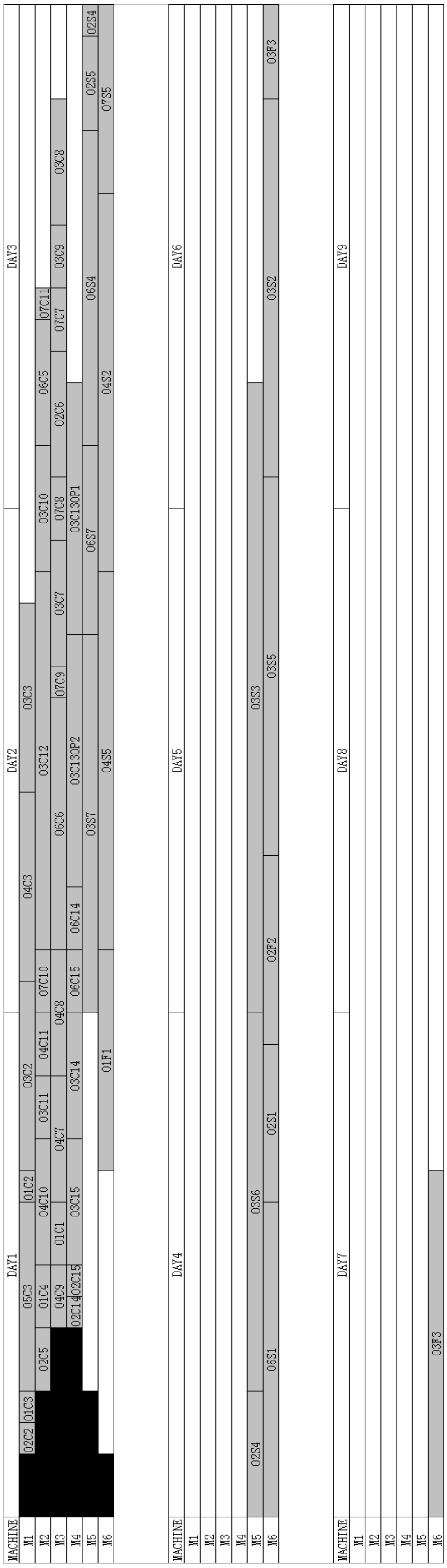


Figure 5.6 The GA results of the representative DAPS example without a frozen interval in the form of Gantt chart

5.3.3.3 Comparison of performance

Table 5.12 summarizes the GA results from different methodologies. The GA with a frozen interval improves the schedule stability by up to 40.45% ((158.816-94.574)/158.816*100%), whereas it degrades efficiency by merely 1.13% ((8875-8775)/8875*100%). The results clearly demonstrate that the proposed dynamic policy is capable of improving the stability much more than it worsens efficiency. Besides, it can be found that the GA with a frozen interval can seek its best schedule more quickly.

Table 5.12 The GA results from different methodologies in the representative DAPS example

Methodology	Objective function	Stability	CPU time (sec.)
GA with a frozen interval	8875	94.574	0.968
GA without a frozen interval	8775	158.816	2.231

5.4 SUMMARY

In this chapter, a periodic policy with a frozen interval is introduced into both the MIP model and the GA method to cope with the Dynamic Advanced Planning and Scheduling (DAPS) problem. A summary of the chapter is provided in the following.

1. Generally, assuming the manufacturing environment is static, Advanced Planning and Scheduling (APS) deals with effectively allocating production resources to complete the multi-level products so that production constraints

are satisfied and production objectives are met. In many real production situations, the problem is even more complicated because of the changing environment, where some unexpected events, such as the arrival of new orders, may arise and disrupt the manufacturing system. Such a problem henceforth is called Dynamic Advanced Planning and Scheduling (DAPS).

2. Much of the previous research on dynamic problems only takes into account efficiency performance to minimize the cost objectives like mean flow time, earliness and tardiness, etc. Usually, doing so will greatly change the production schedule when new conditions occur and induce instability, which is highly undesirable in reality.

3. This chapter studies the issue of Dynamic Advanced Planning and Scheduling (DAPS) to allow for the arrival of new orders. In order to trade off efficiency and stability, a periodic policy with a frozen interval is suggested and introduced into both the MIP in Chapter 3 and the GA in Chapter 4. The objective of the proposed methodology is to minimize cost of both production idle time and earliness-tardiness penalty for all orders including both original orders and new orders at each rescheduling point.

4. The two examples in Chapter 3 with the arrival of new orders are illustrated to examine the effectiveness of the proposed mechanism in the dynamic environment. The numerical results confirm that the presented methodology can improve the schedule stability while retaining efficiency.

In the previous chapters, we have developed a mathematical model and a GA method for the APS problem. Both static and dynamic APS have been studied. In order to apply these results, a prototype of the Advanced Planning and Scheduling Decision Support System (APSDSS) will be constructed in the next chapter. The infrastructural framework and the functional modules included in the system will be discussed. An example will be offered to validate the applicability of the proposed system.

CHAPTER 6

A PROTOTYPE OF THE ADVANCED PLANNING AND SCHEDULING DECISION SUPPORT SYSTEM (APSDSS)

6.1 INTRODUCTION

With the advances in computer technologies, many production management decision support systems have been developed for various manufacturing processes [Tsu91, Art97, Kov05]. As shown in Figure 6.1, conventional decision support systems in production planning and control are structured on the basis of the hierarchical production planning (HPP) principle [Mck95, Per99, Ria01]. The general strategy of HPP is that the higher level (planning) creates a production plan for the lower level (scheduling) to detail. However, the higher level only uses aggregated information and does not consider the inner workings of the lower level. In other words, the hierarchical paradigm deals with different levels with different scopes and objectives. Most of the decision support systems based on the paradigm suffer from incompatibility of decisions at different levels, and create many problems on the shop floor for later production. To be effective, production planning and scheduling should be integrated together rather than separately, when designing such a production decision support system.

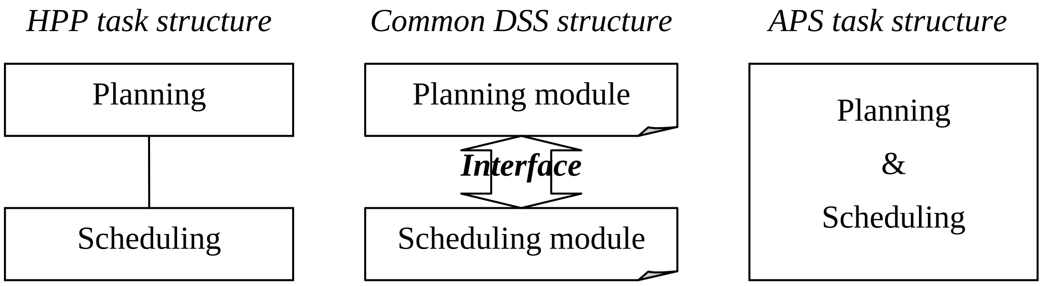


Figure 6.1 Production control tasks and decision support [Mck03]

With reference to developing decision support for Advanced Planning and Scheduling (APS), both academia and commercial companies have invested significant efforts, as reviewed in Section 2.5. Nevertheless, most of the efforts have restricted themselves to embed trial-and-error methods in their computer-based systems. Better production plans and schedules can be generated by decision support tools with the employment of intelligent heuristic approaches, such as genetic algorithm (GA).

This chapter presents a seamless decision support system, on the basis of the developed genetic algorithm (GA), for integrating production planning and scheduling. As in the enclosed CD-ROM, the Advanced Planning and Scheduling Decision Support System (APSDSS) is such a Windows application that can manage the data electronically, handle the Advanced Planning and Scheduling (APS) problem as well as the Dynamic Advanced Planning and Scheduling (DAPS) problem efficiently, and create the production plans and schedules automatically. Moreover, APSDSS is a promising decision support tool for production planners. It will not only free the production planners from the labor-concentrated jobs, such as constructing the production plans and schedules, but also assist them to take effective decisions depending on the various situations of the manufacturing system. The APSDSS was implemented using VC++, an object oriented programming language. The following reasons lead to choose this object oriented approach [Art97, Hai01, Sch05]:

- Simplicity. It is natural to represent the real world objects (products, orders and machines) using VC++.
- Modularity. Each object forms an independent entity whose internal procedures or methods are separated from other entities of the system in VC++.
- Extensibility. It is easy to introduce new functionalities or features to response to changing environments by use of VC++.
- Reusability. An object can be a standard one across systems and can be reused in VC++. Not only does reusing speed the development, but also it improves the quality of the system.

The remainder of this chapter is structured as the following three sections. In Section 6.2, the functional architecture of the decision support system is developed. The use of the system with an illustrative example is presented in Section 6.3. Finally, Section 6.4 is the summary of the chapter.

6.2 FUNCTIONAL ARCHITECTURE OF THE APSDSS

The principle of the Advanced Planning and Scheduling Decision Support System (APSDSS) is to consider operation sequences among items, capacity constraints of the manufacturing system as well as the dynamic arrival of orders, and translate the orders of the various products into detailed production schedules with operation starting time and finish time. The infrastructure of the proposed APSDSS, which aggregates the different levels of the order fulfillment process, is depicted in Figure 6.2. The APSDSS was implemented using VC++, an object oriented programming language, and is composed of four major elements:

- A database
- An advanced planning and scheduling module
- A performance evaluation module
- A set of interfaces for interactively using the APSDSS

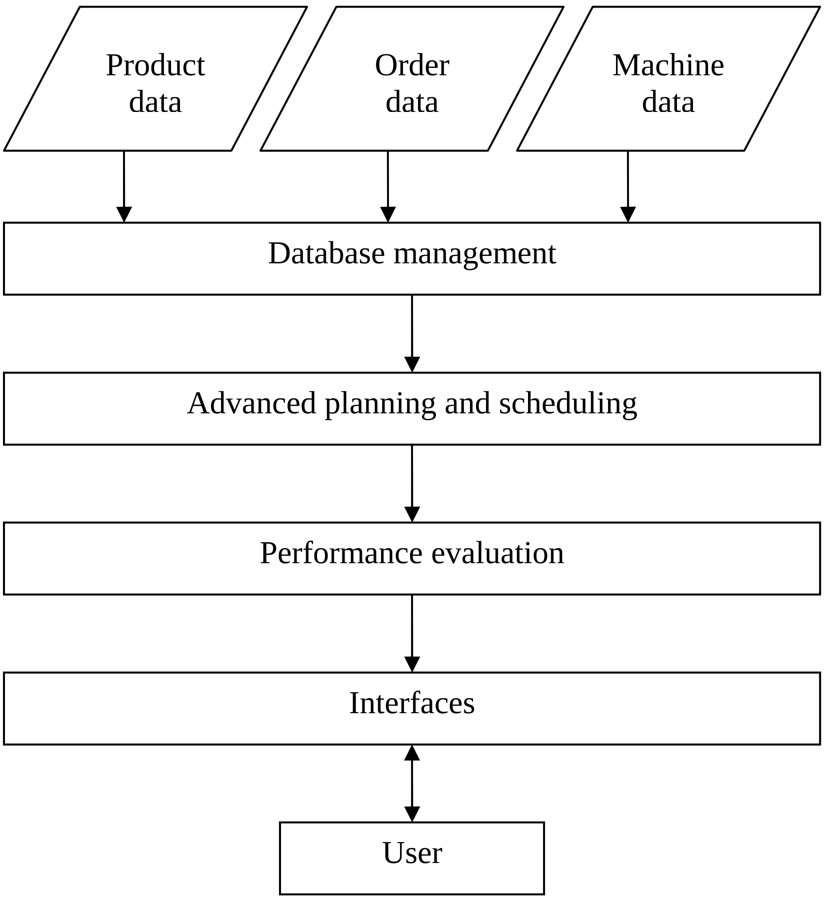


Figure 6.2 Infrastructure of the APSDSS

The following sections introduce each of the subsets of the APSDSS in detail.

6.2.1 Database management

A database plays a crucial role in the functionality of the decision support system. Great efforts are required to make a system's database accurate, consistent and complete. The database of the APSDSS contains information about the manufacturing system under study, and is maintained in Microsoft Access. Specifically, the data information includes:

- Products (names, BOMs, processing times, manufacturing machines)
- Orders (names, ordered products, ordered quantities, due dates)
- Machines (names, ready times, time capacities)

The product information is the basic data required in the APSDSS system. They include items names, BOMs, processing times, and manufacturing machines. A BOM defines the production information of a final product by specifying the precedence constraints among the items needed to make the product, together with the quantities of the items at each operation. Meanwhile, the eligible machines and the processing time for the items are also described in the product data. In terms of the representative example in Section 3.4.3, the corresponding product data are shown in Figure 6.3. For instance, the final product F1, requiring 0.70 hour on M6, is composed of four components C1, C2, C3, and C4; then, it is defined as:

```
{
        Item name = "F1";
        Father = NULL;
        Quantity = 1;
        Child number = 4;
        Processing time = 0.70;
        Machine = 6;
}
```

With regard to the common items, their parent (father) items are also included in their names. Thus, the component C2 of F1 is expressed as:

```
{
        Item name = "C2(F1)";
        Father = F1;
        Quantity = 1;
        Child number = 0;
        Processing time = 0.10;
        Machine = 1;
}
```

The order data are provided by customers and involve the detailed requirements for the products. They are made up of four components: order names, ordered products, ordered quantities, and due dates. The ordered products can be the final products and also any items described in the product data.

The machine-related attributes, such as machine names, ready times, and time capacities, are clarified in the machine data. The ready times specify the times when the machines become available, while the time capacities state how long the machines work a day.

ID	Item name	Father	Quantity	Child Number	Processing time	Machine
1	F1	NULL	1	4	0.70	6
2	C1(F1)	F1	1	0	0.20	3
3	C2(F1)	F1	1	0	0.10	1
4	C3(F1)	F1	1	0	0.10	1
5	C4	F1	1	0	0.20	2
6	F2	NULL	1	2	0.60	6
7	C2(F2)	F2	1	0	0.10	1
8	S1	F2	1	2	0.50	6
9	S4	S1	1	2	0.50	5
10	C5	S1	1	0	0.20	2
11	C6	S4	1	0	0.40	3
12	S7(S4)	S4	1	2	0.30	5
13	C14(S4S7)	S7(S4)	1	0	0.10	4
14	C15(S4S7)	S7(S4)	1	0	0.10	4
15	F3	NULL	1	3	0.70	6
16	S2	F3	1	2	0.60	6
17	S3	F3	2	2	0.50	5
18	C2(F3)	F3	3	0	0.10	1
19	S5	S2	1	5	0.60	6
20	C3(S2)	S2	3	0	0.10	1
21	S6	S3	1	2	0.30	5
22	S7(S3)	S3	1	2	0.30	5
23	C7	S5	1	0	0.20	3
24	C8	S5	1	0	0.20	3
25	C9	S5	1	0	0.10	3
26	C10	S5	1	0	0.20	2
27	C11	S5	1	0	0.10	2
28	C12	S6	1	0	0.30	2
29	C13OP1	S6	1	1	0.20	4
30	C14(S3S7)	S7(S3)	1	0	0.10	4
31	C15(S3S7)	S7(S3)	1	0	0.10	4
32	C13OP2	C13OP1	1	0	0.20	4

Figure 6.3 The product data of the representative example

6.2.2 Advanced planning and scheduling

Both APS and DAPS have been proved to be NP-hard, and many heuristic methods have been widely studied in the literature [Moo03, Vie03]. The plans and schedules generation procedure of the proposed APSDSS is based on the developed genetic algorithm (GA).

Our GA encoding scheme utilizes the concept of random keys as discussed earlier. This scheme encodes a solution with a string of random numbers. Each item in the product structure has one random number generated from the range [0, 1]. For the DAPS problem, a periodic policy with a frozen interval is introduced into the GA. Consequently, the length of the random number string equals to the number of unfrozen items in both original orders and new orders, that is, each unfrozen item takes a random value in the range [0, 1]. These random numbers denote the priorities of the items, while the smaller value represents the higher priority. The random key encoding has the advantage that it eliminates the offspring feasibility problem and is robust to problem structures [Bea94].

The established genetic algorithm starts with generating the chromosomes randomly as much as the desired population size. Each chromosome contains a string of random numbers that represent the priorities of the genes, and is evaluated using the fitness function given in Equation (4.1), which aggregates production idle time, earliness and tardiness penalty. The well-known roulette wheel approach [Gol89] is employed for choosing some chromosomes to conduct genetic operations. Genetic operations such as reproduction, crossover and mutation are executed to produce a new set of chromosomes called offspring. There are many variations of genetic operations that could be used in GAs. Since random key encoding preserves to create feasible solutions, there is no need to design specialized operations. The genetic operations employed here are elitist reproduction, parameterized uniform crossover and immigration, which have been proved very robust in computational tests. Then a new generation is formed by selecting some of the parents and offspring according to their fitness and rejecting the others to keep the population size constant. When the maximum generation is reached, the algorithm converges to the best chromosome.

6.2.3 Performance evaluation

The quality of a generated operation sequence is evaluated using the performance measure of minimizing the total costs of both production idle time and tardiness or earliness penalty of an order. Minimizing production idle time is equivalent to minimizing production flow time or maximizing machine utilization. Meanwhile, earliness and tardiness penalty is chosen as the performance measure because it is able to reflect the just-in-time (JIT) rule, which is on the fact that either early or late delivery of an order results in an increase in the production costs.

With respect to the DAPS problem where new orders arrive on a continuous basis, the APSDSS utilizes the GA method with the dynamic policy to determine the best production plan and schedule such that both production idle time and penalties on tardiness and earliness of both original orders and new orders are minimized at each rescheduling point.

6.2.4 Interfaces

The interface module offers the user the possibility of access to the APSDSS. The menu structure of the APSDSS is illustrated in Figure 6.4.

Figure 6.4 The menu structure of the APSDSS

The user can operate the menu to input his/her APS as well as DAPS problems according to the general definition and assumptions of the manufacturing system given in the previous chapters. The menu helps to specify the suitable attributes of the products, orders and machines. If the user wants to make some changes to the data set of a particular problem, he/she can interactively add, modify, or delete the relevant information in the windows. Once the problem configuration is determined, the user can run the embedded genetic algorithm to obtain the operation sequences. The APSDSS system also gives the user the control over the parameters. When the production schedule is derived by the GA, a Gantt chart window is displayed on the screen. The Gantt chart window graphically illustrates the generated schedule, and provides a good overview of the schedule relative to time. Simultaneously, a text file will be created and reports the produced results. In the text file, the generated schedule information, such as starting time, finish time as well as random key, is listed next to the corresponding operation, together with the performance measures. Such a display has the advantage that when there are many orders and machines, it is easy to recognize all attributes of each processing operation, including the resulting ones.

6.3 USE OF THE APSDSS

A prototype software system has been established based on the APSDSS infrastructure proposed in this chapter. In the following, the representative example in Section 5.3.2, which deals with five orders at the beginning of the planning horizon (t = 0) and then receives two new orders at the first rescheduling point (t = Day 1), is given to illustrate the effectiveness of the developed system.

Actually, there are two ways to input the product, order, and machine data. One is to enter the data by screen dialogues, and the second type is to input the data in the Access database. When the system is launched, the default database is accessed. Then, the data can be easily inputted or modified by operating the menu. The snapshots of the data input screens are demonstrated in Figures 6.5, 6.6 and 6.7.

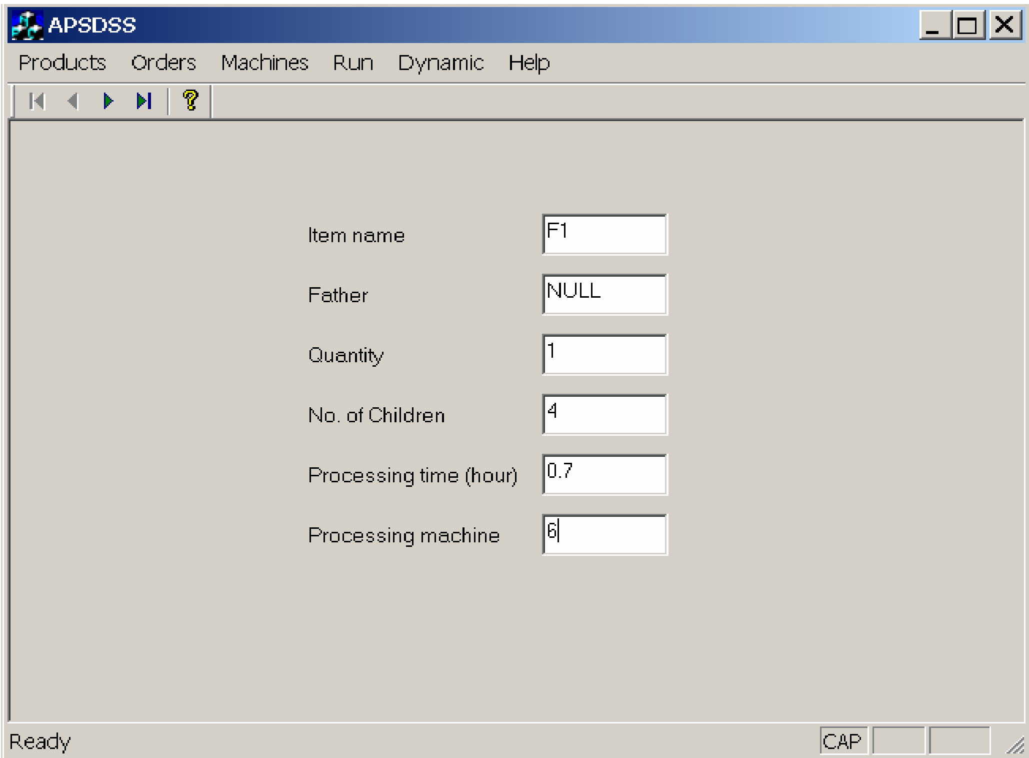


Figure 6.5 The product data input screen

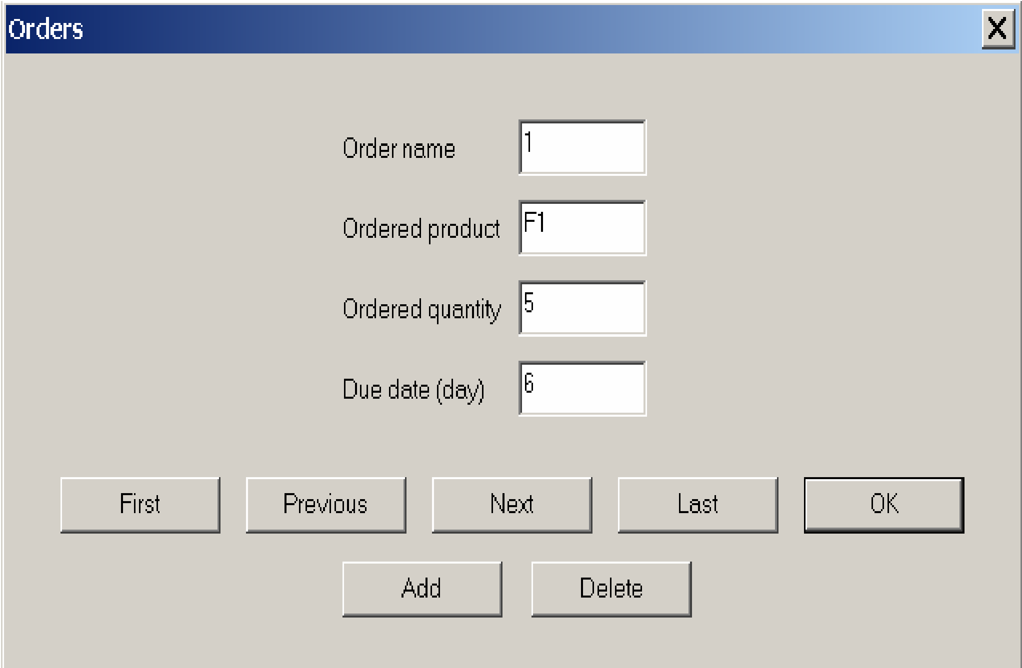


Figure 6.6 The order data input screen

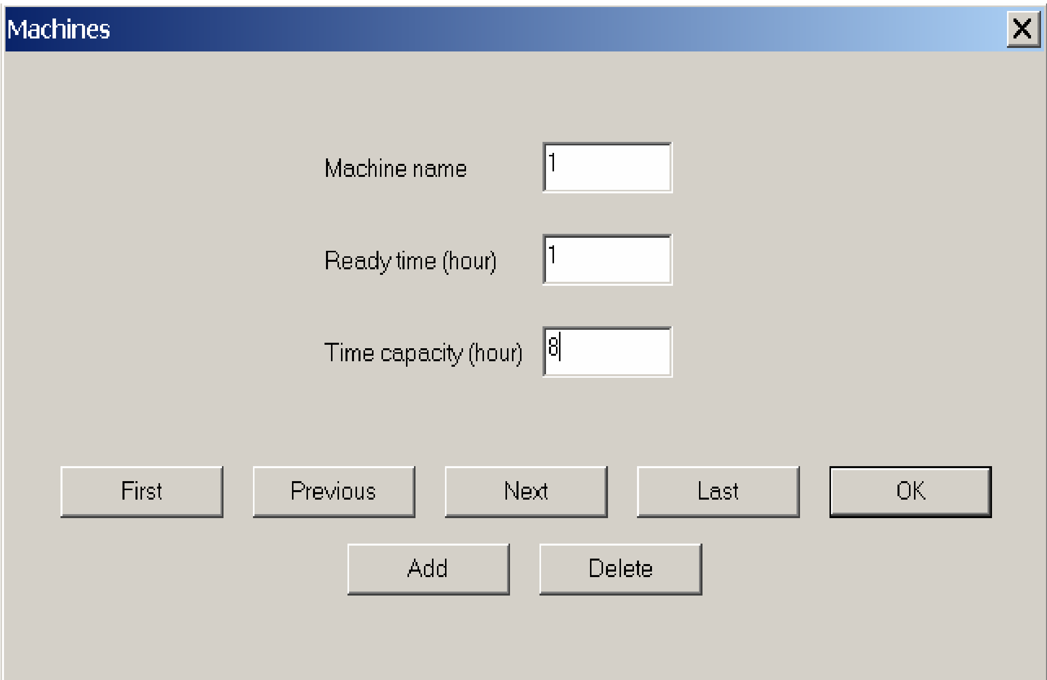


Figure 6.7 The machine data input screen

The original problem is to find a detailed plan and schedule for the orders such that both production idle time and tardiness or earliness penalty of an order are minimized. To solve the problem, the user can choose **Run | Gantt chart** on the menu bar (Figure 6.8).

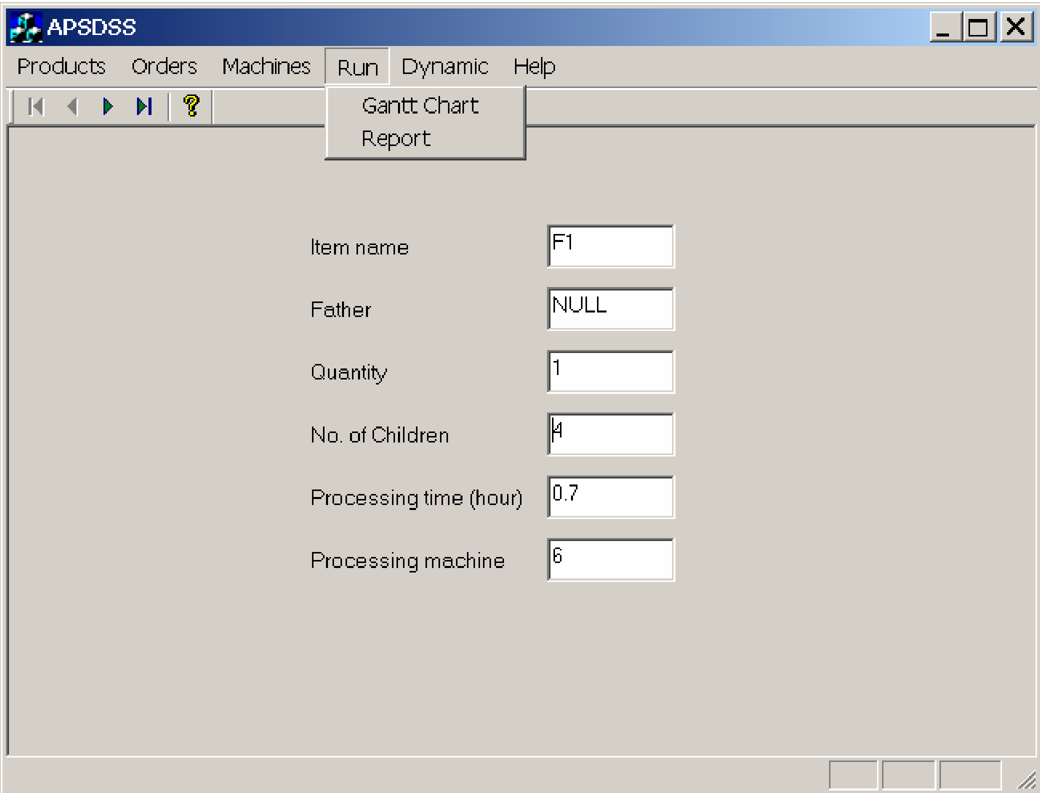


Figure 6.8 The "Run" menu of the APSDSS

Thereafter, it is necessary for the user to input the penalty rates as well as the genetic parameters. The penalty rates are as follows: cost of idle time at $50 per hour, cost of tardiness at $250 per day per order, and cost of earliness at $50 per day per order, while the genetic parameters are set to maximum generation = 200, population size = 100, number of reproduction = 10, number of crossover = 80, crossover probability = 0.7, number of mutation = 10. Figure 6.9 depicts the dialogue box for specifying the parameters.

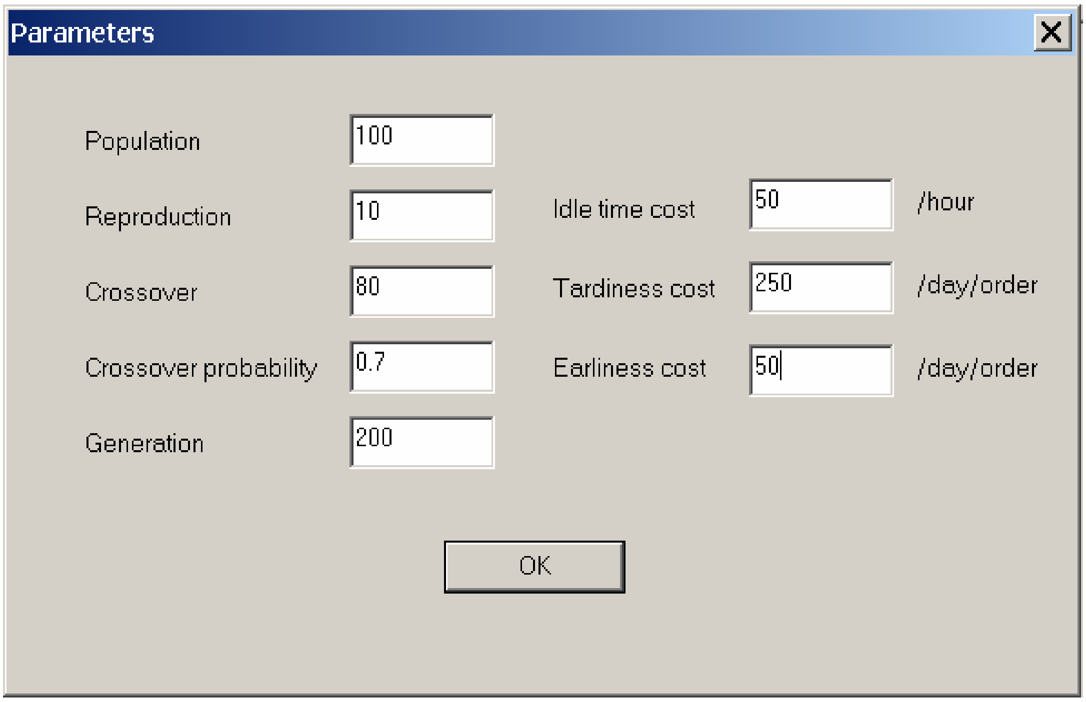


Figure 6.9 The parameters input screen

By calling the embodied GA approach, the system generates the best schedule with operation starting time and finish time, which is graphically represented in a Gantt chart as portrayed in Figure 6.10. When the user positions the mouse cursor on an operation and leaves it there for a certain interval, a tip appears and displays the name of the operation.

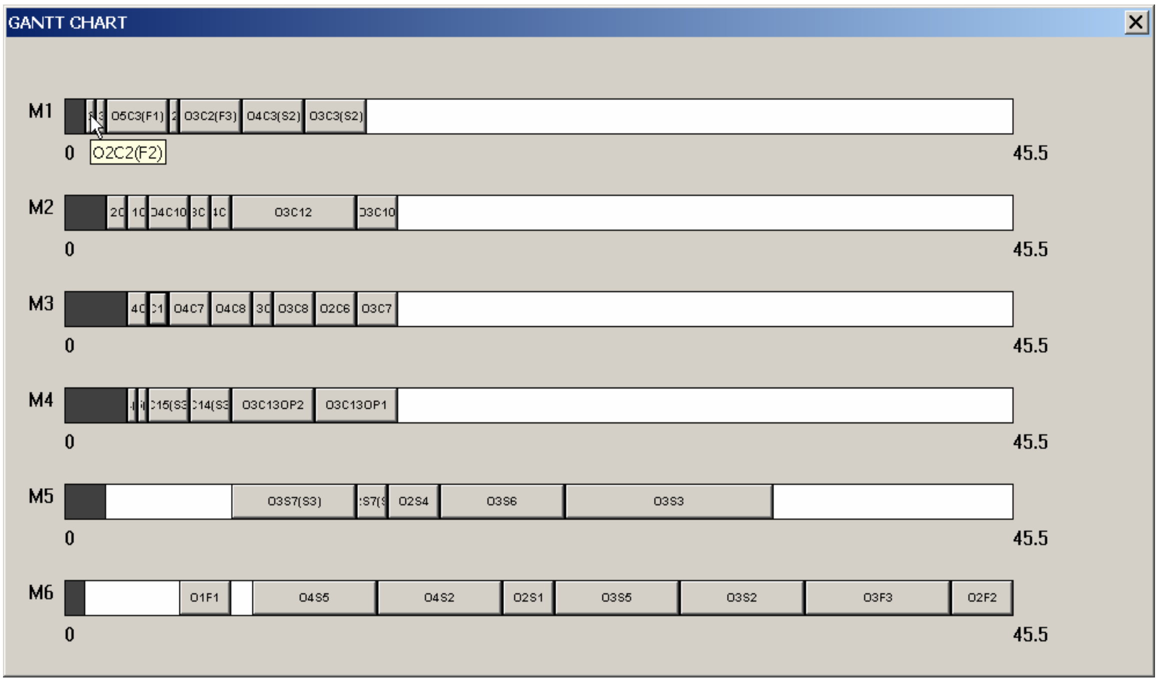


Figure 6.10 The Gantt chart output window of the original problem

If the user wants to know more about the produced results like those in Tables 4.10 and 4.11, he/she can press the menu **Run | Report** (Figure 6.8) and open a text file to see each processing operation's starting time, finish time and random key, together with the resulting performance measure.

On Day 1 when two new orders arrive, one can select **Dynamic | Gantt chart** in the menu and utilize the GA with the dynamic policy to handle the dynamic situation (Figure 6.11). After the parameters in Figure 6.9 and 6.12 have been determined, the system produces the Gantt chart as shown in Figure 6.13. The detailed results like those in Tables 5.9 and 5.10 can be viewed by clicking **Dynamic | Report**.

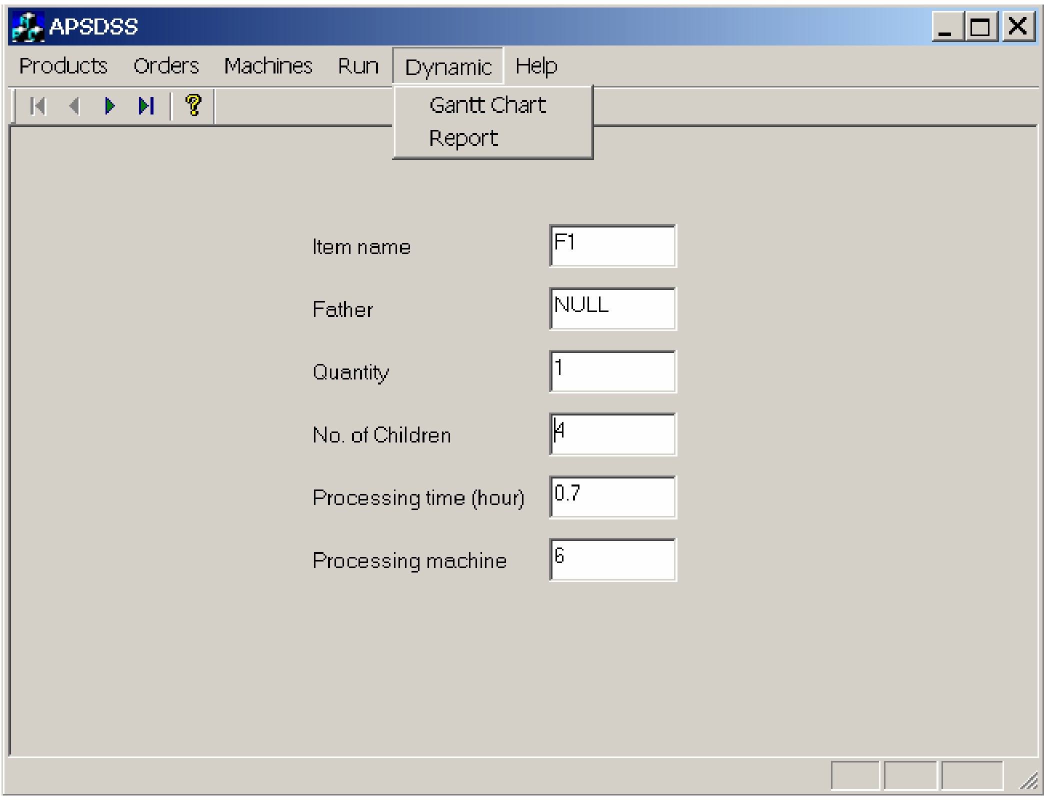


Figure 6.11 The "Dynamic" menu of the APSDSS

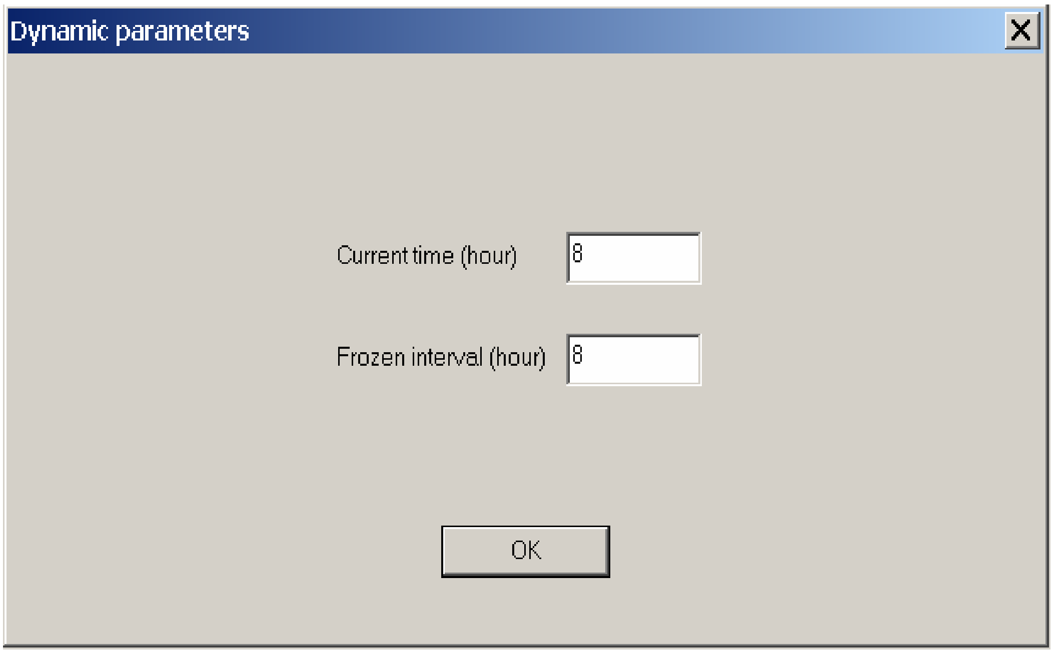


Figure 6.12 The dynamic parameters input screen

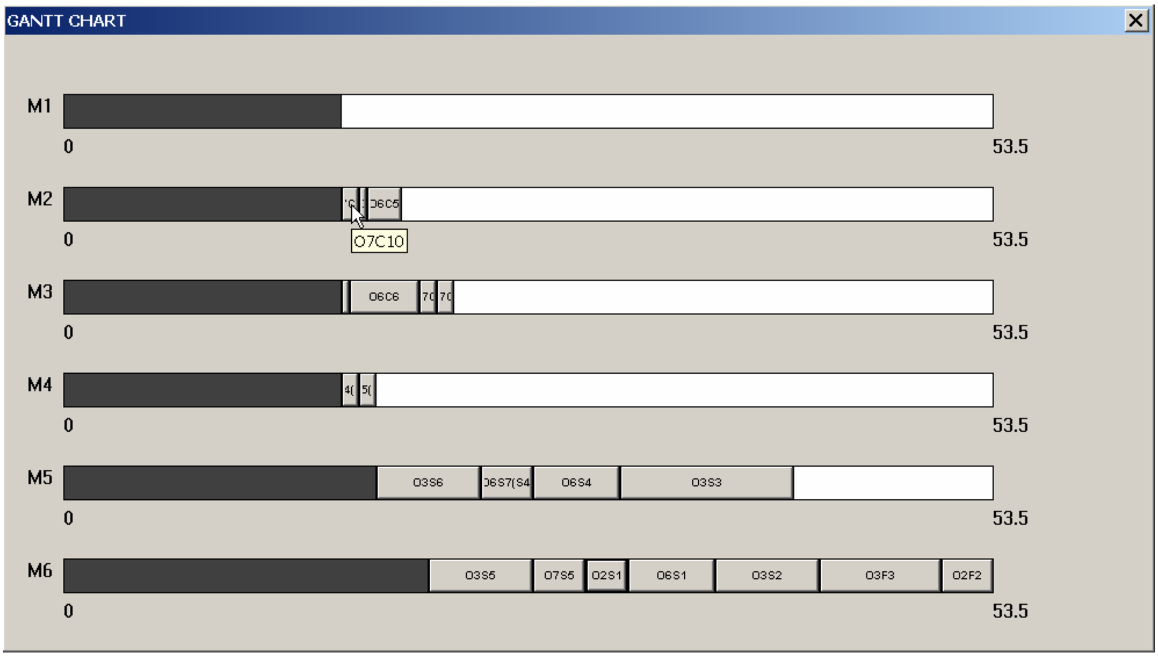


Figure 6.13 The Gantt chart output window of the dynamic problem

The results indicate that the decision support system seamlessly integrates production planning and shop floor scheduling, and efficiently produces effective operation sequences that take into account the real-life constraints as well as the dynamic condition.

6.4 SUMMARY

In this chapter, a prototype of the Advanced Planning and Scheduling Decision Support System (APSDSS) has been established. A brief conclusion is drawn as follows.

1. In order to avoid incompatibility of decisions at different levels, production planning and scheduling should be combined together rather than separately, when designing the production decision support system.
2. This chapter proposes an infrastructural framework, involving various functional modules, for the development of an Advanced Planning and Scheduling Decision Support System (APSDSS). The system employs a GA-based method to generate realistic plans and schedules for the shop floor, and is endowed with a set of interfaces for easy implementation.
3. The same example as in Section 5.3.2 is illustrated to validate the applicability of the constructed system. The implementation indicates that the decision support system seamlessly integrates production planning and shop floor scheduling, and efficiently produces effective operation sequences that take into account the real-life constraints as well as the dynamic condition.
4. The advantages of the APSDSS are:
    - Finding the effective production plans and schedules for the APS problem as well as the DAPS problem in a short time;
    - Allowing extension of the functionalities with ease;
    - Providing user-friendliness and user control;
    - Improving the management of the data.

We have established a decision support system for APS, so it is natural to apply the system to a real situation. A case study for the Advanced Planning and Scheduling (APS) problem in a light source manufacturer will be reported in the next chapter. The case problem and the computational results obtained by use of the developed APSDSS will be described in detail.

CHAPTER 7

A CASE STUDY FOR ADVANCED PLANNING AND SCHEDULING (APS)

7.1 INTRODUCTION

Many manufacturing firms produce products with a multi-level structure, that is, final products comprise several subassemblies and components, and each subassembly may also require subassemblies and components. Such a product structure specifies the dependent relationships and precedence constraints among the items. This structural complexity associated with multi-level products planning and scheduling arouses unique coordination problems that do not exist when scheduling string-type operations in the general job shop. Advanced Planning and Scheduling (APS) deals with effectively allocating production resources to complete the multi-level products so that production constraints are satisfied and production objectives are met [Lee02, Moo04]. This chapter presents a case study for the Advanced Planning and Scheduling (APS) problem encountered in a light source manufacturer.

The chapter is arranged in the following way. Section 7.2 describes the case problem in detail. The computational results obtained on applying the APSDSS to the case study are presented in section 7.3. Finally, section 7.4 concludes the chapter with some remarks.

7.2 THE CASE PROBLEM

The Advanced Planning and Scheduling (APS) problem solved and reported in this chapter is extracted from a company located in China, which is one of the technology and market leaders in the production of specialist light sources. In 2004, the company had an annual turnover of 760 million RMB and employed 626 workers and staff. They develop, manufacture and market infrared heaters and ultraviolet lamps for applications in manufacturing, industrial process technology, environmental protection, medicine and cosmetics, research and analytical measurement technology. Currently, there are approximately 3000 different types of light sources, and the products can be grouped in the following main categories, as samples shown in Figure 7.1.

(a) infrared heaters and systems

(b) ultraviolet light sources for optical and analytical instrumentation

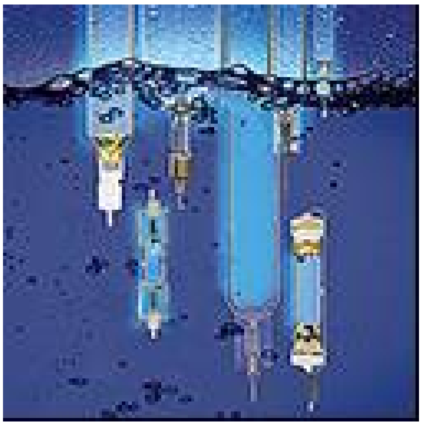
(c) ultraviolet lamps for disinfection and oxidation

(d) ultraviolet lamps for curing and exposure

(e) pulse and continuous wave laser lamps

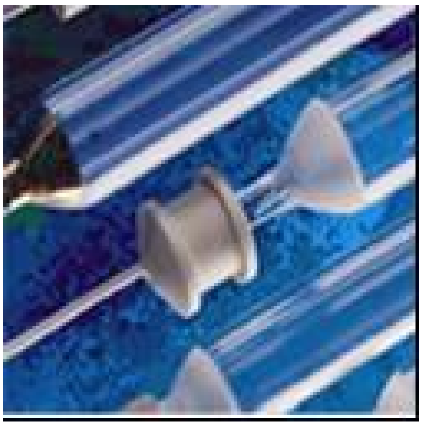
(a)

(b)



(c)

(d)



(e)

Figure 7.1 The sample products

The specialist light source products are manufactured through eight basic operations including cutting, polishing, slotting, welding, finishing, manual processing, grinding, and inspection. A light source may need processing of all eight operations or fewer operations based on combinations of these eight operations. Technological requirements specify an order in which operations must be processed. The maximum operations per product is 30 while the minimum is two. There is a variety of parallel processors to complete the operations in the shop. Some processors are eligible for fulfilling two or more operations. Each operation can be processed on at most one processor at a given time, and once started, must be finished before another operation may be started on that processor. A processor can perform one operation at a time and works for 24 hours a day. In this chapter, we ignore the parallel situation and adopt only one processor of each type with modified processing times to reflect the parallel property.

The production shop operates on a make-to-order (MTO) basis. The unit processing time is multiplied by a factor to cover lost time due to non-availability of processors. A lot-for-lot strategy is employed for making items. The unit processing time is multiplied by the lot size, while the setup times and the transfer times between operations are negligible or are included in the processing times.

The company’s goals are to schedule the production on the shop floor in order to minimize costs of both production idle time and tardiness or earliness penalty of an order. The minimum production idle time implies less production flow time or higher utilization of the machines, which after discussion is identified as an important objective to be considered. The company also feels that to find a schedule with all orders completed as close to their due dates as possible is essential for sustainable business. This stems from the fact that if a product is finished earlier than its due date, inventory carrying costs will increase. On the other hand, when products are late relative to their due dates, tardiness costs will be incurred due to contract penalties, goodwill losses, and so on.

At present, the production planning and scheduling is performed by the staff in the production department. Considering the availability of production capacity and the orders provided by the marketing department, they derive the production plan and schedule based on their own experiences. The planning period for the production is four weeks, rolling week by week. One of the main problems with the current planning procedure is that the production shop is usually behind schedule which results in many tardy jobs. Moreover, it is very difficult for the planners to trace all of about 3000 products. Due to the difficulty of tracing, the production completion times of the orders cannot be exactly determined or anticipated. Now, the company is experiencing a great surge in demands. Better planning and scheduling is required for improving the productivity without a proportional increase in the production facilities.

7.3 COMPUTATIONAL RESULTS

Our APSDSS developed in Chapter 6 is tested using a set of data describing the current scenario of the specialist light source manufacturing company. Due to business confidentiality, the real data have been modified after discussions with the production department of the company.

Three typical product structures were chosen as shown in Figure 7.2. Besides, it should be noted that C12 is a common item, which is shared by subassembly S4 and component C13. A customer may order the final products F1, F2 and F3, and also some major components, like S1 and C13 etc. Six machines, with 24 hours available per day, are eligible to process the items (Table 7.1). The penalty rates are as follows: cost of idle time at $50 per hour, cost of tardiness at $250 per day per order, and cost of earliness at $50 per day per order.

At the beginning of the planning horizon (t = 0), the ready times of these six machines are Hour 16, Hour 4, Hour 10, Hour 4, Hour 2 and Hour 1, respectively. There are 35 orders for the specialist light sources, as depicted in Table 7.2. The rescheduling interval is determined as 1 day, that is, 24 hours. Then, at the first rescheduling point (t = Day 1), three new orders are received: 5 Product F1s with due date Day 100, 10 Product S3s with due date Day 55, and 20 Product C12s with due date Day 14.

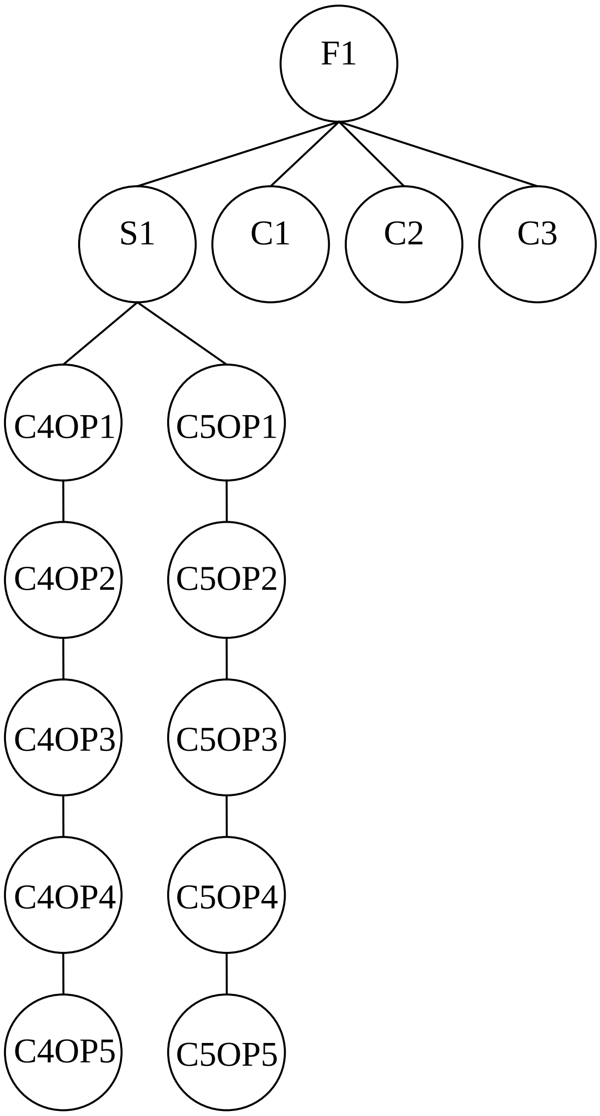


Figure 7.2 (a) The product structure of F1 in the case study

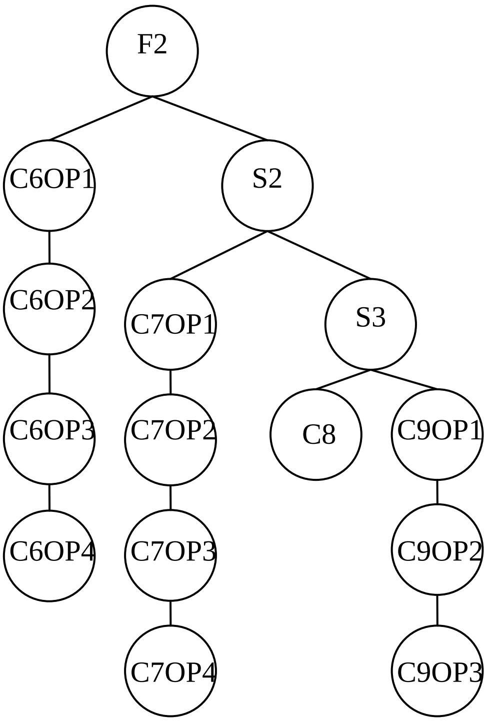


Figure 7.2 (b) The product structure of F2 in the case study

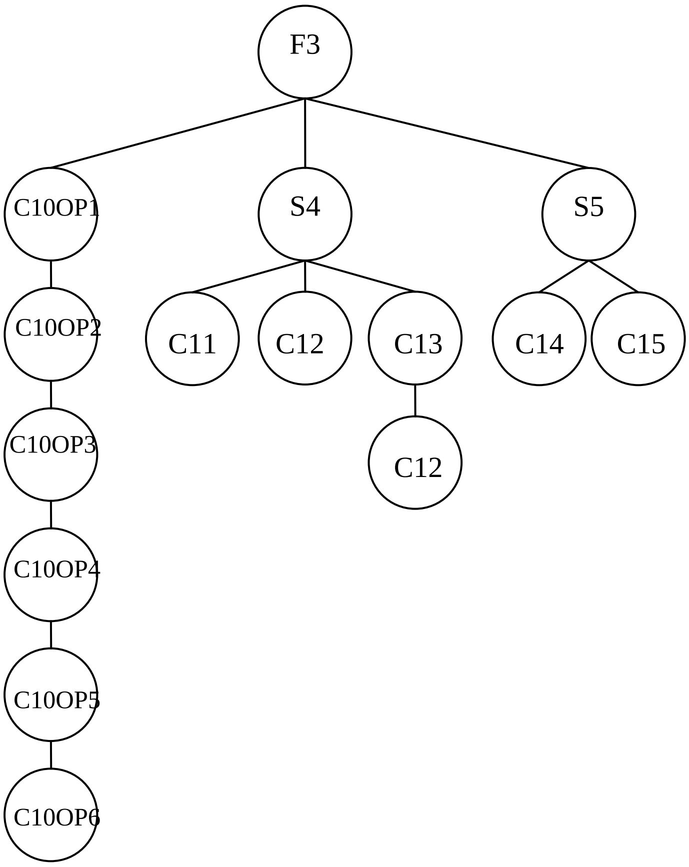


Figure 7.2 (c) The product structure of F3 in the case study

Table 7.1 Machine processing time for the items in the case study

Items	Machine number	Processing time (hours)
F1	M1	0.2
F2	M1	0.2
F3	M1	0.2
S1	M1	0.2
S2	M1	0.2
S3	M1	0.2
S4	M1	0.2
S5	M1	0.2
C1	M2	1
C2	M2	0.5
C3	M2	0.5
C4OP1	M3	8
C4OP2	M4	1
C4OP3	M6	5
C4OP4	M5	3
C4OP5	M4	1
C5OP1	M3	8
C5OP2	M5	3
C5OP3	M4	2
C5OP4	M6	6
C5OP5	M5	3
C6OP1	M3	8
C6OP2	M4	1
C6OP3	M5	3
C6OP4	M6	5
C7OP1	M3	8
C7OP2	M6	6
C7OP3	M5	3
C7OP4	M4	2
C8	M2	1
C9OP1	M4	1
C9OP2	M6	6
C9OP3	M4	2
C10OP1	M3	10
C10OP2	M6	5
C10OP3	M5	3
C10OP4	M4	1
C10OP5	M5	3
C10OP6	M4	1
C11	M2	1
C12	M5	3
C13	M4	1
C14	M2	0.5
C15	M2	0.5

Table 7.2 Specialist light source customer orders

Order	Item	Amount	Due date
1	F1	10	80
2	F1	15	130
3	F2	5	28
4	F2	5	50
5	F3	15	98
6	F3	20	105
7	S1	20	154
8	S1	10	180
9	S2	10	120
10	S2	15	140
11	S4	10	70
12	S4	5	90
13	S4	20	160
14	S5	5	60
15	S5	30	126
16	S3	10	84
17	S3	15	100
18	C4	1	30
19	C4	5	48
20	C5	5	100
21	C5	2	112
22	C5	1	120
23	C6	10	20
24	C6	7	40
25	C7	2	42
26	C7	10	60
27	C7	1	80
28	C9	5	24
29	C9	15	56
30	C10	3	28
31	C10	5	42
32	C12	50	36
33	C12	30	150
34	C13	15	60
35	C13	20	90

To solve the original problem, the genetic parameters were set to maximum generation = 200, population size = 200, number of reproduction = 20, number of crossover = 160, crossover probability = 0.7, number of mutation = 20. The experiments were repeated five times and run on a personal computer with Pentium 2.66 GHz CPU and 512 MB RAM. The best solution obtained by the system over the five runs is illustrated in Appendix X and summarized in Table 7.3. Besides, Table 7.4 presents the more detailed information about the order tardiness and earliness. The best operation sequences are graphically represented in a Gantt chart as shown in Figure 7.3. For convenience, item p of order O_i is denoted Oip.

Table 7.3 The case results obtained by APSDSS and by hand when *t* = 0

Schedule	Makespan (hour)	Number of Tardiness	Number of Earliness	Total Cost	CPU Time (sec.)
APSDSS	2439	268	689	493500	122.953
Manual	3148	820	652	842350	-

Table 7.4 Order tardiness and earliness in the case study when *t* = 0

Order	Tardiness (days)	Earliness (days)
1	0	6
2	0	34
3	46	0
4	12	0
5	0	10
6	0	9
7	0	52
8	0	99
9	0	25
10	0	45
11	0	22
12	0	16
13	0	58
14	14	0
15	0	38
16	0	9
17	0	26
18	46	0
19	42	0
20	0	25
21	0	24
22	0	32
23	0	1
24	20	0
25	0	12
26	0	31
27	0	37
28	21	0
29	0	15
30	30	0
31	36	0
32	1	0
33	0	57
34	0	1
35	0	5
Total	268	689

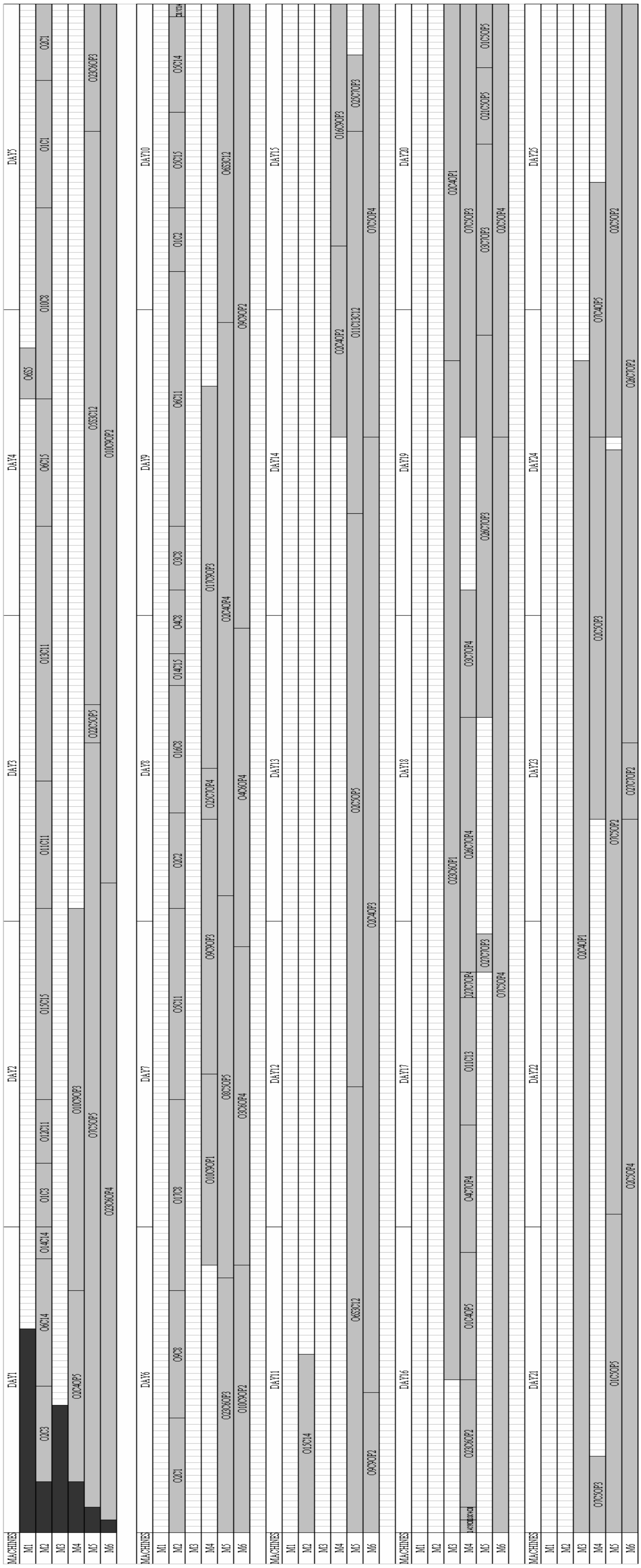


Figure 7.3 (a) The case output in the form of Gantt chart when *t* = 0

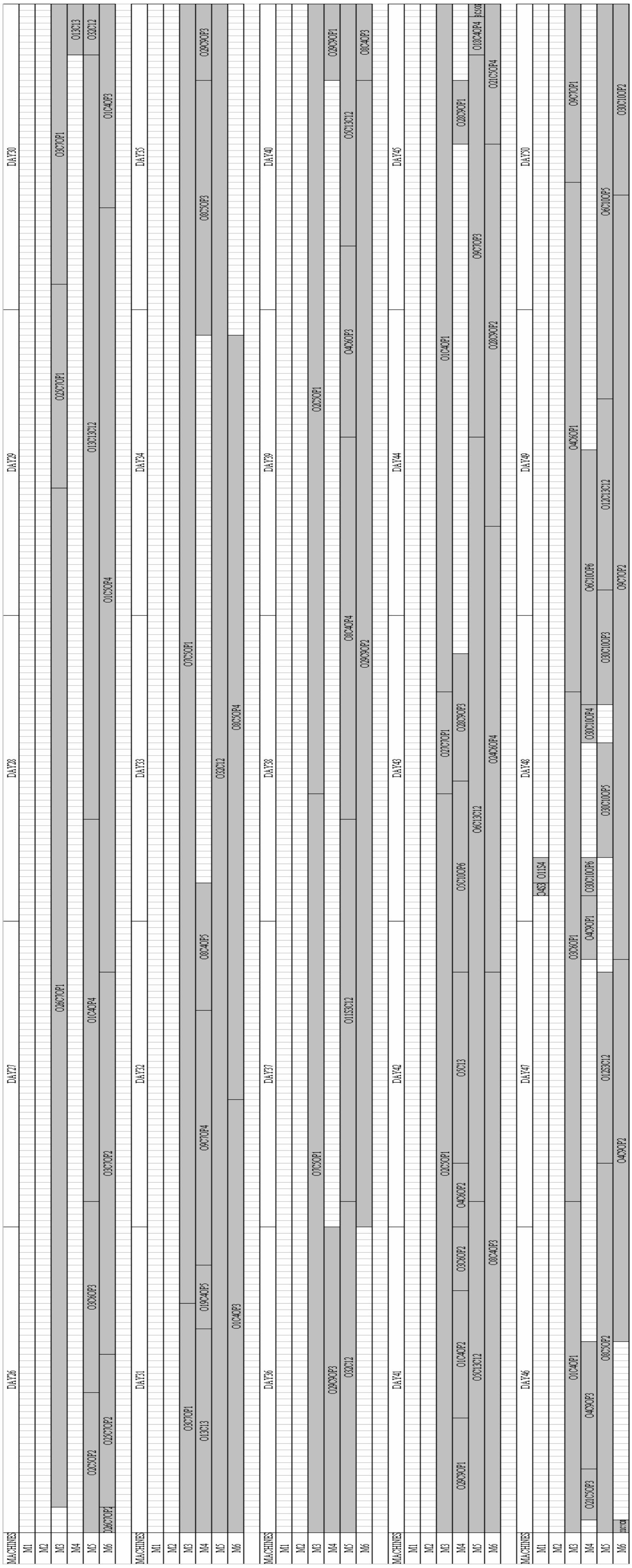


Figure 7.3 (b) The case output in the form of Gantt chart when *t* = 0

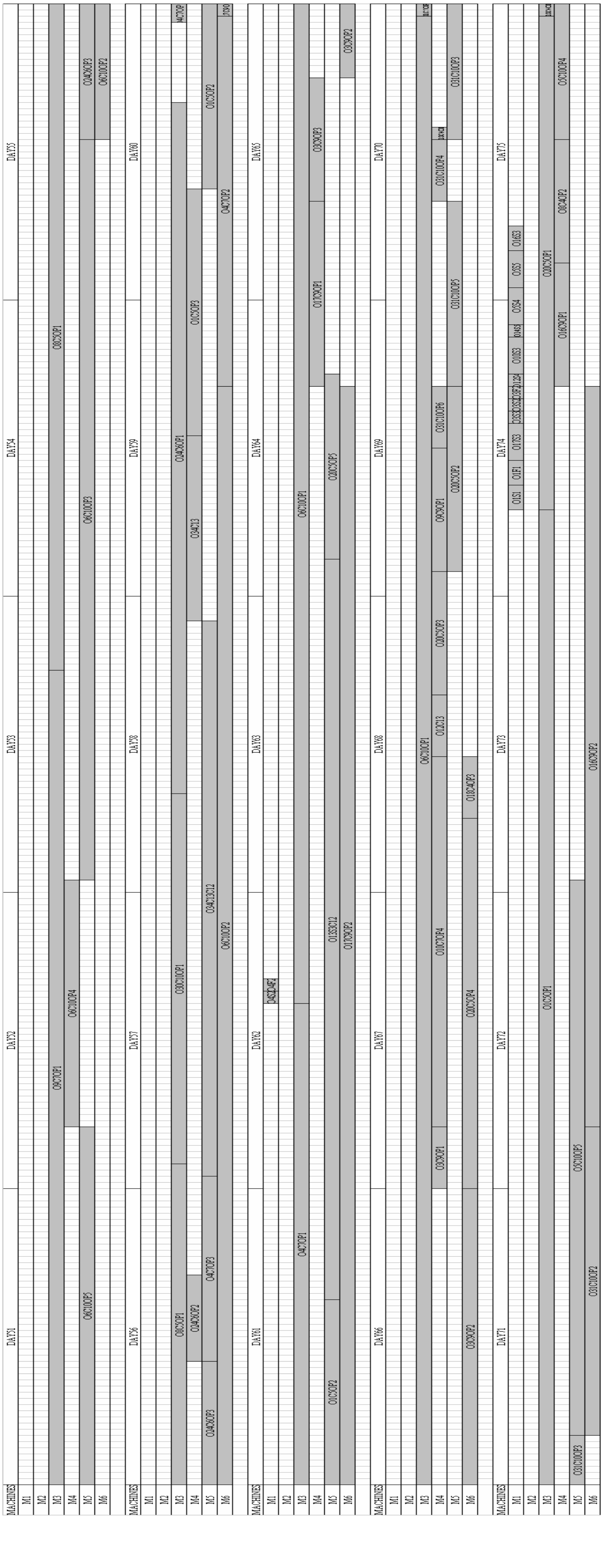


Figure 7.3 (c) The case output in the form of Gantt chart when *t* = 0

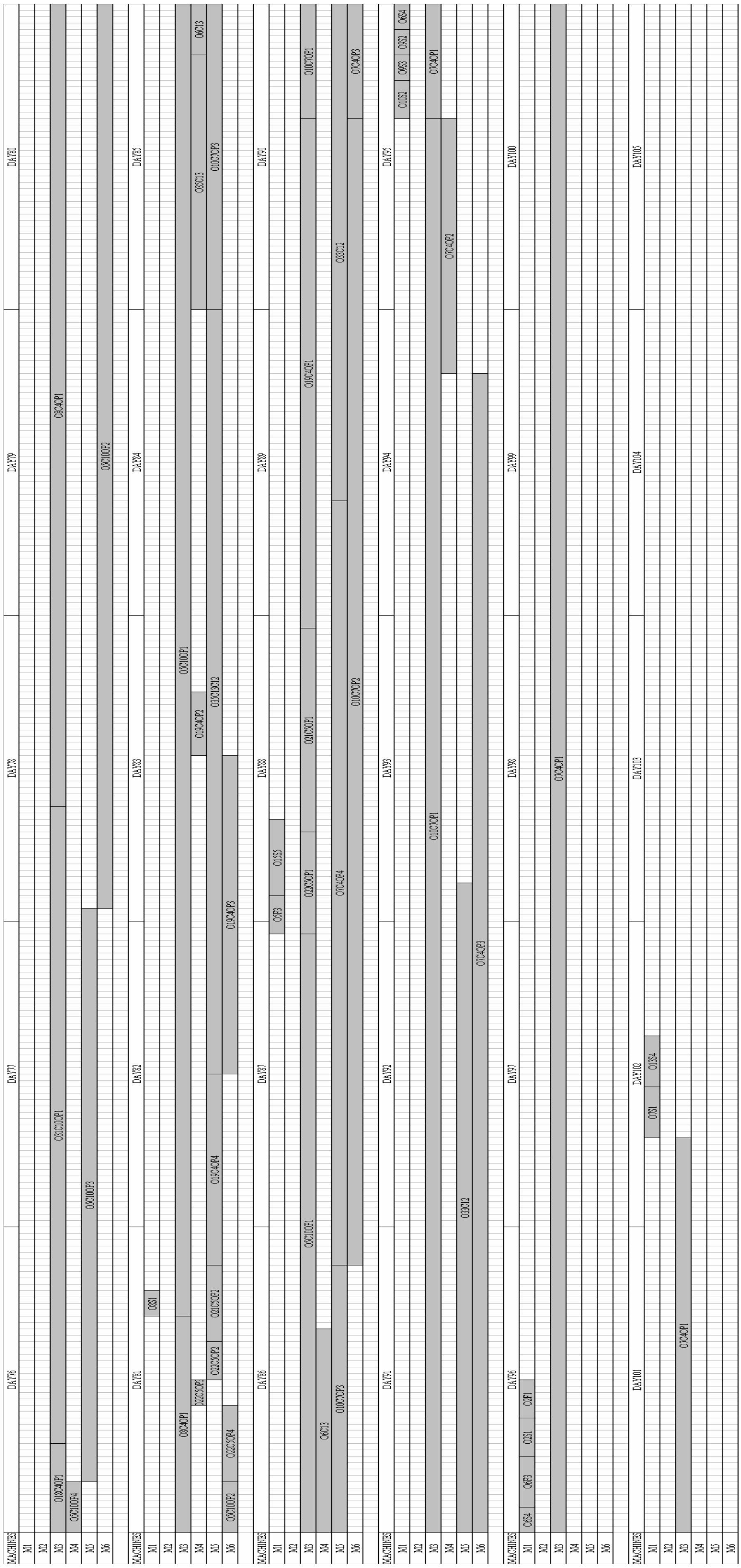


Figure 7.3 (d) The case output in the form of Gantt chart when $t = 0$

In order to compare the results, a production schedule was manually constructed based on the rules the company is using and the results are listed in Table 7.3. It can be found that for the problem instance, the schedule obtained by the APSDSS system indicates a significant improvement and shows an up to 41.4% ((842350-493500)/842350*100%) reduction in the total penalty costs. Meanwhile, the established MIP model was also applied to this practical APS problem. For such a problem, the MIP model requires 15405 constraints, 340 real variables as well as 9914 integer variables, and could not obtain the optimal production plan and schedule in 30 days.

On Day 1 when three new orders arrive, frozen interval = 1 day (24 hours) was adopted and the same genetic parameters as previous were utilized. The best production schedule generated by the system from 5 replications is shown in Appendix X and summarized in Table 7.5. The results of Day 1 to Day 5 in the form of Gantt chart are portrayed in Figure 7.4.

Table 7.5 The case results obtained by APSDSS when *t* = Day 1

Makespan (hour)	Number of tardiness	Number of earliness	Total cost	CPU time (sec.)
2484	197	926	482400	129.453

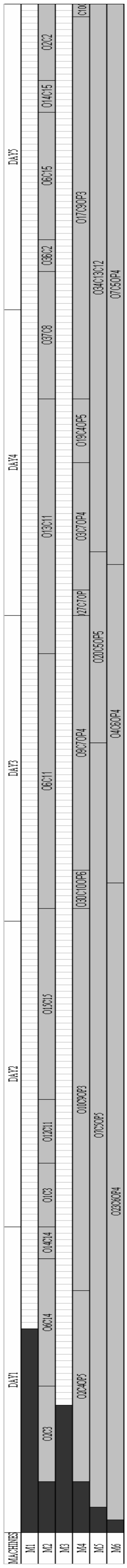


Figure 7.4 The case output (Day 1-5) in the form of Gantt chart when *t* = Day 1

The results further confirm that the suggested approaches and system can find effective schedules with operation starting time and finish time in a reasonable computation time, which is more realistic and useful for the shop floor.

7.4 SUMMARY

In this chapter, the important Advanced Planning and Scheduling (APS) problem arising from a specialist light source manufacturing company is thoroughly investigated. The following observations can be made.

1. The APSDSS proposed in Chapter 6 is applied to the practical APS problem in the company. The computational results indicate that for the problem situation studied, the developed system can generate realistic operation schedules for the shop floor in a reasonable run time and perform very well when compared to the manual schedule, which will be an improvement over a completely intuitive procedure.
2. The implementation of the suggested methodology and system may make it possible to ensure a better customer satisfaction at minimum costs, improve the internal efficiency of the company, and extend the market area. The potential benefits of this work have been communicated to the company, and the system together with the algorithm is being considered for execution.

The next chapter, which is the last part of this dissertation, will present an overall summary of this project and also provide some recommendations for future research.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 DISTINCTIVE ACHIEVEMENTS

Customer driven manufacturing, in which production activities are driven by customer orders, has become a key concept if a manufacturer wishes to be successful in business. To make fast and accurate responses to market demands and ensure reliable delivery for customer orders, manufacturing companies require detailed, realistic, and flexible operational plans and schedules, along with an effective control mechanism for easy tracing of production status of customer orders. In line with this irreversible trend, Advanced Planning and Scheduling (APS), with the integration of production planning and scheduling and the use of holistic and collaborative approaches to provide global optimization, has evolved. With an APS system, manufacturing enterprises offer new potential to increase flexibility and responsiveness to customer orders and market requirements, which thereby enhances customer satisfaction and expands market share.

In this research, the remarkable achievements in the APS study are summarized in the following:

1. Conventional Material Requirements Planning (MRP) dose not sufficiently help the planner in settling production planning and control issues, and create many problems on the shop floor for later production. Hence, in this project, MRP and production scheduling have been considered simultaneously and

integrated together to generate realistic production schedules for the shop floor.

2. A Mixed Integer Programming (MIP) model for Advanced Planning and Scheduling (APS), with the objective of minimizing cost of both production idle time and tardiness or earliness penalty of an order, has been formulated. The proposed model explicitly considers capacity constraints, operation sequences, lead times and due dates in a multi-order environment and generates useful operation schedules for the shop floor, which overcomes the principal difficulty inherent in the existing MRP procedures. Numerical examples extracted or modified from the literature have been illustrated to verify the model and solved adopting the software CPLEX on a personal computer. The numerical results have demonstrated the optimality and effectiveness of the established model.

3. Since the APS problem has been proved to be NP-hard, any exact optimization approach is highly impossible to solve this kind of problem efficiently, and a genetic algorithm (GA) has been built to tackle this issue. Different size problems were utilized to test the established GA approach. The results have indicated that the presented methodology can efficiently find effective schedules with operation starting time and finish time for all of the problems tested. It has been also found that the better results could be produced with higher levels of population size and number of generations. These two factors together determine the amount of search and the algorithm execution time.

4. A series of computational experiments using randomly generated problems were conducted to compare the developed Mixed Integer Programming (MIP) and genetic algorithm (GA). By solving the established MIP, the optimal production schedules can be obtained for all test problems. However, the computational time grows exponentially with the problem size. On the contrary, the GA, as a heuristic method, can reach the global optima for the small size problems, and only achieve the near-optimal solutions for large problems, but it requires much less computation time.
5. In order to cope with the Dynamic Advanced Planning and Scheduling (DAPS) problem where new orders arrive on a continuous basis, a periodic policy with a frozen interval has been adopted to increase stability on the shop floor, and introduced into both the MIP and the GA. The objective of the proposed methodology is to minimize cost of both production idle time and earliness-tardiness penalty for all orders including both original orders and new orders at each rescheduling point. The numerical results proved that the offered methodology can improve the schedule stability while retaining efficiency.
6. A prototype of the Advanced Planning and Scheduling Decision Support System (APSDSS) has been designed. The principle of APSDSS is to consider operation sequences among items, capacity constraints of the manufacturing system as well as the dynamic arrival of orders, and employ a GA-based method to translate the orders of the various products into detailed production schedules with operation starting time and finish time. The system

was implemented using VC++, an object oriented programming language, and is endowed with a set of interfaces for easy use. An example has been illustrated to validate the applicability of the constructed system.

7. For the APS problem arising from a specialist light source manufacturing company, the proposed system has been applied to the practical problem, and the results further confirmed that the suggested approaches and system can find effective schedules with operation starting time and finish time in a reasonable computation time, which is more realistic and useful for the shop floor.

8.2 ACADEMIC CONTRIBUTIONS

This research investigates Advanced Planning and Scheduling (APS) in detail and formulates a mathematical model for APS, with the objective of minimizing cost of both production idle time and tardiness or earliness penalty of an order. A genetic algorithm (GA) is established to solve the APS problem more efficiently. Both the mathematical model and the GA are further extended by incorporating a periodic policy with a frozen interval to settle the Dynamic Advanced Planning and Scheduling (DAPS) problem where new orders arrive on a continuous basis. Furthermore, a prototype of the Advanced Planning and Scheduling Decision Support System (APSDSS) is built on the basis of the GA method and applied to a real case in a light source manufacturing company.

In this project, six academic contributions associated with the optimization of APS are made.

First of all, Advanced Planning and Scheduling (APS) studied in this work performs a higher level of integration on production planning and shop floor scheduling. Traditionally, production planning and scheduling are treated hierarchically and separately, which has been proved to be ineffective and creates many problems on the shop floor for production. It is therefore expected that Advanced Planning and Scheduling (APS) will significantly reduce conflicts on the shop floor and provide better performance.

Secondly, the APS problem is based on a more realistic performance measure. The APS problem investigated in this research is to find a production schedule for the orders that both production idle time and earliness and tardiness penalty are minimized. The minimum production idle time implies less production flow time or higher utilization of the machines. In addition, production idle time is chosen as the objective to be reduced because it is able to reflect two focuses on the shop floor: manufacturing lead time and WIP (work-in-process) inventory level. Another objective of the problem in this research is to seek a schedule with all orders completed as closed to their due date as possible, which is on the basis of the fact that either an early or a late delivery of an order results in an increase in the production costs. Because of the common goal of avoiding either earliness or tardiness of an order to keep the cost as low as possible, the earliness and tardiness problem leads itself to a just-in-time (JIT) production system.

Thirdly, a Mixed Integer Programming (MIP) model, which succeeds in a system integration of the production planning and shop floor scheduling problems and favorably produces optimal operation sequence, is developed. This contribution is believed to be a significant advance in APS optimization since none of the other researchers has provided such a complete and verified mathematical model.

Fourthly, a GA-based method is designed for solving the APS problem more efficiently. The established genetic algorithm adopts the random key encoding mechanism, and constitutes a general approach that can be easily modified to adapt to a variety of APS problems.

Fifthly, a periodic policy with a frozen interval is first introduced into the Dynamic Advanced Planning and Scheduling (DAPS) problem. Much of the previous research on dynamic problems only takes into account efficiency performance to minimize the cost objectives like mean flow time, earliness and tardiness, etc. Usually, doing so will greatly change the production schedule when new conditions occur and induce instability. Meanwhile, it is observed that the stability of the production system will decrease more when changes are made closer to the current period. Therefore, this research adopts a periodic policy with a frozen interval. This policy provides a framework for balancing efficiency and stability, which fills up a gap in the DAPS research.

Finally, the APSDSS prototype built in this project can generate better production plans and schedules for APS. With the employment of an intelligent heuristic approach, genetic algorithm, the APSDSS can provide the desirable solution or hopefully optimal solution to the APS problem. Such a feature is absent in the current computer-based APS systems, most of which only incorporate trial-and-error methods.

8.3 POSSIBLE BENEFITS TO INDUSTRY

Many manufacturing firms have products with a multi-level structure, and encounter the Advanced Planning and Scheduling (APS) problem.

In this project, a prototype of the Advanced Planning and Scheduling Decision Support System (APSDSS) has been established and successfully applied to the practical APS problem in a specialist light source manufacturing company. The case study indicates that developed system can generate realistic operation schedules for the shop floor in a reasonable run time and perform very well when compared to the manual schedule, which will be an improvement over a completely intuitive procedure. The implementation of the suggested methodology and system may make it possible to ensure a better customer satisfaction at minimum costs, improve the internal efficiency of the company, and extend the market area. The potential benefits of this work have been communicated to the company, and the system together with the algorithm is being considered for execution.

The APSDSS prototype is such a Windows application that can manage the data electronically, handle the Advanced Planning and Scheduling (APS) problem as well as the Dynamic Advanced Planning and Scheduling (DAPS) problem efficiently, and create the production plans and schedules automatically. Although some refinements need to be carried out, the APSDSS prototype has been constructed as a promising decision support tool for production planners in manufacturing industry. It will not only free the production planners from the labor-concentrated jobs, such as constructing the production plans and schedules, but also assist them to take effective decisions depending on the various situations of the manufacturing system. Consequently, the company's competitiveness and productivity can be enhanced.

8.4 FUTURE WORK

Some possible further work related to this project in future is suggested as follows:

1. Introducing setup time into the APS problem

   In this APS problem, the setup time, including the transfer time between operations, is assumed to be negligible or included in the processing times. Actually, the machine setup time is common in the shop, and also has an effect on the performance of production. It is worth extending APS by taking into account the setup time.

2. Enriching the genetic algorithm's performance

   The random key encoding strategy with some standard GA techniques has been utilized to tackle the APS problem. Definitely, certain permutation-based encoding methods can be designed to compare the results. Meanwhile, some other genetic operations, for instance heuristic-featured ones, or some local search approaches can be adopted to enrich the GA and to improve the performance.

3. Investigating other dynamic events

   This project has studied the DAPS problem where new orders arrive on a continuous basis. However, in the practical shop, there are many uncertainties that may disturb the production system and require the modification of the existing production plans and schedules. An area of future research might be the investigation of other dynamic events, such as machine failure, order cancellation, due date change, and so on.

4. Extending the APSDSS functionalities

   An APSDSS, on the basis of a GA method, has been presented in this project. The APSDSS can be further strengthened by integrating the formulated MIP model to generate optimal solutions, designing more output windows to report the results, and so forth. After adding all these functions, the system may be viewed as a powerful tool for APS in manufacturing industry.

5. Implementing the research results in manufacturing industry

   Advanced Planning and Scheduling plays an important role in manufacturing companies, and also is a crucial factor for factories' success. Thus, it is useful to apply the research results to the manufacturing industry such that enterprises can sharpen their competitive edge through substantial reduction in production costs and flexible reaction to market requirements.

REFERENCES

[Abd88] Abdul-Razaq, T. S., and Potts, C. N. (1988), "Dynamic programming state-space relaxation for single-machine scheduling", *Journal of the Operational Research Society*, 39(2), 141-152.

[Agr96] Agrawal, A., Harhalakis, G., Minis, I., and Nagi, R. (1996), "'Just-in-time' production of large assemblies", *IIE Transactions*, 28(8), 653-667.

[Alb02] Alberto P. (2002), "Critical factors of MRP implementation in small and medium-sized firms", *International Journal of Operations and Production Management*, 22(3), 329-348.

[And90] Anderson, E. J., and Nyirenda, J. C. (1990), "Two new rules to minimize tardiness in a job shop", *International Journal of Production Research*, 28(12), 2277-2292.

[App91] Applegate, D., and Cook, W. (1991), "A computational study of the job-shop scheduling problem", *ORSA Journal on Computing*, 3(2), 149-156.

[Art97] Artiba, A., and Aghezzaf, E.H. (1997), "An architecture of a multi-model system for planning and scheduling", *International Journal of Computer Integrated Manufacturing*, 10(5), 380-393.

[Ayt03] Aytug, H., Khouja, M., and Vergara, F. E. (2003), "Use of genetic algorithms to solve production and operations management problems: a review", *International Journal of Production Research*, 41(17), 3955-4009.

[Bag86] Bagchi, U., Sullivan, R. S., and Chang, Y. L. (1986), “Minimizing mean absolute deviation of completion times about a common due date”, *Naval Research Logistics Quarterly*, 33(2), 227-240.

[Bag87] Bagchi, U., Chang, Y. L., and Sullivan, R. S. (1987), “Minimizing absolute and squared deviations of completion times with different earliness and tardiness penalties and a common due date”, *Naval Research Logistics*, 34(5), 739-751.

[Bag91] Bagchi, S., Uckun, S., Miyabe, Y., and Kawamura, K. (1991), "Exploring problem-specific recombination operators for job shop scheduling", *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Diego, 10-17.

[Bah84] Bahl, A. C., and Ritzman, L. P. (1984), “An integrated model for master scheduling, lot sizing and capacity requirements planning”, *Journal of the Operational Research Society*, 35(5), 389-399.

[Bak90] Baker, R., and Scudder, G. D. (1990), “Sequencing with earliness and tardiness penalties: a review”, *Operation Research*, 38(1), 22-36.

[Bak93] Bakke, N. A., and Hellberg, R. (1993), “The challenges of capacity planning”, *International Journal of Production Economics*, 30-31, 243-264.

[Bal67] Balas, E. (1967), “Discrete programming by the filter method”, *Operations Research*, 15(5), 915 – 957.

[Bal69] Balas, E. (1969), “Machine sequencing via disjunctive graphs: an implicit enumeration algorithm", *Operations Research*, 17(6), 941 – 957.

[Bar95] Barnes, J. W., and Chambers, J. B. (1995), "Solving the job shop scheduling problem with tabu search", *IIE Transactions*, 27(2), 257-263.

[Bea91] Bean, J. C., Birge, J. R., Mittenthal, J., and Noon, C. E. (1991), "Matchup scheduling with multiple resources, release dates and disruptions", *Operations Research*, 39(3), 470-483.

[Bea94] Bean, J. C. (1994), "Genetic algorithms and random keys for sequencing and optimization", *ORSA Journal on Computing*, 6(2), 154-160.

[Bie99] Bierwirth, C., and Mattfeld, D. C. (1999), "Production scheduling and rescheduling with genetic algorithms", *Evolutionary Computation*, 7(1), 1-17.

[Bil83] Billington, P. J., Mcclain, J. D., and Thomas, L. J. (1983), "Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction", *Management Science*, 29(10), 1126-1141.

[Bla82] Blackstone, J. H., Phillips, D. T., and Hogg, G. L. (1982), "A state-of-the-art survey of dispatching rules for manufacturing job shop operations", *International Journal of Production Research*, 20(1), 27-45.

[Bla88] Blazewicz, J., Finke, G., Haupt, R., and Schmidt, G. (1988), "New trends in machine scheduling", *European Journal of Operational Research*, 37(3), 303-317.

[Bla91] Blazewicz, J., Dror, M., and Weglarz, J. (1991), "Mathematical programming formulations for machine scheduling: a survey", *European Journal of Operational Research*, 51(3), 283-300.

[Bla96] Blazewicz, J. (1996), *Scheduling Computer and Manufacturing Processes*, Springer, New York.

[Bou80] Bourke, R. (1980), "Surveying the software", *Datamation*, 26(10), 101-120.

[Bow59] Bowman, E. (1959), "The scheduling sequence problem", *Operations Research*, 7(5), 621-624.

[Bro65] Brooks, G., and White, C. (1965), "An algorithm for finding optimal or near optimal solutions to the production scheduling problem", *Journal of Industrial Engineering*, 15, 34-40.

[Bru93] Brucker, P., and Jurisch, B. (1993), "A new lower bound for the job shop scheduling problem", *European Journal of Operations Research*, 64(2), 156-167.

[Bru94] Brucker, P., Jurisch, B., and Sievers, B. (1994), "A branch and bound algorithm for the job-shop scheduling problem", *Discrete Applied Mathematics*, 49(1-3), 107-127.

[Bru04] Brucker, P. (2004), *Scheduling Algorithms*, Springer, New York.

[Cal86] Callerman, T. E., and Heyl, J. E. (1986), "A model for material requirements planning implementation", *International Journal of*

Operations and Production Management, 6(5), 30-37.

[Car89] Carlier, J., and Pinson, E. (1989), "An algorithm for solving the job-shop problem", *Management Science*, 35(2), 164-176.

[Car01] Carter, M. W., Price, C. C. (2001), *Operations Research: A Practical Introduction*, CRC Press, Boca Raton.

[Cas02] Castillo, E., Conejo, A. J., Pedregal, P., Garcia, R., and Alguacil, N. (2002), *Building and Solving Mathematical Programming Models in Engineering and Science*, John Wiley & Sons, New York.

[Cer89] Cerveny, R. P., and Scott, L. W. (1989), "A survey of MRP implementation", *Production and Inventory Management Journal*, 30(3), 31-34.

[Cha85] Chase, R. B., and Aquilano, N. J. (1985), *Production and Operations Management: A Life Cycle Approach*, R.D. Irwin, Homewood.

[Cha94] Chang, S. C., and Liao, D.Y. (1994), "Scheduling flexible flow shops with no setup effects", *IEEE Transactions on Robotics and Automation*, 10(2), 112-122.

[Cha96] Chang, Y., Sueyoshi, T., and Sullivan, R. S. (1996), "Ranking dispatching rules by data envelopment analysis in a job shop environment", *IIE Transactions*, 28(8), 631-642.

[Cha99] Chambers, L. (1999), *Practical Handbook of Genetic Algorithms*, CRC Press, Boca Raton.

[Cha05] Chaudhry, S. S., and Luo, W. (2005), "Application of genetic algorithms in production and operations management: a review", *International Journal of Production Research*, 43(19), 4083-4101.

[Che89] Cheng, T. C. E., and Gupta, M. C. (1989), "Survey of scheduling research involving due date determination decisions", *European Journal of Operational Research*, 38(2), 156-166.

[Che96] Cheng, R. W., Gen, M., and Tsujimura, Y. (1996), "A tutorial survey of job-shop scheduling problems using genetic algorithms—I. representation", *Computers and Industrial Engineering*, 30(4), 983-997.

[Che99] Cheng, R. W., Gen, M., and Tsujimura, Y. (1999), "A tutorial survey of job-shop scheduling problems using genetic algorithms: Part II. hybrid genetic search strategies", *Computers and Industrial Engineering*, 37(1-2), 51-55.

[Chu92] Church, L. K., and Uzsoy, R. (1992), "Analysis of periodic and event-driven rescheduling policies in dynamic shops", *International Journal of Computer Integrated Manufacturing*, 5(3), 153-163.

[Cow02] Cowling, P., and Johansson, M. (2002), "Using real time information for effective dynamic scheduling", *European Journal of Operational Research*, 139(2), 230-244.

[Cze94] Czerwinski, C. S., and Luh, P. B. (1994), "Scheduling products with bills of materials using an improved Lagrangian relaxation technique", *IEEE Transactions on Robotics and Automation*, 10(2), 99-111.

[Dav85] Davis, L. (1985), "Job shop scheduling with genetic algorithms", *Proceedings of the 1st International Conference on Genetic*

Algorithms, Hillsdale, NJ, 136 - 140.

[Dav91] Davis, L. (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

[Del93] Dell'Amico, M., and Trubian, M. (1993), "Applying tabu search to the job-shop scheduling problem", *Annals of Operations Research*, 41(3), 231-252.

[Del95] Della Croce, F., Tadei, R., and Volta, G. (1995), "A genetic algorithm for the job shop problem", *Computers and Operations Research*, 22(1), 15-24.

[Dep91] De, P., Ghosh, J. B., and Wells, C. E. (1991), "Scheduling to minimize weighted earliness and tardiness about a common due-date", *Computers and Operations Research*, 18(5), 465-475.

[Dep93] De, P., Ghosh, J. B., and Wells, C. E. (1993), "On general solution for a class of early/tardy problems", *Computer and Operations Research*, 20(2), 141-149.

[Dil94] Dillenberger, C., Escudero, L. F., Wollensak, A., and Zhang, W. (1994), "On practical resource allocation for production planning and scheduling with period overlapping setups", *European Journal of Operational Research*, 75(2), 275-286.

[Ell00] Elliott, M. (2000), "Advanced planning and scheduling software", *IIE Solutions*, 32(10), 48-56.

[Emm87] Emmons, H. (1987), "Scheduling to a common due date on parallel uniform processors", *Naval Research Logistics*, 34(6), 803-810.

[Faa87] Faaland, B., and Schmitt, T. (1987), "Scheduling tasks with due dates in a fabrication/assembly process", *Operations Research*, 35(3), 378-388.

[Fal91] Falkenauer, E., and Bouffouix, S. (1991), "A genetic algorithm for job shop", *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, 824-829.

[Fis73] Fisher, M. L. (1973), "Optimal solution of scheduling problems using Lagrange multipliers: Part I", *Operations Research*, 21(5), 1114-1127.

[Fis83] Fisher, M. L., Lageweg, B. J., Lenstra, J. K., and Rinnooy Kan, A. H. G. (1983), "Surrogate duality relaxation for job shop scheduling", *Discrete Applied Mathematics*, 5(1), 65-75.

[Flo71] Florian, M., Trepant, P., and McMahon, G. (1971), "An implicit enumeration algorithm for the machine sequencing problem", *Management Science*, 17(12), 782-792.

[Fry89] Fry, T. D., Oliff, M. D., Minor, E. D., and Leong, G. K. (1989), "The effects of product structure and sequencing rule on assembly shop performance", *International Journal of Production Research*, 27(4), 671-686.

[Gar76] Garey, M. R., and Johnson, D. S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York.

[Gar88] Garey, M. R., Tarjan, R. E., and Wilfong, G. T. (1988), "One-processor scheduling with symmetric earliness and tardiness penalties", *Mathematics of Operations Research*, 13(2), 330-348.

[Gar03] Garg, V. K., and Venkitakrishnan, N. K. (2003), *Enterprise Resource Planning: Concepts and Practice*, Prentice-Hall of India Private Limited, New Delhi.

[Gen00] Gen, M., and Cheng, R. (2000), *Genetic Algorithms and Engineering Optimization*, Wiley, New York.

[Glo93] Glover, F., Taillard, E., and De Werra, D. (1993), "A user's guide to tabu search", *Annals of Operations Research*, 41(3), 3-28.

[Glo97] Glover, F., and Laguna, M. (1997), *Tabu Search*, Kluwer Academic Publishers, Boston.

[Gol89] Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

[Gor02] Gordon, V., Proth, J. M., and Chu, C. B. (2002), "A survey of the state-of-the-art of common due date assignment and scheduling research", *European Journal of Operational Research*, 139(1), 1-25.

[Gra93] Graves, S. C., Rinnooy Kan, A. H. G., and Zipkin, P. H. (1993), *Logistics of Production and Inventory*, North-Holland, Amsterdam.

[Gif60] Giffler, B., and Thompson, G. L. (1960), "Algorithms for solving production scheduling problems", *Operations Research*, 8(4), 487-503.

[Gup89] Gupta, Y. P., Gupta, M. C., and Bector, C. R. (1989), "A review of scheduling rules in flexible manufacturing systems", *International Journal of Computer Integrated Manufacturing*, 2(6), 356-377.

[Had97] Hadj-Alouane, A. B., and Bean, J. C. (1997), “A genetic algorithm for the multiple-choice integer program”, *Operations Research*, 45(1), 92-101.

[Hai01] Haigh, A. (2001), *Object-Oriented Analysis and Design*, McGraw-Hill, New York.

[Hal86] Hall, N. G. (1986), “Single- and multiple-processor models for minimizing completion time variance”, *Naval Research Logistics Quarterly*, 33(1), 49-54.

[Hal91a] Hall, N. G., and Posner, M. E. (1991), “Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date”, *Operations Research*, 39(5), 836-846.

[Hal91b] Hall, N. G., Kubiak, W., and Sethi, S. P. (1991), “Earliness-tardiness scheduling problems, II: deviation of completion times about a restrictive common due date”, *Operations Research*, 39(5), 847-856.

[Hal04] Hall, N. G., and Potts, C. N. (2004), “Rescheduling for new orders”, *Operations Research*, 52(3), 440-453.

[Har85] Harl, J. E. and Ritzman, L. P. (1985), “A heuristic algorithm for capacity sensitive requirements planning”, *Journal of Operations Management*, 5(3), 309-326.

[Has82] Hastings, N. A. J., Marshall, P., and Willis, R. J. (1982), “Schedule based M.R.P.: an integrated approach to production scheduling and material requirements planning”, *The Journal of the Operational Research Society*, 33(11), 1021-1029.

[Hau89] Haupt, R. (1989), "A survey of priority rule-based scheduling", *OR Spektrum*, 11(1), 3-16.

[Hoo91] Hoogeveen, J. A., and van de Velde, S. L. (1991), "Scheduling around a small common due date", *European Journal of Operational Research*, 55(2), 237-242.

[Jai97] Jain, A. K., and Elmaraghy, H. A. (1997), "Production scheduling/rescheduling in flexible manufacturing", *International Journal of Production Research*, 35(1), 281-309.

[Jen03a] Jensen, M. T. (2003), "Generating robust and flexible job shop schedules using genetic algorithms", *IEEE Transactions on Evolutionary Computation*, 7(3), 275-288.

[Jen03b] Jensen, P. A., and Bard, J. F. (2003), *Operations Research: Models and Methods*, Wiley, New York.

[Kan81] Kanet, J. J. (1981), "Minimizing the average deviation of job completion times about a common due date", *Naval Research Logistics Quarterly*, 28(4), 643-651.

[Kan88] Kanet, J. J. (1988), "MRP 96: time to rethink manufacturing logistics", *Production and Inventory Management Journal*, 29(2), 57-61.

[Kar84] Karmarkar, N. (1984), "A new polynomial-time algorithm for linear programming", *Combinatorica*, 4(4), 373-395.

[Kim96] Kim, J. U., and Kim, Y. D. (1996), "Simulated annealing and genetic algorithms for scheduling products with multi-level product structure", *Computers and Operations Research*, 23(9), 857-868.

[Kir83] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. (1983), "Optimization by simulated annealing", *Science*, 220(4598), 671-680.

[Kol00] Kolish, R. (2000), "Integration of assembly and fabrication for make-to-order production", *International Journal of Production Economics*, 38(3), 287-306.

[Kov05] Kovacs, A., Egri, P., Kis, T., and Vancza, J. (2005), "Proterv-II: An integrated production planning and scheduling system", *Lecture notes in Computer Science*, 3709, 880-880.

[Kra87] Krajewski, L. J., King, B. E., Ritzman, L. P., and Wong, D. S. (1987), "Kanban, MRP, and shaping the manufacturing environment", *Management Science*, 33(1), 39-57.

[Kru84] Krupp, J. A. G. (1984), "Why MRP systems fail: traps to avoid", *Production and Inventory Management Journal*, 25(3), 48-53.

[Kub90] Kubiak, W., Lou, S., and Sethi, S. (1990), "Equivalence of mean flow time problems and mean absolute deviation problems", *Operations Research Letters*, 9(6), 371-374.

[Lag77] Lageweg, B. J., Lenstra, J. K., and Rinnooy Kan, A. H. G. (1977), "Job-shop scheduling by implicit enumeration", *Management Science*, 24(4), 441-449.

[Lan89] Landvater, D. V., and Gray, C. D. (1989), *MRP II Standard System: A Handbook for Manufacturing Software Survival*, Oliver Wight, Essex.

[Las92] Lasserre, J. B. (1992), "An integrated model for job-shop planning and scheduling", *Management Science*, 38(8), 1201-1211.

[Law82] Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G. (1982), "Recent developments in deterministic sequencing and scheduling: a survey," in: *Deterministic and Stochastic Scheduling*, Dempster, M. A. H., Lenstra, J. K., and Rinnooy Kan, A. H. G. (eds.), Reidel, Dordrecht, The Netherlands, 35-73.

[Lee02] Lee, Y.H., Jeong, C.S., and Moon, C. (2002), "Advanced planning and scheduling with outsourcing in manufacturing supply chain", *Computers and Industrial Engineering*, 43(1-2), 351-374.

[Leo94] Leon, V. J., Wu, S. D., and Storer, R. H. (1994), "A game-theoretic control approach for job shops in the presence of disruptions", *International Journal of Production Research*, 32(6), 1451-1476.

[Lic94] Li, C. L., and Cheng, T. C. E. (1994), "The parallel machine min-max weighted absolute lateness scheduling problem", *Naval Research Logistics*, 41(1), 33-46.

[Lin94] Lin, N. P., Krajewski, L., Leong, G. K., and Benton, W. C. (1994), "The effects of environmental factors on the design of master production scheduling systems", *Journal of Operations Management*, 11(4), 367-384.

[Lir93] Li, R. K., Shyu, Y. T., and Adiga, S. (1993), "A heuristic rescheduling algorithm for computer-based production scheduling systems", *International Journal of Production Research*, 31(8), 1815-1826.

[Lus93] Luscombe, M. (1993), *MRP II: Integrating the Business*, Butterworth

Heinemann, Oxford.

[Mac93] MacCarthy, B. L., and Liu, J. (1993), “Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling”, *International Journal of Production Research*, 31(1), 59-79.

[Man60] Manne, A. (1960), “On the job shop scheduling problem”, *Operations Research*, 8(2), 219-223.

[Mar01] Markowitz, D. M., and Wein, L. M. (2001), “Heavy traffic analysis of dynamic cyclic policies: A unified treatment of the single machine scheduling problem”, *Operations Research*, 49(2), 246-270.

[Mat88] Matsuó, H., Suh, C. J., and Sullivan, R. S. (1988), “Controlled search simulated annealing method for the general jobshop scheduling problem”, Working Paper 03-04-88, Department of Management, The University of Texas at Austin.

[Mat93] Matsuura, H., Tsubone, H., and Kanezashi, M. (1993), “Sequencing, dispatching, and switching in a dynamic manufacturing environment”, *International Journal of Production Research*, 31(7), 1671-1688.

[Mcc90] McCarthy, S. W., and Barber, K. D. (1990), “Medium to short term finite capacity scheduling: A planning methodology for capacity constrained workshops”, *Engineering Costs and Production Economics*, 19(1-3), 189-199.

[Mcc92] McClain, J. O., Thomas, L. J., and Mazzola, J. B. (1992), *Operations Management: Production of Goods and Services*, Prentice Hall, Englewood Cliffs, New Jersey.

[Mck95] McKay, K. N., Safayeni, F. R., and Buzacott, J. A. (1995), "A review of hierarchical production planning and its applicability for modern manufacturing", *Production Planning and Control*, 6(5), 384-394.

[Mck03] McKay, K. N., and Wiers, V. C. S. (2003), "Integrated decision support for planning, scheduling, and dispatching tasks in a focused factory", *Computers in Industry*, 50(1), 5-14.

[Mit96] Mitchell, M. (1996), *An Introduction to Genetic Algorithms*, MIT Press, London.

[Moo03] Moon, C., and Lee, Y. H. (2003), "Advanced process planning and scheduling with precedence constraints and machine selection using a genetic algorithm", *International Journal of Industrial Engineering: Theory Applications and Practice*, 10(1), 26-34.

[Moo04] Moon, C., Kim, J. S., and Gen, M. (2004), "Advanced planning and scheduling based on precedence and resource constraints for e-plant chains", *International Journal of Production Research*, 42(15), 2941-2955.

[Mos01] Mosheiov, G., and Shadmon, M. (2001), "Minmax earliness-tardiness costs with unit processing time jobs", *European Journal of Operational Research*, 130(3), 638-652.

[Mos03] Mosheiov, G. (2003), "Scheduling unit processing time jobs on an m-machine flow-shop", *Journal of the Operational Research Society*, 54(4), 437-441.

[Nak91] Nakano, R., and Yamada, T. (1991), “Conventional genetic algorithm for job shop problems”, *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Diego, CA, 474-479.

[Nan93] Nandkeolyar, U., Ahmed, M. U., and Sundararaghavan, P. S. (1993), “Dynamic single-machine-weighted absolute deviation problem: predictive heuristics and evaluation”, *International Journal of Production Research*, 31(6), 1453-1466.

[Nem88] Nemhauser, G. L., and Wolsey, L. A. (1988), *Integer and Combinatorial Optimization*, Wiley, New York.

[New74] New, C. C. (1974), *Requirements Planning*, Gower Press, Essex.

[Nis04] Nishioka, Y. (2004), “Collaborative agents for production planning and scheduling (CAPPS): a challenge to develop a new software system architecture for manufacturing management in Japan”, *International Journal of Production Research*, 42(17), 3355-3368.

[Nof91] Nof, S. Y., and Grant, F. H. (1991), “Adaptive/predictive scheduling: review and a general framework”, *Production Planning and Control*, 2(4), 298-312.

[Now96] Nowicki, E., and Smutnicki, C. (1996), “A fast taboo search algorithm for the job shop problem”, *Management Science*, 42(6), 797-813.

[Ole00] O'Leary, D. E. (2000), *Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk*, Cambridge University Press, Cambridge.

[Orl69] Orlicky, G. (1969), *The Successful Computer System: Its Planning,* *Development, and Management in a Business Enterprise*, McGraw-Hill, New York.

[Pan77] Panwalkar, S. S., Iskander, W. (1977), "A survey of scheduling rules", *Operations Research*, 25(1), 45-61.

[Par92] Paredis, J. (1992), "Exploiting constraints as background knowledge for genetic algorithms: a case-study for scheduling", *Parallel Problem Solving from Nature*, 2, 229-238.

[Pen89] Peng S. O., and Morton, T. E. (1989), "The single machine early/tardy problem", *Management Science*, 35(2), 177-191.

[Per99] Persi, P., Ukovich, W., Pesenti, R., and Nicolich, M. (1999), "A hierarchic approach to production planning and scheduling of a flexible manufacturing system", *Robotics and Computer-integrated Manufacturing*, 15(5), 373-385.

[Pha00] Pham, D. T., and Karaboga, D. (2000), *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, Springer, New York.

[Pin02] Pinedo, M. (2002), *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Upper Saddle River, New Jersey.

[Plo94] Plossl, G. W. (1994), *Orlicky's Material Requirements Planning*, McGraw-Hill, New York.

[Pon01] Ponnambalam, S. G., Aravindan, P., and Rao, P. S. (2001), "Comparative evaluation of genetic algorithms for job-shop scheduling", *Production Planning and Control*, 12(6), 560-574.

[Pon02] Pongcharoen, P., Hicks, C., Braiden, P. M., Stewardson, D. J. (2002), “Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products”, *International Journal of Production Economics*, 78(3), 311-322.

[Pon04] Pongcharoen, P., Hicks, C., and Braiden, P. M. (2004), “The development of genetic algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure”, *European Journal of Operational Research*, 152(1), 215-225.

[Pro98] Proudlove, N. C., Vadera, S., Kobbacy, K. A. H. (1998), “Intelligent management systems in operations: a review”, *Journal of the Operational Research Society*, 49(7), 682-699.

[Rag86] Raghavachari, M. (1986), “A V-shape property of optimal schedule of jobs about a common due date”, *European Journal of Operational Research*, 23(3), 401-402.

[Rag93] Raghu, T. S., and Rajendran, C. (1993), “An efficient dynamic dispatching rule for scheduling in a job shop”, *International Journal of Production Economics*, 32(3), 301-313.

[Ram94] Ramaswamy, S. E., and Joshi, S. B. (1994), “A response surface approach to developing composite dispatching rules for flexible manufacturing systems”, *International Journal of production Research*, 32(11), 2613-2629.

[Ran04] Rangsaritratsamee, R., Ferrell, W. G., and Kurz, M. B. (2004), “Dynamic rescheduling that simultaneously considers efficiency and stability”, *Computers and Industrial Engineering*, 46(1), 1-15.

[Ree00a] Reeja MK, Rajendran C. (2000), “Dispatching rules for scheduling in assembly jobshops - Part 1”, *International Journal of Production Research*, 38(9), 2051-2066.

[Ree00b] Reeja MK, Rajendran C. (2000), “Dispatching rules for scheduling in assembly jobshops - Part 2”, *International Journal of Production Research*, 38(10), 2349-2360.

[Ree95] Reeves, C. R. (1995), “A genetic algorithm for flowshop sequencing”, *Computers and Operations Research*, 22(1), 5-13.

[Ria01] Riane, F., Artiba, A., and Iassinovski, S. (2001), “An integrated production planning and scheduling system for hybrid flowshop organizations”, *International Journal of Production Economics*, 74(1-3), 33-48.

[Rin76] Rinnooy Kan, A. H. G. (1976), “Machine scheduling problem: classification complexity and computations”, *Martinus Nijhoff*, The Hague Holland.

[Rog91] Rogers, R. V., and White, K. P. (1991), “Algebraic, mathematical-programming, and network models of the deterministic job-shop scheduling problem”, *IEEE Transactions on Systems Man and Cybernetics*, 21(3), 693-697.

[Rom02] Rom, W. O., Tukel, O. I., and Muscatello, J. R. (2002), “MRP in a job shop environment using a resource constrained project scheduling model”, *OMEGA-International Journal of Management Science,* 30(4), 275-286.

[Sab00] Sabuncuoglu, I., and Bayiz, M. (2000), "Analysis of reactive scheduling problems in a job shop environment", *European Journal of Operational Research*, 126(3), 567-586.

[Sar95] Sarper, H. (1995), "Minimizing the sum of absolute deviations about a common due date for the two-machine flow shop problem", *Applied Mathematical Modelling*, 19(3), 153-161.

[Sch00] Schubert, W. R., Sr. (2000), *Manufacturing Management Decision Support Tools for Year 2000 and Beyond*, DMgt Dissertation, Colorado Technical University.

[Sch05] Schach, S. R. (2005), *Object-Oriented and Classical Software Engineering*, McGraw-Hill, New York.

[Sen84] Sen, T., and Gupta, S. K. (1984), "A state-of-art survey of static scheduling research involving due dates", *OMEGA-International Journal of Management Science*, 12(1), 63-76.

[Sil98] Silver, E. A., Pyke, D. F., and Peterson, R. (1998), *Inventory Management and Production Planning and Scheduling*, Wiley, New York.

[Siv99] Sivrikaya-Serifoglu, F., and Ulusoy, G. (1999), "Parallel machine scheduling with earliness and tardiness penalties", *Computers and Operations Research*, 26(8), 773-787.

[Spe90] Spearman, M. L., Woodruff, D. J., and Hopp, W. J. (1990), "CONWIP: A pull alternative to kanban", *International Journal of*

Production Research, 28(5), 879-894.

[Spe98] Spearman, M. L., and Hopp, W. J. (1998), “Teaching operations management from a science of manufacturing”, *Production and Operations Management*, 7(2), 132-145.

[Sri71] Srinivasan, V. (1971), “A hybrid algorithm for the one machine sequencing problem to minimize total tardiness”, *Naval Research Logistics Quarterly*, 18(3), 317 – 327.

[Sri96] Sridharan, V., and Zhou, Z. (1996), “A decision theory based scheduling procedure for single-machine weighted earliness and tardiness problems”, *European Journal of Operational Research*, 94(2), 292-301.

[Sta05] Stadtler, H., and Kilger, C. (2005), *Supply Chain Management and Advanced Planning: Concepts, Models, Software and Case Studies*. Springer, Berlin.

[Ste80] Steinberg, E., and Napier, H. A. (1980), “Optimal multi-level lot sizing for requirements planning systems”, *Management Science*, 26(12), 1258-1271.

[Suh98] Suh, M. S., Lee, A., Lee, Y. J., and Ko, Y. K. (1998), “Evaluation of ordering strategies for constraint satisfaction reactive scheduling”, *Decision Support Systems*, 22(2), 187-197.

[Sum93] Sum, C. C., and Hill, A. V. (1993), “A new framework for manufacturing planning and control systems”, *Decision Sciences*, 24(4), 739-760.

[Sun84] Sundararaghavan, P. S., and Ahmed, M. U. (1984), “Minimizing the sum of absolute lateness in single-machine and multimachine scheduling”, *Naval Research Logistics Quarterly*, 31(2), 325-333.

[Sun01a] Sun, J., and Xue, D. (2001), “A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources”, *Computers in Industry*, 46(2), 189-207.

[Sun01b] Sung, C. S., and Min, J. I. (2001), “Scheduling in a two-machine flowshop with batch processing machine(s) for earliness/tardiness measure under a common due date”, *European Journal of Operational Research*, 131(1), 95-106.

[Sur93] Suresh, V., and Chaudhuri, D. (1993), “Dynamic scheduling - a survey of research”, *International Journal of Production Economics*, 32(1), 53-63.

[Szw60] Szwarc W. (1960), “Solution of the Akers-Friedman scheduling problem”, *Operations Research*, 8(6), 782-788.

[Szw89] Szwarc, W. (1989), “Single-machine scheduling to minimize absolute deviation of completion times from a common due date”, *Naval Research Logistics*, 36(5), 663-673.

[Taa97] Taal, M., and Wortmann, J. C. (1997), “Integrating MRP and finite capacity planning”, *Production Planning and Control*, 8(3), 245-254.

[Tah03] Taha, H. A. (2003), *Operations Research: an Introduction*, Prentice Hall, Upper Saddle River, New Jersey.

[Tai94] Taillard, É. D. (1994), "Parallel taboo search techniques for the job shop scheduling problem", *ORSA Journal on Computing*, 6(2), 108-117.

[Tsu91] Tsubone, H., Anzai, M., Sugawara, M., and Matsuura, H. (1991), "An interactive production planning and scheduling system for a flow-type manufacturing process", *International Journal of Production Economics*, 22(1), 43-51.

[Una97] Unal, A.T., Uzsoy, R., and Kiran, A.S. (1997), "Rescheduling on a single machine with part-type dependent setup times and deadlines", *Annals of Operations Research*, 70(1), 93-113.

[Van92] Van Laarhoven, P. J. M., Aarts, E. H. L., and Lenstra, J. K. (1992), "Job shop scheduling by simulated annealing", *Operations Research*, 40(1), 113-125.

[Vie03] Vieira, G. E., Herrmann, J. W., and Lin, E. (2003), "Rescheduling manufacturing systems: a framework of strategies, policies, and methods", *Journal of Scheduling*, 6(1), 39-62.

[Vol88] Vollmann, T. E., Berry, W. L., and Whybark, D. C. (1988), *Manufacturing Planning and Control Systems*, Dow Jones-Irwin, Homewood.

[Vos03] Voss, S., and Woodruff, D. L. (2003), *Introduction to Computational Optimization Models for Production Planning in a Supply Chain*, Springer, Berlin.

[Wal90] Wallace, T. F. (1990), *MRP II: Making it Happen*, Oliver Wight Limited Publications, Essex Junction.

[Wan95] Wang, D. W. (1995), "Earliness/tardiness production planning approaches for manufacturing systems", *Computers and Industrial Engineering*, 28(3), 425-436.

[Wan96] Wang, D. W., Chen, X. Z., and Li, Y. (1996), "Experimental push/pull production planning and control system", *Production Planning and Control*, 7 (3), 236-241.

[Web97] Webster, S. T. (1997), "The complexity of scheduling job families about a common due date", *Operations Research Letters*, 20(2), 65-74.

[Whi82] White, E. M., Anderson, J. C., Schroeder, R. G., and Tupy, S. E. (1982), "A study of the MRP implementation process", *Journal of Operations Management*, 2(3), 145-154.

[Whi90] White, K. P. (1990), "Advances in the theory and practice of production scheduling", in: *Control and Dynamic Systems: Advances in Theory and Applications*, Leondes C. T. (eds), Vol. 37, Academic Press, New York, 115-157.

[Wid91] Widmer, M. (1991), "Job shop scheduling with tooling constraints: a tabu search approach", *Journal of Operations Research Society*, 42(1), 75-82.

[Wig74] Wight, O. W. (1974), *Production and Inventory Management in the Computer Age*, Cahners, Boston.

[Wil99] Williams, H. P. (1999), *Model Building in Mathematical Programming*, Wiley, New York.

[Wus93] Wu, S. D., Storer, R. H., Chang, P. C. (1993), "One-machine rescheduling heuristics with efficiency and stability as criteria", *Computers and Operations Research*, 20(1), 1-14.

[Yan91] Yano, C. A., and Kim, Y. (1991), "Algorithms for a class of single-machine weighted tardiness and earliness problems", *European Journal of Operational Research*, 52(2), 167-178.

[Yeh00] Yeh, C. H. (2000), "A customer-focused planning approach to make-to-order production", *Industrial Management and Data Systems*, 100(4), 180-187.

[Zen05] Zeng, X., and Zhou, Z. (2005), "Discussion of advanced planning and scheduling system", *Industrial Engineering Journal*, 8(1), 53-56.

APPENDIX I

THE PROBLEM FORMULATION

(5 ORDERS, 6 MACHINES AND 5 LEVELS)

Minimize

300Cmax-6575
+250LI1+250LI2+250LI3+250LI4+250LI5
+50EI1+50EI2+50EI3+50EI4+50EI5

Subject to

(3.2):

cons1: C1-Cmax<=0
cons2: C2-Cmax<=0
cons3: C3-Cmax<=0
cons4: C4-Cmax<=0
cons5: C5-Cmax<=0

(3.3):

cons6: S131>=1
cons7: S141>=1
cons8: S221>=1
cons9: S351>=1
cons10: S371>=1
cons11: S431>=1
cons12: S511>=1
cons13: S152>=2
cons14: S252>=2
cons15: S3162>=2
cons16: S3172>=2
cons17: S3182>=2
cons18: S472>=2
cons19: S482>=2
cons20: S123>=3
cons21: S263>=3
cons22: S3133>=3
cons23: S3143>=3
cons24: S3153>=3
cons25: S443>=3
cons26: S453>=3
cons27: S463>=3
cons28: S2124>=3
cons29: S2134>=3
cons30: S3194>=3
cons31: S3204>=3
cons32: S3214>=3
cons33: S3224>=3

(3.4):

cons34: S116-S123>=1.000000
cons35: S116-S131>=0.500000
cons36: S116-S141>=0.500000
cons37: S116-S152>=1.000000
cons38: S216-S221>=0.500000
cons39: S216-S236>=2.500000
cons40: S236-S245>=2.500000
cons41: S236-S252>=1.000000
cons42: S245-S263>=2.000000
cons43: S245-S2115>=1.500000
cons44: S2115-S2124>=0.500000
cons45: S2115-S2134>=0.500000
cons46: S316-S326>=6.000000
cons47: S316-S335>=10.000000
cons48: S316-S351>=3.000000
cons49: S326-S366>=6.000000
cons50: S326-S371>=3.000000
cons51: S335-S385>=6.000000

cons52: S335-S3105>=6.000000
cons53: S366-S3133>=2.000000
cons54: S366-S3143>=2.000000
cons55: S366-S3153>=1.000000
cons56: S366-S3162>=2.000000
cons57: S366-S3172>=1.000000
cons58: S385-S3182>=6.000000
cons59: S385-S3194>=4.000000
cons60: S3105-S3204>=2.000000
cons61: S3105-S3214>=2.000000
cons62: S3194-S3224>=4.000000
cons63: S416-S426>=6.000000
cons64: S416-S431>=3.000000
cons65: S426-S443>=2.000000
cons66: S426-S453>=2.000000
cons67: S426-S463>=1.000000
cons68: S426-S472>=2.000000
cons69: S426-S482>=1.000000

(3.5):
cons70: C1-S116=3.5
cons71: C2-S216=3
cons72: C3-S316=7
cons73: C4-S416=6
cons74: C5-S511=3

(3.6) and (3.7):
cons75: S131-S141+999X13141>=0.500000
cons76: S141-S131+999X14131>=0.500000
cons77: X13141+X14131=1
cons78: S131-S221+999X13221>=0.500000
cons79: S221-S131+999X22131>=0.500000
cons80: X13221+X22131=1
cons81: S131-S351+999X13351>=3.000000
cons82: S351-S131+999X35131>=0.500000
cons83: X13351+X35131=1
cons84: S131-S371+999X13371>=3.000000
cons85: S371-S131+999X37131>=0.500000
cons86: X13371+X37131=1
cons87: S131-S431+999X13431>=3.000000
cons88: S431-S131+999X43131>=0.500000
cons89: X13431+X43131=1
cons90: S131-S511+999X13511>=3.000000
cons91: S511-S131+999X51131>=0.500000
cons92: X13511+X51131=1
cons93: S141-S221+999X14221>=0.500000
cons94: S221-S141+999X22141>=0.500000
cons95: X14221+X22141=1
cons96: S141-S351+999X14351>=3.000000
cons97: S351-S141+999X35141>=0.500000
cons98: X14351+X35141=1
cons99: S141-S371+999X14371>=3.000000
cons100: S371-S141+999X37141>=0.500000
cons101: X14371+X37141=1
cons102: S141-S431+999X14431>=3.000000
cons103: S431-S141+999X43141>=0.500000
cons104: X14431+X43141=1

cons105: S141-S511+999X14511>=3.000000
cons106: S511-S141+999X51141>=0.500000
cons107: X14511+X51141=1
cons108: S221-S351+999X22351>=3.000000
cons109: S351-S221+999X35221>=0.500000
cons110: X22351+X35221=1
cons111: S221-S371+999X22371>=3.000000
cons112: S371-S221+999X37221>=0.500000
cons113: X22371+X37221=1
cons114: S221-S431+999X22431>=3.000000
cons115: S431-S221+999X43221>=0.500000
cons116: X22431+X43221=1
cons117: S221-S511+999X22511>=3.000000
cons118: S511-S221+999X51221>=0.500000
cons119: X22511+X51221=1
cons120: S351-S371+999X35371>=3.000000
cons121: S371-S351+999X37351>=3.000000
cons122: X35371+X37351=1
cons123: S351-S431+999X35431>=3.000000
cons124: S431-S351+999X43351>=3.000000
cons125: X35431+X43351=1
cons126: S351-S511+999X35511>=3.000000
cons127: S511-S351+999X51351>=3.000000
cons128: X35511+X51351=1
cons129: S371-S431+999X37431>=3.000000
cons130: S431-S371+999X43371>=3.000000
cons131: X37431+X43371=1
cons132: S371-S511+999X37511>=3.000000
cons133: S511-S371+999X51371>=3.000000
cons134: X37511+X51371=1
cons135: S431-S511+999X43511>=3.000000
cons136: S511-S431+999X51431>=3.000000
cons137: X43511+X51431=1
cons138: S152-S252+999X15252>=1.000000
cons139: S252-S152+999X25152>=1.000000
cons140: X15252+X25152=1
cons141: S152-S3162+999X153162>=2.000000
cons142: S3162-S152+999X316152>=1.000000
cons143: X153162+X316152=1
cons144: S152-S3172+999X153172>=1.000000
cons145: S3172-S152+999X317152>=1.000000
cons146: X153172+X317152=1
cons147: S152-S3182+999X153182>=6.000000
cons148: S3182-S152+999X318152>=1.000000
cons149: X153182+X318152=1
cons150: S152-S472+999X15472>=2.000000

cons151: S472-S152+999X47152>=1.000000
cons152: X15472+X47152=1
cons153: S152-S482+999X15482>=1.000000
cons154: S482-S152+999X48152>=1.000000
cons155: X15482+X48152=1
cons156: S252-S3162+999X253162>=2.000000
cons157: S3162-S252+999X316252>=1.000000
cons158: X253162+X316252=1
cons159: S252-S3172+999X253172>=1.000000
cons160: S3172-S252+999X317252>=1.000000
cons161: X253172+X317252=1
cons162: S252-S3182+999X253182>=6.000000
cons163: S3182-S252+999X318252>=1.000000
cons164: X253182+X318252=1
cons165: S252-S472+999X25472>=2.000000
cons166: S472-S252+999X47252>=1.000000
cons167: X25472+X47252=1
cons168: S252-S482+999X25482>=1.000000
cons169: S482-S252+999X48252>=1.000000
cons170: X25482+X48252=1
cons171: S3162-S3172+999X3163172>=1.000000
cons172: S3172-S3162+999X3173162>=2.000000
cons173: X3163172+X3173162=1
cons174: S3162-S3182+999X3163182>=6.000000
cons175: S3182-S3162+999X3183162>=2.000000
cons176: X3163182+X3183162=1
cons177: S3162-S472+999X316472>=2.000000
cons178: S472-S3162+999X473162>=2.000000
cons179: X316472+X473162=1
cons180: S3162-S482+999X316482>=1.000000
cons181: S482-S3162+999X483162>=2.000000
cons182: X316482+X483162=1
cons183: S3172-S3182+999X3173182>=6.000000
cons184: S3182-S3172+999X3183172>=1.000000
cons185: X3173182+X3183172=1
cons186: S3172-S472+999X317472>=2.000000
cons187: S472-S3172+999X473172>=1.000000
cons188: X317472+X473172=1
cons189: S3172-S482+999X317482>=1.000000
cons190: S482-S3172+999X483172>=1.000000
cons191: X317482+X483172=1
cons192: S3182-S472+999X318472>=2.000000
cons193: S472-S3182+999X473182>=6.000000
cons194: X318472+X473182=1
cons195: S3182-S482+999X318482>=1.000000
cons196: S482-S3182+999X483182>=6.000000

cons197: X318482+X483182=1
cons198: S472-S482+999X47482>=1.000000
cons199: S482-S472+999X48472>=2.000000
cons200: X47482+X48472=1
cons201: S123-S263+999X12263>=2.000000
cons202: S263-S123+999X26123>=1.000000
cons203: X12263+X26123=1
cons204: S123-S3133+999X123133>=2.000000
cons205: S3133-S123+999X313123>=1.000000
cons206: X123133+X313123=1
cons207: S123-S3143+999X123143>=2.000000
cons208: S3143-S123+999X314123>=1.000000
cons209: X123143+X314123=1
cons210: S123-S3153+999X123153>=1.000000
cons211: S3153-S123+999X315123>=1.000000
cons212: X123153+X315123=1
cons213: S123-S443+999X12443>=2.000000
cons214: S443-S123+999X44123>=1.000000
cons215: X12443+X44123=1
cons216: S123-S453+999X12453>=2.000000
cons217: S453-S123+999X45123>=1.000000
cons218: X12453+X45123=1
cons219: S123-S463+999X12463>=1.000000
cons220: S463-S123+999X46123>=1.000000
cons221: X12463+X46123=1
cons222: S263-S3133+999X263133>=2.000000
cons223: S3133-S263+999X313263>=2.000000
cons224: X263133+X313263=1
cons225: S263-S3143+999X263143>=2.000000
cons226: S3143-S263+999X314263>=2.000000
cons227: X263143+X314263=1
cons228: S263-S3153+999X263153>=1.000000
cons229: S3153-S263+999X315263>=2.000000
cons230: X263153+X315263=1
cons231: S263-S443+999X26443>=2.000000
cons232: S443-S263+999X44263>=2.000000
cons233: X26443+X44263=1
cons234: S263-S453+999X26453>=2.000000
cons235: S453-S263+999X45263>=2.000000
cons236: X26453+X45263=1
cons237: S263-S463+999X26463>=1.000000
cons238: S463-S263+999X46263>=2.000000
cons239: X26463+X46263=1
cons240: S3133-S3143+999X3133143>=2.000000
cons241: S3143-S3133+999X3143133>=2.000000
cons242: X3133143+X3143133=1

cons243: S3133-S3153+999X3133153>=1.000000
cons244: S3153-S3133+999X3153133>=2.000000
cons245: X3133153+X3153133=1
cons246: S3133-S443+999X313443>=2.000000
cons247: S443-S3133+999X443133>=2.000000
cons248: X313443+X443133=1
cons249: S3133-S453+999X313453>=2.000000
cons250: S453-S3133+999X453133>=2.000000
cons251: X313453+X453133=1
cons252: S3133-S463+999X313463>=1.000000
cons253: S463-S3133+999X463133>=2.000000
cons254: X313463+X463133=1
cons255: S3143-S3153+999X3143153>=1.000000
cons256: S3153-S3143+999X3153143>=2.000000
cons257: X3143153+X3153143=1
cons258: S3143-S443+999X314443>=2.000000
cons259: S443-S3143+999X443143>=2.000000
cons260: X314443+X443143=1
cons261: S3143-S453+999X314453>=2.000000
cons262: S453-S3143+999X453143>=2.000000
cons263: X314453+X453143=1
cons264: S3143-S463+999X314463>=1.000000
cons265: S463-S3143+999X463143>=2.000000
cons266: X314463+X463143=1
cons267: S3153-S443+999X315443>=2.000000
cons268: S443-S3153+999X443153>=1.000000
cons269: X315443+X443153=1
cons270: S3153-S453+999X315453>=2.000000
cons271: S453-S3153+999X453153>=1.000000
cons272: X315453+X453153=1
cons273: S3153-S463+999X315463>=1.000000
cons274: S463-S3153+999X463153>=1.000000
cons275: X315463+X463153=1
cons276: S443-S453+999X44453>=2.000000
cons277: S453-S443+999X45443>=2.000000
cons278: X44453+X45443=1
cons279: S443-S463+999X44463>=1.000000
cons280: S463-S443+999X46443>=2.000000
cons281: X44463+X46443=1
cons282: S453-S463+999X45463>=1.000000
cons283: S463-S453+999X46453>=2.000000
cons284: X45463+X46453=1
cons285: S2124-S2134+999X2122134>=0.500000
cons286: S2134-S2124+999X2132124>=0.500000
cons287: X2122134+X2132124=1
cons288: S2124-S3194+999X2123194>=4.000000

cons289: S3194-S2124+999X3192124>=0.500000
cons290: X2123194+X3192124=1
cons291: S2124-S3204+999X2123204>=2.000000
cons292: S3204-S2124+999X3202124>=0.500000
cons293: X2123204+X3202124=1
cons294: S2124-S3214+999X2123214>=2.000000
cons295: S3214-S2124+999X3212124>=0.500000
cons296: X2123214+X3212124=1
cons297: S2124-S3224+999X2123224>=4.000000
cons298: S3224-S2124+999X3222124>=0.500000
cons299: X2123224+X3222124=1
cons300: S2134-S3194+999X2133194>=4.000000
cons301: S3194-S2134+999X3192134>=0.500000
cons302: X2133194+X3192134=1
cons303: S2134-S3204+999X2133204>=2.000000
cons304: S3204-S2134+999X3202134>=0.500000
cons305: X2133204+X3202134=1
cons306: S2134-S3214+999X2133214>=2.000000
cons307: S3214-S2134+999X3212134>=0.500000
cons308: X2133214+X3212134=1
cons309: S2134-S3224+999X2133224>=4.000000
cons310: S3224-S2134+999X3222134>=0.500000
cons311: X2133224+X3222134=1
cons312: S3194-S3204+999X3193204>=2.000000
cons313: S3204-S3194+999X3203194>=4.000000
cons314: X3193204+X3203194=1
cons315: S3194-S3214+999X3193214>=2.000000
cons316: S3214-S3194+999X3213194>=4.000000
cons317: X3193214+X3213194=1
cons318: S3194-S3224+999X3193224>=4.000000
cons319: S3224-S3194+999X3223194>=4.000000
cons320: X3193224+X3223194=1
cons321: S3204-S3214+999X3203214>=2.000000
cons322: S3214-S3204+999X3213204>=2.000000
cons323: X3203214+X3213204=1
cons324: S3204-S3224+999X3203224>=4.000000
cons325: S3224-S3204+999X3223204>=2.000000
cons326: X3203224+X3223204=1
cons327: S3214-S3224+999X3213224>=4.000000
cons328: S3224-S3214+999X3223214>=2.000000
cons329: X3213224+X3223214=1
cons330: S245-S2115+999X242115>=1.500000
cons331: S2115-S245+999X211245>=2.500000
cons332: X242115+X211245=1
cons333: S245-S335+999X24335>=10.000000
cons334: S335-S245+999X33245>=2.500000

cons335: X24335+X33245=1
cons336: S245-S385+999X24385>=6.000000
cons337: S385-S245+999X38245>=2.500000
cons338: X24385+X38245=1
cons339: S245-S3105+999X243105>=6.000000
cons340: S3105-S245+999X310245>=2.500000
cons341: X243105+X310245=1
cons342: S2115-S335+999X211335>=10.000000
cons343: S335-S2115+999X332115>=1.500000
cons344: X211335+X332115=1
cons345: S2115-S385+999X211385>=6.000000
cons346: S385-S2115+999X382115>=1.500000
cons347: X211385+X382115=1
cons348: S2115-S3105+999X2113105>=6.000000
cons349: S3105-S2115+999X3102115>=1.500000
cons350: X2113105+X3102115=1
cons351: S335-S385+999X33385>=6.000000
cons352: S385-S335+999X38335>=10.000000
cons353: X33385+X38335=1
cons354: S335-S3105+999X333105>=6.000000
cons355: S3105-S335+999X310335>=10.000000
cons356: X333105+X310335=1
cons357: S385-S3105+999X383105>=6.000000
cons358: S3105-S385+999X310385>=6.000000
cons359: X383105+X310385=1
cons360: S116-S216+999X11216>=3.000000
cons361: S216-S116+999X21116>=3.500000
cons362: X11216+X21116=1
cons363: S116-S236+999X11236>=2.500000
cons364: S236-S116+999X23116>=3.500000
cons365: X11236+X23116=1
cons366: S116-S316+999X11316>=7.000000
cons367: S316-S116+999X31116>=3.500000
cons368: X11316+X31116=1
cons369: S116-S326+999X11326>=6.000000
cons370: S326-S116+999X32116>=3.500000
cons371: X11326+X32116=1
cons372: S116-S366+999X11366>=6.000000
cons373: S366-S116+999X36116>=3.500000
cons374: X11366+X36116=1
cons375: S116-S416+999X11416>=6.000000
cons376: S416-S116+999X41116>=3.500000
cons377: X11416+X41116=1
cons378: S116-S426+999X11426>=6.000000
cons379: S426-S116+999X42116>=3.500000
cons380: X11426+X42116=1

cons381: S216-S236+999X21236>=2.500000
cons382: S236-S216+999X23216>=3.000000
cons383: X21236+X23216=1
cons384: S216-S316+999X21316>=7.000000
cons385: S316-S216+999X31216>=3.000000
cons386: X21316+X31216=1
cons387: S216-S326+999X21326>=6.000000
cons388: S326-S216+999X32216>=3.000000
cons389: X21326+X32216=1
cons390: S216-S366+999X21366>=6.000000
cons391: S366-S216+999X36216>=3.000000
cons392: X21366+X36216=1
cons393: S216-S416+999X21416>=6.000000
cons394: S416-S216+999X41216>=3.000000
cons395: X21416+X41216=1
cons396: S216-S426+999X21426>=6.000000
cons397: S426-S216+999X42216>=3.000000
cons398: X21426+X42216=1
cons399: S236-S316+999X23316>=7.000000
cons400: S316-S236+999X31236>=2.500000
cons401: X23316+X31236=1
cons402: S236-S326+999X23326>=6.000000
cons403: S326-S236+999X32236>=2.500000
cons404: X23326+X32236=1
cons405: S236-S366+999X23366>=6.000000
cons406: S366-S236+999X36236>=2.500000
cons407: X23366+X36236=1
cons408: S236-S416+999X23416>=6.000000
cons409: S416-S236+999X41236>=2.500000
cons410: X23416+X41236=1
cons411: S236-S426+999X23426>=6.000000
cons412: S426-S236+999X42236>=2.500000
cons413: X23426+X42236=1
cons414: S316-S326+999X31326>=6.000000
cons415: S326-S316+999X32316>=7.000000
cons416: X31326+X32316=1
cons417: S316-S366+999X31366>=6.000000
cons418: S366-S316+999X36316>=7.000000
cons419: X31366+X36316=1
cons420: S316-S416+999X31416>=6.000000
cons421: S416-S316+999X41316>=7.000000
cons422: X31416+X41316=1
cons423: S316-S426+999X31426>=6.000000
cons424: S426-S316+999X42316>=7.000000
cons425: X31426+X42316=1
cons426: S326-S366+999X32366>=6.000000

cons427: S366-S326+999X36326>=6.000000
cons428: X32366+X36326=1
cons429: S326-S416+999X32416>=6.000000
cons430: S416-S326+999X41326>=6.000000
cons431: X32416+X41326=1
cons432: S326-S426+999X32426>=6.000000
cons433: S426-S326+999X42326>=6.000000
cons434: X32426+X42326=1
cons435: S366-S416+999X36416>=6.000000
cons436: S416-S366+999X41366>=6.000000
cons437: X36416+X41366=1
cons438: S366-S426+999X36426>=6.000000
cons439: S426-S366+999X42366>=6.000000
cons440: X36426+X42366=1
cons441: S416-S426+999X41426>=6.000000
cons442: S426-S416+999X42416>=6.000000
cons443: X41426+X42416=1

(3.8) and (3.9):
cons444: 0.125C1-L1<=6
cons445: E1+0.125C1>=6
cons446: 0.125C2-L2<=7
cons447: E2+0.125C2>=7
cons448: 0.125C3-L3<=14
cons449: E3+0.125C3>=14
cons450: 0.125C4-L4<=3
cons451: E4+0.125C4>=3
cons452: 0.125C5-L5<=1
cons453: E5+0.125C5>=1

(3.10) and (3.11):
cons454: L1-LI1<=0
cons455: E1-EI1<=0.99
cons456: L2-LI2<=0
cons457: E2-EI2<=0.99
cons458: L3-LI3<=0
cons459: E3-EI3<=0.99
cons460: L4-LI4<=0
cons461: E4-EI4<=0.99
cons462: L5-LI5<=0
cons463: E5-EI5<=0.99

Bounds
LI1 free LI3 free LI5 free EI2 free EI4 free
LI2 free LI4 free EI1 free EI3 free EI5 free

Integers
EI1 LI1 EI2 LI2 EI3 LI3 EI4 LI4 EI5 LI5

X13141 X14131
X13221 X22131
X13351 X35131
X13371 X37131
X13431 X43131
X13511 X51131
X14221 X22141
X14351 X35141
X14371 X37141
X14431 X43141
X14511 X51141
X22351 X35221
X22371 X37221
X22431 X43221
X22511 X51221
X35371 X37351
X35431 X43351
X35511 X51351
X37431 X43371
X37511 X51371
X43511 X51431

X15252 X25152
X153162 X316152
X153172 X317152
X153182 X318152
X15472 X47152
X15482 X48152
X253162 X316252
X253172 X317252
X253182 X318252
X25472 X47252
X25482 X48252
X3163172 X3173162
X3163182 X3183162
X316472 X473162
X316482 X483162
X3173182 X3183172
X317472 X473172
X317482 X483172
X318472 X473182
X318482 X483182
X47482 X48472
X12263 X26123
X123133 X313123
X123143 X314123
X123153 X315123
X12443 X44123
X12453 X45123
X12463 X46123
X263133 X313263
X263143 X314263
X263153 X315263
X26443 X44263
X26453 X45263
X26463 X46263
X3133143 X3143133
X3133153 X3153133
X313443 X443133
X313453 X453133
X313463 X463133
X3143153 X3153143
X314443 X443143
X314453 X453143
X314463 X463143
X315443 X443153
X315453 X453153
X315463 X463153
X44453 X45443
X44463 X46443
X45463 X46453
X2122134 X2132124
X2123194 X3192124
X2123204 X3202124
X2123214 X3212124
X2123224 X3222124
X2133194 X3192134
X2133204 X3202134
X2133214 X3212134
X2133224 X3222134
X3193204 X3203194
X3193214 X3213194
X3193224 X3223194
X3203214 X3213204
X3203224 X3223204
X3213224 X3223214
X242115 X211245
X24335 X33245
X24385 X38245
X243105 X310245
X211335 X332115
X211385 X382115
X2113105 X3102115
X33385 X38335
X333105 X310335
X383105 X310385
X11216 X21116
X11236 X23116
X11316 X31116
X11326 X32116
X11366 X36116
X11416 X41116
X11426 X42116
X21236 X23216
X21316 X31216
X21326 X32216
X21366 X36216
X21416 X41216
X21426 X42216
X23316 X31236
X23326 X32236
X23366 X36236
X23416 X41236
X23426 X42236
X31326 X32316
X31366 X36316
X31416 X41316
X31426 X42316
X32366 X36326
X32416 X41326
X32426 X42326
X36416 X41366
X36426 X42366
X41426 X42416

End

APPENDIX II

THE OPTIMAL RESULTS

(5 ORDERS, 6 MACHINES AND 5 LEVELS)

Integer optimal
Objective = 7.5750000000e+003
Solution time = 69814.53 sec.
Iterations = 454420772
Nodes = 133237299

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	45.000000	S463	10.000000
EI1	4.000000	S2124	3.000000
EI2	1.000000	S2134	3.500000
EI3	8.000000	S3194	15.000000
C1	8.080000	S3204	6.500000
C2	45.000000	S3214	4.000000
C3	42.000000	S3224	11.000000
C4	23.000000	S116	4.580000
C5	8.000000	S216	42.000000
S131	1.000000	S236	8.500000
S141	1.500000	S245	6.000000
S221	8.000000	S2115	4.000000
S351	32.000000	S316	35.000000
S371	8.500000	S326	29.000000
S431	2.000000	S335	25.000000
S511	5.000000	S366	23.000000
S152	3.000000	S385	19.000000
S252	2.000000	S3105	8.500000
S3162	14.000000	S416	17.000000
S3172	4.000000	S426	11.000000
S3182	8.000000	X13141	1.000000
S472	6.000000	X13221	1.000000
S482	5.000000	X13351	1.000000
S123	3.000000	X13371	1.000000
S263	4.000000	X13431	1.000000
S3133	19.000000	X13511	1.000000
S3143	21.000000	X14221	1.000000
S3153	18.000000	X14351	1.000000
S443	8.000000	X14371	1.000000
S453	6.000000	X14431	1.000000

X14511	1.000000	X3153133	1.000000
X22351	1.000000	X443133	1.000000
X22371	1.000000	X453133	1.000000
X43221	1.000000	X463133	1.000000
X51221	1.000000	X3153143	1.000000
X37351	1.000000	X443143	1.000000
X43351	1.000000	X453143	1.000000
X51351	1.000000	X463143	1.000000
X43371	1.000000	X443153	1.000000
X51371	1.000000	X453153	1.000000
X43511	1.000000	X463153	1.000000
X25152	1.000000	X45443	1.000000
X153162	1.000000	X44463	1.000000
X153172	1.000000	X45463	1.000000
X153182	1.000000	X2122134	1.000000
X15472	1.000000	X2123194	1.000000
X15482	1.000000	X2123204	1.000000
X253162	1.000000	X2123214	1.000000
X253172	1.000000	X2123224	1.000000
X253182	1.000000	X2133194	1.000000
X25472	1.000000	X2133204	1.000000
X25482	1.000000	X2133214	1.000000
X3173162	1.000000	X2133224	1.000000
X3183162	1.000000	X3193224	1.000000
X473162	1.000000	X3203194	1.000000
X483162	1.000000	X3203224	1.000000
X3173182	1.000000	X3213194	1.000000
X317472	1.000000	X3213204	1.000000
X317482	1.000000	X3213224	1.000000
X473182	1.000000	X211245	1.000000
X483182	1.000000	X24335	1.000000
X48472	1.000000	X24385	1.000000
X12263	1.000000	X243105	1.000000
X123133	1.000000	X211335	1.000000
X123143	1.000000	X211385	1.000000
X123153	1.000000	X2113105	1.000000
X12443	1.000000	X38335	1.000000
X12453	1.000000	X310335	1.000000
X12463	1.000000	X310385	1.000000
X263133	1.000000	X11216	1.000000
X263143	1.000000	X11236	1.000000
X263153	1.000000	X11316	1.000000
X26443	1.000000	X11326	1.000000
X26453	1.000000	X11366	1.000000
X26463	1.000000	X11416	1.000000
X3133143	1.000000	X11426	1.000000

X23216	1.000000	X41316	1.000000
X31216	1.000000	X42316	1.000000
X32216	1.000000	X36326	1.000000
X36216	1.000000	X41326	1.000000
X41216	1.000000	X42326	1.000000
X42216	1.000000	X41366	1.000000
X23316	1.000000	X42366	1.000000
X23326	1.000000	X42416	1.000000
X23366	1.000000	E1	4.990000
X23416	1.000000	E2	1.990000
X23426	1.000000	E3	8.990000
X32316	1.000000	E4	0.990000
X36316	1.000000	E5	0.990000

All other variables are zero.

APPENDIX III

THE APS PROBLEM (3 ORDERS, 4 MACHINES AND 5 LEVELS)

The five-level product structure in [Lee02] is adopted (Figure III-1). Four machines, with 8 hours available per day, are eligible to process the items (Table III-1). It should be noted that item S6 has three operations (OP1, OP2, OP3) to process, and then S6 is further divided into three child items: S6OP1, S6OP2, S6OP3. Moreover, M1 and M2 are responsible for assembling, while M3 and M4 deal with the components. The ready times of the four machines are Hour 3, Hour 2, Hour 2 and Hour 1, respectively. There are three orders: 5 Product F1s with due date Day 5, 10 Product S3s with due date Day 3, and 30 Product C4s with due date Day 1. The penalty rates are as follows: cost of idle time at $60 per hour, cost of tardiness at $250 per day per order, and cost of earliness at $50 per day per order.

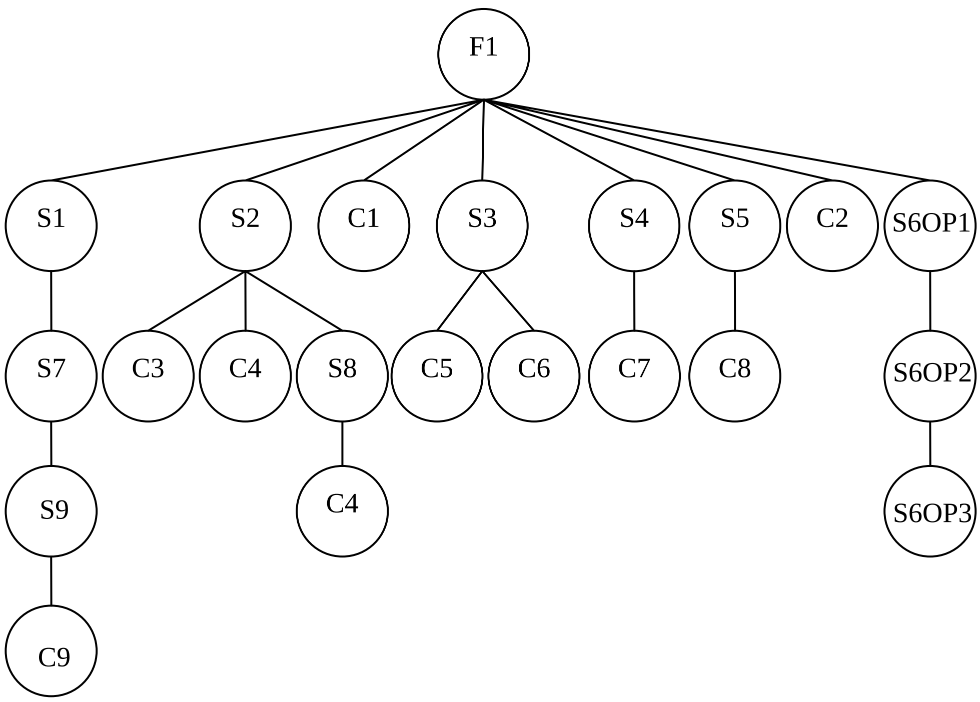


Figure III-1 The product structure in example 3 × 4 × 5

Table III-1 Machine processing time for the items in example 3 × 4 × 5

Items	Machine number	Processing time (hours)
F1	M1	0.7
S1	M1	0.4
S2	M1	0.6
S3	M1	0.5
S4	M2	0.5
S5	M2	0.3
S6OP1	M1	0.4
S6OP2	M2	0.4
S6OP3	M3	0.1
S7	M2	0.4
S8	M2	0.3
S9	M2	0.3
C1	M3	0.2
C2	M3	0.2
C3	M4	0.4
C4	M3	0.1
C5	M4	0.2
C6	M4	0.2
C7	M4	0.3
C8	M4	0.4
C9	M3	0.1

The optimal solution generated by solving the developed MIP using CPLEX 9.1 is listed below. The important data extracted from the optimal results are summarized in Table III-2.

SOLVE SUMMARY

Integer optimal solution

Objective = 1.8100000000e+003

Solution time = 2.68 sec.

Iterations = 26621

Nodes = 5537

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	21.000000	S1114	3.000
EI1	2.000000	S1123	3.000
C1	21.000000	S1132	4.000
C2	17.500000	S1144	1.000
C3	6.500000	S1154	2.000
S111	17.500	S1164	11.000
S121	8.500	S1174	7.000
S131	5.500	S1182	7.500
S143	8.000	S1192	2.500
S151	3.000	S1203	2.500
S162	12.500	S1213	6.500
S172	9.500	S1223	2.000
S183	7.000	S211	12.500
S191	10.500	S224	5.000
S1102	5.500	S234	9.000
		S313	3.500

Table III-2 The optimal result of example 3 × 4 × 5

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
271	186	150	21	2.68

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
0	2	0	100	1810

APPENDIX IV

THE APS PROBLEM (4 ORDERS, 5 MACHINES AND 5 LEVELS)

The product structure modified from [Pon02] is adopted (Figure IV-1). Five machines, with 8 hours available per day, are eligible to process the items (Table IV-1). Moreover, M1 and M2 deal with the components, while M3, M4 and M5 are responsible for assembling. The ready times of the four machines are Hour 2, Hour 2, Hour 3, Hour 4 and Hour 4, respectively. There are four orders: 10 Product F1s with due date Day 14, 5 Product S2s with due date Day 5, 20 Product S8s with due date Day 3, and 10 Product S4s with due date Day 7. The penalty rates are as follows: cost of idle time at $50 per hour, cost of tardiness at $250 per day per order, and cost of earliness at $50 per day per order.

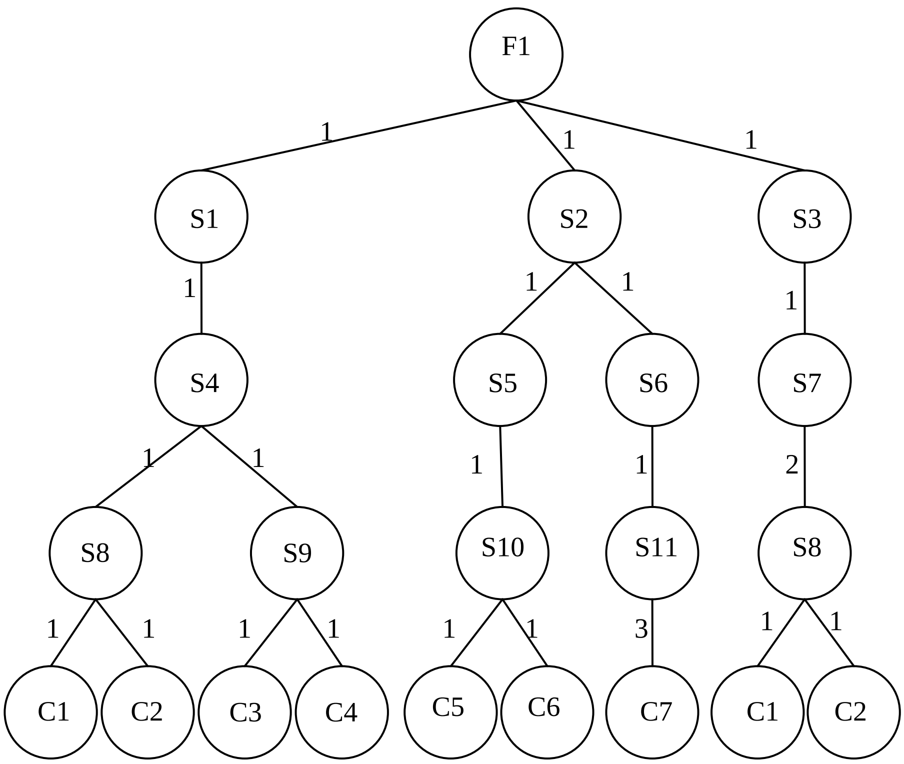


Figure IV-1 The product structure in example 4 × 5 × 5

Table IV-1 Machine processing time for the items in example 4 × 4 × 5

Items	Machine number	Processing time (hours)
F1	M5	0.6
S1	M5	0.4
S2	M4	0.6
S3	M3	0.4
S4	M5	0.5
S5	M3	0.3
S6	M3	0.3
S7	M4	0.4
S8	M4	0.5
S9	M5	0.3
S10	M3	0.3
S11	M3	0.3
C1	M1	0.1
C2	M2	0.2
C3	M2	0.4
C4	M1	0.2
C5	M1	0.2
C6	M2	0.3
C7	M1	0.1

The optimal solution generated by solving the developed MIP using CPLEX 9.1 is listed below. The important data extracted from the optimal results are summarized in Table IV-2.

SOLVE SUMMARY

Integer optimal solution

Objective = 5.5500000000e+003

Solution time = 134.86 sec.

Iterations = 1206081

Nodes = 213748

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	47.000000	S321	25.000000
LI3	3.000000	S441	3.000000
EI1	8.000000	S471	27.000000
EI4	1.000000	S1152	6.000000
C1	47.000000	S1162	17.000000
C2	37.000000	S1192	12.000000
C3	47.000000	S1222	8.000000
C4	41.000000	S272	4.000000
S1141	2.000000	S332	33.000000
S1171	7.000000	S452	2.000000
S1181	13.000000	S462	25.000000
S1201	9.000000	S115	41.000000
S1211	5.000000	S125	32.000000
S261	4.000000	S134	24.000000
S281	23.500000	S143	34.000000

S155	24.000000	S214	34.000000
S163	21.000000	S223	7.000000
S173	18.000000	S233	26.500000
S184	30.000000	S243	5.500000
S194	9.000000	S253	25.000000
S1105	21.000000	S314	37.000000
S1113	15.000000	S415	36.000000
S1123	12.000000	S424	4.000000
S1134	14.000000	S435	29.000000

Table IV-2 The optimal result of example 4 × 5 × 5

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
513	351	298	47	134.86

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
3	9	750	450	5550

APPENDIX V

THE APS PROBLEM (5 ORDERS, 5 MACHINES AND 4 LEVELS)

The product structure in Figure V-1 is adopted. It should be noted that S3, C2, and C3 are common items. Five machines, with 8 hours available per day, are eligible to process the items (Table V-1). The ready times of these machines are Hour 1, Hour 2, Hour 3, Hour 2 and Hour 1, respectively. There are five orders: 10 Product F1s with due date Day 12, 5 Product F2s with due date Day 14, 10 Product S1s with due date Day 11, 30 Product C2s with due date Day 2, and 15 Product C3s with due date Day 10. The penalty rates are as follows: cost of idle time at \$50 per hour, cost of tardiness at \$250 per day per order, and cost of earliness at \$50 per day per order.

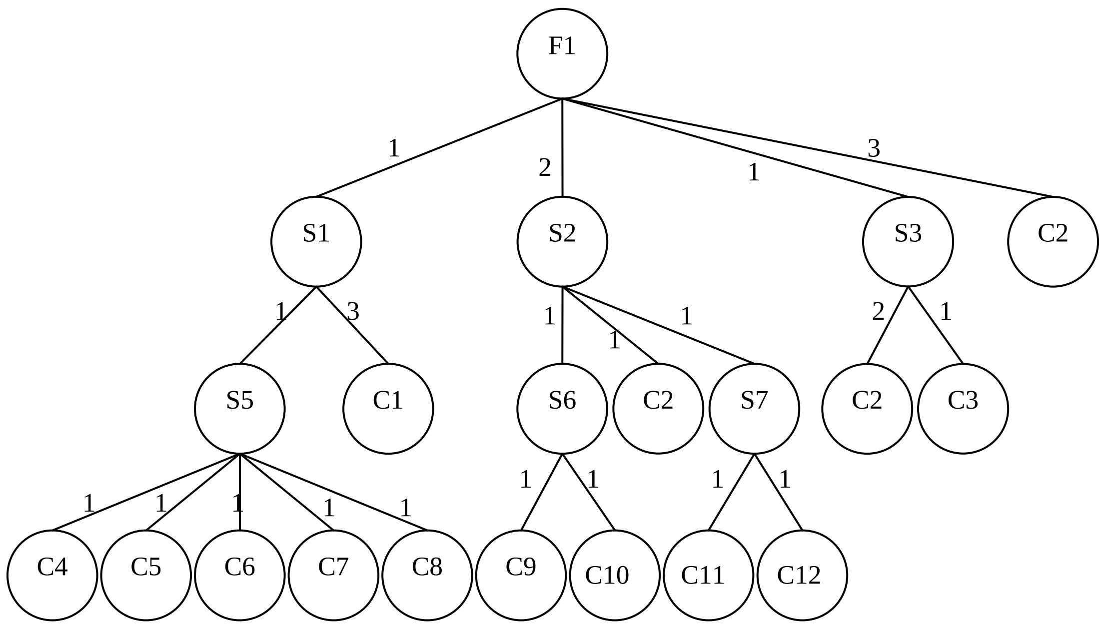


Figure V-1 (a) The product structure of F1 in example 5 × 5 × 4

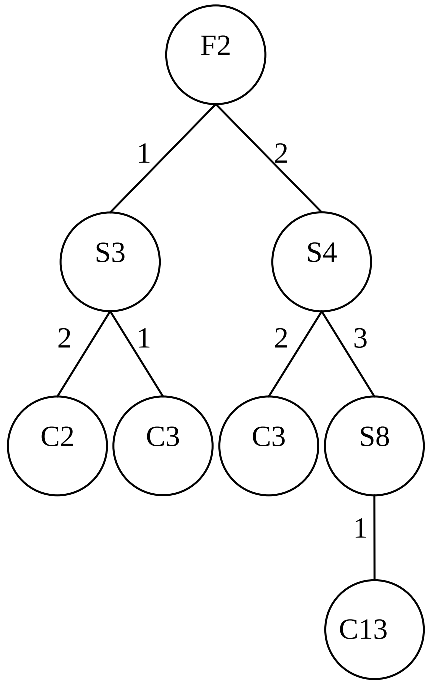


Figure V-1 (b) The product structure of F2 in example 5 × 5 × 4

Table V-1 Machine processing time for the items in example 5 × 5 × 4

Items	Machine No.	Processing Time (hours)	Items	Machine No.	Processing Time (hours)
F1	M5	0.7	C3	M5	0.3
F2	M5	0.6	C4	M1	0.1
S1	M4	0.7	C5	M1	0.2
S2	M5	0.5	C6	M1	0.3
S3	M4	0.6	C7	M2	0.3
S4	M4	0.5	C8	M2	0.4
S5	M5	0.4	C9	M1	0.4
S6	M4	0.4	C10	M2	0.2
S7	M4	0.3	C11	M1	0.1
S8	M3	0.3	C12	M1	0.2
C1	M3	0.2	C13	M2	0.3
C2	M3	0.3			

The optimal solution generated by solving the developed MIP using CPLEX 9.1 is listed below. The important data extracted from the optimal results are summarized in Table V-2.

SOLVE SUMMARY

Integer optimal solution

Objective = 6.5500000000e+003

Solution time = 42726.78 sec.

Iterations = 172306508

Nodes = 52142248

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	57.000000	S1201	5.000000
LI4	6.000000	S1211	9.000000
EI1	5.000000	S1221	21.000000
EI2	6.000000	S1232	21.000000
EI3	3.000000	S1242	2.000000
EI5	3.000000	S1251	12.000000
C1	49.000000	S1262	16.000000
C2	57.000000	S1271	7.000000
C3	57.000000	S1281	1.000000
C4	57.000000	S1155	2.500000
C5	53.500000	S2163	3.000000
S153	24.000000	S2175	1.000000
S1103	12.000000	S2185	5.500000
S1133	18.000000	S2382	24.000000
S1143	6.000000	S383	42.000000

S3141	35.000000	S164	36.000000
S3151	36.000000	S195	24.000000
S3161	32.000000	S1114	20.000000
S3172	35.000000	S1124	9.000000
S3182	6.000000	S225	54.000000
S4113	48.000000	S274	6.000000
S5125	49.000000	S284	45.000000
S115	42.000000	S2193	33.000000
S134	28.000000	S334	50.000000
S145	28.000000	S375	38.000000

Table V-2 The optimal result of example 5 × 5 × 4

Number of constraints	Total number of variables	No. of integer variables	Makespan (hours)	CPU time (sec.)
498	337	282	57	42726.78

Number of tardiness	Number of earliness	Tardiness penalty	Earliness penalty	Total cost
6	17	1500	850	6550

APPENDIX VI

THE PROBLEM FORMULATION OF THE DAPS EXAMPLE

(FROZEN INTERVAL = 1 DAY)

Minimize

300Cmax-7975
+250LI1+250LI2+250LI3+250LI4+250LI5+250LI6+250LI7
+50EI1+50EI2+50EI3+50EI4+50EI5+50EI6+50EI7

Subject to

(3.2):

cons1: C1-Cmax<=0
cons2: C2-Cmax<=0
cons3: C3-Cmax<=0
cons4: C4-Cmax<=0
cons5: C5-Cmax<=0
cons6: C6-Cmax<=0
cons7: C7-Cmax<=0

(3.3):

cons8: S351>=16
cons9: S632>=16
cons10: S752>=16
cons11: S762>=16
cons12: S3133>=16
cons13: S3143>=16
cons14: S3153>=16
cons15: S643>=16
cons16: S723>=16
cons17: S733>=16
cons18: S743>=16
cons19: S664>=19
cons20: S674>=19
cons21: S335>=16
cons22: S385>=16
cons23: S625>=16
cons24: S655>=16
cons25: S216>=17
cons26: S316>=17
cons27: S326>=17
cons28: S366>=17
cons29: S416>=17
cons30: S616>=17
cons31: S716>=17

(3.4):

cons32: S116-S123>=1.000000
cons33: S116-S131>=0.500000
cons34: S116-S141>=0.500000
cons35: S116-S152>=1.000000
cons36: S216-S221>=0.500000
cons37: S216-S236>=2.500000
cons38: S236-S245>=2.500000
cons39: S236-S252>=1.000000
cons40: S245-S263>=2.000000
cons41: S245-S2115>=1.500000
cons42: S2115-S2124>=0.500000
cons43: S2115-S2134>=0.500000
cons44: S316-S326>=6.000000
cons45: S316-S335>=10.000000
cons46: S316-S351>=3.000000
cons47: S326-S366>=6.000000
cons48: S326-S371>=3.000000
cons49: S335-S385>=6.000000
cons50: S335-S3105>=6.000000
cons51: S366-S3133>=2.000000

cons52: S366-S3143>=2.000000
cons53: S366-S3153>=1.000000
cons54: S366-S3162>=2.000000
cons55: S366-S3172>=1.000000
cons56: S385-S3182>=6.000000
cons57: S385-S3194>=4.000000
cons58: S3105-S3204>=2.000000
cons59: S3105-S3214>=2.000000
cons60: S3194-S3224>=4.000000
cons61: S416-S426>=6.000000
cons62: S416-S431>=3.000000
cons63: S426-S443>=2.000000
cons64: S426-S453>=2.000000
cons65: S426-S463>=1.000000
cons66: S426-S472>=2.000000
cons67: S426-S482>=1.000000
cons68: S616-S625>=5.000000
cons69: S616-S632>=2.000000
cons70: S625-S643>=4.000000
cons71: S625-S655>=3.000000
cons72: S655-S664>=1.000000
cons73: S655-S674>=1.000000
cons74: S716-S723>=1.000000
cons75: S716-S733>=1.000000
cons76: S716-S743>=0.500000
cons77: S716-S752>=1.000000
cons78: S716-S712>=0.500000

(3.5):
cons79: C1-S116=3.5
cons80: C2-S216=3
cons81: C3-S316=7
cons82: C4-S416=6
cons83: C5-S511=3
cons84: C6-S616=5
cons85: C7-S716=3

(3.6) and (3.7):
cons86: S131-S141+999X13141>=0.500000
cons87: S141-S131+999X14131>=0.500000
cons88: X13141+X14131=1
cons89: S131-S221+999X13221>=0.500000
cons90: S221-S131+999X22131>=0.500000
cons91: X13221+X22131=1
cons92: S131-S351+999X13351>=3.000000
cons93: S351-S131+999X35131>=0.500000
cons94: X13351+X35131=1
cons95: S131-S371+999X13371>=3.000000
cons96: S371-S131+999X37131>=0.500000
cons97: X13371+X37131=1
cons98: S131-S431+999X13431>=3.000000
cons99: S431-S131+999X43131>=0.500000
cons100: X13431+X43131=1
cons101: S131-S511+999X13511>=3.000000
cons102: S511-S131+999X51131>=0.500000
cons103: X13511+X51131=1
cons104: S141-S221+999X14221>=0.500000
cons105: S221-S141+999X22141>=0.500000
cons106: X14221+X22141=1
cons107: S141-S351+999X14351>=3.000000
cons108: S351-S141+999X35141>=0.500000
cons109: X14351+X35141=1

cons110: S141-S371+999X14371>=3.000000
cons111: S371-S141+999X37141>=0.500000
cons112: X14371+X37141=1
cons113: S141-S431+999X14431>=3.000000
cons114: S431-S141+999X43141>=0.500000
cons115: X14431+X43141=1
cons116: S141-S511+999X14511>=3.000000
cons117: S511-S141+999X51141>=0.500000
cons118: X14511+X51141=1
cons119: S221-S351+999X22351>=3.000000
cons120: S351-S221+999X35221>=0.500000
cons121: X22351+X35221=1
cons122: S221-S371+999X22371>=3.000000
cons123: S371-S221+999X37221>=0.500000
cons124: X22371+X37221=1
cons125: S221-S431+999X22431>=3.000000
cons126: S431-S221+999X43221>=0.500000
cons127: X22431+X43221=1
cons128: S221-S511+999X22511>=3.000000
cons129: S511-S221+999X51221>=0.500000
cons130: X22511+X51221=1
cons131: S351-S371+999X35371>=3.000000
cons132: S371-S351+999X37351>=3.000000
cons133: X35371+X37351=1
cons134: S351-S431+999X35431>=3.000000
cons135: S431-S351+999X43351>=3.000000
cons136: X35431+X43351=1
cons137: S351-S511+999X35511>=3.000000
cons138: S511-S351+999X51351>=3.000000
cons139: X35511+X51351=1
cons140: S371-S431+999X37431>=3.000000
cons141: S431-S371+999X43371>=3.000000
cons142: X37431+X43371=1
cons143: S371-S511+999X37511>=3.000000
cons144: S511-S371+999X51371>=3.000000
cons145: X37511+X51371=1
cons146: S431-S511+999X43511>=3.000000
cons147: S511-S431+999X51431>=3.000000
cons148: X43511+X51431=1
cons149: S152-S252+999X15252>=1.000000
cons150: S252-S152+999X25152>=1.000000
cons151: X15252+X25152=1
cons152: S152-S3162+999X153162>=2.000000
cons153: S3162-S152+999X316152>=1.000000
cons154: X153162+X316152=1
cons155: S152-S3172+999X153172>=1.000000

cons156: S3172-S152+999X317152>=1.000000
cons157: X153172+X317152=1
cons158: S152-S3182+999X153182>=6.000000
cons159: S3182-S152+999X318152>=1.000000
cons160: X153182+X318152=1
cons161: S152-S472+999X15472>=2.000000
cons162: S472-S152+999X47152>=1.000000
cons163: X15472+X47152=1
cons164: S152-S482+999X15482>=1.000000
cons165: S482-S152+999X48152>=1.000000
cons166: X15482+X48152=1
cons167: S152-S632+999X15632>=2.000000
cons168: S632-S152+999X63152>=1.000000
cons169: X15632+X63152=1
cons170: S152-S752+999X15752>=1.000000
cons171: S752-S152+999X75152>=1.000000
cons172: X15752+X75152=1
cons173: S152-S762+999X15762>=0.500000
cons174: S762-S152+999X76152>=1.000000
cons175: X15762+X76152=1
cons176: S252-S3162+999X253162>=2.000000
cons177: S3162-S252+999X316252>=1.000000
cons178: X253162+X316252=1
cons179: S252-S3172+999X253172>=1.000000
cons180: S3172-S252+999X317252>=1.000000
cons181: X253172+X317252=1
cons182: S252-S3182+999X253182>=6.000000
cons183: S3182-S252+999X318252>=1.000000
cons184: X253182+X318252=1
cons185: S252-S472+999X25472>=2.000000
cons186: S472-S252+999X47252>=1.000000
cons187: X25472+X47252=1
cons188: S252-S482+999X25482>=1.000000
cons189: S482-S252+999X48252>=1.000000
cons190: X25482+X48252=1
cons191: S252-S632+999X25632>=2.000000
cons192: S632-S252+999X63252>=1.000000
cons193: X25632+X63252=1
cons194: S252-S752+999X25752>=1.000000
cons195: S752-S252+999X75252>=1.000000
cons196: X25752+X75252=1
cons197: S252-S762+999X25762>=0.500000
cons198: S762-S252+999X76252>=1.000000
cons199: X25762+X76252=1
cons200: S3162-S3172+999X3163172>=1.000000
cons201: S3172-S3162+999X3173162>=2.000000

cons202: X3163172+X3173162=1
cons203: S3162-S3182+999X3163182>=6.000000
cons204: S3182-S3162+999X3183162>=2.000000
cons205: X3163182+X3183162=1
cons206: S3162-S472+999X316472>=2.000000
cons207: S472-S3162+999X473162>=2.000000
cons208: X316472+X473162=1
cons209: S3162-S482+999X316482>=1.000000
cons210: S482-S3162+999X483162>=2.000000
cons211: X316482+X483162=1
cons212: S3162-S632+999X316632>=2.000000
cons213: S632-S3162+999X633162>=2.000000
cons214: X316632+X633162=1
cons215: S3162-S752+999X316752>=1.000000
cons216: S752-S3162+999X753162>=2.000000
cons217: X316752+X753162=1
cons218: S3162-S762+999X316762>=0.500000
cons219: S762-S3162+999X763162>=2.000000
cons220: X316762+X763162=1
cons221: S3172-S3182+999X3173182>=6.000000
cons222: S3182-S3172+999X3183172>=1.000000
cons223: X3173182+X3183172=1
cons224: S3172-S472+999X317472>=2.000000
cons225: S472-S3172+999X473172>=1.000000
cons226: X317472+X473172=1
cons227: S3172-S482+999X317482>=1.000000
cons228: S482-S3172+999X483172>=1.000000
cons229: X317482+X483172=1
cons230: S3172-S632+999X317632>=2.000000
cons231: S632-S3172+999X633172>=1.000000
cons232: X317632+X633172=1
cons233: S3172-S752+999X317752>=1.000000
cons234: S752-S3172+999X753172>=1.000000
cons235: X317752+X753172=1
cons236: S3172-S762+999X317762>=0.500000
cons237: S762-S3172+999X763172>=1.000000
cons238: X317762+X763172=1
cons239: S3182-S472+999X318472>=2.000000
cons240: S472-S3182+999X473182>=6.000000
cons241: X318472+X473182=1
cons242: S3182-S482+999X318482>=1.000000
cons243: S482-S3182+999X483182>=6.000000
cons244: X318482+X483182=1
cons245: S3182-S632+999X318632>=2.000000
cons246: S632-S3182+999X633182>=6.000000
cons247: X318632+X633182=1

cons248: S3182-S752+999X318752>=1.000000
cons249: S752-S3182+999X753182>=6.000000
cons250: X318752+X753182=1
cons251: S3182-S762+999X318762>=0.500000
cons252: S762-S3182+999X763182>=6.000000
cons253: X318762+X763182=1
cons254: S472-S482+999X47482>=1.000000
cons255: S482-S472+999X48472>=2.000000
cons256: X47482+X48472=1
cons257: S472-S632+999X47632>=2.000000
cons258: S632-S472+999X63472>=2.000000
cons259: X47632+X63472=1
cons260: S472-S752+999X47752>=1.000000
cons261: S752-S472+999X75472>=2.000000
cons262: X47752+X75472=1
cons263: S472-S762+999X47762>=0.500000
cons264: S762-S472+999X76472>=2.000000
cons265: X47762+X76472=1
cons266: S482-S632+999X48632>=2.000000
cons267: S632-S482+999X63482>=1.000000
cons268: X48632+X63482=1
cons269: S482-S752+999X48752>=1.000000
cons270: S752-S482+999X75482>=1.000000
cons271: X48752+X75482=1
cons272: S482-S762+999X48762>=0.500000
cons273: S762-S482+999X76482>=1.000000
cons274: X48762+X76482=1
cons275: S632-S752+999X63752>=1.000000
cons276: S752-S632+999X75632>=2.000000
cons277: X63752+X75632=1
cons278: S632-S762+999X63762>=0.500000
cons279: S762-S632+999X76632>=2.000000
cons280: X63762+X76632=1
cons281: S752-S762+999X75762>=0.500000
cons282: S762-S752+999X76752>=1.000000
cons283: X75762+X76752=1
cons284: S123-S263+999X12263>=2.000000
cons285: S263-S123+999X26123>=1.000000
cons286: X12263+X26123=1
cons287: S123-S3133+999X123133>=2.000000
cons288: S3133-S123+999X313123>=1.000000
cons289: X123133+X313123=1
cons290: S123-S3143+999X123143>=2.000000
cons291: S3143-S123+999X314123>=1.000000
cons292: X123143+X314123=1
cons293: S123-S3153+999X123153>=1.000000

cons294: S3153-S123+999X315123>=1.000000
cons295: X123153+X315123=1
cons296: S123-S443+999X12443>=2.000000
cons297: S443-S123+999X44123>=1.000000
cons298: X12443+X44123=1
cons299: S123-S453+999X12453>=2.000000
cons300: S453-S123+999X45123>=1.000000
cons301: X12453+X45123=1
cons302: S123-S463+999X12463>=1.000000
cons303: S463-S123+999X46123>=1.000000
cons304: X12463+X46123=1
cons305: S123-S643+999X12643>=4.000000
cons306: S643-S123+999X64123>=1.000000
cons307: X12643+X64123=1
cons308: S123-S723+999X12723>=1.000000
cons309: S723-S123+999X72123>=1.000000
cons310: X12723+X72123=1
cons311: S123-S733+999X12733>=1.000000
cons312: S733-S123+999X73123>=1.000000
cons313: X12733+X73123=1
cons314: S123-S743+999X12743>=0.500000
cons315: S743-S123+999X74123>=1.000000
cons316: X12743+X74123=1
cons317: S263-S3133+999X263133>=2.000000
cons318: S3133-S263+999X313263>=2.000000
cons319: X263133+X313263=1
cons320: S263-S3143+999X263143>=2.000000
cons321: S3143-S263+999X314263>=2.000000
cons322: X263143+X314263=1
cons323: S263-S3153+999X263153>=1.000000
cons324: S3153-S263+999X315263>=2.000000
cons325: X263153+X315263=1
cons326: S263-S443+999X26443>=2.000000
cons327: S443-S263+999X44263>=2.000000
cons328: X26443+X44263=1
cons329: S263-S453+999X26453>=2.000000
cons330: S453-S263+999X45263>=2.000000
cons331: X26453+X45263=1
cons332: S263-S463+999X26463>=1.000000
cons333: S463-S263+999X46263>=2.000000
cons334: X26463+X46263=1
cons335: S263-S643+999X26643>=4.000000
cons336: S643-S263+999X64263>=2.000000
cons337: X26643+X64263=1
cons338: S263-S723+999X26723>=1.000000
cons339: S723-S263+999X72263>=2.000000

cons340: X26723+X72263=1
cons341: S263-S733+999X26733>=1.000000
cons342: S733-S263+999X73263>=2.000000
cons343: X26733+X73263=1
cons344: S263-S743+999X26743>=0.500000
cons345: S743-S263+999X74263>=2.000000
cons346: X26743+X74263=1
cons347: S3133-S3143+999X3133143>=2.000000
cons348: S3143-S3133+999X3143133>=2.000000
cons349: X3133143+X3143133=1
cons350: S3133-S3153+999X3133153>=1.000000
cons351: S3153-S3133+999X3153133>=2.000000
cons352: X3133153+X3153133=1
cons353: S3133-S443+999X313443>=2.000000
cons354: S443-S3133+999X443133>=2.000000
cons355: X313443+X443133=1
cons356: S3133-S453+999X313453>=2.000000
cons357: S453-S3133+999X453133>=2.000000
cons358: X313453+X453133=1
cons359: S3133-S463+999X313463>=1.000000
cons360: S463-S3133+999X463133>=2.000000
cons361: X313463+X463133=1
cons362: S3133-S643+999X313643>=4.000000
cons363: S643-S3133+999X643133>=2.000000
cons364: X313643+X643133=1
cons365: S3133-S723+999X313723>=1.000000
cons366: S723-S3133+999X723133>=2.000000
cons367: X313723+X723133=1
cons368: S3133-S733+999X313733>=1.000000
cons369: S733-S3133+999X733133>=2.000000
cons370: X313733+X733133=1
cons371: S3133-S743+999X313743>=0.500000
cons372: S743-S3133+999X743133>=2.000000
cons373: X313743+X743133=1
cons374: S3143-S3153+999X3143153>=1.000000
cons375: S3153-S3143+999X3153143>=2.000000
cons376: X3143153+X3153143=1
cons377: S3143-S443+999X314443>=2.000000
cons378: S443-S3143+999X443143>=2.000000
cons379: X314443+X443143=1
cons380: S3143-S453+999X314453>=2.000000
cons381: S453-S3143+999X453143>=2.000000
cons382: X314453+X453143=1
cons383: S3143-S463+999X314463>=1.000000
cons384: S463-S3143+999X463143>=2.000000
cons385: X314463+X463143=1

cons386: S3143-S643+999X314643>=4.000000
cons387: S643-S3143+999X643143>=2.000000
cons388: X314643+X643143=1
cons389: S3143-S723+999X314723>=1.000000
cons390: S723-S3143+999X723143>=2.000000
cons391: X314723+X723143=1
cons392: S3143-S733+999X314733>=1.000000
cons393: S733-S3143+999X733143>=2.000000
cons394: X314733+X733143=1
cons395: S3143-S743+999X314743>=0.500000
cons396: S743-S3143+999X743143>=2.000000
cons397: X314743+X743143=1
cons398: S3153-S443+999X315443>=2.000000
cons399: S443-S3153+999X443153>=1.000000
cons400: X315443+X443153=1
cons401: S3153-S453+999X315453>=2.000000
cons402: S453-S3153+999X453153>=1.000000
cons403: X315453+X453153=1
cons404: S3153-S463+999X315463>=1.000000
cons405: S463-S3153+999X463153>=1.000000
cons406: X315463+X463153=1
cons407: S3153-S643+999X315643>=4.000000
cons408: S643-S3153+999X643153>=1.000000
cons409: X315643+X643153=1
cons410: S3153-S723+999X315723>=1.000000
cons411: S723-S3153+999X723153>=1.000000
cons412: X315723+X723153=1
cons413: S3153-S733+999X315733>=1.000000
cons414: S733-S3153+999X733153>=1.000000
cons415: X315733+X733153=1
cons416: S3153-S743+999X315743>=0.500000
cons417: S743-S3153+999X743153>=1.000000
cons418: X315743+X743153=1
cons419: S443-S453+999X44453>=2.000000
cons420: S453-S443+999X45443>=2.000000
cons421: X44453+X45443=1
cons422: S443-S463+999X44463>=1.000000
cons423: S463-S443+999X46443>=2.000000
cons424: X44463+X46443=1
cons425: S443-S643+999X44643>=4.000000
cons426: S643-S443+999X64443>=2.000000
cons427: X44643+X64443=1
cons428: S443-S723+999X44723>=1.000000
cons429: S723-S443+999X72443>=2.000000
cons430: X44723+X72443=1
cons431: S443-S733+999X44733>=1.000000

cons432: S733-S443+999X73443>=2.000000
cons433: X44733+X73443=1
cons434: S443-S743+999X44743>=0.500000
cons435: S743-S443+999X74443>=2.000000
cons436: X44743+X74443=1
cons437: S453-S463+999X45463>=1.000000
cons438: S463-S453+999X46453>=2.000000
cons439: X45463+X46453=1
cons440: S453-S643+999X45643>=4.000000
cons441: S643-S453+999X64453>=2.000000
cons442: X45643+X64453=1
cons443: S453-S723+999X45723>=1.000000
cons444: S723-S453+999X72453>=2.000000
cons445: X45723+X72453=1
cons446: S453-S733+999X45733>=1.000000
cons447: S733-S453+999X73453>=2.000000
cons448: X45733+X73453=1
cons449: S453-S743+999X45743>=0.500000
cons450: S743-S453+999X74453>=2.000000
cons451: X45743+X74453=1
cons452: S463-S643+999X46643>=4.000000
cons453: S643-S463+999X64463>=1.000000
cons454: X46643+X64463=1
cons455: S463-S723+999X46723>=1.000000
cons456: S723-S463+999X72463>=1.000000
cons457: X46723+X72463=1
cons458: S463-S733+999X46733>=1.000000
cons459: S733-S463+999X73463>=1.000000
cons460: X46733+X73463=1
cons461: S463-S743+999X46743>=0.500000
cons462: S743-S463+999X74463>=1.000000
cons463: X46743+X74463=1
cons464: S643-S723+999X64723>=1.000000
cons465: S723-S643+999X72643>=4.000000
cons466: X64723+X72643=1
cons467: S643-S733+999X64733>=1.000000
cons468: S733-S643+999X73643>=4.000000
cons469: X64733+X73643=1
cons470: S643-S743+999X64743>=0.500000
cons471: S743-S643+999X74643>=4.000000
cons472: X64743+X74643=1
cons473: S723-S733+999X72733>=1.000000
cons474: S733-S723+999X73723>=1.000000
cons475: X72733+X73723=1
cons476: S723-S743+999X72743>=0.500000
cons477: S743-S723+999X74723>=1.000000

cons478: X72743+X74723=1
cons479: S733-S743+999X73743>=0.500000
cons480: S743-S733+999X74733>=1.000000
cons481: X73743+X74733=1
cons482: S2124-S2134+999X2122134>=0.500000
cons483: S2134-S2124+999X2132124>=0.500000
cons484: X2122134+X2132124=1
cons485: S2124-S3194+999X2123194>=4.000000
cons486: S3194-S2124+999X3192124>=0.500000
cons487: X2123194+X3192124=1
cons488: S2124-S3204+999X2123204>=2.000000
cons489: S3204-S2124+999X3202124>=0.500000
cons490: X2123204+X3202124=1
cons491: S2124-S3214+999X2123214>=2.000000
cons492: S3214-S2124+999X3212124>=0.500000
cons493: X2123214+X3212124=1
cons494: S2124-S3224+999X2123224>=4.000000
cons495: S3224-S2124+999X3222124>=0.500000
cons496: X2123224+X3222124=1
cons497: S2124-S664+999X212664>=1.000000
cons498: S664-S2124+999X662124>=0.500000
cons499: X212664+X662124=1
cons500: S2124-S674+999X212674>=1.000000
cons501: S674-S2124+999X672124>=0.500000
cons502: X212674+X672124=1
cons503: S2134-S3194+999X2133194>=4.000000
cons504: S3194-S2134+999X3192134>=0.500000
cons505: X2133194+X3192134=1
cons506: S2134-S3204+999X2133204>=2.000000
cons507: S3204-S2134+999X3202134>=0.500000
cons508: X2133204+X3202134=1
cons509: S2134-S3214+999X2133214>=2.000000
cons510: S3214-S2134+999X3212134>=0.500000
cons511: X2133214+X3212134=1
cons512: S2134-S3224+999X2133224>=4.000000
cons513: S3224-S2134+999X3222134>=0.500000
cons514: X2133224+X3222134=1
cons515: S2134-S664+999X213664>=1.000000
cons516: S664-S2134+999X662134>=0.500000
cons517: X213664+X662134=1
cons518: S2134-S674+999X213674>=1.000000
cons519: S674-S2134+999X672134>=0.500000
cons520: X213674+X672134=1
cons521: S3194-S3204+999X3193204>=2.000000
cons522: S3204-S3194+999X3203194>=4.000000
cons523: X3193204+X3203194=1

cons524: S3194-S3214+999X3193214>=2.000000
cons525: S3214-S3194+999X3213194>=4.000000
cons526: X3193214+X3213194=1
cons527: S3194-S3224+999X3193224>=4.000000
cons528: S3224-S3194+999X3223194>=4.000000
cons529: X3193224+X3223194=1
cons530: S3194-S664+999X319664>=1.000000
cons531: S664-S3194+999X663194>=4.000000
cons532: X319664+X663194=1
cons533: S3194-S674+999X319674>=1.000000
cons534: S674-S3194+999X673194>=4.000000
cons535: X319674+X673194=1
cons536: S3204-S3214+999X3203214>=2.000000
cons537: S3214-S3204+999X3213204>=2.000000
cons538: X3203214+X3213204=1
cons539: S3204-S3224+999X3203224>=4.000000
cons540: S3224-S3204+999X3223204>=2.000000
cons541: X3203224+X3223204=1
cons542: S3204-S664+999X320664>=1.000000
cons543: S664-S3204+999X663204>=2.000000
cons544: X320664+X663204=1
cons545: S3204-S674+999X320674>=1.000000
cons546: S674-S3204+999X673204>=2.000000
cons547: X320674+X673204=1
cons548: S3214-S3224+999X3213224>=4.000000
cons549: S3224-S3214+999X3223214>=2.000000
cons550: X3213224+X3223214=1
cons551: S3214-S664+999X321664>=1.000000
cons552: S664-S3214+999X663214>=2.000000
cons553: X321664+X663214=1
cons554: S3214-S674+999X321674>=1.000000
cons555: S674-S3214+999X673214>=2.000000
cons556: X321674+X673214=1
cons557: S3224-S664+999X322664>=1.000000
cons558: S664-S3224+999X663224>=4.000000
cons559: X322664+X663224=1
cons560: S3224-S674+999X322674>=1.000000
cons561: S674-S3224+999X673224>=4.000000
cons562: X322674+X673224=1
cons563: S664-S674+999X66674>=1.000000
cons564: S674-S664+999X67664>=1.000000
cons565: X66674+X67664=1
cons566: S245-S2115+999X242115>=1.500000
cons567: S2115-S245+999X211245>=2.500000
cons568: X242115+X211245=1
cons569: S245-S335+999X24335>=10.000000

cons570: S335-S245+999X33245>=2.500000
cons571: X24335+X33245=1
cons572: S245-S385+999X24385>=6.000000
cons573: S385-S245+999X38245>=2.500000
cons574: X24385+X38245=1
cons575: S245-S3105+999X243105>=6.000000
cons576: S3105-S245+999X310245>=2.500000
cons577: X243105+X310245=1
cons578: S245-S625+999X24625>=5.000000
cons579: S625-S245+999X62245>=2.500000
cons580: X24625+X62245=1
cons581: S245-S655+999X24655>=3.000000
cons582: S655-S245+999X65245>=2.500000
cons583: X24655+X65245=1
cons584: S2115-S335+999X211335>=10.000000
cons585: S335-S2115+999X332115>=1.500000
cons586: X211335+X332115=1
cons587: S2115-S385+999X211385>=6.000000
cons588: S385-S2115+999X382115>=1.500000
cons589: X211385+X382115=1
cons590: S2115-S3105+999X2113105>=6.000000
cons591: S3105-S2115+999X3102115>=1.500000
cons592: X2113105+X3102115=1
cons593: S2115-S625+999X211625>=5.000000
cons594: S625-S2115+999X622115>=1.500000
cons595: X211625+X622115=1
cons596: S2115-S655+999X211655>=3.000000
cons597: S655-S2115+999X652115>=1.500000
cons598: X211655+X652115=1
cons599: S335-S385+999X33385>=6.000000
cons600: S385-S335+999X38335>=10.000000
cons601: X33385+X38335=1
cons602: S335-S3105+999X333105>=6.000000
cons603: S3105-S335+999X310335>=10.000000
cons604: X333105+X310335=1
cons605: S335-S625+999X33625>=5.000000
cons606: S625-S335+999X62335>=10.000000
cons607: X33625+X62335=1
cons608: S335-S655+999X33655>=3.000000
cons609: S655-S335+999X65335>=10.000000
cons610: X33655+X65335=1
cons611: S385-S3105+999X383105>=6.000000
cons612: S3105-S385+999X310385>=6.000000
cons613: X383105+X310385=1
cons614: S385-S625+999X38625>=5.000000
cons615: S625-S385+999X62385>=6.000000

cons616: X38625+X62385=1
cons617: S385-S655+999X38655>=3.000000
cons618: S655-S385+999X65385>=6.000000
cons619: X38655+X65385=1
cons620: S3105-S625+999X310625>=5.000000
cons621: S625-S3105+999X623105>=6.000000
cons622: X310625+X623105=1
cons623: S3105-S655+999X310655>=3.000000
cons624: S655-S3105+999X653105>=6.000000
cons625: X310655+X653105=1
cons626: S625-S655+999X62655>=3.000000
cons627: S655-S625+999X65625>=5.000000
cons628: X62655+X65625=1
cons629: S116-S216+999X11216>=3.000000
cons630: S216-S116+999X21116>=3.500000
cons631: X11216+X21116=1
cons632: S116-S236+999X11236>=2.500000
cons633: S236-S116+999X23116>=3.500000
cons634: X11236+X23116=1
cons635: S116-S316+999X11316>=7.000000
cons636: S316-S116+999X31116>=3.500000
cons637: X11316+X31116=1
cons638: S116-S326+999X11326>=6.000000
cons639: S326-S116+999X32116>=3.500000
cons640: X11326+X32116=1
cons641: S116-S366+999X11366>=6.000000
cons642: S366-S116+999X36116>=3.500000
cons643: X11366+X36116=1
cons644: S116-S416+999X11416>=6.000000
cons645: S416-S116+999X41116>=3.500000
cons646: X11416+X41116=1
cons647: S116-S426+999X11426>=6.000000
cons648: S426-S116+999X42116>=3.500000
cons649: X11426+X42116=1
cons650: S116-S616+999X11616>=5.000000
cons651: S616-S116+999X61116>=3.500000
cons652: X11616+X61116=1
cons653: S116-S716+999X11716>=3.000000
cons654: S716-S116+999X71116>=3.500000
cons655: X11716+X71116=1
cons656: S216-S236+999X21236>=2.500000
cons657: S236-S216+999X23216>=3.000000
cons658: X21236+X23216=1
cons659: S216-S316+999X21316>=7.000000
cons660: S316-S216+999X31216>=3.000000
cons661: X21316+X31216=1

cons662: S216-S326+999X21326>=6.000000
cons663: S326-S216+999X32216>=3.000000
cons664: X21326+X32216=1
cons665: S216-S366+999X21366>=6.000000
cons666: S366-S216+999X36216>=3.000000
cons667: X21366+X36216=1
cons668: S216-S416+999X21416>=6.000000
cons669: S416-S216+999X41216>=3.000000
cons670: X21416+X41216=1
cons671: S216-S426+999X21426>=6.000000
cons672: S426-S216+999X42216>=3.000000
cons673: X21426+X42216=1
cons674: S216-S616+999X21616>=5.000000
cons675: S616-S216+999X61216>=3.000000
cons676: X21616+X61216=1
cons677: S216-S716+999X21716>=3.000000
cons678: S716-S216+999X71216>=3.000000
cons679: X21716+X71216=1
cons680: S236-S316+999X23316>=7.000000
cons681: S316-S236+999X31236>=2.500000
cons682: X23316+X31236=1
cons683: S236-S326+999X23326>=6.000000
cons684: S326-S236+999X32236>=2.500000
cons685: X23326+X32236=1
cons686: S236-S366+999X23366>=6.000000
cons687: S366-S236+999X36236>=2.500000
cons688: X23366+X36236=1
cons689: S236-S416+999X23416>=6.000000
cons690: S416-S236+999X41236>=2.500000
cons691: X23416+X41236=1
cons692: S236-S426+999X23426>=6.000000
cons693: S426-S236+999X42236>=2.500000
cons694: X23426+X42236=1
cons695: S236-S616+999X23616>=5.000000
cons696: S616-S236+999X61236>=2.500000
cons697: X23616+X61236=1
cons698: S236-S716+999X23716>=3.000000
cons699: S716-S236+999X71236>=2.500000
cons700: X23716+X71236=1
cons701: S316-S326+999X31326>=6.000000
cons702: S326-S316+999X32316>=7.000000
cons703: X31326+X32316=1
cons704: S316-S366+999X31366>=6.000000
cons705: S366-S316+999X36316>=7.000000
cons706: X31366+X36316=1
cons707: S316-S416+999X31416>=6.000000

cons708: S416-S316+999X41316>=7.000000
cons709: X31416+X41316=1
cons710: S316-S426+999X31426>=6.000000
cons711: S426-S316+999X42316>=7.000000
cons712: X31426+X42316=1
cons713: S316-S616+999X31616>=5.000000
cons714: S616-S316+999X61316>=7.000000
cons715: X31616+X61316=1
cons716: S316-S716+999X31716>=3.000000
cons717: S716-S316+999X71316>=7.000000
cons718: X31716+X71316=1
cons719: S326-S366+999X32366>=6.000000
cons720: S366-S326+999X36326>=6.000000
cons721: X32366+X36326=1
cons722: S326-S416+999X32416>=6.000000
cons723: S416-S326+999X41326>=6.000000
cons724: X32416+X41326=1
cons725: S326-S426+999X32426>=6.000000
cons726: S426-S326+999X42326>=6.000000
cons727: X32426+X42326=1
cons728: S326-S616+999X32616>=5.000000
cons729: S616-S326+999X61326>=6.000000
cons730: X32616+X61326=1
cons731: S326-S716+999X32716>=3.000000
cons732: S716-S326+999X71326>=6.000000
cons733: X32716+X71326=1
cons734: S366-S416+999X36416>=6.000000
cons735: S416-S366+999X41366>=6.000000
cons736: X36416+X41366=1
cons737: S366-S426+999X36426>=6.000000
cons738: S426-S366+999X42366>=6.000000
cons739: X36426+X42366=1
cons740: S366-S616+999X36616>=5.000000
cons741: S616-S366+999X61366>=6.000000
cons742: X36616+X61366=1
cons743: S366-S716+999X36716>=3.000000
cons744: S716-S366+999X71366>=6.000000
cons745: X36716+X71366=1
cons746: S416-S426+999X41426>=6.000000
cons747: S426-S416+999X42416>=6.000000
cons748: X41426+X42416=1
cons749: S416-S616+999X41616>=5.000000
cons750: S616-S416+999X61416>=6.000000
cons751: X41616+X61416=1
cons752: S416-S716+999X41716>=3.000000
cons753: S716-S416+999X71416>=6.000000

cons754: X41716+X71416=1
cons755: S426-S616+999X42616>=5.000000
cons756: S616-S426+999X61426>=6.000000
cons757: X42616+X61426=1
cons758: S426-S716+999X42716>=3.000000
cons759: S716-S426+999X71426>=6.000000
cons760: X42716+X71426=1
cons761: S616-S716+999X61716>=3.000000
cons762: S716-S616+999X71616>=5.000000
cons763: X61716+X71616=1

(3.8) and (3.9):
cons764: 0.125C1-L1<=6
cons765: E1+0.125C1>=6
cons766: 0.125C2-L2<=7
cons767: E2+0.125C2>=7
cons768: 0.125C3-L3<=14
cons769: E3+0.125C3>=14
cons770: 0.125C4-L4<=3
cons771: E4+0.125C4>=3
cons772: 0.125C5-L5<=1
cons773: E5+0.125C5>=1
cons774: 0.125C6-L6<=5
cons775: E6+0.125C6>=5
cons776: 0.125C7-L7<=3
cons777: E7+0.125C7>=3

(3.10) and (3.11):
cons778: L1-LI1<=0
cons779: E1-EI1<=0.99
cons780: L2-LI2<=0
cons781: E2-EI2<=0.99
cons782: L3-LI3<=0
cons783: E3-EI3<=0.99
cons784: L4-LI4<=0
cons785: E4-EI4<=0.99
cons786: L5-LI5<=0
cons787: E5-EI5<=0.99
cons788: L6-LI6<=0
cons789: E6-EI6<=0.99
cons790: L7-LI7<=0
cons791: E7-EI7<=0.99

(5.1):
cons792: S131=1.0
cons793: S141=1.5
cons794: S221=8.0
cons795: S371=8.5
cons796: S431=2.0
cons797: S511=5.0
cons798: S152=3.0
cons799: S252=2.0
cons800: S3162=14.0
cons801: S3172=4.0
cons802: S3182=8.0
cons803: S472=6.0
cons804: S482=5.0
cons805: S123=3.0
cons806: S263=4.0
cons807: S443=8.0
cons808: S453=6.0
cons809: S463=10.0
cons810: S2124=3.0
cons811: S2134=3.5
cons812: S3194=15.0
cons813: S3204=6.5
cons814: S3214=4.0
cons815: S3224=11.0
cons816: S245=6.0
cons817: S2115=4.0
cons818: S3105=8.5
cons819: S116=4.58
cons820: S236=8.5
cons821: S426=11.0

Bounds

LI1 free
LI2 free
LI3 free
LI4 free
LI5 free
LI6 free
LI7 free
EI1 free
EI2 free
EI3 free
EI4 free
EI5 free
EI6 free
EI7 free

Integers

EI1 LI1
EI2 LI2
EI3 LI3
EI4 LI4
EI5 LI5
EI6 LI6
EI7 LI7

X13141 X14131
X13221 X22131
X13351 X35131
X13371 X37131
X13431 X43131
X13511 X51131
X14221 X22141
X14351 X35141
X14371 X37141
X14431 X43141
X14511 X51141
X22351 X35221
X22371 X37221
X22431 X43221
X22511 X51221
X35371 X37351
X35431 X43351
X35511 X51351
X37431 X43371
X37511 X51371
X43511 X51431
X15252 X25152
X153162 X316152
X153172 X317152
X153182 X318152
X15472 X47152
X15482 X48152
X15632 X63152
X15752 X75152
X15762 X76152
X253162 X316252
X253172 X317252
X253182 X318252
X25472 X47252
X25482 X48252
X25632 X63252
X25752 X75252
X25762 X76252
X3163172 X3173162
X3163182 X3183162
X316472 X473162
X316482 X483162
X316632 X633162
X316752 X753162
X316762 X763162
X3173182 X3183172
X317472 X473172
X317482 X483172
X317632 X633172
X317752 X753172
X317762 X763172
X318472 X473182
X318482 X483182
X318632 X633182
X318752 X753182
X318762 X763182
X47482 X48472
X47632 X63472
X47752 X75472
X47762 X76472
X48632 X63482
X48752 X75482
X48762 X76482
X63752 X75632
X63762 X76632
X75762 X76752
X12263 X26123
X123133 X313123
X123143 X314123
X123153 X315123
X12443 X44123
X12453 X45123
X12463 X46123
X12643 X64123
X12723 X72123
X12733 X73123
X12743 X74123
X263133 X313263
X263143 X314263
X263153 X315263
X26443 X44263
X26453 X45263
X26463 X46263
X26643 X64263
X26723 X72263
X26733 X73263
X26743 X74263
X3133143 X3143133
X3133153 X3153133
X313443 X443133
X313453 X453133
X313463 X463133
X313643 X643133
X313723 X723133
X313733 X733133
X313743 X743133
X3143153 X3153143
X314443 X443143
X314453 X453143

X314463 X463143
X314643 X643143
X314723 X723143
X314733 X733143
X314743 X743143
X315443 X443153
X315453 X453153
X315463 X463153
X315643 X643153
X315723 X723153
X315733 X733153
X315743 X743153
X44453 X45443
X44463 X46443
X44643 X64443
X44723 X72443
X44733 X73443
X44743 X74443
X45463 X46453
X45643 X64453
X45723 X72453
X45733 X73453
X45743 X74453
X46643 X64463
X46723 X72463
X46733 X73463
X46743 X74463
X64723 X72643
X64733 X73643
X64743 X74643
X72733 X73723
X72743 X74723
X73743 X74733
X2122134 X2132124
X2123194 X3192124
X2123204 X3202124
X2123214 X3212124
X2123224 X3222124
X212664 X662124
X212674 X672124
X2133194 X3192134
X2133204 X3202134
X2133214 X3212134
X2133224 X3222134
X213664 X662134
X213674 X672134
X3193204 X3203194
X3193214 X3213194
X3193224 X3223194
X319664 X663194
X319674 X673194
X3203214 X3213204
X3203224 X3223204
X320664 X663204
X320674 X673204
X3213224 X3223214
X321664 X663214
X321674 X673214
X322664 X663224
X322674 X673224
X66674 X67664
X242115 X211245
X24335 X33245
X24385 X38245
X243105 X310245
X24625 X62245
X24655 X65245
X211335 X332115
X211385 X382115
X2113105 X3102115
X211625 X622115
X211655 X652115
X33385 X38335
X333105 X310335
X33625 X62335
X33655 X65335
X383105 X310385
X38625 X62385
X38655 X65385
X310625 X623105
X310655 X653105
X62655 X65625
X11216 X21116
X11236 X23116
X11316 X31116
X11326 X32116
X11366 X36116
X11416 X41116
X11426 X42116
X11616 X61116
X11716 X71116
X21236 X23216
X21316 X31216
X21326 X32216
X21366 X36216
X21416 X41216
X21426 X42216
X21616 X61216
X21716 X71216
X23316 X31236
X23326 X32236
X23366 X36236
X23416 X41236
X23426 X42236
X23616 X61236
X23716 X71236
X31326 X32316
X31366 X36316
X31416 X41316
X31426 X42316
X31616 X61316
X31716 X71316
X32366 X36326
X32416 X41326
X32426 X42326
X32616 X61326
X32716 X71326
X36416 X41366
X36426 X42366
X36616 X61366
X36716 X71366
X41426 X42416
X41616 X61416
X41716 X71416
X42616 X61426
X42716 X71426
X61716 X71616

End

APPENDIX VII

THE OPTIMAL RESULTS OF THE DAPS EXAMPLE

(FROZEN INTERVAL = 1 DAY)

Integer optimal
Objective = 8.7750000000e+003
Solution time = 0.45 sec.
Iterations = 5069
Nodes = 1463

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	53.000000	S316	46.000000
LI7	1.000000	S326	37.000000
EI1	4.000000	S366	23.000000
EI2	1.000000	S416	17.000000
EI3	7.000000	S616	32.000000
C1	8.080000	S716	29.000000
C2	46.000000	S116	4.580000
C3	53.000000	S123	3.000000
C4	23.000000	S131	1.000000
C5	8.000000	S141	1.500000
C6	37.000000	S152	3.000000
C7	32.000000	S221	8.000000
S351	16.000000	S236	8.500000
S632	16.500000	S245	6.000000
S752	28.000000	S252	2.000000
S762	16.000000	S263	4.000000
S3133	19.000000	S2115	4.000000
S3143	17.000000	S2124	3.000000
S3153	16.000000	S2134	3.500000
S643	21.000000	S371	8.500000
S723	27.000000	S3105	8.500000
S733	25.000000	S3162	14.000000
S743	26.000000	S3172	4.000000
S664	19.000000	S3182	8.000000
S674	20.000000	S3194	15.000000
S335	36.000000	S3204	6.500000
S385	30.000000	S3214	4.000000
S625	25.000000	S3224	11.000000
S655	21.000000	S426	11.000000
S216	43.000000	S431	2.000000

S443	8.000000	X473162	1.000000
S453	6.000000	X483162	1.000000
S463	10.000000	X316632	1.000000
S472	6.000000	X316752	1.000000
S482	5.000000	X316762	1.000000
S511	5.000000	X3173182	1.000000
X13141	1.000000	X317472	1.000000
X13221	1.000000	X317482	1.000000
X13351	1.000000	X317632	1.000000
X13371	1.000000	X317752	1.000000
X13431	1.000000	X317762	1.000000
X13511	1.000000	X473182	1.000000
X14221	1.000000	X483182	1.000000
X14351	1.000000	X318632	1.000000
X14371	1.000000	X318752	1.000000
X14431	1.000000	X318762	1.000000
X14511	1.000000	X48472	1.000000
X22351	1.000000	X47632	1.000000
X22371	1.000000	X47752	1.000000
X43221	1.000000	X47762	1.000000
X51221	1.000000	X48632	1.000000
X37351	1.000000	X48752	1.000000
X43351	1.000000	X48762	1.000000
X51351	1.000000	X63752	1.000000
X43371	1.000000	X76632	1.000000
X51371	1.000000	X76752	1.000000
X43511	1.000000	X12263	1.000000
X25152	1.000000	X123133	1.000000
X153162	1.000000	X123143	1.000000
X153172	1.000000	X123153	1.000000
X153182	1.000000	X12443	1.000000
X15472	1.000000	X12453	1.000000
X15482	1.000000	X12463	1.000000
X15632	1.000000	X12643	1.000000
X15752	1.000000	X12723	1.000000
X15762	1.000000	X12733	1.000000
X253162	1.000000	X12743	1.000000
X253172	1.000000	X263133	1.000000
X253182	1.000000	X263143	1.000000
X25472	1.000000	X263153	1.000000
X25482	1.000000	X26443	1.000000
X25632	1.000000	X26453	1.000000
X25752	1.000000	X26463	1.000000
X25762	1.000000	X26643	1.000000
X3173162	1.000000	X26723	1.000000
X3183162	1.000000	X26733	1.000000

X26743	1.000000	X2122134	1.000000
X3143133	1.000000	X2123194	1.000000
X3153133	1.000000	X2123204	1.000000
X443133	1.000000	X2123214	1.000000
X453133	1.000000	X2123224	1.000000
X463133	1.000000	X212664	1.000000
X313643	1.000000	X212674	1.000000
X313723	1.000000	X2133194	1.000000
X313733	1.000000	X2133204	1.000000
X313743	1.000000	X2133214	1.000000
X3153143	1.000000	X2133224	1.000000
X443143	1.000000	X213664	1.000000
X453143	1.000000	X213674	1.000000
X463143	1.000000	X3203194	1.000000
X314643	1.000000	X3213194	1.000000
X314723	1.000000	X3223194	1.000000
X314733	1.000000	X319664	1.000000
X314743	1.000000	X319674	1.000000
X443153	1.000000	X3213204	1.000000
X453153	1.000000	X3203224	1.000000
X463153	1.000000	X320664	1.000000
X315643	1.000000	X320674	1.000000
X315723	1.000000	X3213224	1.000000
X315733	1.000000	X321664	1.000000
X315743	1.000000	X321674	1.000000
X45443	1.000000	X322664	1.000000
X44463	1.000000	X322674	1.000000
X44643	1.000000	X66674	1.000000
X44723	1.000000	X67664	0.000000
X44733	1.000000	X211245	1.000000
X44743	1.000000	X24335	1.000000
X45463	1.000000	X24385	1.000000
X45643	1.000000	X243105	1.000000
X45723	1.000000	X24625	1.000000
X45733	1.000000	X24655	1.000000
X45743	1.000000	X211335	1.000000
X46643	1.000000	X211385	1.000000
X46723	1.000000	X2113105	1.000000
X46733	1.000000	X211625	1.000000
X46743	1.000000	X211655	1.000000
X64723	1.000000	X38335	1.000000
X64733	1.000000	X310335	1.000000
X64743	1.000000	X62335	1.000000
X73723	1.000000	X65335	1.000000
X74723	1.000000	X310385	1.000000
X73743	1.000000	X62385	1.000000

X65385	1.000000	X36316	1.000000
X310625	1.000000	X41316	1.000000
X310655	1.000000	X42316	1.000000
X65625	1.000000	X61316	1.000000
X11216	1.000000	X71316	1.000000
X11236	1.000000	X36326	1.000000
X11316	1.000000	X41326	1.000000
X11326	1.000000	X42326	1.000000
X11366	1.000000	X61326	1.000000
X11416	1.000000	X71326	1.000000
X11426	1.000000	X41366	1.000000
X11616	1.000000	X42366	1.000000
X11716	1.000000	X36616	1.000000
X23216	1.000000	X36716	1.000000
X21316	1.000000	X42416	1.000000
X32216	1.000000	X41616	1.000000
X36216	1.000000	X41716	1.000000
X41216	1.000000	X42616	1.000000
X42216	1.000000	X42716	1.000000
X61216	1.000000	X71616	1.000000
X71216	1.000000	E1	4.990000
X23316	1.000000	E2	1.990000
X23326	1.000000	E3	7.990000
X23366	1.000000	E4	0.990000
X23416	1.000000	E5	0.990000
X23426	1.000000	E6	0.990000
X23616	1.000000	L7	1.000000
X23716	1.000000	E7	0.990000
X32316	1.000000		

All other variables are zero.

APPENDIX VIII

THE PROBLEM FORMULATION OF THE DAPS EXAMPLE

(FROZEN INTERVAL = 0)

Minimize

300Cmax-7975
+250LI1+250LI2+250LI3+250LI4+250LI5+250LI6+250LI7
+50EI1+50EI2+50EI3+50EI4+50EI5+50EI6+50EI7

Subject to

(3.2):

cons1: C1-Cmax<=0
cons2: C2-Cmax<=0
cons3: C3-Cmax<=0
cons4: C4-Cmax<=0
cons5: C5-Cmax<=0
cons6: C6-Cmax<=0
cons7: C7-Cmax<=0

(3.3):

cons8: S221>=8
cons9: S351>=8
cons10: S371>=8
cons11: S3162>=8
cons12: S3182>=8
cons13: S632>=8
cons14: S752>=8
cons15: S762>=8
cons16: S3133>=8
cons17: S3143>=8
cons18: S3153>=8
cons19: S443>=8
cons20: S463>=8
cons21: S643>=8
cons22: S723>=8
cons23: S733>=8
cons24: S743>=8
cons25: S3194>=8.5
cons26: S3224>=8.5
cons27: S664>=8.5
cons28: S674>=8.5
cons29: S335>=8.5
cons30: S385>=8.5
cons31: S3105>=8.5
cons32: S625>=8.5
cons33: S655>=8.5
cons34: S216>=8
cons35: S236>=8
cons36: S316>=8
cons37: S326>=8
cons38: S366>=8
cons39: S416>=8
cons40: S426>=8
cons41: S616>=8
cons42: S716>=8

(3.4):

cons43: S116-S123>=1.000000
cons44: S116-S131>=0.500000
cons45: S116-S141>=0.500000
cons46: S116-S152>=1.000000
cons47: S216-S221>=0.500000
cons48: S216-S236>=2.500000
cons49: S236-S245>=2.500000
cons50: S236-S252>=1.000000
cons51: S245-S263>=2.000000
cons52: S245-S2115>=1.500000

cons53: S2115-S2124>=0.500000
cons54: S2115-S2134>=0.500000
cons55: S316-S326>=6.000000
cons56: S316-S335>=10.000000
cons57: S316-S351>=3.000000
cons58: S326-S366>=6.000000
cons59: S326-S371>=3.000000
cons60: S335-S385>=6.000000
cons61: S335-S3105>=6.000000
cons62: S366-S3133>=2.000000
cons63: S366-S3143>=2.000000
cons64: S366-S3153>=1.000000
cons65: S366-S3162>=2.000000
cons66: S366-S3172>=1.000000
cons67: S385-S3182>=6.000000
cons68: S385-S3194>=4.000000
cons69: S3105-S3204>=2.000000
cons70: S3105-S3214>=2.000000
cons71: S3194-S3224>=4.000000
cons72: S416-S426>=6.000000
cons73: S416-S431>=3.000000
cons74: S426-S443>=2.000000
cons75: S426-S453>=2.000000
cons76: S426-S463>=1.000000
cons77: S426-S472>=2.000000
cons78: S426-S482>=1.000000
cons79: S616-S625>=5.000000
cons80: S616-S632>=2.000000
cons81: S625-S643>=4.000000
cons82: S625-S655>=3.000000
cons83: S655-S664>=1.000000
cons84: S655-S674>=1.000000
cons85: S716-S723>=1.000000
cons86: S716-S733>=1.000000
cons87: S716-S743>=0.500000
cons88: S716-S752>=1.000000
cons89: S716-S712>=0.500000

(3.5):
cons90: C1-S116=3.5
cons91: C2-S216=3
cons92: C3-S316=7
cons93: C4-S416=6
cons94: C5-S511=3
cons95: C6-S616=5
cons96: C7-S716=3

(3.6) and (3.7):
cons97: S131-S141+999X13141>=0.500000
cons98: S141-S131+999X14131>=0.500000
cons99: X13141+X14131=1
cons100: S131-S221+999X13221>=0.500000
cons101: S221-S131+999X22131>=0.500000
cons102: X13221+X22131=1
cons103: S131-S351+999X13351>=3.000000
cons104: S351-S131+999X35131>=0.500000
cons105: X13351+X35131=1
cons106: S131-S371+999X13371>=3.000000
cons107: S371-S131+999X37131>=0.500000
cons108: X13371+X37131=1
cons109: S131-S431+999X13431>=3.000000
cons110: S431-S131+999X43131>=0.500000
cons111: X13431+X43131=1
cons112: S131-S511+999X13511>=3.000000
cons113: S511-S131+999X51131>=0.500000
cons114: X13511+X51131=1
cons115: S141-S221+999X14221>=0.500000

cons116: S221-S141+999X22141>=0.500000
cons117: X14221+X22141=1
cons118: S141-S351+999X14351>=3.000000
cons119: S351-S141+999X35141>=0.500000
cons120: X14351+X35141=1
cons121: S141-S371+999X14371>=3.000000
cons122: S371-S141+999X37141>=0.500000
cons123: X14371+X37141=1
cons124: S141-S431+999X14431>=3.000000
cons125: S431-S141+999X43141>=0.500000
cons126: X14431+X43141=1
cons127: S141-S511+999X14511>=3.000000
cons128: S511-S141+999X51141>=0.500000
cons129: X14511+X51141=1
cons130: S221-S351+999X22351>=3.000000
cons131: S351-S221+999X35221>=0.500000
cons132: X22351+X35221=1
cons133: S221-S371+999X22371>=3.000000
cons134: S371-S221+999X37221>=0.500000
cons135: X22371+X37221=1
cons136: S221-S431+999X22431>=3.000000
cons137: S431-S221+999X43221>=0.500000
cons138: X22431+X43221=1
cons139: S221-S511+999X22511>=3.000000
cons140: S511-S221+999X51221>=0.500000
cons141: X22511+X51221=1
cons142: S351-S371+999X35371>=3.000000
cons143: S371-S351+999X37351>=3.000000
cons144: X35371+X37351=1
cons145: S351-S431+999X35431>=3.000000
cons146: S431-S351+999X43351>=3.000000
cons147: X35431+X43351=1
cons148: S351-S511+999X35511>=3.000000
cons149: S511-S351+999X51351>=3.000000
cons150: X35511+X51351=1
cons151: S371-S431+999X37431>=3.000000
cons152: S431-S371+999X43371>=3.000000
cons153: X37431+X43371=1
cons154: S371-S511+999X37511>=3.000000
cons155: S511-S371+999X51371>=3.000000
cons156: X37511+X51371=1
cons157: S431-S511+999X43511>=3.000000
cons158: S511-S431+999X51431>=3.000000
cons159: X43511+X51431=1
cons160: S152-S252+999X15252>=1.000000
cons161: S252-S152+999X25152>=1.000000

cons162: X15252+X25152=1
cons163: S152-S3162+999X153162>=2.000000
cons164: S3162-S152+999X316152>=1.000000
cons165: X153162+X316152=1
cons166: S152-S3172+999X153172>=1.000000
cons167: S3172-S152+999X317152>=1.000000
cons168: X153172+X317152=1
cons169: S152-S3182+999X153182>=6.000000
cons170: S3182-S152+999X318152>=1.000000
cons171: X153182+X318152=1
cons172: S152-S472+999X15472>=2.000000
cons173: S472-S152+999X47152>=1.000000
cons174: X15472+X47152=1
cons175: S152-S482+999X15482>=1.000000
cons176: S482-S152+999X48152>=1.000000
cons177: X15482+X48152=1
cons178: S152-S632+999X15632>=2.000000
cons179: S632-S152+999X63152>=1.000000
cons180: X15632+X63152=1
cons181: S152-S752+999X15752>=1.000000
cons182: S752-S152+999X75152>=1.000000
cons183: X15752+X75152=1
cons184: S152-S762+999X15762>=0.500000
cons185: S762-S152+999X76152>=1.000000
cons186: X15762+X76152=1
cons187: S252-S3162+999X253162>=2.000000
cons188: S3162-S252+999X316252>=1.000000
cons189: X253162+X316252=1
cons190: S252-S3172+999X253172>=1.000000
cons191: S3172-S252+999X317252>=1.000000
cons192: X253172+X317252=1
cons193: S252-S3182+999X253182>=6.000000
cons194: S3182-S252+999X318252>=1.000000
cons195: X253182+X318252=1
cons196: S252-S472+999X25472>=2.000000
cons197: S472-S252+999X47252>=1.000000
cons198: X25472+X47252=1
cons199: S252-S482+999X25482>=1.000000
cons200: S482-S252+999X48252>=1.000000
cons201: X25482+X48252=1
cons202: S252-S632+999X25632>=2.000000
cons203: S632-S252+999X63252>=1.000000
cons204: X25632+X63252=1
cons205: S252-S752+999X25752>=1.000000
cons206: S752-S252+999X75252>=1.000000
cons207: X25752+X75252=1

cons208: S252-S762+999X25762>=0.500000
cons209: S762-S252+999X76252>=1.000000
cons210: X25762+X76252=1
cons211: S3162-S3172+999X3163172>=1.000000
cons212: S3172-S3162+999X3173162>=2.000000
cons213: X3163172+X3173162=1
cons214: S3162-S3182+999X3163182>=6.000000
cons215: S3182-S3162+999X3183162>=2.000000
cons216: X3163182+X3183162=1
cons217: S3162-S472+999X316472>=2.000000
cons218: S472-S3162+999X473162>=2.000000
cons219: X316472+X473162=1
cons220: S3162-S482+999X316482>=1.000000
cons221: S482-S3162+999X483162>=2.000000
cons222: X316482+X483162=1
cons223: S3162-S632+999X316632>=2.000000
cons224: S632-S3162+999X633162>=2.000000
cons225: X316632+X633162=1
cons226: S3162-S752+999X316752>=1.000000
cons227: S752-S3162+999X753162>=2.000000
cons228: X316752+X753162=1
cons229: S3162-S762+999X316762>=0.500000
cons230: S762-S3162+999X763162>=2.000000
cons231: X316762+X763162=1
cons232: S3172-S3182+999X3173182>=6.000000
cons233: S3182-S3172+999X3183172>=1.000000
cons234: X3173182+X3183172=1
cons235: S3172-S472+999X317472>=2.000000
cons236: S472-S3172+999X473172>=1.000000
cons237: X317472+X473172=1
cons238: S3172-S482+999X317482>=1.000000
cons239: S482-S3172+999X483172>=1.000000
cons240: X317482+X483172=1
cons241: S3172-S632+999X317632>=2.000000
cons242: S632-S3172+999X633172>=1.000000
cons243: X317632+X633172=1
cons244: S3172-S752+999X317752>=1.000000
cons245: S752-S3172+999X753172>=1.000000
cons246: X317752+X753172=1
cons247: S3172-S762+999X317762>=0.500000
cons248: S762-S3172+999X763172>=1.000000
cons249: X317762+X763172=1
cons250: S3182-S472+999X318472>=2.000000
cons251: S472-S3182+999X473182>=6.000000
cons252: X318472+X473182=1
cons253: S3182-S482+999X318482>=1.000000

cons254: S482-S3182+999X483182>=6.000000
cons255: X318482+X483182=1
cons256: S3182-S632+999X318632>=2.000000
cons257: S632-S3182+999X633182>=6.000000
cons258: X318632+X633182=1
cons259: S3182-S752+999X318752>=1.000000
cons260: S752-S3182+999X753182>=6.000000
cons261: X318752+X753182=1
cons262: S3182-S762+999X318762>=0.500000
cons263: S762-S3182+999X763182>=6.000000
cons264: X318762+X763182=1
cons265: S472-S482+999X47482>=1.000000
cons266: S482-S472+999X48472>=2.000000
cons267: X47482+X48472=1
cons268: S472-S632+999X47632>=2.000000
cons269: S632-S472+999X63472>=2.000000
cons270: X47632+X63472=1
cons271: S472-S752+999X47752>=1.000000
cons272: S752-S472+999X75472>=2.000000
cons273: X47752+X75472=1
cons274: S472-S762+999X47762>=0.500000
cons275: S762-S472+999X76472>=2.000000
cons276: X47762+X76472=1
cons277: S482-S632+999X48632>=2.000000
cons278: S632-S482+999X63482>=1.000000
cons279: X48632+X63482=1
cons280: S482-S752+999X48752>=1.000000
cons281: S752-S482+999X75482>=1.000000
cons282: X48752+X75482=1
cons283: S482-S762+999X48762>=0.500000
cons284: S762-S482+999X76482>=1.000000
cons285: X48762+X76482=1
cons286: S632-S752+999X63752>=1.000000
cons287: S752-S632+999X75632>=2.000000
cons288: X63752+X75632=1
cons289: S632-S762+999X63762>=0.500000
cons290: S762-S632+999X76632>=2.000000
cons291: X63762+X76632=1
cons292: S752-S762+999X75762>=0.500000
cons293: S762-S752+999X76752>=1.000000
cons294: X75762+X76752=1
cons295: S123-S263+999X12263>=2.000000
cons296: S263-S123+999X26123>=1.000000
cons297: X12263+X26123=1
cons298: S123-S3133+999X123133>=2.000000
cons299: S3133-S123+999X313123>=1.000000

cons300: X123133+X313123=1
cons301: S123-S3143+999X123143>=2.000000
cons302: S3143-S123+999X314123>=1.000000
cons303: X123143+X314123=1
cons304: S123-S3153+999X123153>=1.000000
cons305: S3153-S123+999X315123>=1.000000
cons306: X123153+X315123=1
cons307: S123-S443+999X12443>=2.000000
cons308: S443-S123+999X44123>=1.000000
cons309: X12443+X44123=1
cons310: S123-S453+999X12453>=2.000000
cons311: S453-S123+999X45123>=1.000000
cons312: X12453+X45123=1
cons313: S123-S463+999X12463>=1.000000
cons314: S463-S123+999X46123>=1.000000
cons315: X12463+X46123=1
cons316: S123-S643+999X12643>=4.000000
cons317: S643-S123+999X64123>=1.000000
cons318: X12643+X64123=1
cons319: S123-S723+999X12723>=1.000000
cons320: S723-S123+999X72123>=1.000000
cons321: X12723+X72123=1
cons322: S123-S733+999X12733>=1.000000
cons323: S733-S123+999X73123>=1.000000
cons324: X12733+X73123=1
cons325: S123-S743+999X12743>=0.500000
cons326: S743-S123+999X74123>=1.000000
cons327: X12743+X74123=1
cons328: S263-S3133+999X263133>=2.000000
cons329: S3133-S263+999X313263>=2.000000
cons330: X263133+X313263=1
cons331: S263-S3143+999X263143>=2.000000
cons332: S3143-S263+999X314263>=2.000000
cons333: X263143+X314263=1
cons334: S263-S3153+999X263153>=1.000000
cons335: S3153-S263+999X315263>=2.000000
cons336: X263153+X315263=1
cons337: S263-S443+999X26443>=2.000000
cons338: S443-S263+999X44263>=2.000000
cons339: X26443+X44263=1
cons340: S263-S453+999X26453>=2.000000
cons341: S453-S263+999X45263>=2.000000
cons342: X26453+X45263=1
cons343: S263-S463+999X26463>=1.000000
cons344: S463-S263+999X46263>=2.000000
cons345: X26463+X46263=1

cons346: S263-S643+999X26643>=4.000000
cons347: S643-S263+999X64263>=2.000000
cons348: X26643+X64263=1
cons349: S263-S723+999X26723>=1.000000
cons350: S723-S263+999X72263>=2.000000
cons351: X26723+X72263=1
cons352: S263-S733+999X26733>=1.000000
cons353: S733-S263+999X73263>=2.000000
cons354: X26733+X73263=1
cons355: S263-S743+999X26743>=0.500000
cons356: S743-S263+999X74263>=2.000000
cons357: X26743+X74263=1
cons358: S3133-S3143+999X3133143>=2.000000
cons359: S3143-S3133+999X3143133>=2.000000
cons360: X3133143+X3143133=1
cons361: S3133-S3153+999X3133153>=1.000000
cons362: S3153-S3133+999X3153133>=2.000000
cons363: X3133153+X3153133=1
cons364: S3133-S443+999X313443>=2.000000
cons365: S443-S3133+999X443133>=2.000000
cons366: X313443+X443133=1
cons367: S3133-S453+999X313453>=2.000000
cons368: S453-S3133+999X453133>=2.000000
cons369: X313453+X453133=1
cons370: S3133-S463+999X313463>=1.000000
cons371: S463-S3133+999X463133>=2.000000
cons372: X313463+X463133=1
cons373: S3133-S643+999X313643>=4.000000
cons374: S643-S3133+999X643133>=2.000000
cons375: X313643+X643133=1
cons376: S3133-S723+999X313723>=1.000000
cons377: S723-S3133+999X723133>=2.000000
cons378: X313723+X723133=1
cons379: S3133-S733+999X313733>=1.000000
cons380: S733-S3133+999X733133>=2.000000
cons381: X313733+X733133=1
cons382: S3133-S743+999X313743>=0.500000
cons383: S743-S3133+999X743133>=2.000000
cons384: X313743+X743133=1
cons385: S3143-S3153+999X3143153>=1.000000
cons386: S3153-S3143+999X3153143>=2.000000
cons387: X3143153+X3153143=1
cons388: S3143-S443+999X314443>=2.000000
cons389: S443-S3143+999X443143>=2.000000
cons390: X314443+X443143=1
cons391: S3143-S453+999X314453>=2.000000

cons392: S453-S3143+999X453143>=2.000000
cons393: X314453+X453143=1
cons394: S3143-S463+999X314463>=1.000000
cons395: S463-S3143+999X463143>=2.000000
cons396: X314463+X463143=1
cons397: S3143-S643+999X314643>=4.000000
cons398: S643-S3143+999X643143>=2.000000
cons399: X314643+X643143=1
cons400: S3143-S723+999X314723>=1.000000
cons401: S723-S3143+999X723143>=2.000000
cons402: X314723+X723143=1
cons403: S3143-S733+999X314733>=1.000000
cons404: S733-S3143+999X733143>=2.000000
cons405: X314733+X733143=1
cons406: S3143-S743+999X314743>=0.500000
cons407: S743-S3143+999X743143>=2.000000
cons408: X314743+X743143=1
cons409: S3153-S443+999X315443>=2.000000
cons410: S443-S3153+999X443153>=1.000000
cons411: X315443+X443153=1
cons412: S3153-S453+999X315453>=2.000000
cons413: S453-S3153+999X453153>=1.000000
cons414: X315453+X453153=1
cons415: S3153-S463+999X315463>=1.000000
cons416: S463-S3153+999X463153>=1.000000
cons417: X315463+X463153=1
cons418: S3153-S643+999X315643>=4.000000
cons419: S643-S3153+999X643153>=1.000000
cons420: X315643+X643153=1
cons421: S3153-S723+999X315723>=1.000000
cons422: S723-S3153+999X723153>=1.000000
cons423: X315723+X723153=1
cons424: S3153-S733+999X315733>=1.000000
cons425: S733-S3153+999X733153>=1.000000
cons426: X315733+X733153=1
cons427: S3153-S743+999X315743>=0.500000
cons428: S743-S3153+999X743153>=1.000000
cons429: X315743+X743153=1
cons430: S443-S453+999X44453>=2.000000
cons431: S453-S443+999X45443>=2.000000
cons432: X44453+X45443=1
cons433: S443-S463+999X44463>=1.000000
cons434: S463-S443+999X46443>=2.000000
cons435: X44463+X46443=1
cons436: S443-S643+999X44643>=4.000000
cons437: S643-S443+999X64443>=2.000000

cons438: X44643+X64443=1
cons439: S443-S723+999X44723>=1.000000
cons440: S723-S443+999X72443>=2.000000
cons441: X44723+X72443=1
cons442: S443-S733+999X44733>=1.000000
cons443: S733-S443+999X73443>=2.000000
cons444: X44733+X73443=1
cons445: S443-S743+999X44743>=0.500000
cons446: S743-S443+999X74443>=2.000000
cons447: X44743+X74443=1
cons448: S453-S463+999X45463>=1.000000
cons449: S463-S453+999X46453>=2.000000
cons450: X45463+X46453=1
cons451: S453-S643+999X45643>=4.000000
cons452: S643-S453+999X64453>=2.000000
cons453: X45643+X64453=1
cons454: S453-S723+999X45723>=1.000000
cons455: S723-S453+999X72453>=2.000000
cons456: X45723+X72453=1
cons457: S453-S733+999X45733>=1.000000
cons458: S733-S453+999X73453>=2.000000
cons459: X45733+X73453=1
cons460: S453-S743+999X45743>=0.500000
cons461: S743-S453+999X74453>=2.000000
cons462: X45743+X74453=1
cons463: S463-S643+999X46643>=4.000000
cons464: S643-S463+999X64463>=1.000000
cons465: X46643+X64463=1
cons466: S463-S723+999X46723>=1.000000
cons467: S723-S463+999X72463>=1.000000
cons468: X46723+X72463=1
cons469: S463-S733+999X46733>=1.000000
cons470: S733-S463+999X73463>=1.000000
cons471: X46733+X73463=1
cons472: S463-S743+999X46743>=0.500000
cons473: S743-S463+999X74463>=1.000000
cons474: X46743+X74463=1
cons475: S643-S723+999X64723>=1.000000
cons476: S723-S643+999X72643>=4.000000
cons477: X64723+X72643=1
cons478: S643-S733+999X64733>=1.000000
cons479: S733-S643+999X73643>=4.000000
cons480: X64733+X73643=1
cons481: S643-S743+999X64743>=0.500000
cons482: S743-S643+999X74643>=4.000000
cons483: X64743+X74643=1

cons484: S723-S733+999X72733>=1.000000
cons485: S733-S723+999X73723>=1.000000
cons486: X72733+X73723=1
cons487: S723-S743+999X72743>=0.500000
cons488: S743-S723+999X74723>=1.000000
cons489: X72743+X74723=1
cons490: S733-S743+999X73743>=0.500000
cons491: S743-S733+999X74733>=1.000000
cons492: X73743+X74733=1
cons493: S2124-S2134+999X2122134>=0.500000
cons494: S2134-S2124+999X2132124>=0.500000
cons495: X2122134+X2132124=1
cons496: S2124-S3194+999X2123194>=4.000000
cons497: S3194-S2124+999X3192124>=0.500000
cons498: X2123194+X3192124=1
cons499: S2124-S3204+999X2123204>=2.000000
cons500: S3204-S2124+999X3202124>=0.500000
cons501: X2123204+X3202124=1
cons502: S2124-S3214+999X2123214>=2.000000
cons503: S3214-S2124+999X3212124>=0.500000
cons504: X2123214+X3212124=1
cons505: S2124-S3224+999X2123224>=4.000000
cons506: S3224-S2124+999X3222124>=0.500000
cons507: X2123224+X3222124=1
cons508: S2124-S664+999X212664>=1.000000
cons509: S664-S2124+999X662124>=0.500000
cons510: X212664+X662124=1
cons511: S2124-S674+999X212674>=1.000000
cons512: S674-S2124+999X672124>=0.500000
cons513: X212674+X672124=1
cons514: S2134-S3194+999X2133194>=4.000000
cons515: S3194-S2134+999X3192134>=0.500000
cons516: X2133194+X3192134=1
cons517: S2134-S3204+999X2133204>=2.000000
cons518: S3204-S2134+999X3202134>=0.500000
cons519: X2133204+X3202134=1
cons520: S2134-S3214+999X2133214>=2.000000
cons521: S3214-S2134+999X3212134>=0.500000
cons522: X2133214+X3212134=1
cons523: S2134-S3224+999X2133224>=4.000000
cons524: S3224-S2134+999X3222134>=0.500000
cons525: X2133224+X3222134=1
cons526: S2134-S664+999X213664>=1.000000
cons527: S664-S2134+999X662134>=0.500000
cons528: X213664+X662134=1
cons529: S2134-S674+999X213674>=1.000000

cons530: S674-S2134+999X672134>=0.500000
cons531: X213674+X672134=1
cons532: S3194-S3204+999X3193204>=2.000000
cons533: S3204-S3194+999X3203194>=4.000000
cons534: X3193204+X3203194=1
cons535: S3194-S3214+999X3193214>=2.000000
cons536: S3214-S3194+999X3213194>=4.000000
cons537: X3193214+X3213194=1
cons538: S3194-S3224+999X3193224>=4.000000
cons539: S3224-S3194+999X3223194>=4.000000
cons540: X3193224+X3223194=1
cons541: S3194-S664+999X319664>=1.000000
cons542: S664-S3194+999X663194>=4.000000
cons543: X319664+X663194=1
cons544: S3194-S674+999X319674>=1.000000
cons545: S674-S3194+999X673194>=4.000000
cons546: X319674+X673194=1
cons547: S3204-S3214+999X3203214>=2.000000
cons548: S3214-S3204+999X3213204>=2.000000
cons549: X3203214+X3213204=1
cons550: S3204-S3224+999X3203224>=4.000000
cons551: S3224-S3204+999X3223204>=2.000000
cons552: X3203224+X3223204=1
cons553: S3204-S664+999X320664>=1.000000
cons554: S664-S3204+999X663204>=2.000000
cons555: X320664+X663204=1
cons556: S3204-S674+999X320674>=1.000000
cons557: S674-S3204+999X673204>=2.000000
cons558: X320674+X673204=1
cons559: S3214-S3224+999X3213224>=4.000000
cons560: S3224-S3214+999X3223214>=2.000000
cons561: X3213224+X3223214=1
cons562: S3214-S664+999X321664>=1.000000
cons563: S664-S3214+999X663214>=2.000000
cons564: X321664+X663214=1
cons565: S3214-S674+999X321674>=1.000000
cons566: S674-S3214+999X673214>=2.000000
cons567: X321674+X673214=1
cons568: S3224-S664+999X322664>=1.000000
cons569: S664-S3224+999X663224>=4.000000
cons570: X322664+X663224=1
cons571: S3224-S674+999X322674>=1.000000
cons572: S674-S3224+999X673224>=4.000000
cons573: X322674+X673224=1
cons574: S664-S674+999X66674>=1.000000
cons575: S674-S664+999X67664>=1.000000

cons576: X66674+X67664=1
cons577: S245-S2115+999X242115>=1.500000
cons578: S2115-S245+999X211245>=2.500000
cons579: X242115+X211245=1
cons580: S245-S335+999X24335>=10.000000
cons581: S335-S245+999X33245>=2.500000
cons582: X24335+X33245=1
cons583: S245-S385+999X24385>=6.000000
cons584: S385-S245+999X38245>=2.500000
cons585: X24385+X38245=1
cons586: S245-S3105+999X243105>=6.000000
cons587: S3105-S245+999X310245>=2.500000
cons588: X243105+X310245=1
cons589: S245-S625+999X24625>=5.000000
cons590: S625-S245+999X62245>=2.500000
cons591: X24625+X62245=1
cons592: S245-S655+999X24655>=3.000000
cons593: S655-S245+999X65245>=2.500000
cons594: X24655+X65245=1
cons595: S2115-S335+999X211335>=10.000000
cons596: S335-S2115+999X332115>=1.500000
cons597: X211335+X332115=1
cons598: S2115-S385+999X211385>=6.000000
cons599: S385-S2115+999X382115>=1.500000
cons600: X211385+X382115=1
cons601: S2115-S3105+999X2113105>=6.000000
cons602: S3105-S2115+999X3102115>=1.500000
cons603: X2113105+X3102115=1
cons604: S2115-S625+999X211625>=5.000000
cons605: S625-S2115+999X622115>=1.500000
cons606: X211625+X622115=1
cons607: S2115-S655+999X211655>=3.000000
cons608: S655-S2115+999X652115>=1.500000
cons609: X211655+X652115=1
cons610: S335-S385+999X33385>=6.000000
cons611: S385-S335+999X38335>=10.000000
cons612: X33385+X38335=1
cons613: S335-S3105+999X333105>=6.000000
cons614: S3105-S335+999X310335>=10.000000
cons615: X333105+X310335=1
cons616: S335-S625+999X33625>=5.000000
cons617: S625-S335+999X62335>=10.000000
cons618: X33625+X62335=1
cons619: S335-S655+999X33655>=3.000000
cons620: S655-S335+999X65335>=10.000000
cons621: X33655+X65335=1

cons622: S385-S3105+999X383105>=6.000000
cons623: S3105-S385+999X310385>=6.000000
cons624: X383105+X310385=1
cons625: S385-S625+999X38625>=5.000000
cons626: S625-S385+999X62385>=6.000000
cons627: X38625+X62385=1
cons628: S385-S655+999X38655>=3.000000
cons629: S655-S385+999X65385>=6.000000
cons630: X38655+X65385=1
cons631: S3105-S625+999X310625>=5.000000
cons632: S625-S3105+999X623105>=6.000000
cons633: X310625+X623105=1
cons634: S3105-S655+999X310655>=3.000000
cons635: S655-S3105+999X653105>=6.000000
cons636: X310655+X653105=1
cons637: S625-S655+999X62655>=3.000000
cons638: S655-S625+999X65625>=5.000000
cons639: X62655+X65625=1
cons640: S116-S216+999X11216>=3.000000
cons641: S216-S116+999X21116>=3.500000
cons642: X11216+X21116=1
cons643: S116-S236+999X11236>=2.500000
cons644: S236-S116+999X23116>=3.500000
cons645: X11236+X23116=1
cons646: S116-S316+999X11316>=7.000000
cons647: S316-S116+999X31116>=3.500000
cons648: X11316+X31116=1
cons649: S116-S326+999X11326>=6.000000
cons650: S326-S116+999X32116>=3.500000
cons651: X11326+X32116=1
cons652: S116-S366+999X11366>=6.000000
cons653: S366-S116+999X36116>=3.500000
cons654: X11366+X36116=1
cons655: S116-S416+999X11416>=6.000000
cons656: S416-S116+999X41116>=3.500000
cons657: X11416+X41116=1
cons658: S116-S426+999X11426>=6.000000
cons659: S426-S116+999X42116>=3.500000
cons660: X11426+X42116=1
cons661: S116-S616+999X11616>=5.000000
cons662: S616-S116+999X61116>=3.500000
cons663: X11616+X61116=1
cons664: S116-S716+999X11716>=3.000000
cons665: S716-S116+999X71116>=3.500000
cons666: X11716+X71116=1
cons667: S216-S236+999X21236>=2.500000

cons668: S236-S216+999X23216>=3.000000
cons669: X21236+X23216=1
cons670: S216-S316+999X21316>=7.000000
cons671: S316-S216+999X31216>=3.000000
cons672: X21316+X31216=1
cons673: S216-S326+999X21326>=6.000000
cons674: S326-S216+999X32216>=3.000000
cons675: X21326+X32216=1
cons676: S216-S366+999X21366>=6.000000
cons677: S366-S216+999X36216>=3.000000
cons678: X21366+X36216=1
cons679: S216-S416+999X21416>=6.000000
cons680: S416-S216+999X41216>=3.000000
cons681: X21416+X41216=1
cons682: S216-S426+999X21426>=6.000000
cons683: S426-S216+999X42216>=3.000000
cons684: X21426+X42216=1
cons685: S216-S616+999X21616>=5.000000
cons686: S616-S216+999X61216>=3.000000
cons687: X21616+X61216=1
cons688: S216-S716+999X21716>=3.000000
cons689: S716-S216+999X71216>=3.000000
cons690: X21716+X71216=1
cons691: S236-S316+999X23316>=7.000000
cons692: S316-S236+999X31236>=2.500000
cons693: X23316+X31236=1
cons694: S236-S326+999X23326>=6.000000
cons695: S326-S236+999X32236>=2.500000
cons696: X23326+X32236=1
cons697: S236-S366+999X23366>=6.000000
cons698: S366-S236+999X36236>=2.500000
cons699: X23366+X36236=1
cons700: S236-S416+999X23416>=6.000000
cons701: S416-S236+999X41236>=2.500000
cons702: X23416+X41236=1
cons703: S236-S426+999X23426>=6.000000
cons704: S426-S236+999X42236>=2.500000
cons705: X23426+X42236=1
cons706: S236-S616+999X23616>=5.000000
cons707: S616-S236+999X61236>=2.500000
cons708: X23616+X61236=1
cons709: S236-S716+999X23716>=3.000000
cons710: S716-S236+999X71236>=2.500000
cons711: X23716+X71236=1
cons712: S316-S326+999X31326>=6.000000
cons713: S326-S316+999X32316>=7.000000

cons714: X31326+X32316=1
cons715: S316-S366+999X31366>=6.000000
cons716: S366-S316+999X36316>=7.000000
cons717: X31366+X36316=1
cons718: S316-S416+999X31416>=6.000000
cons719: S416-S316+999X41316>=7.000000
cons720: X31416+X41316=1
cons721: S316-S426+999X31426>=6.000000
cons722: S426-S316+999X42316>=7.000000
cons723: X31426+X42316=1
cons724: S316-S616+999X31616>=5.000000
cons725: S616-S316+999X61316>=7.000000
cons726: X31616+X61316=1
cons727: S316-S716+999X31716>=3.000000
cons728: S716-S316+999X71316>=7.000000
cons729: X31716+X71316=1
cons730: S326-S366+999X32366>=6.000000
cons731: S366-S326+999X36326>=6.000000
cons732: X32366+X36326=1
cons733: S326-S416+999X32416>=6.000000
cons734: S416-S326+999X41326>=6.000000
cons735: X32416+X41326=1
cons736: S326-S426+999X32426>=6.000000
cons737: S426-S326+999X42326>=6.000000
cons738: X32426+X42326=1
cons739: S326-S616+999X32616>=5.000000
cons740: S616-S326+999X61326>=6.000000
cons741: X32616+X61326=1
cons742: S326-S716+999X32716>=3.000000
cons743: S716-S326+999X71326>=6.000000
cons744: X32716+X71326=1
cons745: S366-S416+999X36416>=6.000000
cons746: S416-S366+999X41366>=6.000000
cons747: X36416+X41366=1
cons748: S366-S426+999X36426>=6.000000
cons749: S426-S366+999X42366>=6.000000
cons750: X36426+X42366=1
cons751: S366-S616+999X36616>=5.000000
cons752: S616-S366+999X61366>=6.000000
cons753: X36616+X61366=1
cons754: S366-S716+999X36716>=3.000000
cons755: S716-S366+999X71366>=6.000000
cons756: X36716+X71366=1
cons757: S416-S426+999X41426>=6.000000
cons758: S426-S416+999X42416>=6.000000
cons759: X41426+X42416=1

cons760: S416-S616+999X41616>=5.000000
cons761: S616-S416+999X61416>=6.000000
cons762: X41616+X61416=1
cons763: S416-S716+999X41716>=3.000000
cons764: S716-S416+999X71416>=6.000000
cons765: X41716+X71416=1
cons766: S426-S616+999X42616>=5.000000
cons767: S616-S426+999X61426>=6.000000
cons768: X42616+X61426=1
cons769: S426-S716+999X42716>=3.000000
cons770: S716-S426+999X71426>=6.000000
cons771: X42716+X71426=1
cons772: S616-S716+999X61716>=3.000000
cons773: S716-S616+999X71616>=5.000000
cons774: X61716+X71616=1

(3.8) and (3.9):
cons775: 0.125C1-L1<=6
cons776: E1+0.125C1>=6
cons777: 0.125C2-L2<=7
cons778: E2+0.125C2>=7
cons779: 0.125C3-L3<=14
cons780: E3+0.125C3>=14
cons781: 0.125C4-L4<=3
cons782: E4+0.125C4>=3
cons783: 0.125C5-L5<=1
cons784: E5+0.125C5>=1
cons785: 0.125C6-L6<=5
cons786: E6+0.125C6>=5
cons787: 0.125C7-L7<=3
cons788: E7+0.125C7>=3

(3.10) and (3.11):
cons789: L1-LI1<=0
cons790: E1-EI1<=0.99
cons791: L2-LI2<=0
cons792: E2-EI2<=0.99
cons793: L3-LI3<=0
cons794: E3-EI3<=0.99
cons795: L4-LI4<=0
cons796: E4-EI4<=0.99
cons797: L5-LI5<=0
cons798: E5-EI5<=0.99
cons799: L6-LI6<=0
cons800: E6-EI6<=0.99
cons801: L7-LI7<=0
cons802: E7-EI7<=0.99

(5.1):
cons803: S131=1.000000
cons804: S141=1.500000
cons805: S431=2.000000
cons806: S511=5.000000
cons807: S152=3.000000
cons808: S252=2.000000
cons809: S3172=4.000000
cons810: S472=6.000000
cons811: S482=5.000000
cons812: S123=3.000000
cons813: S263=4.000000
cons814: S453=6.000000
cons815: S2124=3.000000
cons816: S2134=3.500000
cons817: S3204=6.500000
cons818: S3214=4.000000
cons819: S116=4.580000
cons820: S245=6.000000
cons821: S2115=4.000000

Bounds

LI1 free
LI2 free
LI3 free
LI4 free
LI5 free
LI6 free
LI7 free
EI1 free
EI2 free
EI3 free
EI4 free
EI5 free
EI6 free
EI7 free

Integers

EI1 LI1
EI2 LI2
EI3 LI3
EI4 LI4
EI5 LI5
EI6 LI6
EI7 LI7

X13141 X14131
X13221 X22131
X13351 X35131
X13371 X37131
X13431 X43131
X13511 X51131
X14221 X22141
X14351 X35141
X14371 X37141
X14431 X43141
X14511 X51141
X22351 X35221
X22371 X37221
X22431 X43221
X22511 X51221
X35371 X37351
X35431 X43351
X35511 X51351
X37431 X43371
X37511 X51371
X43511 X51431
X15252 X25152
X153162 X316152
X153172 X317152
X153182 X318152
X15472 X47152
X15482 X48152
X15632 X63152
X15752 X75152
X15762 X76152
X253162 X316252
X253172 X317252
X253182 X318252
X25472 X47252
X25482 X48252
X25632 X63252
X25752 X75252
X25762 X76252
X3163172 X3173162
X3163182 X3183162
X316472 X473162
X316482 X483162
X316632 X633162
X316752 X753162
X316762 X763162
X3173182 X3183172
X317472 X473172
X317482 X483172
X317632 X633172
X317752 X753172
X317762 X763172
X318472 X473182
X318482 X483182
X318632 X633182
X318752 X753182
X318762 X763182
X47482 X48472
X47632 X63472
X47752 X75472
X47762 X76472
X48632 X63482
X48752 X75482
X48762 X76482
X63752 X75632
X63762 X76632
X75762 X76752
X12263 X26123
X123133 X313123
X123143 X314123
X123153 X315123
X12443 X44123
X12453 X45123
X12463 X46123
X12643 X64123
X12723 X72123
X12733 X73123
X12743 X74123
X263133 X313263
X263143 X314263
X263153 X315263
X26443 X44263
X26453 X45263
X26463 X46263
X26643 X64263
X26723 X72263
X26733 X73263
X26743 X74263
X3133143 X3143133
X3133153 X3153133
X313443 X443133
X313453 X453133
X313463 X463133
X313643 X643133
X313723 X723133
X313733 X733133
X313743 X743133
X3143153 X3153143
X314443 X443143
X314453 X453143
X314463 X463143
X314643 X643143
X314723 X723143
X314733 X733143
X314743 X743143
X315443 X443153

X315453 X453153
X315463 X463153
X315643 X643153
X315723 X723153
X315733 X733153
X315743 X743153
X44453 X45443
X44463 X46443
X44643 X64443
X44723 X72443
X44733 X73443
X44743 X74443
X45463 X46453
X45643 X64453
X45723 X72453
X45733 X73453
X45743 X74453
X46643 X64463
X46723 X72463
X46733 X73463
X46743 X74463
X64723 X72643
X64733 X73643
X64743 X74643
X72733 X73723
X72743 X74723
X73743 X74733
X2122134 X2132124
X2123194 X3192124
X2123204 X3202124
X2123214 X3212124
X2123224 X3222124
X212664 X662124
X212674 X672124
X2133194 X3192134
X2133204 X3202134
X2133214 X3212134
X2133224 X3222134
X213664 X662134
X213674 X672134
X3193204 X3203194
X3193214 X3213194
X3193224 X3223194
X319664 X663194
X319674 X673194
X3203214 X3213204
X3203224 X3223204
X320664 X663204
X320674 X673204
X3213224 X3223214
X321664 X663214
X321674 X673214
X322664 X663224
X322674 X673224
X66674 X67664
X242115 X211245
X24335 X33245
X24385 X38245
X243105 X310245
X24625 X62245
X24655 X65245
X211335 X332115
X211385 X382115
X2113105 X3102115
X211625 X622115
X211655 X652115
X33385 X38335
X333105 X310335
X33625 X62335
X33655 X65335
X383105 X310385
X38625 X62385
X38655 X65385
X310625 X623105
X310655 X653105
X62655 X65625
X11216 X21116
X11236 X23116
X11316 X31116
X11326 X32116
X11366 X36116
X11416 X41116
X11426 X42116
X11616 X61116
X11716 X71116
X21236 X23216
X21316 X31216
X21326 X32216
X21366 X36216
X21416 X41216
X21426 X42216
X21616 X61216
X21716 X71216
X23316 X31236
X23326 X32236
X23366 X36236
X23416 X41236
X23426 X42236
X23616 X61236
X23716 X71236
X31326 X32316
X31366 X36316
X31416 X41316
X31426 X42316
X31616 X61316
X31716 X71316
X32366 X36326
X32416 X41326
X32426 X42326
X32616 X61326
X32716 X71326
X36416 X41366
X36426 X42366
X36616 X61366
X36716 X71366
X41426 X42416
X41616 X61416
X41716 X71416
X42616 X61426
X42716 X71426
X61716 X71616

End

APPENDIX IX

THE OPTIMAL RESULTS OF THE DAPS EXAMPLE

(FROZEN INTERVAL = 0)

Integer optimal
Objective = 8.7250000000e+003
Solution time = 426.41 sec.
Iterations = 3894808
Nodes =1117938

Variable Name	Solution Value	Variable Name	Solution Value
Cmax	53.000000	S664	23.000000
LI4	1.000000	S674	22.000000
EI1	4.000000	S335	33.000000
EI3	7.000000	S385	17.500000
C1	8.080000	S3105	8.500000
C2	53.000000	S625	27.000000
C3	50.000000	S655	24.000000
C4	26.000000	S216	50.000000
C5	8.000000	S236	8.500000
C6	37.000000	S316	43.000000
C7	20.000000	S326	37.000000
S221	8.000000	S366	26.000000
S351	8.500000	S416	20.000000
S371	11.500000	S426	11.000000
S3162	9.500000	S616	32.000000
S3182	11.500000	S716	17.000000
S632	17.500000	S116	4.580000
S752	8.500000	S123	3.000000
S762	8.000000	S131	1.000000
S3133	17.000000	S141	1.500000
S3143	24.000000	S152	3.000000
S3153	15.000000	S245	6.000000
S443	9.000000	S252	2.000000
S463	8.000000	S263	4.000000
S643	19.000000	S2115	4.000000
S723	16.000000	S2124	3.000000
S733	11.000000	S2134	3.500000
S743	12.000000	S3172	4.000000
S3194	12.500000	S3204	6.500000
S3224	8.500000	S3214	4.000000

S431	2.000000	X483162	1.000000
S453	6.000000	X316632	1.000000
S472	6.000000	X753162	1.000000
S482	5.000000	X763162	1.000000
S511	5.000000	X3173182	1.000000
X13141	1.000000	X317472	1.000000
X13221	1.000000	X317482	1.000000
X13351	1.000000	X317632	1.000000
X13371	1.000000	X317752	1.000000
X13431	1.000000	X317762	1.000000
X13511	1.000000	X473182	1.000000
X14221	1.000000	X483182	1.000000
X14351	1.000000	X318632	1.000000
X14371	1.000000	X753182	1.000000
X14431	1.000000	X763182	1.000000
X14511	1.000000	X48472	1.000000
X22351	1.000000	X47632	1.000000
X22371	1.000000	X47752	1.000000
X43221	1.000000	X47762	1.000000
X51221	1.000000	X48632	1.000000
X35371	1.000000	X48752	1.000000
X43351	1.000000	X48762	1.000000
X51351	1.000000	X75632	1.000000
X43371	1.000000	X76632	1.000000
X51371	1.000000	X76752	1.000000
X43511	1.000000	X12263	1.000000
X25152	1.000000	X123133	1.000000
X153162	1.000000	X123143	1.000000
X153172	1.000000	X123153	1.000000
X153182	1.000000	X12443	1.000000
X15472	1.000000	X12453	1.000000
X15482	1.000000	X12463	1.000000
X15632	1.000000	X12643	1.000000
X15752	1.000000	X12723	1.000000
X15762	1.000000	X12733	1.000000
X253162	1.000000	X12743	1.000000
X253172	1.000000	X263133	1.000000
X253182	1.000000	X263143	1.000000
X25472	1.000000	X263153	1.000000
X25482	1.000000	X26443	1.000000
X25632	1.000000	X26453	1.000000
X25752	1.000000	X26463	1.000000
X25762	1.000000	X26643	1.000000
X3173162	1.000000	X26723	1.000000
X3163182	1.000000	X26733	1.000000
X473162	1.000000	X26743	1.000000

X3133143	1.000000	X2123194	1.000000
X3153133	1.000000	X2123204	1.000000
X443133	1.000000	X2123214	1.000000
X453133	1.000000	X2123224	1.000000
X463133	1.000000	X212664	1.000000
X313643	1.000000	X212674	1.000000
X723133	1.000000	X2133194	1.000000
X733133	1.000000	X2133204	1.000000
X743133	1.000000	X2133214	1.000000
X3153143	1.000000	X2133224	1.000000
X443143	1.000000	X213664	1.000000
X453143	1.000000	X213674	1.000000
X463143	1.000000	X3203194	1.000000
X643143	1.000000	X3213194	1.000000
X723143	1.000000	X3223194	1.000000
X733143	1.000000	X319664	1.000000
X743143	1.000000	X319674	1.000000
X443153	1.000000	X3213204	1.000000
X453153	1.000000	X3203224	1.000000
X463153	1.000000	X320664	1.000000
X315643	1.000000	X320674	1.000000
X315723	1.000000	X3213224	1.000000
X733153	1.000000	X321664	1.000000
X743153	1.000000	X321674	1.000000
X45443	1.000000	X322664	1.000000
X46443	1.000000	X322674	1.000000
X44643	1.000000	X67664	1.000000
X44723	1.000000	X211245	1.000000
X44733	1.000000	X24335	1.000000
X44743	1.000000	X24385	1.000000
X45463	1.000000	X243105	1.000000
X45643	1.000000	X24625	1.000000
X45723	1.000000	X24655	1.000000
X45733	1.000000	X211335	1.000000
X45743	1.000000	X211385	1.000000
X46643	1.000000	X2113105	1.000000
X46723	1.000000	X211625	1.000000
X46733	1.000000	X211655	1.000000
X46743	1.000000	X38335	1.000000
X72643	1.000000	X310335	1.000000
X73643	1.000000	X62335	1.000000
X74643	1.000000	X65335	1.000000
X73723	1.000000	X310385	1.000000
X74723	1.000000	X38625	1.000000
X73743	1.000000	X38655	1.000000
X2122134	1.000000	X310625	1.000000

X310655	1.000000	X41316	1.000000
X65625	1.000000	X42316	1.000000
X11216	1.000000	X61316	1.000000
X11236	1.000000	X71316	1.000000
X11316	1.000000	X36326	1.000000
X11326	1.000000	X41326	1.000000
X11366	1.000000	X42326	1.000000
X11416	1.000000	X61326	1.000000
X11426	1.000000	X71326	1.000000
X11616	1.000000	X41366	1.000000
X11716	1.000000	X42366	1.000000
X23216	1.000000	X36616	1.000000
X31216	1.000000	X71366	1.000000
X32216	1.000000	X42416	1.000000
X36216	1.000000	X41616	1.000000
X41216	1.000000	X71416	1.000000
X42216	1.000000	X42616	1.000000
X61216	1.000000	X42716	1.000000
X71216	1.000000	X71616	1.000000
X23316	1.000000	E1	4.990000
X23326	1.000000	E2	0.990000
X23366	1.000000	E3	7.990000
X23416	1.000000	L4	1.000000
X23426	1.000000	E4	0.990000
X23616	1.000000	E5	0.990000
X23716	1.000000	E6	0.990000
X32316	1.000000	E7	0.990000
X36316	1.000000		

All other variables are zero.

APPENDIX X

THE RESULTS FOR THE CASE STUDY

Table X-1 (a) The results obtained by APSDSS when *t* = 0

Items	Starting time (hour)	Finish time (hour)	Random number
O1F1	1761.0	1763.0	0.029
O1S1	1759.0	1761.0	0.907
O2F1	2289.0	2292.0	0.987
O2S1	2286.0	2289.0	0.994
O3F2	1768.0	1769.0	0.181
O3S2	1767.0	1768.0	0.178
O3S3	1766.0	1767.0	0.293
O4F2	1480.0	1481.0	0.822
O4S2	1479.0	1480.0	0.550
O4S3	1130.0	1131.0	0.606
O5F3	2087.0	2090.0	0.698
O5S4	1774.0	1777.0	0.950
O5S5	1777.0	1780.0	0.801
O6F3	2282.0	2286.0	0.595
O6S4	2278.0	2282.0	0.090
O6S5	89.0	93.0	0.255
O7S1	2431.0	2435.0	0.556
O8S1	1937.0	1939.0	0.771
O9S2	2276.0	2278.0	0.452
O9S3	2274.0	2276.0	0.991
O10S2	2271.0	2274.0	0.275
O10S3	1770.0	1773.0	0.932
O11S4	1131.0	1133.0	0.862
O12S4	1769.0	1770.0	0.902
O13S4	2435.0	2439.0	0.997
O14S5	1773.0	1774.0	0.942
O15S5	2090.0	2096.0	0.185
O16S3	1780.0	1782.0	0.837
O17S3	1763.0	1766.0	0.077
O1C1	104.0	114.0	0.315
O1C2	219.0	224.0	0.906
O1C3	24.0	29.0	0.151

Table X-1 (b) The results obtained by APSDSS when *t* = 0

Items	Starting time (hour)	Finish time (hour)	Random number
O2C1	114.0	129.0	0.374
O2C2	169.0	176.5	0.437
O2C3	4.0	11.5	0.046
O3C8	194.0	199.0	0.601
O4C8	189.0	194.0	0.586
O5C11	154.0	169.0	0.423
O5C14	231.5	239.0	0.952
O5C15	224.0	231.5	0.923
O6C11	199.0	219.0	0.622
O6C14	11.5	21.5	0.098
O6C15	79.0	89.0	0.287
O9C8	129.0	139.0	0.399
O10C8	89.0	104.0	0.305
O11C11	49.0	59.0	0.273
O12C11	29.0	34.0	0.213
O13C11	59.0	79.0	0.278
O14C14	21.5	24.0	0.116
O14C15	186.5	189.0	0.560
O15C14	239.0	254.0	0.974
O15C15	34.0	49.0	0.242
O16C8	176.5	186.5	0.440
O17C8	139.0	154.0	0.421
O1C4OP1	1026.0	1106.0	0.309
O1C5OP1	1679.0	1759.0	0.259
O2C4OP1	452.0	572.0	0.506
O2C5OP1	898.0	1018.0	0.503
O3C6OP1	1106.0	1146.0	0.302
O3C7OP1	698.0	738.0	0.573
O4C6OP1	1146.0	1186.0	0.187
O4C7OP1	1439.0	1479.0	0.671
O5C10OP1	1937.0	2087.0	0.141
O6C10OP1	1479.0	1679.0	0.899
O7C4OP1	2271.0	2431.0	0.715
O7C5OP1	738.0	898.0	0.516
O8C4OP1	1857.0	1937.0	0.767
O8C5OP1	1266.0	1346.0	0.664
O9C7OP1	1186.0	1266.0	0.301
O10C7OP1	2151.0	2271.0	0.062
O18C4OP1	1799.0	1807.0	0.245
O19C4OP1	2111.0	2151.0	0.732

Table X-1 (c) The results obtained by APSDSS when *t* = 0

Items	Starting time (hour)	Finish time (hour)	Random number
O20C5OP1	1759.0	1799.0	0.193
O21C5OP1	2095.0	2111.0	0.250
O22C5OP1	2087.0	2095.0	0.857
O23C6OP1	372.0	452.0	0.419
O24C6OP1	1376.0	1432.0	0.511
O25C7OP1	682.0	698.0	0.330
O26C7OP1	602.0	682.0	0.084
O27C7OP1	1018.0	1026.0	0.675
O30C10OP1	1346.0	1376.0	0.818
O31C10OP1	1807.0	1857.0	0.763
O1C4OP2	969.0	979.0	0.793
O1C5OP3	1405.0	1425.0	0.907
O1C4OP5	372.0	382.0	0.443
O2C4OP2	326.0	341.0	0.032
O2C5OP3	536.0	566.0	0.621
O2C4OP5	4.0	19.0	0.089
O3C6OP2	979.0	984.0	0.793
O3C9OP1	1584.0	1589.0	0.272
O3C7OP4	424.0	434.0	0.594
O3C9OP3	1544.0	1554.0	0.914
O4C6OP2	984.0	989.0	0.329
O4C9OP1	1125.0	1130.0	0.662
O4C7OP4	382.0	392.0	0.450
O4C9OP3	1085.0	1095.0	0.854
O5C13	989.0	1004.0	0.806
O5C10OP4	1789.0	1804.0	0.832
O5C10OP6	1004.0	1019.0	0.851
O6C13	2036.0	2056.0	0.991
O6C10OP4	1229.0	1249.0	0.685
O6C10OP6	1145.0	1165.0	0.881
O7C4OP2	2251.0	2271.0	0.256
O7C5OP3	446.0	486.0	0.621
O7C4OP5	566.0	586.0	0.642
O8C4OP2	1779.0	1789.0	0.955
O8C5OP3	814.0	834.0	0.395
O8C4OP5	761.0	771.0	0.717
O9C9OP1	1634.0	1644.0	0.931
O9C7OP4	741.0	761.0	0.689
O9C9OP3	156.0	176.0	0.277
O10C9OP1	141.0	156.0	0.016

Table X-1 (d) The results obtained by APSDSS when *t* = 0

Items	Starting time (hour)	Finish time (hour)	Random number
O10C7OP4	1589.0	1619.0	0.922
O10C9OP3	19.0	49.0	0.222
O11C13	392.0	402.0	0.298
O12C13	1619.0	1624.0	0.922
O13C13	716.0	736.0	0.113
O16C9OP1	1769.0	1779.0	0.047
O16C9OP3	341.0	361.0	0.360
O17C9OP1	1529.0	1544.0	0.592
O17C9OP3	180.0	210.0	0.316
O18C4OP2	1669.0	1670.0	0.206
O18C4OP5	361.0	362.0	0.373
O19C4OP2	1981.0	1986.0	0.184
O19C4OP5	736.0	741.0	0.684
O20C5OP3	1624.0	1634.0	0.165
O21C5OP3	1081.0	1085.0	0.368
O22C5OP3	1930.0	1932.0	0.659
O23C6OP2	362.0	372.0	0.391
O24C6OP2	1330.0	1337.0	0.687
O25C7OP4	176.0	180.0	0.313
O26C7OP4	404.0	424.0	0.524
O27C7OP4	402.0	404.0	0.521
O28C9OP1	1069.0	1074.0	0.485
O28C9OP3	1019.0	1029.0	0.853
O29C9OP1	954.0	969.0	0.303
O29C9OP3	834.0	864.0	0.738
O30C10OP4	1142.0	1145.0	0.145
O30C10OP6	1130.0	1133.0	0.878
O31C10OP4	1664.0	1669.0	0.608
O31C10OP6	1644.0	1649.0	0.938
O34C13	1390.0	1405.0	0.073
O35C13	2016.0	2036.0	0.618
O1C5OP2	1425.0	1455.0	0.721
O1C4OP4	626.0	656.0	0.645
O1C5OP5	475.0	505.0	0.613
O2C5OP2	566.0	611.0	0.401
O2C4OP4	170.0	215.0	0.299
O2C5OP5	275.0	320.0	0.415
O3C6OP3	611.0	626.0	0.641
O3C7OP3	454.0	469.0	0.140
O4C6OP3	926.0	941.0	0.818

Table X-1 (e) The results obtained by APSDSS when *t* = 0

Items	Starting time (hour)	Finish time (hour)	Random number
O4C7OP3	1330.0	1345.0	0.891
O5S3C12	65.0	110.0	0.216
O5C10OP3	1804.0	1849.0	0.722
O5C13C12	941.0	986.0	0.821
O5C10OP5	1684.0	1729.0	0.963
O6S3C12	215.0	275.0	0.343
O6C10OP3	1249.0	1309.0	0.714
O6C13C12	986.0	1046.0	0.838
O6C10OP5	1169.0	1229.0	0.719
O7C5OP2	505.0	565.0	0.565
O7C4OP4	2061.0	2121.0	0.995
O7C5OP5	2.0	62.0	0.067
O8C5OP2	1079.0	1109.0	0.865
O8C4OP4	896.0	926.0	0.752
O8C5OP5	140.0	170.0	0.283
O9C7OP3	1046.0	1076.0	0.857
O10C7OP3	2016.0	2061.0	0.990
O11S3C12	866.0	896.0	0.710
O11C13C12	320.0	350.0	0.470
O12S3C12	1109.0	1124.0	0.867
O12C13C12	1154.0	1169.0	0.881
O13S3C12	1455.0	1515.0	0.923
O13C13C12	656.0	716.0	0.656
O18C4OP4	1076.0	1079.0	0.865
O19C4OP4	1941.0	1956.0	0.974
O20C5OP2	1634.0	1649.0	0.904
O20C5OP5	1515.0	1530.0	0.923
O21C5OP2	1935.0	1941.0	0.964
O21C5OP5	469.0	475.0	0.598
O22C5OP2	1932.0	1935.0	0.385
O22C5OP5	62.0	65.0	0.089
O23C6OP3	110.0	140.0	0.226
O24C6OP3	1309.0	1330.0	0.883
O25C7OP3	350.0	356.0	0.490
O26C7OP3	424.0	454.0	0.188
O27C7OP3	404.0	407.0	0.147
O30C10OP3	1145.0	1154.0	0.204
O30C10OP5	1133.0	1142.0	0.878
O31C10OP3	1669.0	1684.0	0.303
O31C10OP5	1649.0	1664.0	0.296

Table X-1 (f) The results obtained by APSDSS when *t* = 0

Items	Starting time (hour)	Finish time (hour)	Random number
O32C12	716.0	866.0	0.693
O33C12	2121.0	2211.0	0.995
O34C13C12	1345.0	1390.0	0.894
O35C13C12	1956.0	2016.0	0.987
O1C4OP3	704.0	754.0	0.193
O1C5OP4	644.0	704.0	0.177
O2C4OP3	251.0	326.0	0.340
O2C5OP4	446.0	536.0	0.082
O3C7OP2	614.0	644.0	0.467
O3C6OP4	141.0	166.0	0.229
O3C9OP2	1554.0	1584.0	0.824
O4C7OP2	1409.0	1439.0	0.332
O4C6OP4	166.0	191.0	0.257
O4C9OP2	1095.0	1125.0	0.713
O5C10OP2	1849.0	1924.0	0.068
O6C10OP2	1309.0	1409.0	0.525
O7C4OP3	2151.0	2251.0	0.053
O7C5OP4	326.0	446.0	0.394
O8C4OP3	954.0	1004.0	0.747
O8C5OP4	754.0	814.0	0.724
O9C7OP2	1125.0	1185.0	0.314
O9C9OP2	191.0	251.0	0.062
O10C7OP2	2061.0	2151.0	0.206
O10C9OP2	51.0	141.0	0.139
O16C9OP2	1709.0	1769.0	0.954
O17C9OP2	1439.0	1529.0	0.913
O18C4OP3	1614.0	1619.0	0.945
O19C4OP3	1956.0	1981.0	0.912
O20C5OP4	1584.0	1614.0	0.908
O21C5OP4	1069.0	1081.0	0.853
O22C5OP4	1924.0	1930.0	0.963
O23C6OP4	1.0	51.0	0.070
O24C6OP4	1004.0	1039.0	0.823
O25C7OP2	602.0	614.0	0.566
O26C7OP2	542.0	602.0	0.154
O27C7OP2	536.0	542.0	0.149
O28C9OP2	1039.0	1069.0	0.301
O29C9OP2	864.0	954.0	0.038
O30C10OP2	1185.0	1200.0	0.344
O31C10OP2	1684.0	1709.0	0.948

Table X-2 (a) The results obtained by APSDSS when *t* = Day 1

Items	Starting time (hour)	Finish time (hour)	Random number
O1F1	2122.0	2124.0	0.044
O1S1	2120.0	2122.0	0.827
O2F1	2324.0	2327.0	0.417
O2S1	2040.0	2043.0	0.613
O3F2	2048.0	2049.0	0.744
O3S2	2047.0	2048.0	0.251
O3S3	2046.0	2047.0	0.886
O4F2	1245.0	1246.0	0.247
O4S2	1244.0	1245.0	0.825
O4S3	938.0	939.0	0.589
O5F3	2058.0	2061.0	0.879
O5S4	2055.0	2058.0	0.441
O5S5	196.5	199.5	0.544
O6F3	2320.0	2324.0	0.485
O6S4	623.0	627.0	0.600
O6S5	1181.0	1185.0	0.722
O7S1	2480.0	2484.0	0.195
O8S1	1920.0	1922.0	0.276
O9S2	1587.0	1589.0	0.852
O9S3	1177.0	1179.0	0.026
O10S2	2330.0	2333.0	0.190
O10S3	2327.0	2330.0	0.354
O11S4	1179.0	1181.0	0.625
O12S4	1243.0	1244.0	0.396
O13S4	1922.0	1926.0	0.181
O14S5	195.5	196.5	0.231
O15S5	2049.0	2055.0	0.111
O16S3	2043.0	2045.0	0.923
O17S3	2124.0	2127.0	0.073
O36F1	2045.0	2046.0	0.673
O36S1	1586.0	1587.0	0.814
O37S3	1111.0	1113.0	0.406
O1C1	201.5	211.5	0.822
O1C2	131.5	136.5	0.430
O2C1	259.0	274.0	0.991
O2C2	114.0	121.5	0.337
O3C8	196.5	201.5	0.700
O4C8	126.5	131.5	0.430
O5C11	211.5	226.5	0.883
O5C14	161.5	169.0	0.560
O5C15	189.0	196.5	0.589

Table X-2 (b) The results obtained by APSDSS when *t* = Day 1

Items	Starting time (hour)	Finish time (hour)	Random number
O6C11	49.0	69.0	0.055
O6C15	101.5	111.5	0.293
O9C8	179.0	189.0	0.583
O10C8	136.5	151.5	0.511
O11C11	151.5	161.5	0.549
O13C11	69.0	89.0	0.074
O14C15	111.5	114.0	0.335
O15C14	229.0	244.0	0.962
O16C8	169.0	179.0	0.582
O17C8	244.0	259.0	0.984
O36C1	121.5	126.5	0.351
O36C2	99.0	101.5	0.275
O36C3	226.5	229.0	0.928
O37C8	89.0	99.0	0.256
O1C4OP1	2040.0	2120.0	0.975
O1C5OP1	601.0	681.0	0.250
O2C4OP1	1920.0	2040.0	0.921
O2C5OP1	1426.0	1546.0	0.251
O3C6OP1	1018.0	1058.0	0.202
O3C7OP1	441.0	481.0	0.165
O4C6OP1	1058.0	1098.0	0.630
O4C7OP1	1098.0	1138.0	0.025
O5C10OP1	1586.0	1736.0	0.415
O6C10OP1	2120.0	2320.0	0.454
O7C4OP1	2320.0	2480.0	0.727
O7C5OP1	1154.0	1314.0	0.455
O8C4OP1	1736.0	1816.0	0.159
O8C5OP1	1840.0	1920.0	0.813
O9C7OP1	286.0	366.0	0.082
O10C7OP1	848.0	968.0	0.131
O18C4OP1	1314.0	1322.0	0.415
O19C4OP1	561.0	601.0	0.080
O20C5OP1	681.0	721.0	0.113
O21C5OP1	1816.0	1832.0	0.257
O22C5OP1	1322.0	1330.0	0.695
O23C6OP1	481.0	561.0	0.138
O24C6OP1	1370.0	1426.0	0.614
O25C7OP1	1138.0	1154.0	0.090
O26C7OP1	758.0	838.0	0.183
O27C7OP1	1832.0	1840.0	0.379
O30C10OP1	721.0	751.0	0.548

Table X-2 (c) The results obtained by APSDSS when *t* = Day 1

Items	Starting time (hour)	Finish time (hour)	Random number
O31C10OP1	968.0	1018.0	0.137
O36C4OP1	1330.0	1370.0	0.100
O36C5OP1	1546.0	1586.0	0.742
O1C4OP2	1793.0	1803.0	0.666
O1C5OP3	371.0	391.0	0.373
O1C4OP5	209.0	219.0	0.171
O2C4OP2	606.0	621.0	0.005
O2C5OP3	1187.0	1217.0	0.723
O3C6OP2	773.0	778.0	0.315
O3C9OP1	1898.0	1903.0	0.961
O3C7OP4	74.0	84.0	0.111
O3C9OP3	1614.0	1624.0	0.873
O4C6OP2	219.0	224.0	0.178
O4C9OP1	933.0	938.0	0.260
O4C7OP4	274.0	284.0	0.267
O4C9OP3	778.0	788.0	0.661
O5C13	1217.0	1232.0	0.725
O5C10OP4	1313.0	1328.0	0.546
O5C10OP6	259.0	274.0	0.267
O6C13	421.0	441.0	0.245
O6C10OP4	1713.0	1733.0	0.498
O6C10OP6	1624.0	1644.0	0.893
O7C4OP2	2198.0	2218.0	0.650
O7C5OP3	441.0	481.0	0.422
O7C4OP5	224.0	244.0	0.217
O8C4OP2	1598.0	1608.0	0.604
O8C5OP3	1393.0	1413.0	0.844
O8C4OP5	1177.0	1187.0	0.705
O9C9OP1	1167.0	1177.0	0.703
O9C7OP4	52.0	72.0	0.062
O9C9OP3	486.0	506.0	0.429
O10C9OP1	1903.0	1918.0	0.992
O10C7OP4	661.0	691.0	0.593
O11C13	941.0	951.0	0.678
O12C13	1238.0	1243.0	0.002
O13C13	1803.0	1823.0	0.253
O16C9OP1	1353.0	1363.0	0.128
O16C9OP3	621.0	641.0	0.465
O17C9OP1	1883.0	1898.0	0.883
O17C9OP3	89.0	119.0	0.132
O18C4OP2	1166.0	1167.0	0.158

Table X-2 (d) The results obtained by APSDSS when *t* = Day 1

Items	Starting time (hour)	Finish time (hour)	Random number
O18C4OP5	294.0	295.0	0.359
O19C4OP2	481.0	486.0	0.426
O19C4OP5	84.0	89.0	0.128
O20C5OP3	284.0	294.0	0.281
O21C5OP3	1610.0	1614.0	0.259
O22C5OP3	939.0	941.0	0.509
O23C6OP2	295.0	305.0	0.365
O24C6OP2	1334.0	1341.0	0.315
O25C7OP4	696.0	700.0	0.600
O26C7OP4	641.0	661.0	0.565
O27C7OP4	72.0	74.0	0.070
O28C9OP1	1243.0	1248.0	0.340
O28C9OP3	305.0	315.0	0.373
O29C9OP1	1363.0	1378.0	0.803
O29C9OP3	391.0	421.0	0.381
O30C10OP4	206.0	209.0	0.165
O30C10OP6	49.0	52.0	0.060
O31C10OP4	691.0	696.0	0.596
O31C10OP6	119.0	124.0	0.160
O34C13	244.0	259.0	0.266
O35C13	1413.0	1433.0	0.849
O36C4OP2	1293.0	1298.0	0.198
O36C4OP5	1248.0	1253.0	0.780
O36C5OP3	1383.0	1393.0	0.384
O37C9OP1	1101.0	1111.0	0.340
O37C9OP3	951.0	971.0	0.682
O1C5OP2	503.0	533.0	0.480
O1C4OP4	1713.0	1743.0	0.899
O1C5OP5	218.0	248.0	0.199
O2C5OP2	1349.0	1394.0	0.829
O2C4OP4	458.0	503.0	0.455
O2C5OP5	836.0	881.0	0.672
O3C6OP3	758.0	773.0	0.602
O3C7OP3	338.0	353.0	0.336
O4C6OP3	152.0	167.0	0.137
O4C7OP3	443.0	458.0	0.446
O5S4C12	1893.0	1938.0	0.966
O5C10OP3	1409.0	1454.0	0.861
O5C13C12	623.0	668.0	0.522
O5C10OP5	1268.0	1313.0	0.787
O6S4C12	563.0	623.0	0.506

Table X-2 (e) The results obtained by APSDSS when *t* = Day 1

Items	Starting time (hour)	Finish time (hour)	Random number
O6C10OP3	1938.0	1998.0	0.985
O6C13C12	353.0	413.0	0.391
O6C10OP5	1653.0	1713.0	0.550
O7C5OP2	887.0	947.0	0.681
O7C4OP4	1998.0	2058.0	0.994
O8C5OP2	1623.0	1653.0	0.887
O8C4OP4	1454.0	1484.0	0.862
O8C5OP5	947.0	977.0	0.687
O9C7OP3	167.0	197.0	0.164
O10C7OP3	698.0	743.0	0.197
O11S4C12	980.0	1010.0	0.706
O11C13C12	413.0	443.0	0.416
O12S4C12	122.0	137.0	0.058
O12C13C12	1223.0	1238.0	0.744
O13S4C12	1163.0	1223.0	0.728
O13C13C12	1743.0	1803.0	0.908
O18C4OP4	977.0	980.0	0.702
O19C4OP4	137.0	152.0	0.054
O20C5OP2	548.0	563.0	0.492
O20C5OP5	62.0	77.0	0.013
O21C5OP2	1614.0	1620.0	0.864
O21C5OP5	1484.0	1490.0	0.865
O22C5OP2	1160.0	1163.0	0.721
O22C5OP5	833.0	836.0	0.669
O23C6OP3	248.0	278.0	0.280
O24C6OP3	1313.0	1334.0	0.792
O25C7OP3	881.0	887.0	0.675
O26C7OP3	668.0	698.0	0.020
O27C7OP3	1620.0	1623.0	0.879
O30C10OP3	209.0	218.0	0.001
O30C10OP5	197.0	206.0	0.168
O31C10OP3	743.0	758.0	0.307
O31C10OP5	533.0	548.0	0.481
O32C12	1010.0	1160.0	0.713
O33C12	1803.0	1893.0	0.940
O34C13C12	77.0	122.0	0.030
O35C13C12	278.0	338.0	0.296
O36C4OP4	1253.0	1268.0	0.721
O36C5OP2	1394.0	1409.0	0.846
O36C5OP5	1334.0	1349.0	0.822
O38C12	773.0	833.0	0.609

Table X-2 (f) The results obtained by APSDSS when *t* = Day 1

Items	Starting time (hour)	Finish time (hour)	Random number
O1C4OP3	1743.0	1793.0	0.595
O1C5OP4	311.0	371.0	0.292
O2C4OP3	531.0	606.0	0.312
O2C5OP4	939.0	1029.0	0.583
O3C7OP2	411.0	441.0	0.039
O3C6OP4	286.0	311.0	0.235
O3C9OP2	1624.0	1654.0	0.001
O4C7OP2	873.0	903.0	0.637
O4C6OP4	51.0	76.0	0.016
O4C9OP2	903.0	933.0	0.280
O5C10OP2	1473.0	1548.0	0.559
O6C10OP2	1998.0	2098.0	0.159
O7C4OP3	2098.0	2198.0	0.191
O7C5OP4	76.0	196.0	0.065
O8C4OP3	1548.0	1598.0	0.843
O8C5OP4	1101.0	1161.0	0.086
O9C7OP2	226.0	286.0	0.117
O9C9OP2	606.0	666.0	0.497
O10C7OP2	758.0	848.0	0.426
O10C9OP2	1383.0	1473.0	0.841
O16C9OP2	1293.0	1353.0	0.802
O17C9OP2	1793.0	1883.0	0.952
O18C4OP3	1161.0	1166.0	0.688
O19C4OP3	386.0	411.0	0.327
O20C5OP4	196.0	226.0	0.081
O21C5OP4	1598.0	1610.0	0.438
O22C5OP4	933.0	939.0	0.137
O24C6OP4	1166.0	1201.0	0.746
O25C7OP2	1029.0	1041.0	0.641
O26C7OP2	698.0	758.0	0.541
O27C7OP2	1654.0	1660.0	0.487
O28C9OP2	1201.0	1231.0	0.764
O29C9OP2	441.0	531.0	0.204
O30C10OP2	371.0	386.0	0.310
O31C10OP2	848.0	873.0	0.160
O36C4OP3	1268.0	1293.0	0.189
O36C5OP4	1353.0	1383.0	0.373
O37C9OP2	1041.0	1101.0	0.032