2009

AN ALGORITHM TO SOLVE THE ASSOCIATIVE PARALLEL MACHINE SCHEDULING PROBLEM

Mohannad Abdelrahman Shuaib
University of Kentucky, m.shuaib@uky.edu

ABSTRACT OF THESIS

AN ALGORITHM TO SOLVE THE ASSOCIATIVE PARALLEL MACHINE SCHEDULING PROBLEM

Effective production scheduling is essential for improved performance. Scheduling strategies for various shop configurations and performance criteria have been widely studied. Scheduling in parallel machines (PM) is one among the many scheduling problems that has received considerable attention in the literature. An even more complex scheduling problem arises when there are several PM families and jobs are capable of being processed in more than one such family. This research addresses such a situation, which is defined as an Associative Parallel Machine scheduling (APMS) problem. This research presents the SAPT-II algorithm that solves a highly constrained APMS problem with the objective to minimize average flow time. A case example from a make-to-order industrial product manufacturer is used to illustrate the complexity of the problem and evaluate the effectiveness of the scheduling algorithm.

KEYWORDS: Associative Parallel Machines, Algorithm, Scheduling, Sequence-Depending Setups, Eligibility Constraint.

Mohannad Abdelrahman Shuaib

07/16/2009

AN ALGORITHM TO SOLVE THE ASSOCIATIVE PARALLEL MACHINE SCHEDULING PROBLEM

By

Mohannad Abdelrahman Shuaib

Dr. Fazleena Badurdeen

Director of Thesis

Dr. Dusan Sekulic

Director of Graduate Studies

07/16/2009

RULES FOR THE USE OF THESES

Unpublished theses submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name	Date

THESIS

Mohannad Abdelrahman Shuaib

The Graduate School

University of Kentucky

2009

AN ALGORITHM TO SOLVE THE ASSOCIATIVE PARALLEL MACHINE SCHEDULING PROBLEM

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in Manufacturing
Systems Engineering in the College of Engineering

By

Mohannad Abdelrahman Shuaib

Lexington, Kentucky

Director: Dr. Fazleena Badurdeen, Assistant Professor of Mechanical Engineering

Lexington, Kentucky

2009

Dedicated to my family: Dad, Mom, Nasr, Mohammed, Sara, Nada and Moataz

ACKNOWLEDGEMENTS

I would like to acknowledge and thank everyone for the guidance and assistance in completing my thesis. First, thanks to my advisor, Dr. Fazleena Badurdeen, for providing continuous guidance and support. She inspired me by challenging me to achieve higher standards, providing continuous and valuable feedback, maintaining the highest level of professionalism, and being a role model that I respect. Next, thanks to my committee members, Dr. Ibrahim Jawahir and Dr. Larry Holloway, for the valuable supervision and feedback during my thesis defense. Also, thanks to four anonymous resources for the support in my experimental work. I would also like to extend my deepest gratitude to the rest of the faculty in the Manufacturing Systems Engineering program in the University of Kentucky. Finally, I would also like to thank my family and friends for believing in me, and for always being there for me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Effective production scheduling is essential for improved performance. Meeting customer expectations through on time delivery, better resource utilization and increasing profitability are all dependent upon effectively scheduling the jobs on available resources. Job scheduling is an even greater challenge for companies involved in make-to-order manufacturing, where the product variety is high and the customer demand is dynamic.

Scheduling strategies for various shop configurations and performance criteria have been widely studied. Nevertheless, many companies—particularly small and medium-sized ones—continue to experience difficulty in effectively scheduling jobs. The reason, on the one hand, appears to be difficulties in translating the academic findings into practical applications. On the other hand, as product mixes evolve and additional capacity is added, companies are faced with new and more complex manufacturing environments to which existing scheduling strategies cannot be applied.

Scheduling in parallel machines (PM) is one among the many scheduling problems that has received considerable attention in the literature. Many real-life manufacturing environments can be equated to a PM situation. However, even the PM scheduling (PMS) problem with two machines has been shown to be NP hard (Garey and Johnson 1979). Thus, as the number of machines increase and other constraints are considered, solving this problem to find optimal solutions gets more complicated.

An even more complex scheduling problem arises when there are several PM families and jobs are capable of being processed in more than one such family. This research addresses such a situation, which is defined as an Associative Parallel Machine scheduling (APMS) problem. In this situation, there are several PM families with different production capabilities. While each job can be processed in one of several PM families the processing time can vary depending on which PM family it is assigned. In this context, the scheduling problem involves (1) selecting one of the PM families and a machine in the family to assign every job to as well as (2) determining the processing

sequence on each machine in every family to optimize the desired performance criteria overall. In an APMS environment, some PM families could be capable of processing only a limited set of jobs, while others may be more flexible to process a broader set of jobs. Setup times could also depend on the job sequence on each machine. The APMS problem will be presented in detail in the next chapter.

This research presents the SAPT-II algorithm that solves a highly constrained APMS problem with the objective to minimize average flow time. While APMS problems can be observed in many industry sectors, a case example from a make-to-order industrial product manufacturer is used to illustrate the complexity of the problem and evaluate the effectiveness of the scheduling algorithm.

2 PROBLEM STATEMENT

This chapter presents the APMS problem studied in this research. Section 2.1 presents the APMS problem and provides the problem notation in the manner commonly used to present scheduling problems. Section 2.2 presents the case study that defines the scheduling problem. Details disclosing information about the company targeted by the case study or the description of its products will be withheld due to privacy reasons.

2.1 The Associative Parallel Machine Scheduling Problem

The problem presented in this thesis involves job scheduling in a unique machine configuration, which is a hybrid between Identical and Unrelated Parallel Machines. A new term is introduced to classify this unique configuration: Associative Parallel Machines (A_m), and the scheduling problem becomes Associative Parallel Machine Scheduling Problem (APMS). Figure 2-1 illustrates the configuration in an APMS problem.

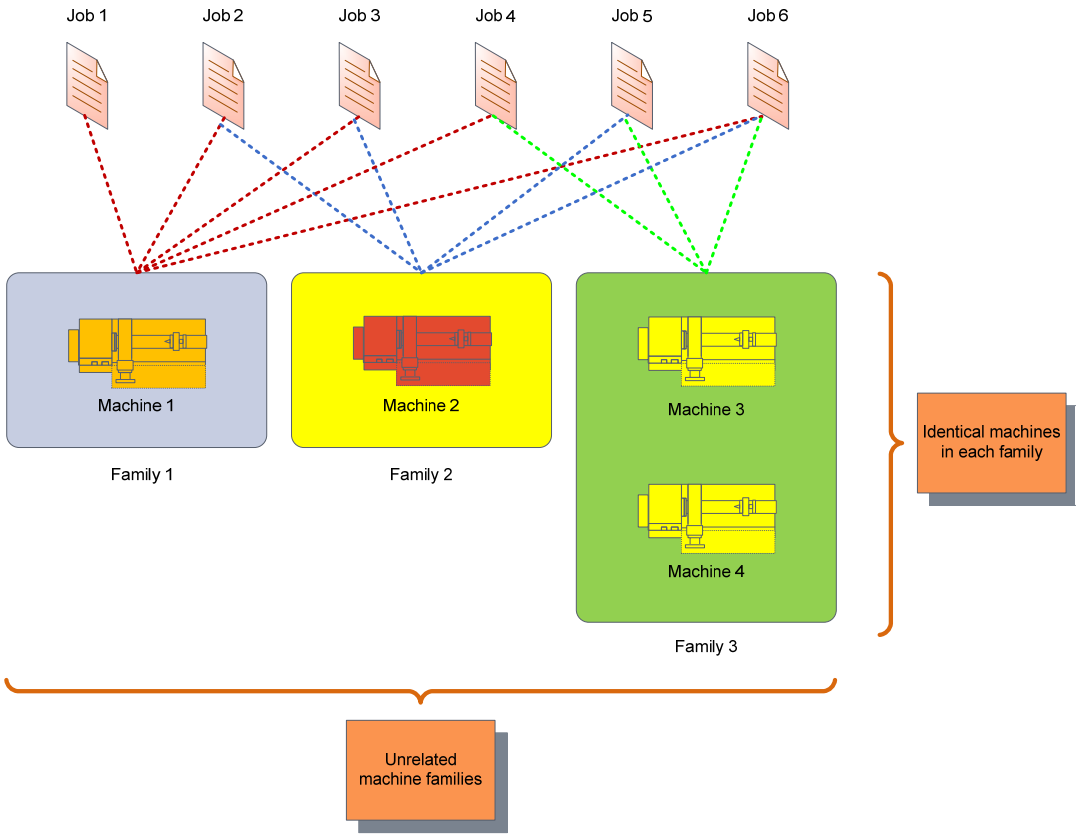


Figure 2-1 Machine Classification in APMS Problem

In the APMS problem, the machines are grouped into families which are denoted as M_1, M_2, …, M_f. Each family M_l has m_l identical machines. The APMS problem presented also has a set of constraints. For each machine, there can be periods of non-availability. This introduces the non-availability constraint, which is denoted by NC_{win}. These windows represent the times that the machines are not available for production depending on shifts worked and/or human resource availability. There are n jobs in each weekly scheduling bucket. Each job represents as single order from the production bucket, and can be processed in one or more families. The eligibility constraint is denoted by the set $E = \{e_{j,l}: j = 1 \rightarrow n, l = 1 \rightarrow f\}$. The processing time for each job j is family-dependent and is denoted by $p_{j,l}$. The setup times are sequence-dependent and machine-independent. When job j is processed after job i the setup time for job j is denoted as $s_{i,j}$.

2.1.1 Research Objective

The flow time for job j is denoted as Fj. For a job j, the flow time (F_j) equals the completion time (C_j) minus the ready time (R_j). Since the ready time for all the jobs is zero, the completion time and flow time are equal as shown in equation 1-1.

$$(Rj = 0; \forall\ j = 1 \rightarrow n); \quad (Fj = Cj) \tag{2-1}$$

The goal in the scheduling problem is to minimize the average flow time (or average completion time) $\sum_{j=1}^{n} C_j/n$. The APMS problem can be presented using the 3-field notation adapted by (Graham et al., 1979) where the problem is described by three fields: $\alpha \mid \beta \mid \gamma$. The first field ($\alpha$) describes the production environment and the machine classification. The second field (β) describes the constraints, production conditions and other details of the production process. The third field (γ) describes the objective of the scheduling problem. Using the three-field notation, the APMS problem is described as

$$A_m, NC_{win} \mid s_{i,j}, R_j=0\ , e_{j,l} \mid \sum_{j=1}^{n} C_j/n \tag{2-2}$$

The notations used in the problem formulation are summarized in Table 2-1.

Table 2-1 Notations in Problem Formulation

Notation	Representation
subscript l	machine family *l*
subscript i	reference to job *i*
subscript j	reference to job *j*
subscript f	number of machines in family *l*
m	total number of machines
n	total number of jobs
A_m	associative parallel machines
NC_{win}	non-availability windows
$s_{i,j}$	sequence-dependent setup time
$p_{j,l}$	processing time for job *j* in machine family *l*
$e_{j,l}$	eligibility of job *j* on family *l*
C_j	completion time for job *j*
F_j	flow time for job *j*
R_j	ready time for job *j*
C_{avg}	Average completion time
C_{max}	Makespan
PT_{tot}	Total processing time
CO_{tot}	Total changeover time
$CO\%$	Percentage of changeover

In addition to the main objective, several performance measures were also tracked to understand their relation with the main objective. The following points list the measures observed and how they are calculated:

- Makespan; the time between the start and finish of the production bucket

$$\boldsymbol{C_{max} = \max_{1 \le j \le n} C_j} \qquad 2\text{-}3$$

- Total Processing Time

$$\boldsymbol{PT_{tot}} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{C_j} \qquad 2\text{-}4$$

- Total Changeover Time; the sum of all the changeover times

$$\boldsymbol{CO_{tot}} = \sum \boldsymbol{s_{ij}} \quad \forall\, \boldsymbol{i} = \boldsymbol{1} \rightarrow \boldsymbol{n}, \boldsymbol{j} = \boldsymbol{1} \rightarrow \boldsymbol{n} : \boldsymbol{jobs\ i}\ \&\ j\ are\ sequential \qquad 2\text{-}5$$

- Changeover %

$$\boldsymbol{CO}\% = {}^{\boldsymbol{CO_{tot}}}/_{\boldsymbol{PT_{tot}}} \qquad 2\text{-}6$$

2.1.2 Assumptions

Several assumptions are made in this research to solve the APMS problem. These are:

- All jobs are available at time *t = 0*
- Preemption is not allowed; once a job is loaded on a machine, it cannot be removed until it is complete
- Job splitting is not allowed; if a job consists of multiple units, all the units go on the same machine
- Combining different jobs with identical unit specifications is not allowed
- There are no unplanned windows of non-availability (e.g. breakdowns, unplanned maintenance, crew absence, raw material shortage)

2.2 Case Study

The scheduling problem presented in this thesis relates to a real-life scheduling problem at a company that produces electrical equipment. To maintain anonymity, the company

will be referred to as the **Electric Company**. The production environment for this company is best described as a mass customization environment. The products are highly customized and are made-to-order. Once the customized order is received by marketing and released for production, each product passes through seven production stages. The process flow is illustrated in Figure 2-2. Though the orders are released to all the Fabrication Departments simultaneously, the operations in Fab 1 (see Figure 2-2) are the most time consuming with large lead times. The machine configuration in this department is quite unique (as will be discussed later), leading to scheduling complexities. As a result, this department has become the bottleneck, delaying operations in Fab 4 and leading to high Work-In-Progress (WIP) build-up before that. Therefore, scheduling jobs in the Fab 1 Department is the focus of this research.

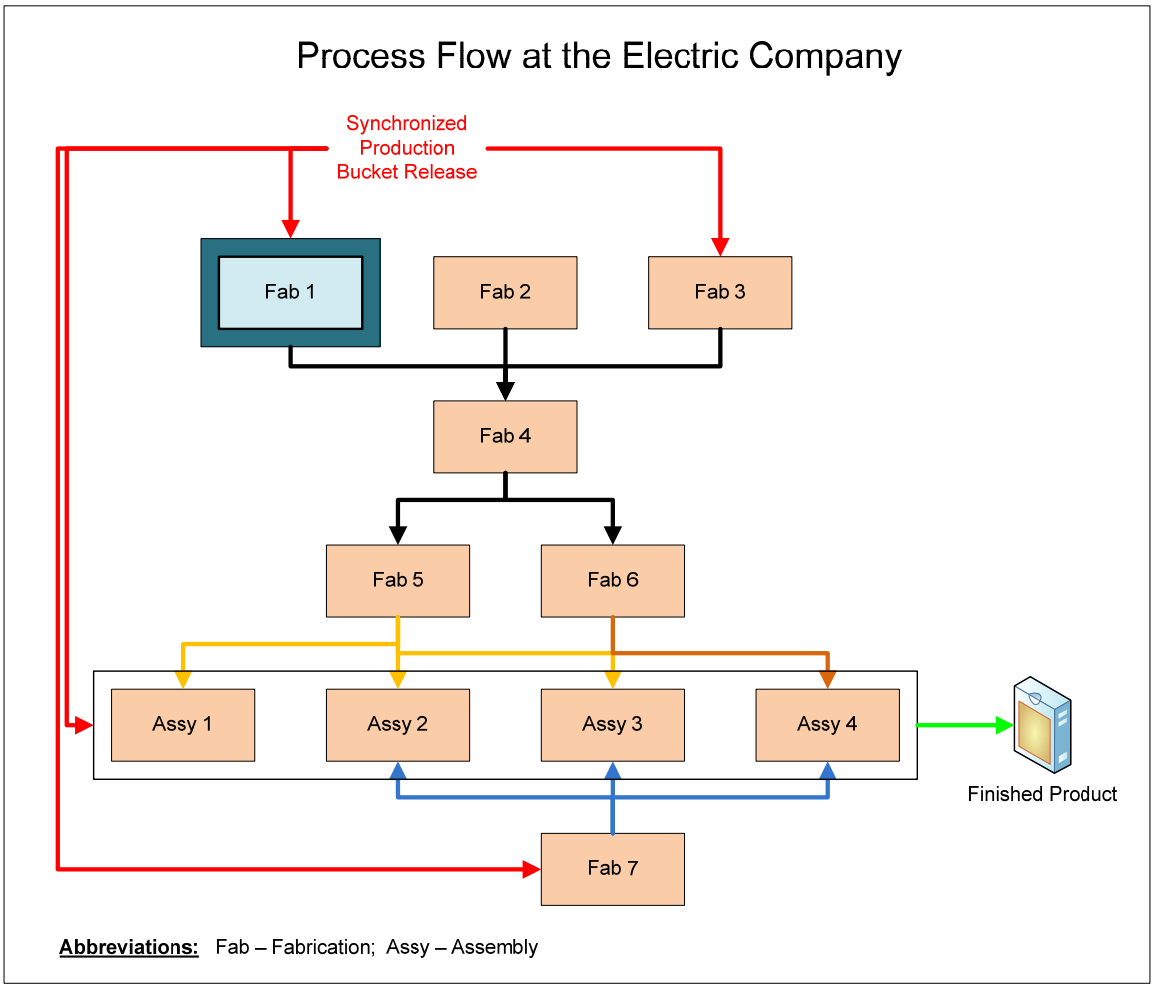


Figure 2-2 Process Flow Chart

Machine Setup

The Fab-1 Department has a single-pass production system. All products are processed similarly, but there is high variety in the specifications. To accommodate this requirement there are a large number of machines in the department. There are several machines of each type (i.e. parallel) which are grouped into families. Any two machines from the same family are identical, while two machines from different families are unrelated.

The department operates 2 shifts a day, 5 days a week. Each machine is operated by one person. The machines are not fully crewed; some machines have two operators assigned

and run both shifts every day, while others have only one operator assigned and run only one shift every day.

Product Description

Each order consists of a number of identical units ordered by a single customer. The structure of the units is similar. The major difference is in the size of each unit, which is dependent on the unit specifications. All the machines have an identical manufacturing process. The different between machines from different families is in the number of stations; the number of stations in a machine equals the number of units that machine can process simultaneously in each production run. For example, 3-station machines can process 3 units simultaneously in each production run. If a job consisting of 8 units is processed on a machine from that family, the sequence of production would be: produce 3 units – unload – produce 3 units – unload – produce 2 units. As mentioned above, the specifications of the singular unit will determine the compatible machine families, i.e., the number of machine families eligible to manufacture each order. In most cases, each order is compatible with more than one machine family. Processing time is generally a function of the machine family (i.e. machine family capability), since the production sequence for each job changes when the number of stations in the machine chage; time taken to complete an order will depend on the machine family it is assigned to. There are no restrictions on sequencing two orders on a machine that they are both compatible with. While the setup times are machine-independent, they are sequence-dependent; the greater the difference in specifications between the two successive orders, the greater the setup time.

Planning and Scheduling

Production is managed on a make-to-order basis. The planning and scheduling cycle starts at the marketing department. Customers initiate the process by contacting the marketing team. Then, with assistance from the engineering team, product specifications are defined and orders are placed. Marketing then divides all the existing orders to

weekly production buckets, based on a first-come first-served basis. These weekly order buckets are transformed to weekly production buckets and passed to each production department. The weekly schedules are subsequently created independently by each department based on the available capacity.

Current Situation

In the focal department, Fab 1, currently the management uses an ad-hoc approach to scheduling by manually assigning jobs to the machines as they become available. There is no systematic approach to scheduling, and jobs are picked arbitrarily from the production bucket. The poor balance in scheduling leads to a decrease in the overall production efficiency. Random sequencing leads to increased setup times further decreasing efficiency. Also, there is no special consideration in scheduling for orders with higher eligibility constraints, that is, orders that can be processed on a fewer number of families. When only these jobs are remaining in the planning bucket, some machines might become idle since there are no compatible jobs remaining. As a result, the efficiency of the idle machines drops drastically.

A major drawback of using ad-hoc scheduling in Fab-1 department is excessive delays and waiting in downstream operations. Furthermore, there is no coordination between production departments. Since the production buckets cover a whole week of production, a component may be manufactured in one department in the beginning of the week and lay as WIP until the matching components are produced in other departments at the end of the week before moving to downstream assembly operations.

Objective

The goal at the Electric Company is to improve production scheduling at the Fab-1 department. The management expects that once scheduling is improved, the department will no longer be a bottleneck in the production operations. Better assignment and sequencing will lead to improving machine efficiencies and utilization. Improvement in

scheduling will be measured by monitoring several performance measures including the average completion time, makespan, total processing time and changeover time. The objective function to this specific case of APMS problem will be to minimize the average flow time. Another objective is to develop a scheduling tool that generates a schedule that can be visually managed. The schedule will be used as a pacer to coordinate the production operations in other departments. As a result, the inventory levels are expected to drop in all production departments. The schedule will also be used as a daily management tool to track and improve resource efficiencies.

3 LITERATURE REVIEW

The systematic approach to scheduling production operations started in the 1950's (Allahverdi, et al. 2008). Since then, much research has been aimed at developing practical and efficient optimization methods for different scheduling problems. Most of the research focused on real life problems. Therefore, these problems came in a wide variety depending on the production environment (or machine configuration), production system conditions, characteristics of the production system and the objective to be optimized. The focus in this literature review will be on scheduling problems focused on scheduling problems for one machine configuration: Parallel Machine Scheduling (PMS).

The PMS problem belongs to the wide class of Combinational Optimization problems. Many of these problems are known to be NP-hard. This means that it is hard to find efficient optimization solutions for these problems (Mokotoff 2001). Because PMS problems are practical applications which are NP-hard, researchers were led to develop more efficient solution methods. For general scheduling problems, the most common methods used are branch-and-bound algorithms, mathematical programming formulations, dynamic programming algorithms, heuristics and meta-heuristics (Allahverdi, et al. 2008). In the special case of PMS problems, little research has been aimed at at developing optimization methods. Instead, most research focused on heuristics (Zhu and Wilhem 2006).

The focus in the literature research was structured as follows. First, for the production environment, the focus was on PMS problems, with an emphasis on unrelated parallel machines. In terms of limitations and constraints, the focus was also on problems with sequence-dependent setup times, machine availability constraints and compatibility constraints. Finally, in terms of the objectives studied, the research was limited to methods that minimize flow time and makespan. The next section, surveys on various scheduling problems are presented, grouped based on the production environment studied; the constraints and objectives studied are discussed within each section.

3.1 Surveys on PMS Problem Research

There are several surveys that connect different research done to solve diverse PMS problems. Lam and Xing (1997) studied the new trends in PMS affected by JIT (Just-in-time), with pre-emption, set-up times and capacitated machines. Allahaverdia, et al. (1999), Allahaverdi, et al. (2008) and Zhu and Wilhem (2006) discussed methods used to solve scheduling problems with sequence-dependent setup times or costs applied to single machine, parallel machine, flow-shop and job shop systems. Mokotoff (2001) presented a survey of research done to study the Identical, Uniform and Unrelated PMS problems with an emphasis on makespan minimization. He also discussed the NP-hardness of the PMS problem. Leung and Li (2008) studied scheduling problems with processing set restrictions. They studied Identical, Parallel and Unrelated PMS problems and their emphasis was on polynomial-time algorithms, complexity issues, and approximation schemes. They mainly looked at problems with a goal to minimize the makespan in addition to discussing other performance criteria. Li and Yang (2009) surveyed the research done on non-identical parallel-machine scheduling problems with the objective to minimize total (or mean) weighted (or un-weighted) completion time. They collected and classified mathematical models and relaxations. They also presented the heuristics and optimization techniques used to solve the scheduling problems surveyed.

3.2 Identical PMS Problem

Considerable research has been done to address the Identical PMS problem. Gao, et al. (1998) solved the Identical PMS problem to minimize total weighted tardiness using a four-step algorithm. The solution was applied by an electrical appliance company. Kurz, and Askin (2001) solved the $P_m|r_i, s_{i,j}|C_{max}$ scheduling problem. They developed four heuristics. In their experimentation, they examined a scheduling problem with 100 jobs to be scheduled on 10 identical machines. They also developed a Mixed Integer Linear Programming (MILP) model incorporating the $P_m| |C_{max}$ scheduling problem. Mendes, et al. (2002) also attempt to solve the Identical PMS problem with sequence-dependent

setup times and the goal to minimize the makespan by proposing and comparing the performance of two meta-heuristic methods. The first is a tabu search-based heuristic and the second is a memetic approach, which combines a population-based method with local search methods. Yalaoui and Chu (2003) discussed the $P_m|S_{ij}|C_{max}$ with job splitting scheduling problem. They applied a heuristic to decompose the scheduling problem into independent one-machine problems and then used a step-by-step improvement methodology. They were able to determine a lower bound for the solution and demonstrate that the heuristic method shows a good performance in comparison to the lower bound solution. Chang, et al. (2005) applied the Two-phase Sub Population Genetic Algorithm (TSPGA) to solve the multi-objective Identical PMS problem. They compared the solution to the Non-dominated Sorting Genetic Algorithm (NSGA2) by Deb, et al. (2002) and the Multi Objective Genetic Algorithm (MOGA) by Muruta and Ishibuchi (1996). Nessah, et al. (2007) addressed an identical parallel machine scheduling problem, with sequence-dependent setup times and release dates to minimize total completion time. They developed a branch-and-bound algorithm that can handle 40 jobs on 2 machines.

3.3 Uniform PMS problems

Lin and Liao (2008) studied the Uniform PMS problem with the goal to minimize the makespan by proposing an optimal algorithm that transforms the scheduling problem to an identical machine problem and applying an algorithm to solve the new single machine scheduling problem. Although the computational time for their algorithm is exponential, it is still efficient for various sizes of scheduling problems. Their results are compared to the LPT heuristic.

3.4 Unrelated PMS problems

Some of the research addressed the unrelated PMS problem. Piersma and Van Dijk (1996) studied the Unrelated PMS problem with the objective to minimize makespan. They proposed and compared the performance of new local search algorithms. Suresh

and Chaudhuri (1996) studied the Unrelated PMS problem on two machines while considering machine availability and interruptions. They developed algorithms to solve the problem in two cases: when the machine availability is known, and when the machine availability is not known. Suresh and Chaudhuri (1996) solved the Unrelated PMS problem with no preemption. The objectives are to minimize the makespan and minimize the maximum tardiness. They proposed an algorithm based on tabu search and compared its performance to a heuristic they have proposed earlier. Weng, et al. (2001) addressed the Unrelated PMS problem with sequence-dependent setup times and the objective of minimizing the weighted mean completion time. They proposed and compared the performance of seven heuristic algorithms. Mokotoff and Chretienne (2002) developed an exact algorithm and an approximate algorithm to solve the $R||C_{max}$ problem. Yu, et al. (2002) tried to eliminate the bottleneck operation in a Printed Wiring Board manufacturing line. The problem was defined as an Unrelated PMS problem with negligible setup times and multiple performance measures including: makespan, average finish time, mean flow time, utilization, number of lots and total amount of overtime beyond the release interval. They proposed a heuristic to solve the problem and compared its performance with a network model and a modified FIFO method. Kim, et al. (2002) solved the Unrelated PMS problem with sequence-dependent setup times using simulated annealing. Arnaout and Rabadi (2005) addressed the batch scheduling problem in Unrelated Parallel Machines with the objective of minimizing the weighted mean completion time. They developed a solution heuristic and compare it to three other algorithms, including Weng's Algorithm 7 in Weng, et al. (2001). Rabadi, et al. (2006) addressed the $R_m|S_{ijk}|C_{max}$ scheduling problem. They applied the Meta-heuristic for Randomized Priority Search (Meta-RaPS) to minimize the makespan. Their results were compared to the Partitioning Heuristic by Al-Salem (2004). He and Hui (2007) presented a genetic algorithm for the single-stage multi-product scheduling problem (SMSP). Their research looked at a high-constrained large-size SMSP. They found that it is difficult for MILP to obtain acceptable solutions to the large-size problems within reasonable time. To solve the problem, they proposed a Genetic Algorithm (GA) based on heuristic rules.

3.5 Summary

In the majority of the literature studied, researchers used algorithms and heuristic methods to solve various scheduling problems. Some of the heuristic methods and algorithms were based on simpler, more trivial scheduling algorithms like the SPT and LPT rules. Another observation was that many researchers presented their scheduling problems with assumptions or relaxations. By doing so, they were able to present solution methods that were more efficient in computation time. Also, most research tackled one or two constraints at a time. In real-life scheduling problems – including the scheduling problem addressed in this thesis – the production environment is complicated and there are many constraints and limitations to the production and scheduling environment. Few researchers have experimented with large size, highly constrained PMS problems.

The Meta-heuristic for Randomized Priority Search (Meta-RaPS) applied by Rabadi, et al. (2006) was a base to the work done to solve the scheduling problem in hand. There were two motives. First, the scheduling problem in hand and the problem discussed in their research have a similar constraint; sequence dependent setup times. In their approach, they sum up the processing and setup times to produce 'adjusted processing times' as an input to the scheduling problem. Second, the scheduling problem in hand has the objective of minimizing the average flow time. Accordingly, it is expected that heuristics that are based on SPT rules would perform better. The Meta-RaPS heuristic proposed schedules the jobs based on combining the sequence-dependent setup time and the processing time and picking the job with the shortest combined time.

4 METHODOLOGY

In this chapter, the methodology to solve the APMS problem is presented. The objective of the research is to,

1. Solve the presented APMS problem.
2. Develop an automated scheduling tool to be applied at the Electric Company.

Section 4.1 presents the methods used to solve the APMS problem. Section 4.2 presents the method used to develop the automated scheduling tool.

4.1 Solving the APMS Problem

There are two approaches to solve scheduling problems. The first is to use mathematical modeling to find the optimum solution. However, solving mathematical models for complex (NP-hard) scheduling problems can be time consuming. In such situations, the use of heuristics would be the alternative to follow. Though such algorithms cannot find the optimal solution, an effective algorithm that can find a near-optimal solution in a reasonable time is a more practical choice.

4.1.1 Mathematical Modeling

A Mixed Integer Programming (MIP) model is developed to optimally solve the APMS problem. The mathematical model includes the eligibility and sequence-dependent setup constraints in the APMS problem. However, the model was relaxed by excluding the non-availability constraint because it would complicate the mathematical model. The goal was to establish the NP-hardness of the relaxed mathematical model to confirm the NP-hardness of the complete APMS problem. The MIP formulation is presented in the following formula set. The model will be applied to actual production data from the Fab-1 department.

4.1.1.1 MIP Model

- Objective function:

$$\text{Minimize } \sum_{l=1}^{F} \sum_{m=1}^{M_l} \sum_{k=1}^{N} c(l, m, k)/N \quad 4\text{-}1$$

- Subject to:

$$\mathbf{x(l,m,j,k) \le e(j,l)} \quad \mathbf{\forall l = 1 \to F, \forall m = 1 \to M_l, \forall j = 1 \to N, \forall k = 1 \to N} \quad 4\text{-}2$$

$$\mathbf{\sum_{j=0}^{N} x(l,m,j,k) \le 1} \quad \mathbf{\forall l = 1 \to F, \forall m = 1 \to M_l, \forall k = 0 \to N} \quad 4\text{-}3$$

$$\mathbf{\sum_{l=1}^{F} \sum_{m=1}^{M_l} \sum_{k=1}^{N} x(l,m,j,k) = 1} \quad \mathbf{\forall j = 1 \to N} \quad 4\text{-}4$$

$$\mathbf{\sum_{j=0}^{N} x(l,m,j,k) \le \sum_{j=0}^{N} x(l,m,j,k-1)} \; \mathbf{\forall l = 1 \to F, \forall m = 1 \to M_l, \forall k = 1 \to N} \quad 4\text{-}5$$

$$\mathbf{c(l,m,0) = 0} \quad \mathbf{\forall l = 1 \to F, \forall m = 1 \to M_l} \quad 4\text{-}6$$

$$\mathbf{c(l,m,k) \ge 0} \quad \mathbf{\forall l = 1 \to F, \forall m = 1 \to M_l, \forall k = 0 \to N} \quad 4\text{-}7$$

$$\mathbf{c(l,m,k) \ge c(l,m,k-1) + p(j,l)\, x(l,m,j,k) + \sum_{i=1}^{N} x(l,m,i,k-1)\, s(i,j) +}$$

$$\mathbf{A[x(l,m,j,k) - 1]} \quad \mathbf{\forall l = 1 \to F,} \quad \mathbf{\forall m = 1 \to M_l, \forall j = 1 \to N, \forall k = 1 \to N} \quad 4\text{-}8$$

$$\mathbf{x(l,m,j,k) \in \{0,1\}} \quad \mathbf{\forall l = 1 \to F, \forall m = 1 \to M_l, \forall j = 1 \to N, \forall k = 1 \to N} \quad 4\text{-}9$$

$$\mathbf{x(l,m,0,0) = 1} \quad \mathbf{\forall l = 1 \to F, \forall m = 1 \to M_l} \quad 4\text{-}10$$

The representations of the mathematical notations used in the formula set are presented in Table 4-1.

Table 4-1 Notations in Mathematical Model

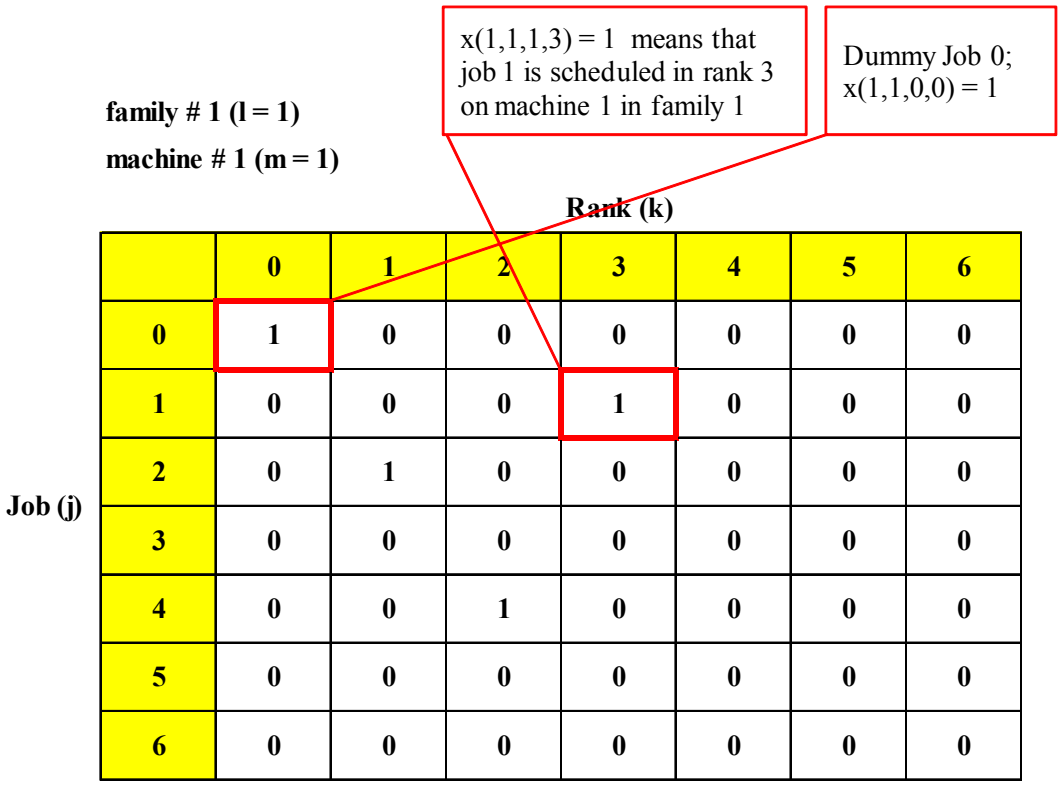
Notation	Representation	Category
i,j	Refer to jobs	Subscript, used for reference
k	Refers to the rank (the order in which a job is processed on a machine)	Subscript, used for reference
l	Refers to machine family	Subscript, used for reference
m	Refers to machines	
A	Very large integer	User input
N	Number of jobs	User input
F	Number of families	User input
M_l	Number of machines in family l	User input
$x(l,m,j,k)$	Assignment of a job = { 1 if job j is scheduled on machine m from family l in rank k 0 otherwise	Decision variable
$c(l,m,k)$	Completion time for job the job scheduled on machine m from family l in rank k	Decision variable, objective
$s(i,j)$	Setup time for job j when it is scheduled after job i	User Input
$e(j,l)$	Eligibility of job j on family l = { 1 if job j is eligible for scheduling on family l 0 otherwise	User input
$p(j,l)$	Processing time for job j in family l	User input

Each formula represents a constraint that the MIP solver must follow when generating the optimum solution. The following explains the objective of each formula.

- Formula 1-1 states the objective to be minimized; the average completion time (equal to the average flow time).
- Formula 1-2 enforces the eligibility constraint, ensuring that each job *j* can be scheduled on family *f* **only if** it is eligible to run on that family.
- Formula 1-3 ensures that a maximum of one job is scheduled in each rank for all the machines.
- Formula 1-4 ensures that each job is scheduled only once.
- Formula 1-5 ensures that each job is scheduled in the rank immediately following the rank of the preceding job.
- Formula 1-6 sets the completion of the dummy job $j = 0$ to zero.
- Formula 1-7 ensures that completion times are non-negative.
- Formula 1-8 is used to calculate the completion times. Here, the completion time for a job j is compared to the completion time of every other job plus its processing time and setup time. If the job compared was not actually scheduled before job j, its completion time is multiplied by a large negative number making the right side of the inequality a negative number. For the job that was scheduled before job j, its completion time is multiplied by 1.
- Formula 1-9 ensures that *x* is binary.
- Formula 1-10 introduces dummy job $j = 0$ that is scheduled in rank $k = 0$ on each machine. The job scheduled in rank $k = 1$ follows the dummy job.

The outcome of the model is a schedule matrix for each machine as shown in Table 4-2.

Table 4-2 MIP Model Schedule Matrix



	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0
2	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0

4.1.1.2 Problem Sets

The MIP model was applied to several data sets varying in the number of families, machines and jobs. A limitation to the size of the problem sets applied in mathematical modeling was caused by the NP-hardness of the APMS problem. To handle within practical processing times, the problem sets had to be much smaller compared to the actual production environment. The experimentation conducted with these data sets and results obtained are explained in Chapters 5 and 6.

4.1.2 Scheduling Algorithms

Another approach to solve the APMS problem is to apply scheduling algorithms. Such algorithms are often more efficient than mathematical models in finding a reasonable

solution with low computational time. The main steps are applicable to any APMS problem, but the specifics mentioned address the specific case study at the Electric Company.

The following algorithms were applied in the experimentation:

- *Shortest Processing Time (SPT)* – selected because SPT minimizes the average lead time in less complicated scheduling problems.
- *Longest Processing Time (LPT)* – selected because LPT results in a better balanced load, which is crucial to prevent overloading one or more families due to poor assignment with the eligibility constraint in effect.
- *Longest Processing Time by Family (LPT*-F) – a special adaptation from the LPT customized to achieve a better load balance.
- *Shortest Changeover (or setup) Time (Min CO)* – selected because setup time is a significant part of the APT, and varies depending on the sequence of assignment.
- *Shortest Adjusted Processing Time (SAPT)* – new algorithm introduced to solve the APMS problem.
- *Shortest Adjusted Processing Time, Improved Version (SAPT-II)* – based on the SAPT algorithm, improved after the second phase of experimentation.

The actual execution of the bucket used in the problem set is also presented to verify the assumptions made in the experiments. A different combination of these algorithms was used during the different phases of experimentation. Figure 4-1 shows the combination at each experimentation phase.

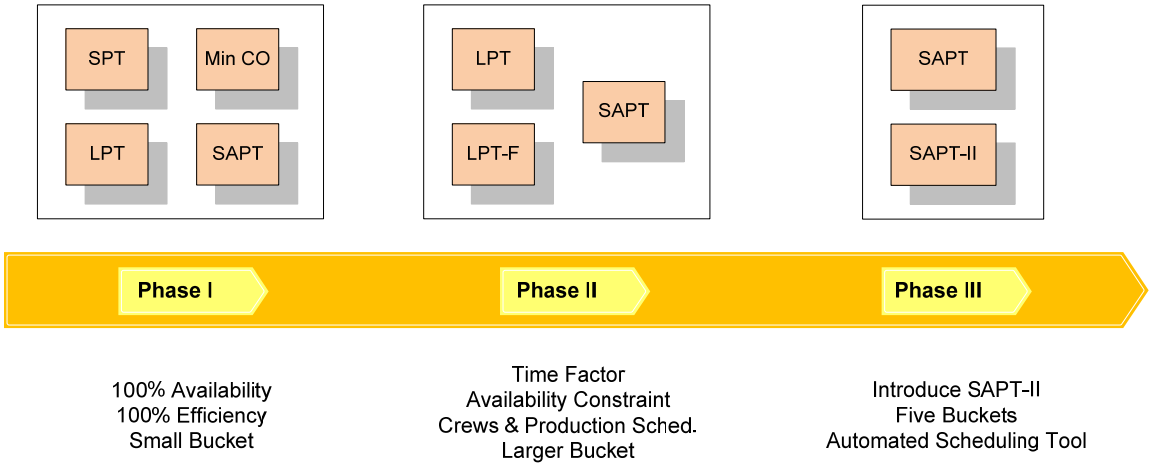


Figure 4-1 Three phases of experimentation

Figure 4-2 summarizes the method of scheduling using the algorithms mentioned above. The steps to apply each algorithm will be presented in Chapter 5. The Data sheet, Sequence-Dependent Setup Matrix, Ranking Tables and APT Matrix contain specially formatted data containing information about job specification, processing times, setup times and eligibility. The format and content of these elements will also be presented in the Appendix.

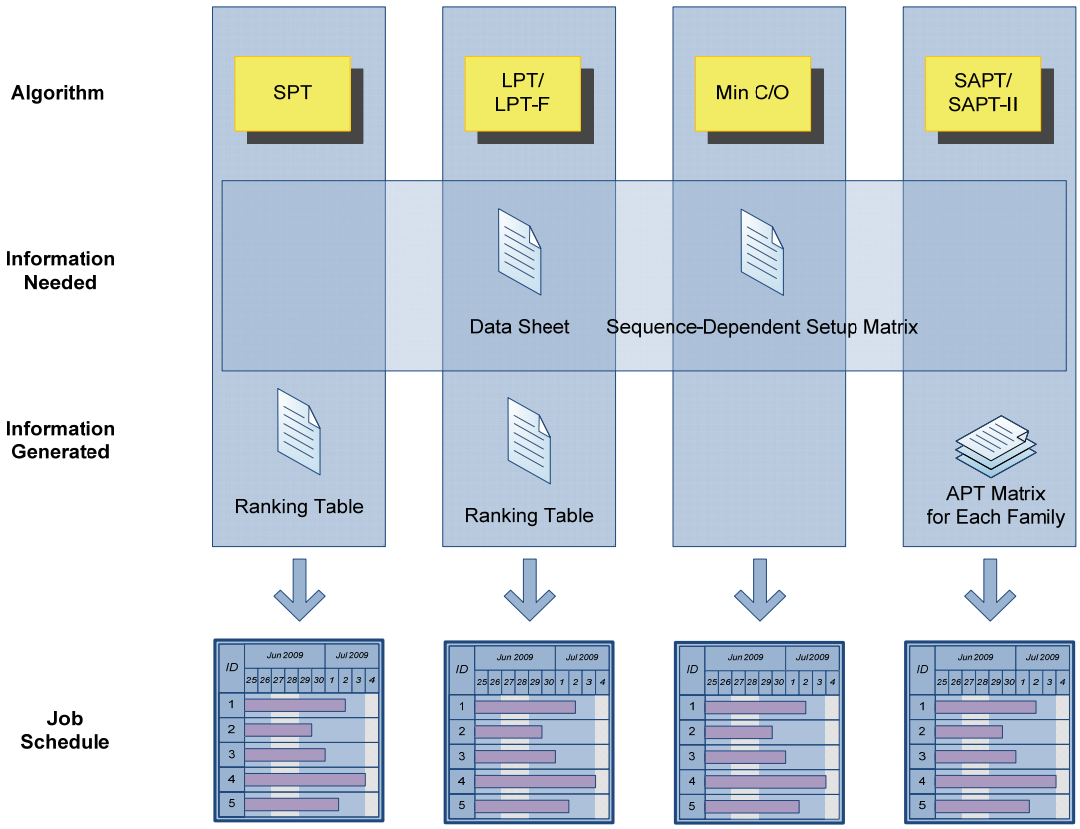


Figure 4-2 Scheduling with Different Algorithms

4.2 Automated Scheduling Tool

Since the research addresses a real industrial problem, the application part was a key deliverable. The goal was to develop a scheduling tool in Visual Basic 2005 Express that interfaces with data files and automatically generates production schedules. The tool must be user-friendly, and able to generate an output that can easily be visually controlled.

5 EXPERIMENTATION DESIGN

This chapter presents the details of the methods followed during experimentation work. Section 5.1 shows how the MIP model was applied, and section 5.2 shows how the scheduling algorithms were applied to solve the APMS problem.

5.1 MIP Model

The first step in the experimentation was to apply the MIP model to optimally solve the APMS problem. The trials were done on different data sets with different dimensions (i.e. different number of families, machines and jobs) in each set. The processing times and sequence-dependent setup times were randomly generated and followed a uniform distribution. The eligibility matrices were also randomly generated, with each machine capable of processing 70-80% of the jobs. Five different sizes of the APMS problem with one or more trials for each size were studied, as shown in Table 5-1 Dimensions of APMS Problem Sets and Figure 5-1.

Table 5-1 Dimensions of APMS Problem Sets

	Case 1	Case 2	Case 3	Case 4	Case 5
# of Families	2	2	2	3	4
# of Machines	2	3	4	4	6
# of Jobs	6	9	12	12	18
# of Trials	3	3	1	1	1

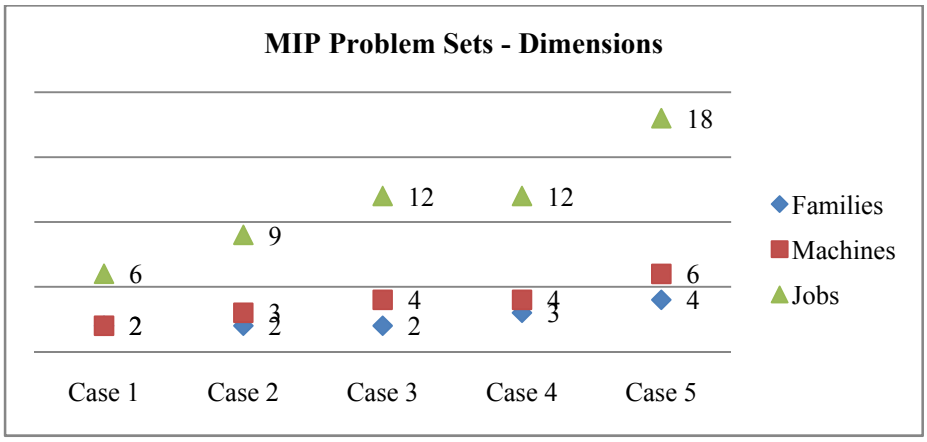


Figure 5-1 Dimensions of APMS Problem Sets

5.2 Scheduling Algorithms

Here, the details of the method followed to apply different scheduling algorithms to the APMS problem are explained. Each section presents a step in the process.

5.2.1 Identifying Resources

In this step, the number of families and the number of machines in each family is defined. Also, the number of crews staffing each machine is specified. Carefully defining the available resources is key to simulate an environment similar to the actual operation environment.

Another attribute that is critical in the APMS problem in hand is the concept of *flexibility of machine families*. In this problem, jobs can be processed in multiple families. The families that can process a higher number of jobs in the production bucket are considered more flexible. In contrast, the families that can process a lower number of jobs are considered less flexible. Family flexibility is used in the scheduling algorithms presented

to determine machine assignment. Table 5-2 below shows how flexibility is determined for 3 families and 6 jobs.

Table 5-2 Flexibility

	Family 1	Family 2	Family 3
Job	Compatibility		
1	x	x	x
2		x	x
3	x		x
4			x
5	x	x	x
6	x		x
	Flexibility		
# of Comp. Jobs	4	3	6
Flexibility	Medium	Lowest	Highest

5.2.2 Order Set

The order set is an input of raw information about the bucket (batch) of jobs to be scheduled. It includes information about each job including:

- Each job's specifications and standard unit processing times
- The number of units in each job

5.2.3 Data Set

The order set is the input for the data generation process. The outcome of this step is information about:

- Eligibility of each job on each family
- Sequence-dependent setup times for any possible job sequence
- Processing time for each job on each family

In the specific case of the Electric Company; based on the nature of the operations setup, the eligibility of each job on each machine family is determined based on each job's unit specifications. The rules determining the eligibility are presented in Table 5-3. Then the processing time for each job on all the compatible PM families is computed. Processing time of an order depends on the processing time for a single unit on each family, the number of stations in each family and the number of units in that order. The points below summarize the method to calculate the processing time for each job on all families. Table 5-4 shows a sample of one job's processing times on different families.

- Processing time per unit on each family is recorded from the order specifications sheet.
- # of full runs = round (order qty / # of stations), rounded to the smaller integer. Example: order qty = 8, running on machine with 5 stations. # of full runs = round (8/5) = 1
- For specific orders: on machines with 3 stations – 1 winding per run; on machines with 5 stations – 2 windings per run
- # in final run = Mod (order qty / # of stations); the remainder in the division
- Time for full runs = # of full runs * # of windings in full run * unit processing time
- Time for last partial run = # of windings in final partial run * unit processing time
- Time for machine reloading = (# of full runs + # of partial runs [1 or 0] – 1) * 10 / 60
- Total run time = time for full runs + time for last partial run + time for reloading

Table 5-3 Determining Eligibility*

Family	Spec 1	Spec 2			Spec 3	Spec 4	Spec 5	Spec 6			Spec 7	
		1	2	3				1	2	3	1	2
1	≥ 167			x	≥ 22.51	≥ 12.0	A - #6 or #7, B - all	x	x	x		x
2	≤ 100	x	x		≤ 22.50	9.0 - 12.0	#8 or smaller	x	x		x	
3	≤ 100	x	x		≤ 20.00	7.0 - 9.0	#8 or smaller	x	x		x	
4	≤ 100	x	x		≤ 20.00	7.0 - 9.0	#8 or smaller		x		x	
5	≤ 100	x			≤ 17.00	≤ 7.0	#8 or smaller		x		x	

* To maintain anonymity, the units of measurement are not shown in the table

Table 5-4 Processing Times

Order #	Units (#)	Fam 1	Fam 2	Fam 3	Fam 4	Fam 5
1	6	8 h 9 m	5 h 6 m	5 h 7 m	5 h 7 m	4 h 30 m
2	5	6 h 34 m	3 h 29 m	3 h 3 m	3 h 3 m	3 h 3 m
3	12	9 h 31 m	5 h 24 m	5 h 2 m	5 h 2 m	4 h 41 m

Standard setup times depend on the number of specification changes (therefore, setup changes) between any two sequential jobs. Based on the difference in specifications between each two sequential jobs, one or more of three components must be changed during the changeover. The number of components changed determines the sequence-dependent setup time. The elements in the changeover matrix are denoted by s_{ij}, where i represents the preceding job and j represents the succeeding job, and $i \neq j$. For the SAPT algorithm, the two sets of data are then combined to generate adjusted processing time

(APT) (= setup time + processing time) matrices for each PM family. A separate APT matrix for each PM family is necessary due to the variation in processing time for jobs on each family. The elements in the APT matrix are denoted by APT_{ij}, where i represents the preceding job and j represents the succeeding job, and $i \neq j$. APT_{jj} represents the processing time for job j when it is first in the sequence of jobs scheduled on a machine.

Table 5-5 Determining Sequence-Dependent Changeover Time

Component 1	Component 2	Component 3	C/O Time
x			**10**
x	x		**15**
x		x	**45**
x	x	x	**60**

The process described above is demonstrated in the following tables in an example with 6 jobs that can be processed in one or more of three families.

The job compatibility with each family and the corresponding processing times in those families is presented in Table 5-6. Table 5-7 shows the sequence-dependent setup times. Rows represent the preceding job, and columns represent the succeeding job. Table 5-8 shows the APT matrix for Family 1. Again, rows represent the preceding job, and columns represent the succeeding job.

Table 5-6 Compatibility and Processing Times for Sample Data Set

	Compatibility			Processing Time (hrs)		
Job	Family 1	Family 2	Family 3	Family 1	Family 2	Family 3
1	x	x	x	2	3	4
2		x	x	-	5	5
3	x		x	5	-	4
4			x	-	-	2
5	x	x		5	5	-
6	x			5	-	-

Table 5-7 Sample Sequence-Dependent Setup Time Matrix

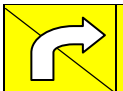
	**1**	**2**	**3**	**4**	**5**	**6**
1	-	0.5	0.25	0.25	1	0.75
2	0.5	-	0.25	0.25	1	0.75
3	0.25	0.25	-	0.25	1	1
4	0.25	0.25	0.25	-	1	1
5	1	1	1	1	-	0.25
6	0.75	0.75	1	1	0.25	-

Table 5-8 Sample APT Matrix for Family 1

	1	2	3	4	5	6
1	-	-	5.25	-	6	5.75
2	-	-	-	-	-	-
3	2.25	-	-	-	6	6
4	-	-	-	-	-	-
5	3	-	6	-	-	5.25
6	2.75	-	6	-	5.25	-

5.2.4 Scheduling Algorithms

Applying the scheduling algorithms involves (1) selecting one of the PM families and a machine in that family to assign each job to as well as (2) determining the processing sequence on each machine in every family. The outcome is presented in a Gantt chart showing the assignment and sequence of the jobs in the available resources. Once the schedule is produced, the performance measures are calculated and the solution set becomes ready for analysis.

The following is an explanation of each algorithm.

SPT

1. Starting with the data sheet, find the minimum processing time for each job.

2. Highlight the families with processing time equal to the minimum processing time for each job.
3. Rank the jobs in increasing order of minimum processing time.
4. All the machines are flagged as available
5. Begin scheduling process by picking the machine from the pool of available machines with the minimum load.
6. If there is a tie between two or more machines, the tie is broken by selecting the machine from the least flexible family. If the machines are from the same family, the tie is broken arbitrarily.
7. Next, pick the job to be scheduled on the selected machine by looking at all the compatible jobs highlighted in step 2 and picking the job with the lowest rank (lowest processing time).
8. If all the highlighted jobs have been scheduled, pick the compatible job with the lowest rank.
9. If there are zero compatible jobs remaining, the selected machine is flagged as full, and removed from the pool of available machines.
10. Schedule the selected job on the selected machine. Setup time depends on the preceding job and is retrieved from the sequence-dependent setup matrix.
11. Repeat steps 5 to 10 until all the machines are full.

LPT

1. Starting with the data sheet, find the minimum processing time for each job.
2. Highlight the families with processing time equal to the minimum processing time for each job.
3. Rank the jobs in decreasing order of minimum processing time.
4. All the machines are flagged as available
5. Begin scheduling process by picking the machine from the pool of available machines with the minimum load.

6. If there is a tie between two or more machines, the tie is broken by selecting the machine from the least flexible family. If the machines are from the same family, the tie is broken arbitrarily.
7. Next, pick the job to be scheduled on the selected machine by looking at all the compatible jobs highlighted in step 2 and picking the job with the lowest rank (highest processing time).
8. If all the highlighted jobs have been scheduled, pick the compatible job with the lowest rank.
9. If there are zero compatible jobs remaining, the selected machine is flagged as full, and removed from the pool of available machines.
10. Schedule the selected job on the selected machine. Setup time depends on the preceding job and is retrieved from the sequence-dependent setup matrix.
11. Repeat steps 5 to 10 until all the machines are full.

Minimum Changeover Time (Min CO)

1. Start with the data sheet and the sequence-dependent setup matrix.
2. All the machines are flagged as available
3. Begin scheduling process by picking the machine from the pool of available machines with the minimum load.
4. If there is a tie between two or more machines, the tie is broken by selecting the machine from the least flexible family. If the machines are from the same family, the tie is broken arbitrarily.
5. Next, look at the sequence-dependent setup matrix to pick the next job. The job j that satisfies

$$\boldsymbol{CO_{ij} = min\{CO_{ix}\}} \qquad \forall\, \boldsymbol{x = 1 \rightarrow n} \tag{5-1}$$

is selected where i is the preceding job on the selected machine. In the special case that the job is the first in the sequence of scheduled jobs on the selected

machines, the job j with the shortest processing time on that machine is selected. Ties between jobs are broken arbitrarily

6. If there are zero compatible jobs remaining, the selected machine is flagged as full, and removed from the pool of available machines.
7. Schedule the selected job on the selected machine.
8. Repeat steps 3 to 7 until all the machines are full.

LPT-F

1. Starting with the data sheet, find the minimum processing time for each job.
2. Highlight the families with processing time equal to the minimum processing time for each job.
3. Rank the jobs in decreasing order of minimum processing time.
4. All the machines are flagged as available
5. All the families are flagged as not available
6. Begin by selecting the family flagged as not available with the lowest flexibility and flagging it as available
7. Pick the machine from the pool of available machines in the available family with the minimum load.
8. If there is a tie between two or more machines, the tie is broken by selecting the machine from the least flexible family. If the machines are from the same family, the tie is broken arbitrarily.
9. Next, pick the job to be scheduled on the selected machine by looking at all the compatible jobs highlighted in step 2 and picking the job with the lowest rank (highest processing time).
10. If all the highlighted jobs have been scheduled, pick the compatible job with the lowest rank.
11. If there are zero compatible jobs remaining, the selected machine is flagged as full, and removed from the pool of available machines. In this case, go back to step 7 until all the machines in that family are full.

12. Schedule the selected job on the selected machine. Setup time depends on the preceding job and is retrieved from the sequence-dependent setup matrix.
13. Once all the machines in the selected family are full, flag this family as full and go back to step 6.
14. Repeat steps 6 to 13 until all the families are full.

SAPT

1. Starting with the data sheet and the sequence-dependent setup matrix, generate the APT matrix for each machine family.
2. The APT matrix for each machine will be a copy of the APT matrix for its family.
3. All the machines are flagged as available
4. Begin scheduling process by picking the machine from the pool of available machines with the minimum load.
5. If there is a tie between a couple of machines, the tie is broken by selecting the machine from the least flexible family. If the machines are from the same family, the tie is broken arbitrarily.
6. Next, look at APT matrix for the selected machine to pick the next job. The job j that satisfies

$$\boldsymbol{APT_{ij} = min\{APT_{ix}\}} \qquad \forall\, \boldsymbol{x = 1 \rightarrow n} \tag{5-2}$$

is selected where i is the preceding job on the selected machine. In the special case that the job is the first in the sequence of scheduled jobs on the selected machines, the job j that satisfies

$$\boldsymbol{APT_{jj} = min\{APT_{xx}\}} \qquad \forall\, \boldsymbol{x = 1 \rightarrow n} \tag{5-3}$$

is selected. Ties between jobs are broken arbitrarily

7. The selected job j cannot be scheduled again, so column j in all the APT matrices is deleted. Also, job j cannot be the preceding job in any machine other than the selected machine, so the row *j* in all the other machines' APT matrices is deleted.
8. If there are zero compatible jobs remaining, the selected machine is flagged as full, and removed from the pool of available machines.
9. Schedule the selected job on the selected machine.
10. Repeat steps 4 to 9 until all the machines are full.

SAPT-II

1. Starting with the data sheet and the sequence-dependent setup matrix, generate the APT matrix for each machine family.
2. The APT matrix for each machine will be a copy of the APT matrix for its family.
3. All the machines are flagged as available
4. Start with the first available machine. Look at APT matrix for each machine to pick the next potential job. The job j that satisfies

$$\boldsymbol{APT_{ij} = min\{APT_{ix}\}} \qquad \forall\, \boldsymbol{x = 1 \rightarrow n} \tag{5-4}$$

   is selected where i is the preceding job on the selected machine. In the special case that the job is the first in the sequence of scheduled jobs on the selected machines, the job j that satisfies

$$\boldsymbol{APT_{jj} = min\{APT_{xx}\}} \qquad \forall\, \boldsymbol{x = 1 \rightarrow n} \tag{5-5}$$

   is selected. Ties between jobs are broken arbitrarily.
5. If there are zero compatible jobs remaining for any machine, that machine is flagged as full, and removed from the pool of available machines.
6. Repeat step 4 for all the available machines.
7. The loads of the machines are compared after the potential job picked for each available machine is scheduled.

8. The machine/job pair with the lowest load is selected and that job is actually scheduled on that machine.
9. If there is a tie between a couple of machines, the tie is broken by selecting the machine from the least flexible family. If the machines are from the same family, the tie is broken arbitrarily.
10. The selected job j cannot be scheduled again, so column j in all the APT matrices is deleted. Also, job j cannot be the preceding job in any machine other than the selected machine, so the row *j* in all the other machines' APT matrices is deleted.
11. Repeat steps 4 to 10 until all the machines are full.

5.2.5 Assumptions

- All jobs are available at time $t = 0$
- All machines are available at time $t = 0$
- Machines are running on a continuous schedule; 24 hours/day, 7 days/week
- Preemption is not allowed; once a job is loaded on a machine, it cannot be removed until it is complete
- Job splitting is not allowed
- There are no unplanned windows of non-availability (e.g. breakdowns, unplanned maintenance, crew absence, raw material shortage)
- Unit processing times are equal to standard processing times provided in job specification sheets

6 RESULTS

In this chapter, the results of the experimentation work done are presented. Section 6.1 presents the results from the MIP. Section 6.1 presents the results from the mathematical model. Section 0 presents the results of applying scheduling algorithms to the APMS problem. Section 4.2 presents the automated scheduling tool developed to solve the APMS problem.

6.1 MIP Results

The MIP model was used to solve several problem sets varying in their dimensions, as presented in the previous chapter. Each problem set was also solved using the SAPT algorithm to provide some sense to whether the SAPT algorithm would be an appropriate method to apply to the APMS problem. A total of 9 trials were executed, and the results are summarized as follows:

MIP Processing Time (seconds)

Table 6-1 MIP Trials Processing Time in Seconds

	Case 1	Case 2	Case 3	Case 4	Case 5
Trial 1	2.15	638.40	16361.70*	16361.70*	53640.00*
Trial 2	1.09	200.11			
Trial 3	1.43	134.74			
Average	1.56	324.42	16361.70	16361.70	53640.00

* Size of problem set cause optimizer to stop run before 100% completion, out-of-memory error

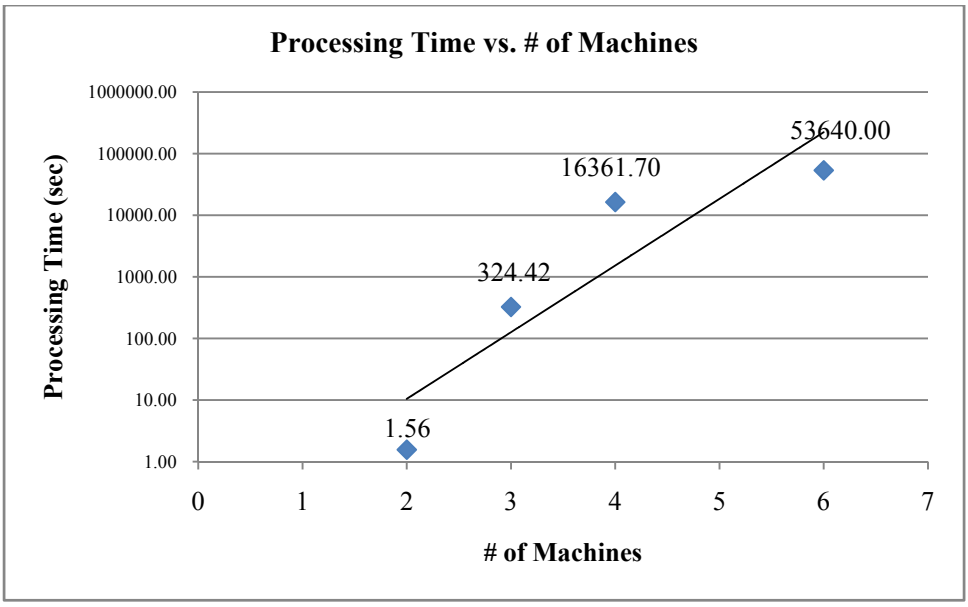


Figure 6-1 Processing Times

As noted in Table 6-1, the optimizer stopped running before reaching the 100% optimized solution. The reason is that the number of variables and constraints was huge and caused the optimizer to run out of memory. Figure 6-1 presents the same data in a chart with a log-scale x-axis. From the chart, it can be concluded that the processing time exponentially increases with increasing number of machines.

Schedule Charts

Each generated APMS problem set was applied to the MIP model and the SAPT algorithm. The schedule charts generated by the model and the algorithm are presented in Appendix C.

Objective Function Ratio

To compare the performance of the SAPT algorithm to the optimal schedules generated by the MIP, the ratio of the objective function, the average completion time, for the

SAPT algorithm versus the optimal schedule was calculated. The results are presented in Table 6-2 and Figure 6-2.

Table 6-2 Comparing SAPT to MIP – Objective Function Ratio

Case	Trial	MIP	SAPT	SAPT/MIP	Average
Case1	Trial 1	30.61	30.67	1.00	1.03
	Trial 2	37.01	39.94	1.08	
	Trial 3	36.48	36.48	1.00	
Case 2	Trial 1	38.22	43.55	1.14	1.07
	Trial 2	35.91	38.34	1.07	
	Trial 3	39.88	40.59	1.02	
Case 3		30.11	32.46	1.08	1.08
Case 4		29.36	26.99	0.92	0.92
Case 5		6.72	8.06	1.20	1.20

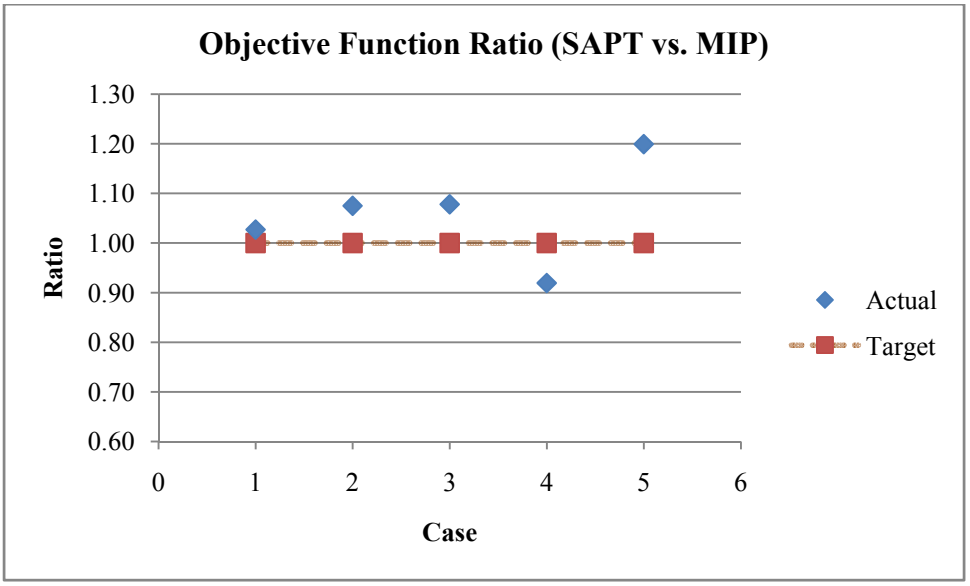


Figure 6-2 Comparing SAPT to MIP – Objective Function Ratio

Observations

After the trials using the MIP and the SAPT algorithm, the following was observed.

- The exponential increase in the processing times confirmed the NP-hardness of the APMS problem. The appropriate approach to solve larger APMS problems is to apply scheduling algorithms.
- By observing the objective function ratio comparison between the MIP and the SAPT algorithm, it was confirmed that the SAPT would result in near-optimum solutions to the APMS problem; so the SAPT algorithm will be the base to start the experimentation in the next phase.
- By observing the schedule charts, it is obvious that the less balanced a schedule is, the further the objective function is from the optimum. The emphasis on load balancing will be carried to the next step in the experimentation.

6.2 Algorithms Results

After observing the results from the mathematical model, it was confirmed that the MIP model was effective for really small APMS problems. The more practical approach to solve larger APMS problems would be the application of scheduling algorithms. The SAPT algorithm was introduced, and its performance was compared to the optimal schedule produced by the MIP. Based on that, it was verified that the SAPT algorithm could solve the APMS problem with a high level of effectiveness. To validate that assumption, the SAPT algorithm will be applied to larger APMS problem sets, and compared to several other scheduling algorithms.

Experimentation to apply and compare the scheduling algorithms was done in three phases. The experimentation was progressive, and result analysis in each phase was used to make adjustments and modifications to the experimentation design and data sets. The sections are divided to present the results from each phase separately. Details about the data sets, resources and scheduling results are available in the appendix.

6.2.1 Phase I

Algorithms

- SPT
- LPT
- Min CO
- SAPT

Bucket Set – Set # 1 [total number of jobs: 29, total number of units: 340]

Schedule Charts

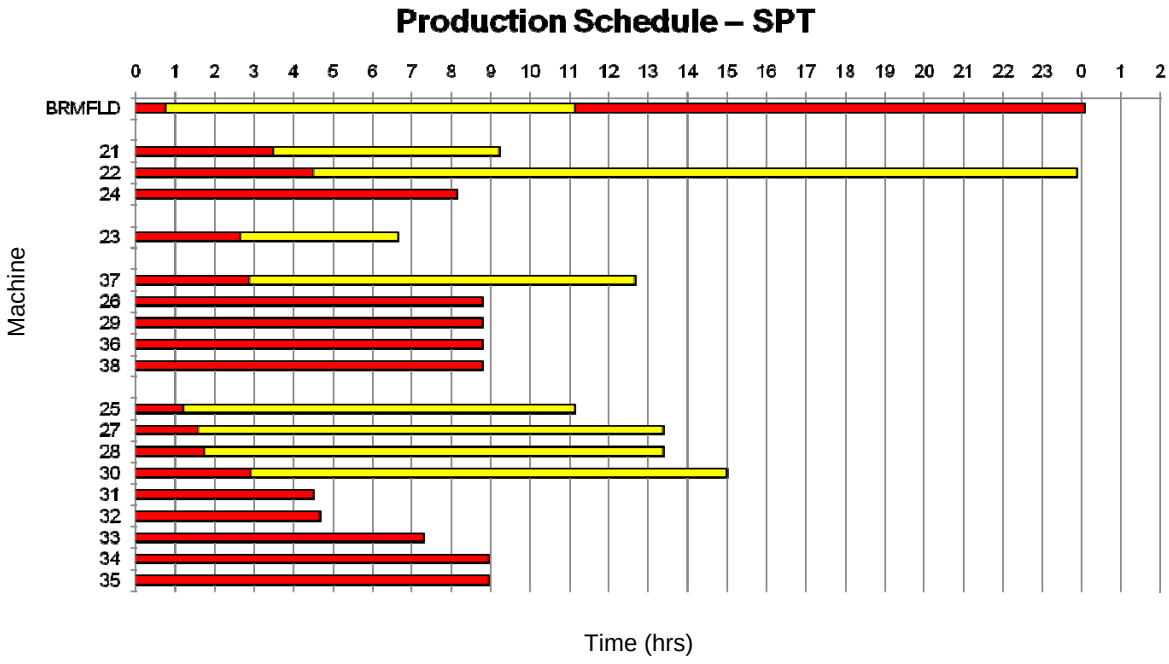


Figure 6-3 Production Schedule, Phase I, SPT

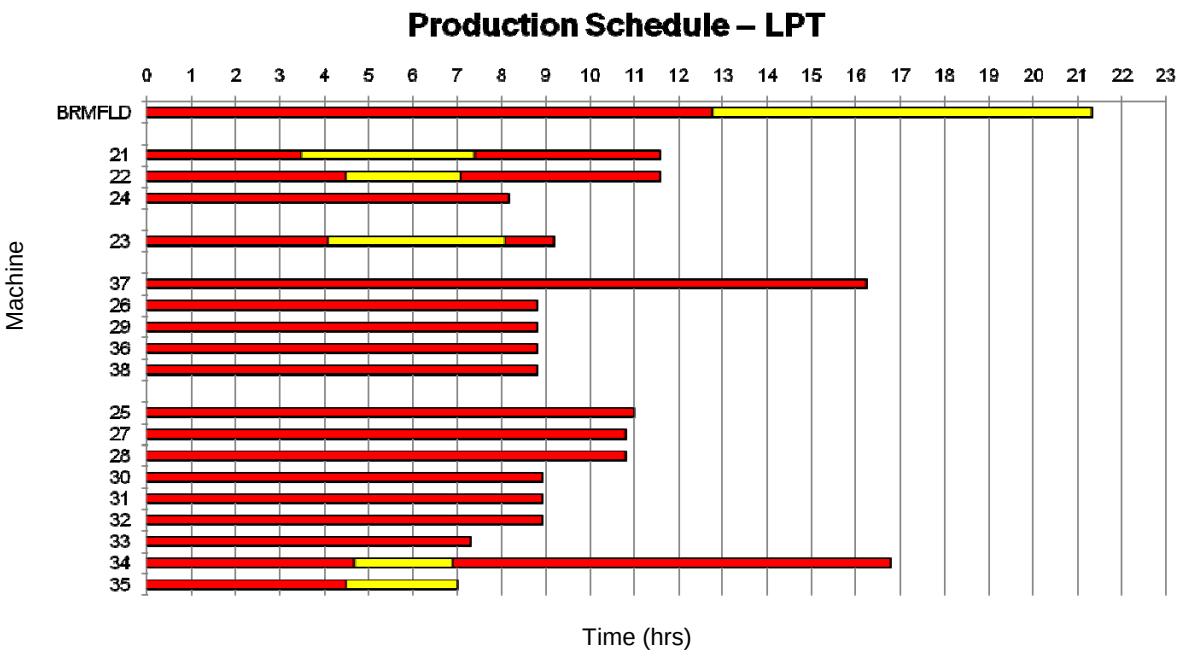


Figure 6-4 Production Schedule, Phase I, LPT

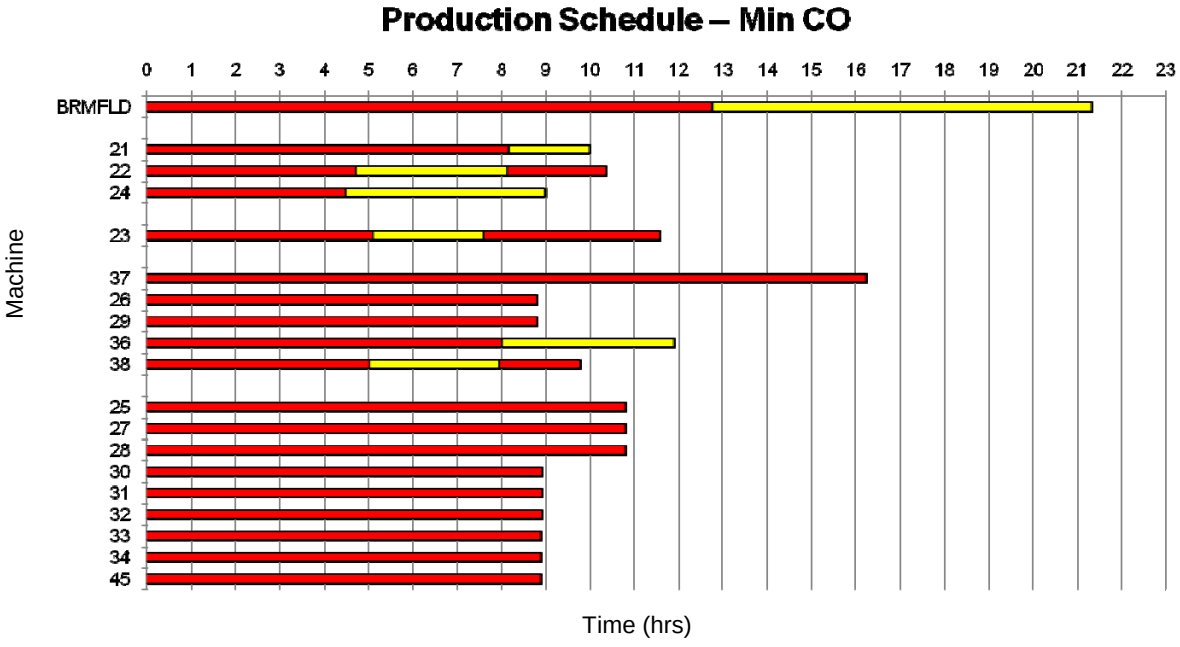


Figure 6-5 Production Schedule, Phase I, Min CO

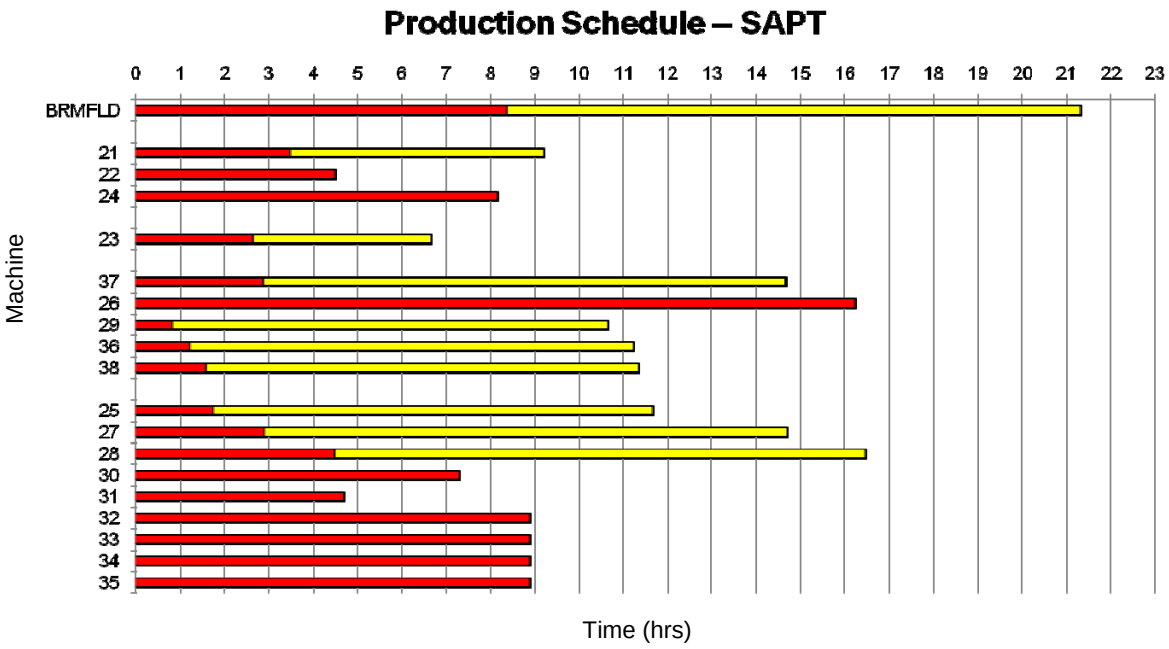


Figure 6-6 Production Schedule, Phase I, SAPT

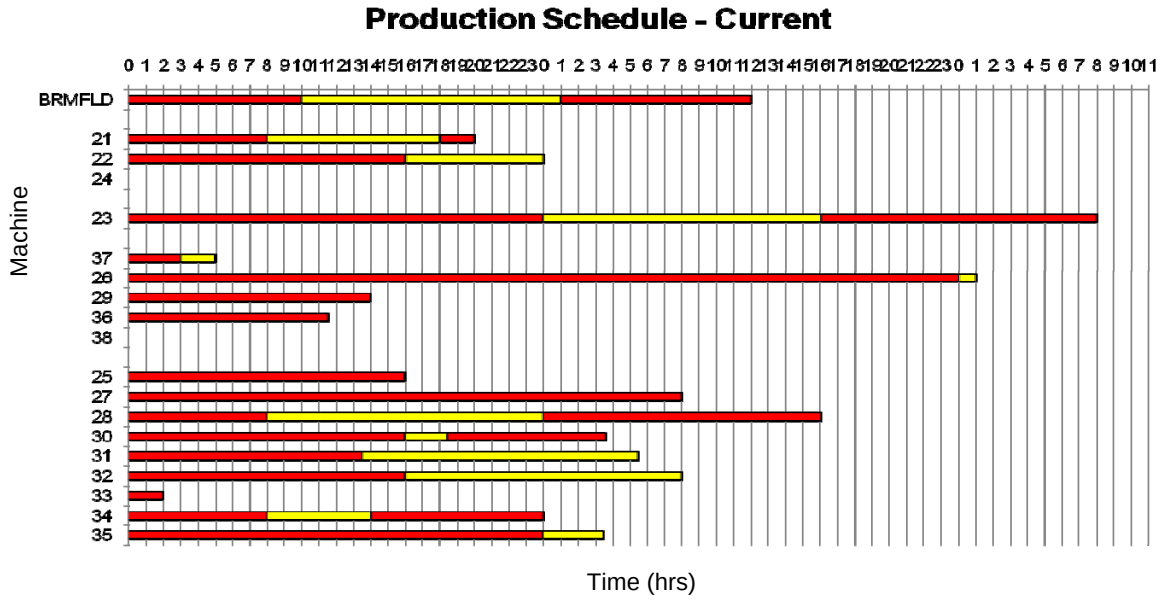


Figure 6-7 Production Schedule, Phase I, Actual Execution

Performance Measures

The performance of the four scheduling algorithms along with the actual execution of the bucket used in the problem set at this phase is presented as a summary of the performance measures in Table 6-3.

Table 6-3 Result for Each Algorithm and Actual Execution

	SPT	LPT	SAPT	Min CO	Current
Avg. Completion time*	**8.30**	**9.24**	**8.09**	**9.51**	**24.54**
Makespan*	24.09	21.32	21.32	21.32	56.00
Total Time*	207.34	203.77	204.19	203.86	446.1
CO Time*	9.17	8.12	8.92	7.42	-

* All times are in hours

Observations

The following observations were made after comparing the results of the experimental work done in phase I:

- *SAPT* algorithm was best for reducing the average completion time per order.
- *LPT* policy was best for reducing bucket completion time and makespan. However, the difference is not significant between the four scheduling methods evaluated ($\Delta T_{run,\ max} < 4$ hrs). Except for a few machines, LPT application also helps distribute the load evenly across all the machine families (can be seen in schedule Gantt charts).
- *Min CO* algorithm was best for achieving minimal setup time and maximum machine utilization, particularly when applied together with APT. Again, the difference in total setup time was small ($\Delta T_{C/O,\ max} < 4$ hrs).

- The actual processing times were found to be significantly different from the actual times recorded in the bucket execution records. This shows that comparing the theoretically generated schedules to the actual bucket execution might not possible. The assumptions regarding the processing times must be revised and adjusted to have the right comparison.

6.2.2 Phase II

Adjustments

The same set of assumptions in phase I apply to the second phase in addition to:

- Theoretical processing times from job specification sheets are multiplied by 1.8 to reflect estimates of actual processing times.
- Machines are operating with availability constraints according to the actual production schedule. The schedule is presented in Appendix A.

Algorithms

- LPT
- LPT-F
- SAPT

Bucket Set – Set # 2 [total number of jobs: 73, total number of units: 560]

Schedule Charts

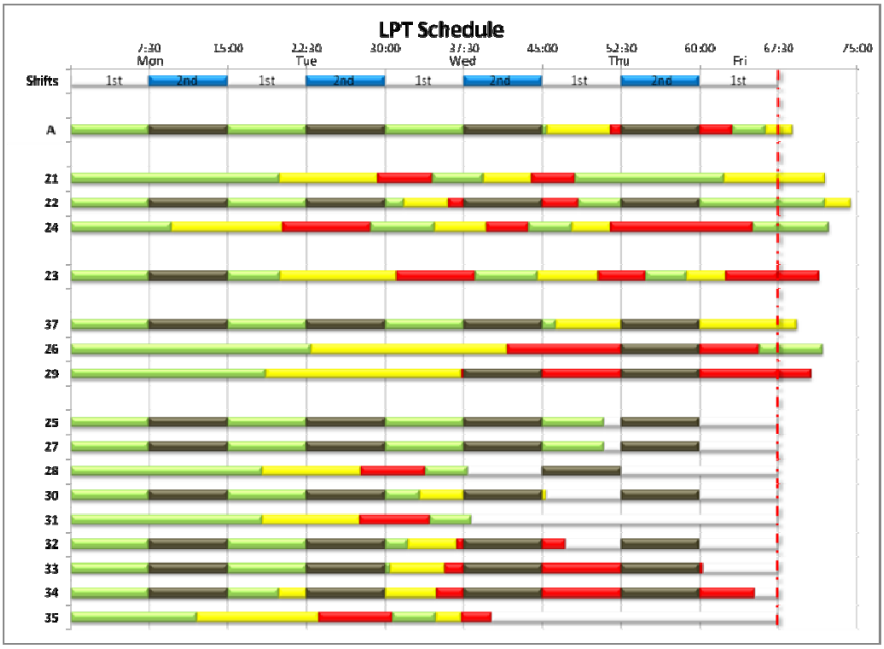


Figure 6-8 Production Schedule, Phase II, LPT

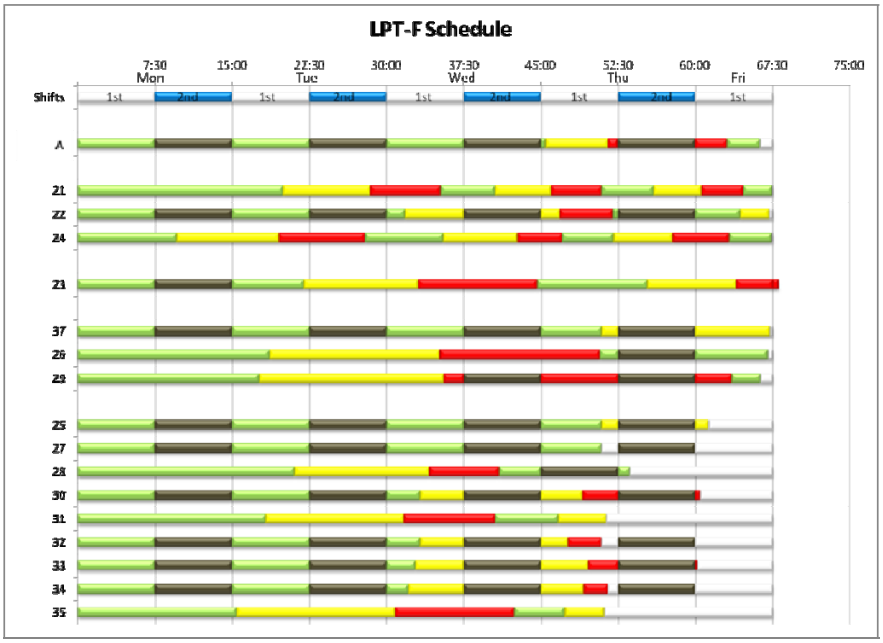


Figure 6-9 Production Schedule, Phase II, LPT-F

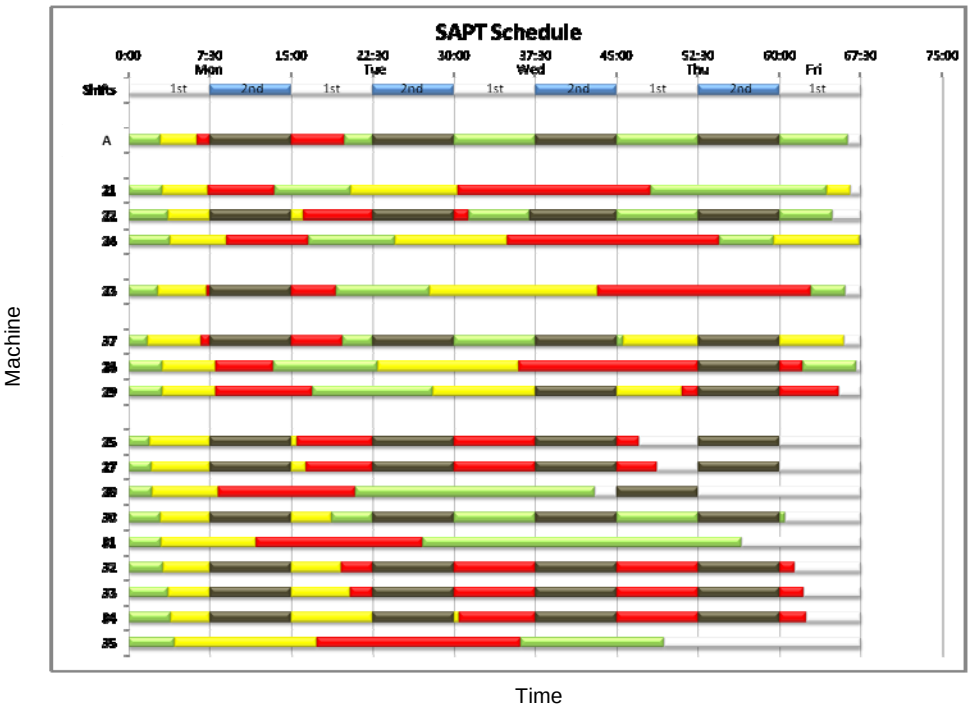


Figure 6-10 Production Schedule, Phase II, SAPT

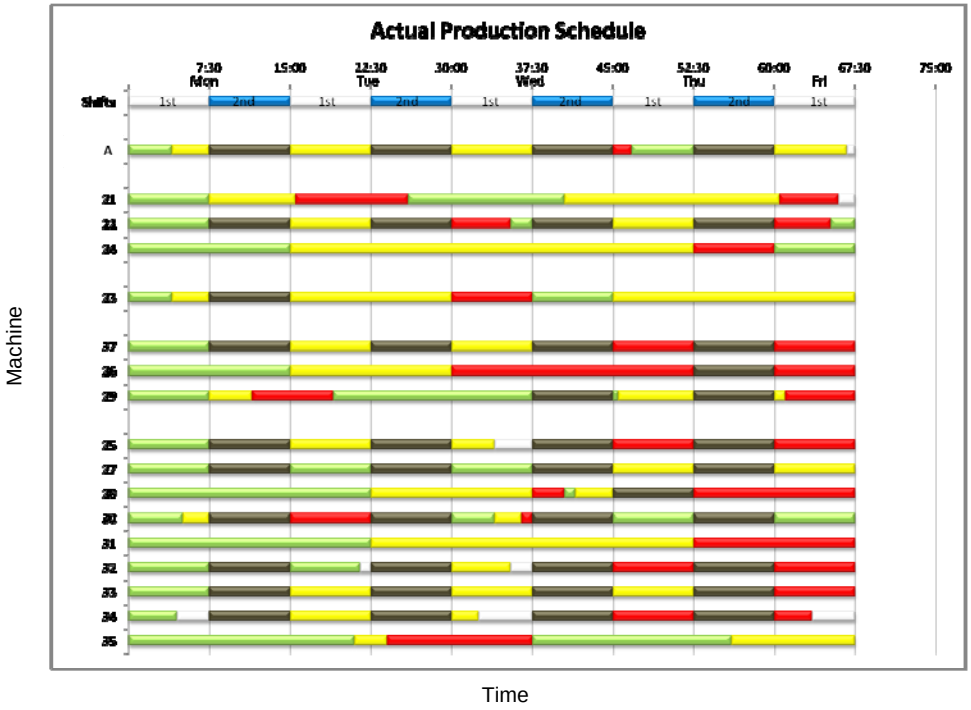


Figure 6-11 Production Schedule, Phase II, Actual

Performance Measures

The performance of the four scheduling algorithms along with the actual execution of the bucket used in the problem set at this phase is presented as a summary of the performance measures in Table 6-4.

Table 6-4 Result for Each Algorithm and Actual Execution

	Actual	LPT	LPT-F	SAPT
AVG Completion Time*	38.48	43.20	45.37	25.72
Makespan*	67.5	74.41	68.09	67.41
Total Time*	834	669	744	736
C/O %	-	7.2%	6.1%	5.9%
Utilization %	-	80%	89%	88%

* All times are in hours

Observations

The following observations were made after comparing the results of the experimental work done in phase II:

- The superiority of the *SAPT* algorithm was verified versus the *LPT* and the *LPT-F* algorithms
- Comparing the *LPT* and *LPT-F* algorithms, the *LPT-F* performed better in minimizing the makespan and maximizing the utilization. This fact is useful for future studies tackling the APMS problems with a different objective.
- The *SAPT* algorithm did not perform well in balancing the machine loads and improving the utilization. The presumption is that this was caused by the introduction of the availability constraint; where the *SAPT* algorithm does not compare machine availability when selecting machines in each scheduling

iteration. In the next phase, there will be an attempt to improve the performance of the *SAPT* algorithm by looking into the future. This improvement should also help minimize the average lead time.

6.2.3 Phase III

Adjustments

The experimentation in this phase builds on Phase II. In this phase, the problem set used in the previous phase was applied, in addition to five more problem sets. The automated scheduling tool was used to generate the production schedules and summarize the performance measures. Starting with the problem set used in the previous set allowed the validation of the functionality of the scheduling tool. Adding the five problem sets increased the number of trials; consequently providing additional results to confirm the observations and conclusions made at the end of this phase.

Algorithms

- SAPT
- SAPT-II

Bucket Sets

- Set # 2 [total number of jobs: 73, total number of units: 560]
- Set # 3 [total number of jobs: 75, total number of units: 642]
- Set # 4 [total number of jobs: 55, total number of units: 654]
- Set # 5 [total number of jobs: 75, total number of units: 739]
- Set # 6 [total number of jobs: 60, total number of units: 604]
- Set # 7 [total number of jobs: 65, total number of units: 607]

Schedule Charts

There are six schedule charts for each algorithm; a total of twelve charts. These charts are presented in Appendix D.

Performance Measures

The table below compares the SAPT-II algorithm to the SAPT algorithm. The color of the cell highlights the improvement or decline in performance (green – better, yellow – same, red – worse). The direction of the arrow shows the direction of the change (up – increase, down – decrease).

Table 6-5 Comparing SAPT-II to SAPT

Bucket Set	Set # 2			Set # 3			Set # 4		
Algorithm	SAPT	SAPT-II	Δ	SAPT	SAPT-II	Δ	SAPT	SAPT-II	Δ
# of Jobs	73	73		75	75		55	55	
Avg. Completion Time (hrs)	25.39	23.19		27.07	22.62		29.31	25.21	
# of Jobs Scheduled	72	73		75	75		54	54	
Makespan (hrs)	84.88	83.82		87.68	68.05		82.24	86.65	
Total Proc. Time (hrs)	709.71	776.11		749.14	795.16		722.08	768.09	
CO Time (hrs)	42.08	42.42		49.42	51.83		31.58	29.17	
CO %	5.90%	5.50%		6.60%	6.50%		4.30%	3.80%	

Bucket Set	Set # 5			Set # 6			Set # 7		
Algorithm	SAPT	SAPT-II	Δ	SAPT	SAPT-II	Δ	SAPT	SAPT-II	Δ
# of Jobs	75	75		60	60		65	65	
Avg. Completion Time (hrs)	27.96	26.07		27.3	24		27.18	23.32	
# of Jobs Scheduled	72	75		60	60		64	65	
Makespan (hrs)	83.67	83.82		84.31	83.03		86.79	86.86	
Total Proc. Time (hrs)	756.26	893.83		732.14	780.82		722.68	798.41	
CO Time (hrs)	43.33	43.83		39.17	41.5		43.92	42.67	
CO %	5.70%	4.90%		5.30%	5.30%		6.10%	5.30%	

Observations

- The SAPT-II algorithm performs better than SAPT in minimizing the average lead time.
- For the other monitored performance measures, SAPT-II performs similar to SAPT except for minimizing the total processing time. This is because the SAPT algorithm gives priority only to APT, while the SAPT-II also gives consideration to load balancing across families; so more often, jobs are scheduled on families that do have the minimal APT when compared to the other families.
- The experimentation also used to validate the scheduling software. A copy was delivered and installed at the company mentioned in the case study.

After observing the results, it was concluded that the SAPT-II is more efficient than the SAPT algorithm in scheduling the APMS problem.

6.3 Automated Scheduling Tool

After the three phases of experimentation, it was concluded that the SAPT-II algorithm will be used to solve the APMS problem. The tool was developed and validated using the buckets presented in Phase III of the scheduling algorithms experimentation. The features of the automated scheduling tool are explained in the following sections.

6.3.1 Main Interface Window

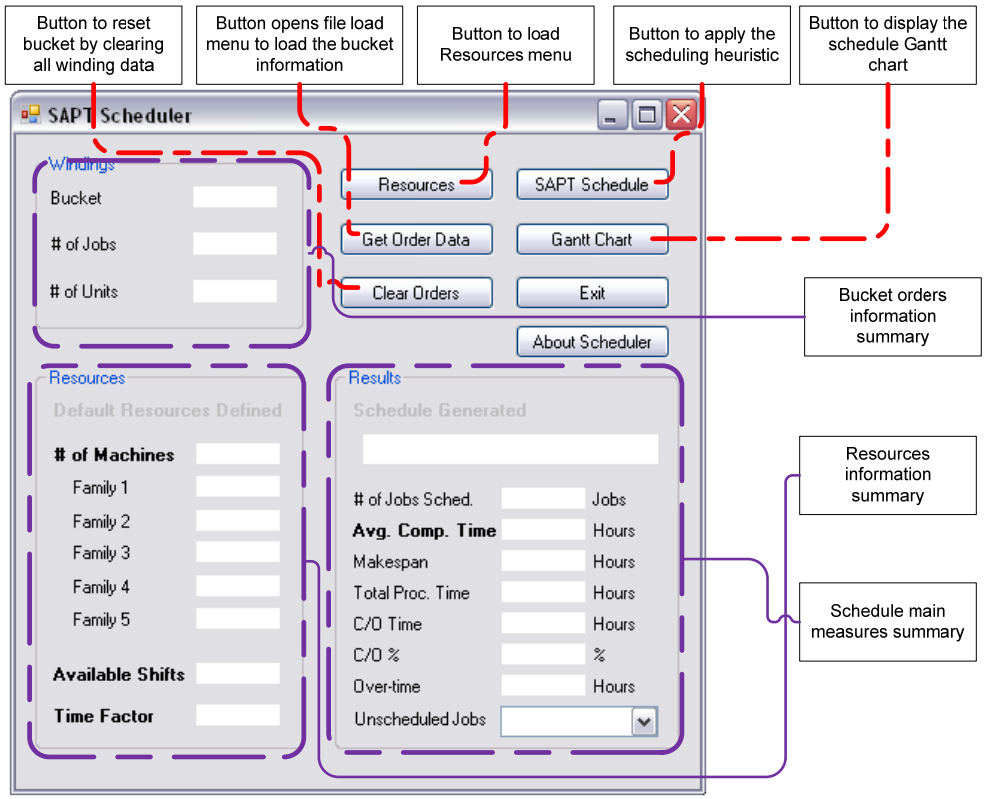


Figure 6-12 Scheduling software main window

6.3.2 Data Input

The scheduling software was developed to interface with the output of a program currently used by the company in the case study to analyze the information in the production bucket and calculate loads and capacities. That program accesses the company data-base, gives full details about the orders within a bucket and outputs that data to a text file. The program accesses the company's data base, opens the text file, and stores all the details in an array that is used to generate APT and sequence dependent setup time matrices.

An example to the format of the information in the text file is shown in below. The file has one line for each order that contains all the specifications and theoretical processing times. The information is comma-delimited.

```
464204,07/11,  3,EC 107600N865    , 10.0,1777750210, AL,
17.0, 1,R,AL, 0.015, 5.00,1,12.94,KOLD,6 Stn     ,Sm Pole
, 0.87 , 0.99 , 0.67 , 0.56 , 0.51 , 0.48 , 0.46
```

6.3.3 Resources

The default resources – or machines – available are user-defined. The time factor used to adjust theoretical processing times is also user-defined. The assumption is that the available production time is 5 days/week and 2 shifts/day. When the program starts, it loads the default resources which are stored in a master file edited and updated by the user. The user can access the resources menu to change any of the machine resources and click on '**Update'** to save the adjustment, or to return to the default resources defined in the program by clicking on '**Default Res.**'. To notify the user that the scheduling software is using the default resources, a tab in the main window that says '**Default Resources Defined**' is visible. Once the user changes one or more of the resources, the text in the tab becomes transparent.

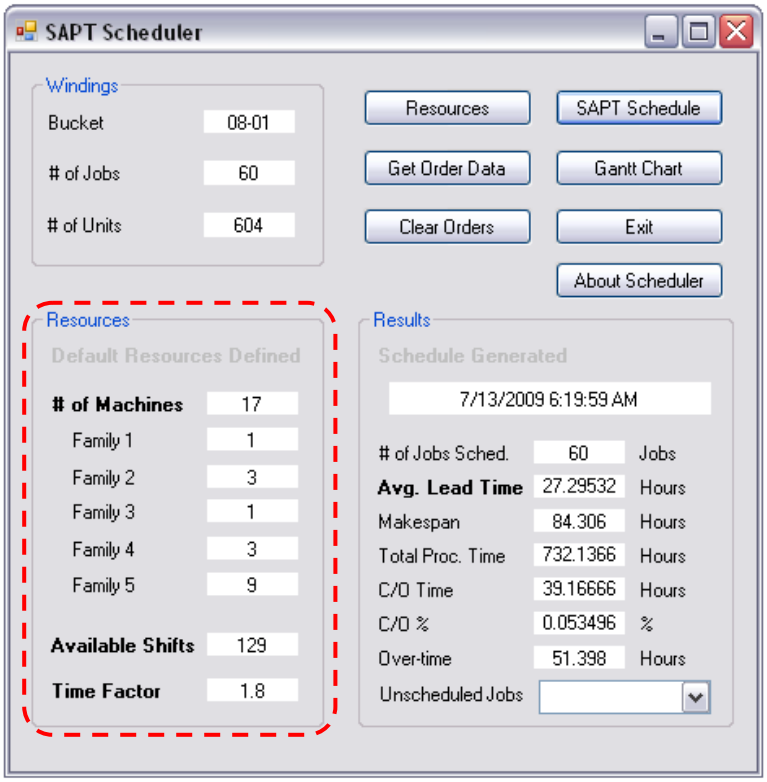


Figure 6-13 Summary information about the available resources

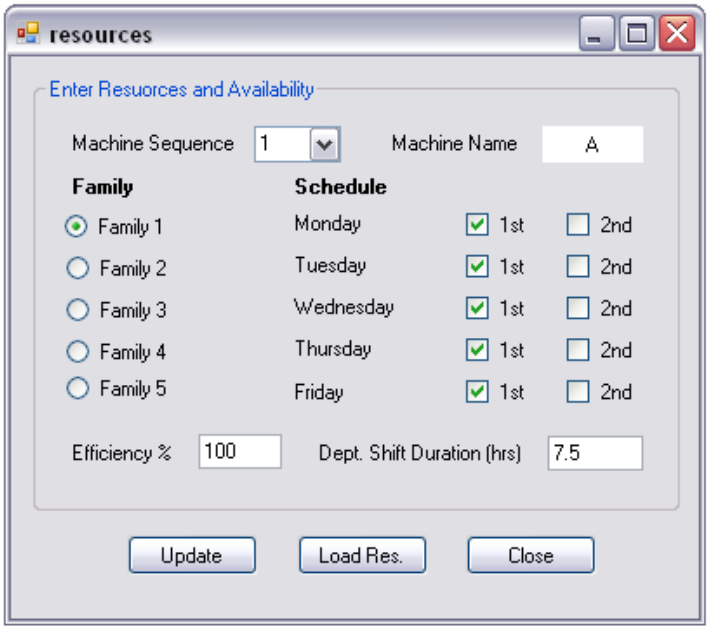


Figure 6-14 Resources menu allows user to redefine the machine information

6.3.4 Output

The output of the scheduling software is a summary of the important measures of the schedule (including # of jobs scheduled, make-span and average lead time), and a Gantt chart showing the machine assignment and sequence of each order scheduled to run. In the Gantt chart the user can click on '**Show/Hide End Time**' to toggle the view of the end time for each order.

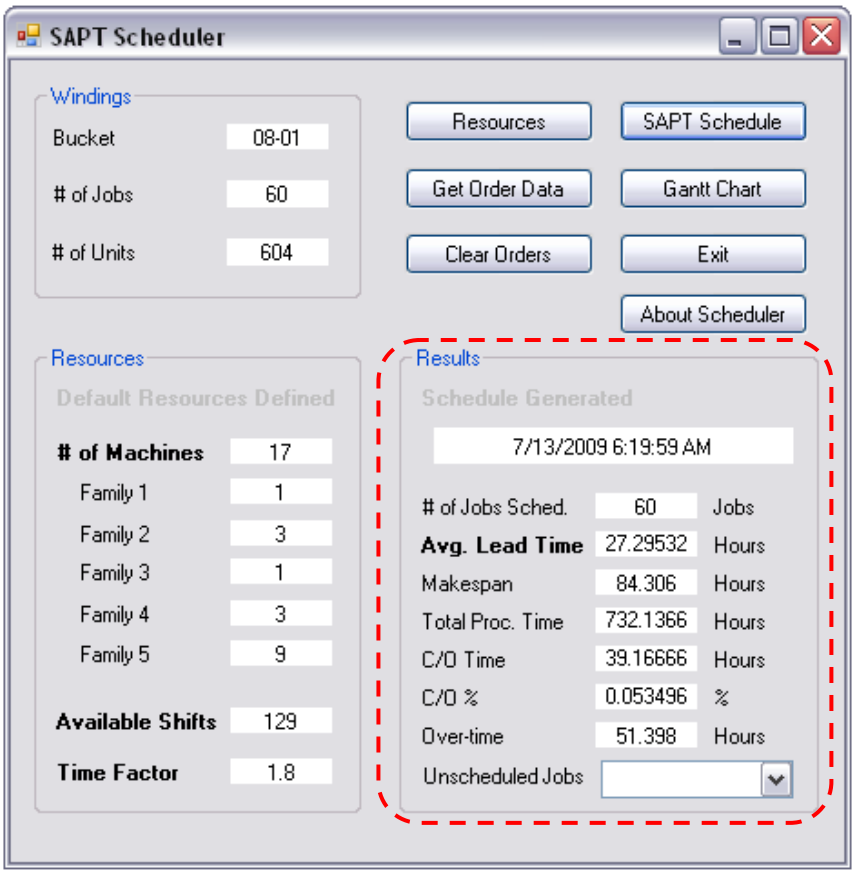


Figure 6-15 Summary of the schedule performance measures

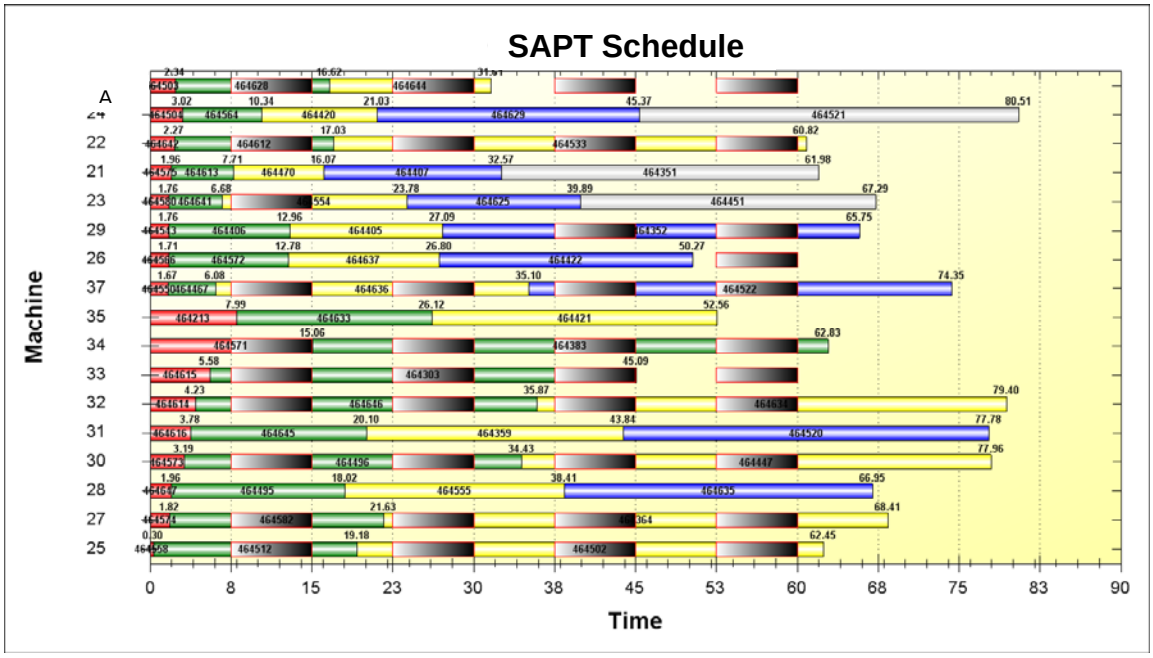


Figure 6-16 Schedule Gantt chart with end times shown

7 CONCLUSIONS AND FUTURE RESEARCH

The experimentation work, results and observations for the tested problem sets were presented in chapters 5 and 6. This chapter states the conclusions and future research opportunities.

7.1 Conclusions

In this research, the unique APMS problem with the objective of minimizing the average flow time was solved. The constraints in this problem included machine availability constraints, sequence-dependent setup times, and eligibility constraints. A mathematical model for the APMS problem was developed to check the feasibility of finding an optimal solution for the NP-hard scheduling problem. The SAPT-II algorithm was introduced to solve the APMS problem.

The research was motivated by a real scheduling problem at a make-to-order industrial product manufacturer. The first objective was to remove the bottleneck in their operations by improving production scheduling. The second objective was to develop an automated scheduling tool to allow better shop floor management and improved coordination between different production departments.

The first approach was to check the feasibility to apply mathematical modeling to solve the APMS problem. As expected, the processing times for the MIP model developed increased exponentially with the increase in the size of the APMS problem. It was confirmed that it was not feasible to solve the APMS problem optimally using mathematical modeling. The SAPT algorithm was introduced as an alternative, and its performance was compared to the optimum solution generated by the MIP.

The second approach to solve the APMS problem was to use scheduling algorithms. Initially, the SAPT algorithm was introduced. After comparing its performance to several other scheduling algorithms in two phases of experimentation, the algorithm was tuned

and improved to produce the SAPT-II algorithm. In the third phase of experimentation, the SAPT-II algorithm was applied and compared to the SAPT algorithm, and it was concluded that the SAPT-II produced the lowest average lead times.

The SAPT-II algorithm was selected, and an automated scheduling tool was developed. The tool is user friendly, and produces the production schedules in the form of Gantt charts. The resources and job sets can be fully customized by the user. The tool provides a critical visual management tool, and it can also be used in forecasting and capacity planning operations.

Overall, the outcome of the research demonstrates that the SAPT-II algorithm is able to solve the heavily constrained, unique APMS problem and produce acceptable results at very low computation times. The data presented was inspired by a specific case study, but the outcome can be applied to any APMS problem; typically single-pass production with several groups of identical machines. An example is in packaging operations with multiple machines dedicated to each package size, where the bulk is produced upstream and packed into different package sizes on different machine families. The AMPS problem can also extend past production operations. An example is yard and docking management, where docks can be classified into families depending on the type or size of loads and unloading or discharge times are dependent on the load and the dock.

7.2 Future Research

The performance of the SAPT-II algorithm is a single-pass algorithm; the schedule produced by the algorithm is considered the final output, and no improvement activities are performed. Its performance could be improved by applying an improvement heuristic to the output of the algorithm. Another approach would be to allow job splitting; which could improve the performance of the algorithm, but complicate the scheduling problem at the same time.

Also, the APMS problem presented in this research had a specific set of constraints and objective function. Other manufacturing environments might have different constraints and objective. This presents a great opportunity to explore APMS problems with different constraint sets and objective functions.

APPENDIX A: RESOURCES

Phase I

Family	# of Machines	Machines
Family 1	1	A
Family 2	3	21, 22, 24
Family 3	2	23, 37
Family 4	4	26, 29, 36, 38
Family 5	9	25, 27, 28, 30, 31, 32, 33, 34, 35

Phases II & III

In these phases, the availability constraint was introduced. The number of machines in each family and the number of shifts each machine was crewed to operate reflected the staffing during the actual execution of the data set. At this point, the problem set represented the actual APMS problem presented in this research in terms of production environment. The resources and staffing are presented in each period in the schedule represents an 8-hour shift.

Machine	Family	Stations	Monday		Tuesday		Wednesday		Thursday		Friday	
			1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
A	1	1	x		x		x		x		x	
21	2	3	x	x	x	x	x	x	x	x	x	
22			x		x		x		x		x	
24			x	x	x	x	x	x	x	x	x	
23	3	5	x		x	x	x	x	x	x	x	
26	4	5	x	x	x	x	x	x	x		x	
29			x	x	x	x	x		x		x	
37			x		x		x		x		x	
25	5	6	x		x		x		x		x	
27			x		x		x		x		x	
28			x	x	x	x	x	x		x	x	
30			x		x		x		x		x	
31			x	x	x	x	x	x	x	x	x	
32			x		x		x		x		x	
33			x		x		x		x		x	
34			x		x		x		x		x	
35			x	x	x	x	x	x	x	x	x	

APPENDIX B: PERFORMANCE MEASURES

Phase I

Family	M/C	SPT # of jobs	SPT running time	SPT CO time	SPT CO %	SPT Utilization				LPT # of jobs	LPT running time	LPT CO time	LPT CO %	LPT Utilization			
1	A	3	24.09	1.17	5%	100%		100%		2	21.32	0.17	1%	100%		100%	
2	21	2	9.21	1	11%	38%	39%			3	11.58	2	17%	54%	69%		
	22	2	23.89	1	4%	99%	100%	57%	58%	3	11.58	2	17%	54%	69%	49%	62%
	24	1	8.16		0%	34%	34%			1	8.16		0%	38%	49%		
3	23	2	6.66	1	15%	28%	28%	28%	28%	3	9.19	1.25	14%	43%	55%	43%	55%
4	37	2	12.68	1	8%	53%	53%			1	16.25		0%	76%	97%		
	26	1	8.8		0%	37%	37%			1	8.8		0%	41%	52%		
	29	1	8.8		0%	37%	37%	40%	40%	1	8.8		0%	41%	52%	48%	61%
	36	1	8.8		0%	37%	37%			1	8.8		0%	41%	52%		
	38	1	8.8		0%	37%	37%			1	8.8		0%	41%	52%		
5	25	2	11.16	1	9%	46%	47%			1	10.98		0%	52%	65%		
	27	2	13.39	1	7%	56%	56%			1	10.8		0%	51%	64%		
	28	2	13.54	1	7%	56%	57%			1	10.8		0%	51%	64%		
	30	2	14.98	1	7%	62%	63%			1	8.94		0%	42%	53%		
	31	1	4.5		0%	19%	19%	40%	41%	1	8.94		0%	42%	53%	47%	60%
	32	1	4.68		0%	19%	20%			1	8.94		0%	42%	53%		
	33	1	7.32		0%	30%	31%			1	7.32		0%	34%	44%		
	34	1	8.94		0%	37%	37%			3	16.78	2	12%	79%	100%		
	35	1	8.94		0%	37%	37%			2	6.99	0.7	10%	33%	42%		
Total		**29**	**207.34**	**9.17**	**4.42%**	**45.30%**	**42.61%**			**29**	**203.77**	**8.12**	**3.98%**	**50.30%**	**60.41%**		
Family	M/C	Min CO								SAPT							

		# of jobs	running time	CO time	CO %	Utilization				# of jobs	running time	CO time	CO %	Utilization			
1	A	2	21.32	0.17	1%	100%		100%		2	21.32	0.17	1%	100%		100%	
2	21	2	10	1	10%	47%	62%	46%	60%	2	9.21	1	11%	43%	56%	34%	44%
	22	3	10.37	1.25	12%	49%	64%			1	4.5		0%	21%	27%		
	24	2	8.99	1	11%	42%	55%			1	8.16		0%	38%	50%		
3	23	3	11.6	1.75	15%	54%	71%	54%	71%	2	6.66	1	15%	31%	40%	31%	40%
4	37	1	16.25		0%	76%	100%	52%	68%	2	14.68	1	7%	69%	89%	60%	78%
	26	1	8.8		0%	41%	54%			1	16.25		0%	76%	99%		
	29	1	8.8		0%	41%	54%			2	10.46	1	10%	49%	63%		
	36	2	11.9	1	8%	56%	73%			2	11.22	1	9%	53%	68%		
	38	3	9.79	1.25	13%	46%	60%			2	11.34	0.75	7%	53%	69%		
5	25	1	10.98		0%	52%	68%	45%	59%	2	11.68	1	9%	55%	71%	47%	61%
	27	1	10.8		0%	51%	66%			2	14.71	1	7%	69%	89%		
	28	1	10.8		0%	51%	66%			2	16.48	1	6%	77%	100%		
	30	1	8.94		0%	42%	55%			1	7.32		0%	34%	44%		
	31	1	8.94		0%	42%	55%			1	4.68		0%	22%	28%		
	32	1	8.94		0%	42%	55%			1	8.88		0%	42%	54%		
	33	1	8.88		0%	42%	55%			1	8.88		0%	42%	54%		
	34	1	8.88		0%	42%	55%			1	8.88		0%	42%	54%		
	35	1	8.88		0%	42%	55%			1	8.88		0%	42%	54%		
Total		**29**	**203.86**	**7.42**	**3.64%**	**50.33%**	**62.41%**			**29**	**204.19**	**8.92**	**4.37%**	**50.41%**	**61.65%**		

Family	M/C	Current							
		# of jobs	running time	CO time	CO %	Utilization			
1	A	2.5	36		0%	169%		64%	
2	21	1	20		0%	94%	121%	26%	26%
	22	2	24		0%	113%	146%		
	24	0	0		-	0%	0%		
3	23	3	56		0%	263%	340%	100%	100%
4	37	1.5	5		0%	23%	30%	28%	28%
	26	2	49		0%	230%	297%		
	29	1	14		0%	66%	85%		
	36	0.5	11.5		0%	54%	70%		
	38	0	0		-	0%	0%		
5	25	1	16		0%	75%	97%	46%	46%
	27	1	32		0%	150%	194%		
	28	3	40		0%	188%	243%		
	30	3	27.6		0%	129%	167%		
	31	1	29.5		0%	138%	179%		
	32	1	32		0%	150%	194%		
	33	1	2		0%	9%	12%		
	34	3	24		0%	113%	146%		
	35	1.5	27.5		0%	129%	167%		
Total		**29**	**446.1**		**0.00%**	**41.93%**	**40.68%**		

Phase II

LPT									
Family	M/C	# of jobs	# of shifts	Available time	Running time	# of CO's	CO time	Utilization	Utilization
1	A	5	5	37.5	38.88	4	3.25	100%	104%
2	21	8	9	67.5	71.95	7	7	100%	109%
	22	5	5	37.5	44.41	4	3.25	100%	
	24	10	9	67.5	72.31	9	8.75	100%	
3	23	9	8	60	63.87	8	4.92	100%	106%
4	37	2	5	37.5	39.29	1	1	100%	106%
	26	4	8	60	64.24	3	3	100%	
	29	3	7	52.5	55.66	2	1.75	100%	
5	25	1	5	37.5	20.9	0	0	56%	52%
	27	1	5	37.5	20.9	0	0	56%	
	28	4	8	60	30.36	3	3	51%	
	30	2	5	37.5	15.39	1	1	41%	
	31	4	9	67.5	30.71	3	3	45%	
	32	3	5	37.5	17.21	2	1.75	46%	
	33	3	5	37.5	22.85	2	2	61%	
	34	3	5	37.5	27.77	2	1.25	74%	
	35	6	9	67.5	32.68	5	3.5	48%	
Total		**73**	**112**	**840**	**669.38**	**56**	**48.42**	**80%**	

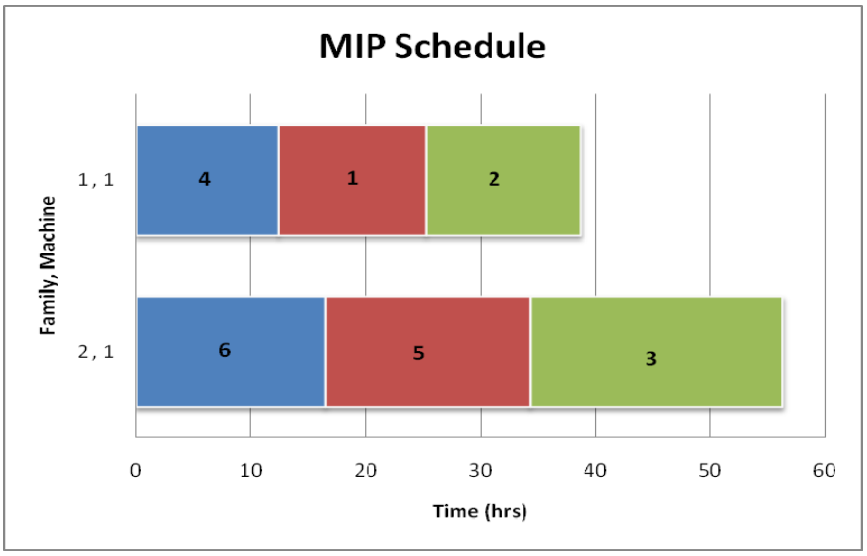
LPT-F									
Family	M/C	# of jobs	# of shifts	Available time	Running time	# of CO's	CO time	Utilization	Utilization
1	A	4	5	37.5	36.3	3	2.25	97%	97%
2	21	10	9	67.5	67.48	9	6.25	100%	100%
	22	5	5	37.5	37.17	4	3.25	99%	
	24	10	9	67.5	67.41	9	7.92	100%	
3	23	6	8	60	60.59	5	4	100%	101%
4	37	2	5	37.5	37.25	1	1	99%	99%
	26	4	8	60	59.56	3	2.17	99%	
	29	4	7	52.5	51.31	3	2.17	98%	
5	25	2	5	37.5	31.31	1	1	83%	78%
	27	1	5	37.5	28.4	0		76%	
	28	4	8	60	46.15	3	3	77%	
	30	3	5	37.5	30.5	2	2	81%	
	31	5	9	67.5	51.35	4	3.25	76%	
	32	3	5	37.5	28.41	2	1.25	76%	
	33	3	5	37.5	30.22	2	2	81%	
	34	3	5	37.5	29	2	1.25	77%	
	35	5	9	67.5	51.16	4	3.25	76%	
Total		**74**	**112**	**840**	**743.57**	**57**	**46.01**	**89%**	

SAPT									
Family	M/C	# of jobs	# of shifts	Available time	Running time	# of CO's	CO time	Utilization	Utilization
1	A	4	5	37.5	36.34	3	2.25	97%	97%
2	21	8	9	67.5	66.54	7	3.25	99%	98%
	22	4	5	37.5	34.88	3	1.5	93%	
	24	8	9	67.5	67.41	7	6.25	100%	
3	23	7	8	60	58.55	6	3.59	98%	98%
4	37	5	5	37.5	35.97	4	3	96%	97%
	26	7	8	60	59.56	6	5.25	99%	
	29	6	7	52.5	50.46	5	5	96%	
5	25	3	5	37.5	24.52	2	2	65%	78%
	27	3	5	37.5	26.18	2	1.17	70%	
	28	4	8	60	42.98	3	1.5	72%	
	30	3	5	37.5	30.5	2	2	81%	
	31	4	9	67.5	56.45	3	2.25	84%	
	32	3	5	37.5	31.38	2	1.25	84%	
	33	3	5	37.5	32.24	2	2	86%	
	34	3	5	37.5	32.35	2	2	86%	
	35	4	9	67.5	49.39	3	3	73%	
Total		**79**	**112**	**840**	**735.7**	**62**	**47.26**	**88%**	

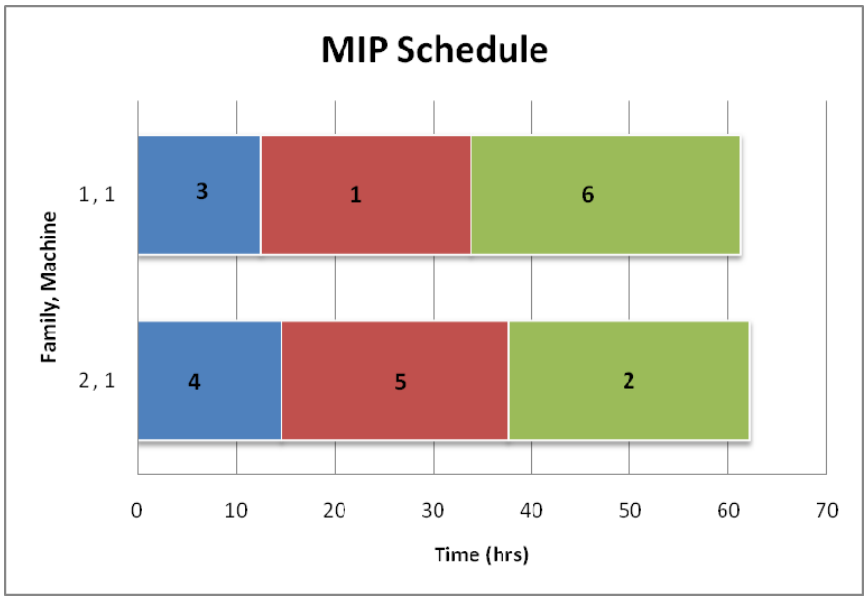
APPENDIX C: SCHEDULE CHARTS, MIP MODEL

MIP Model Schedule Charts

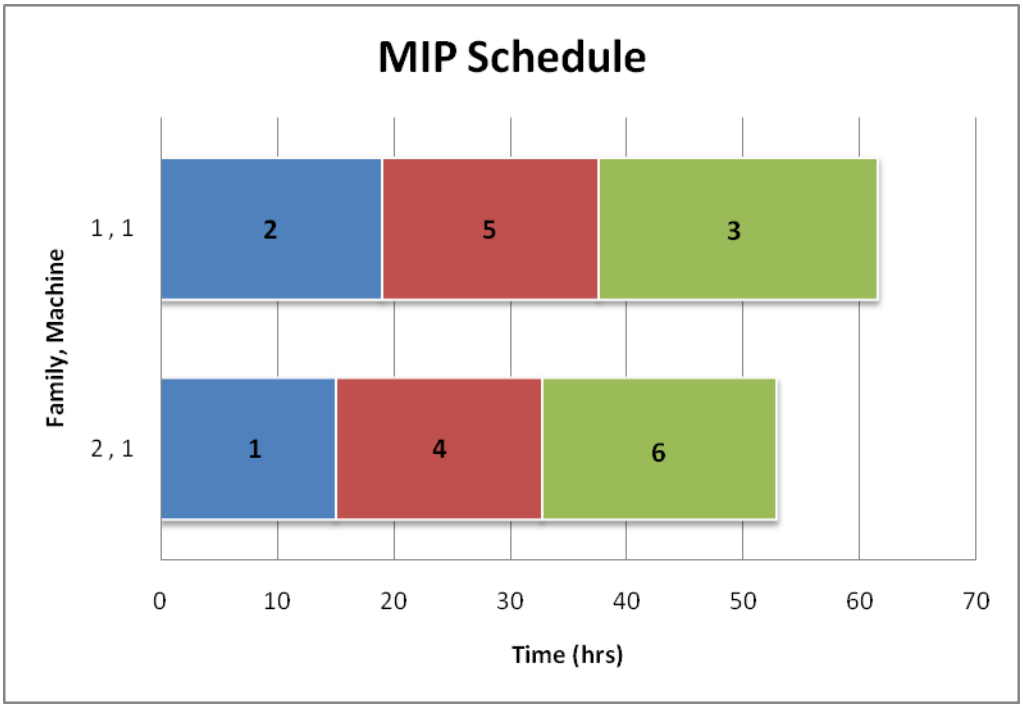
Case 1, Trial 1



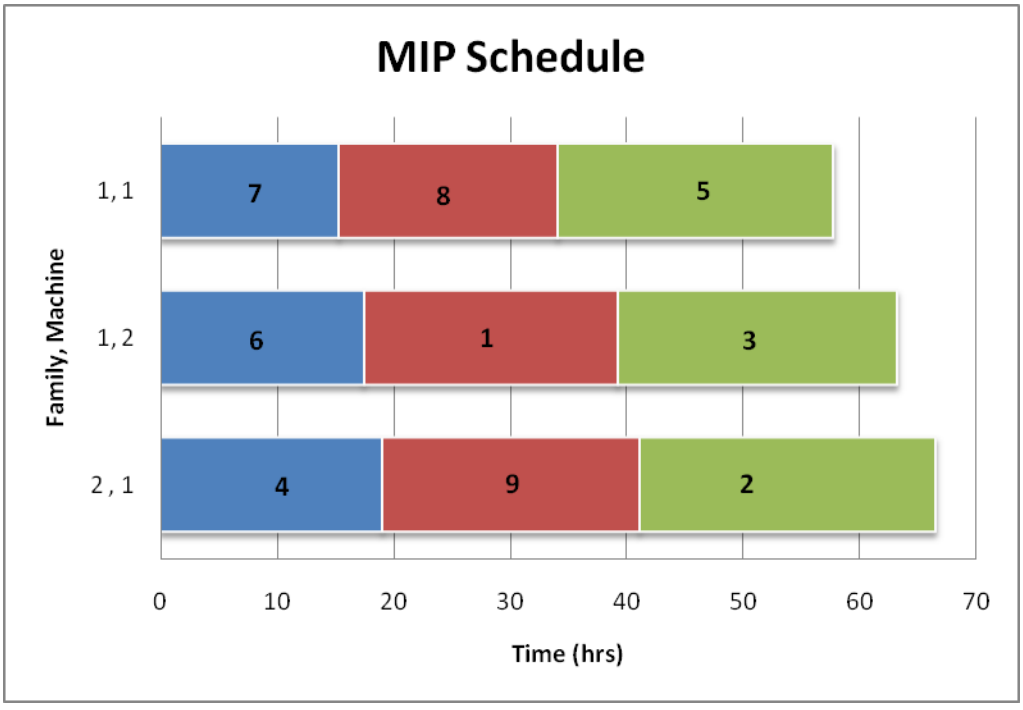
Case 1, Trial 2



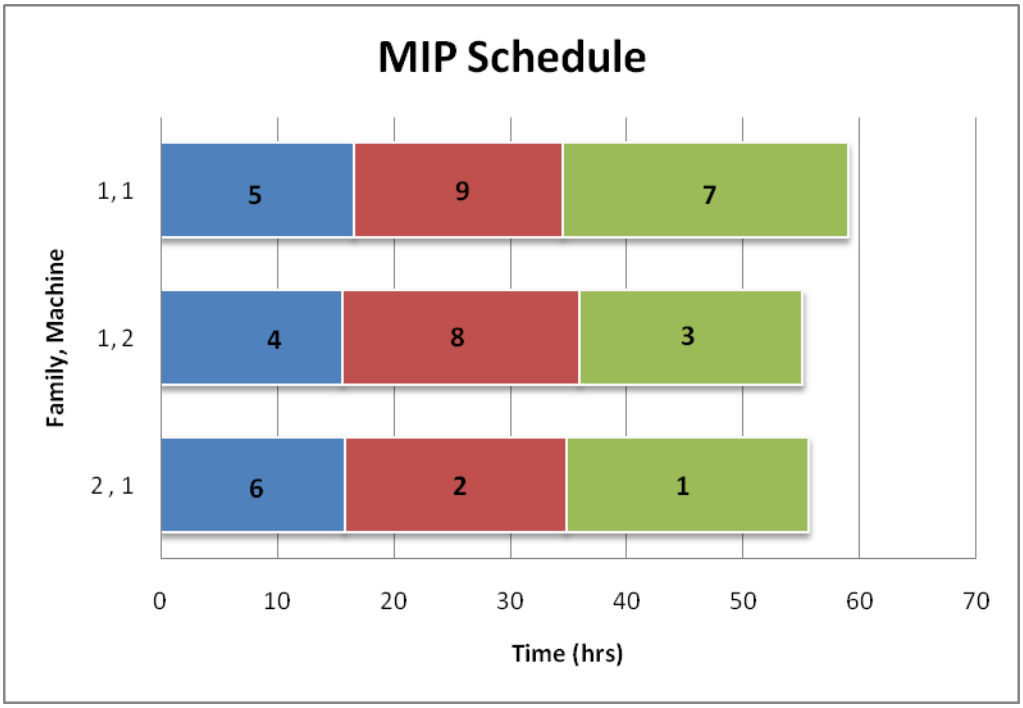
Case 1, Trial 3



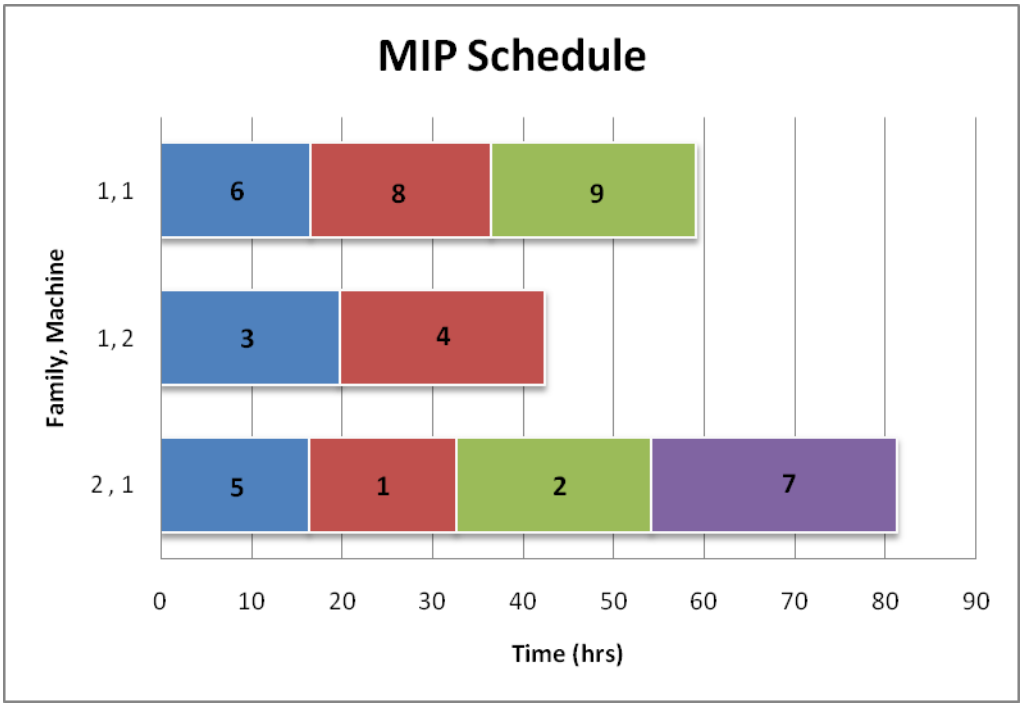
Case 2, Trial 1



Case 2, Trial 2

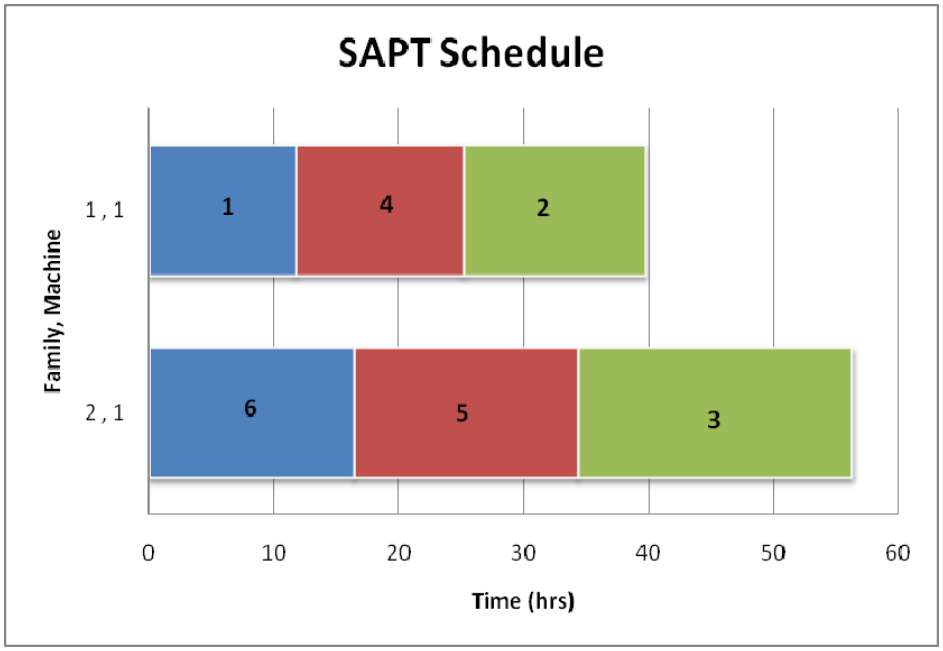


Case 2, Trial 3

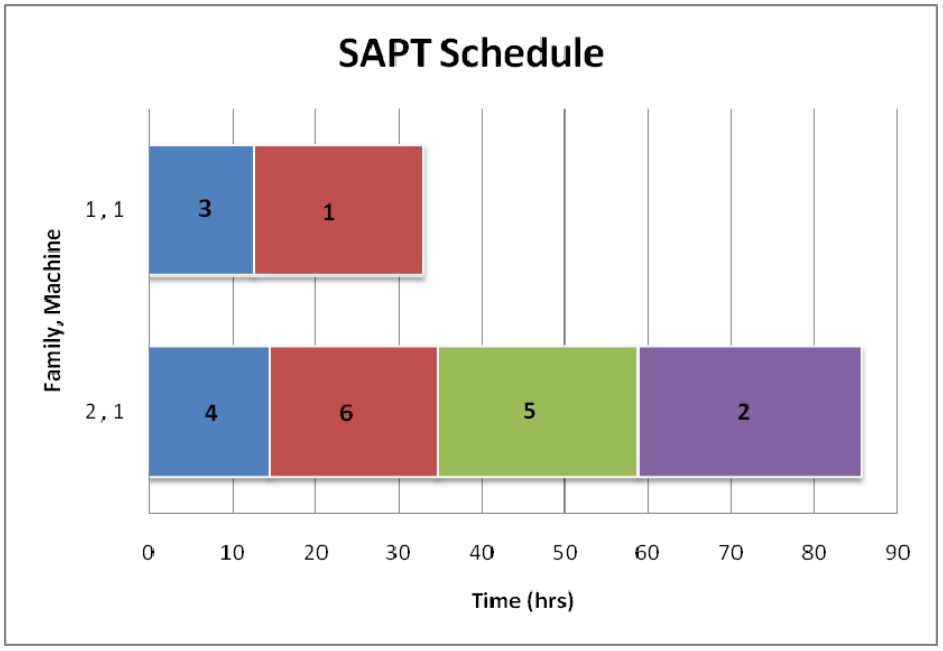


SAPT Schedule Charts

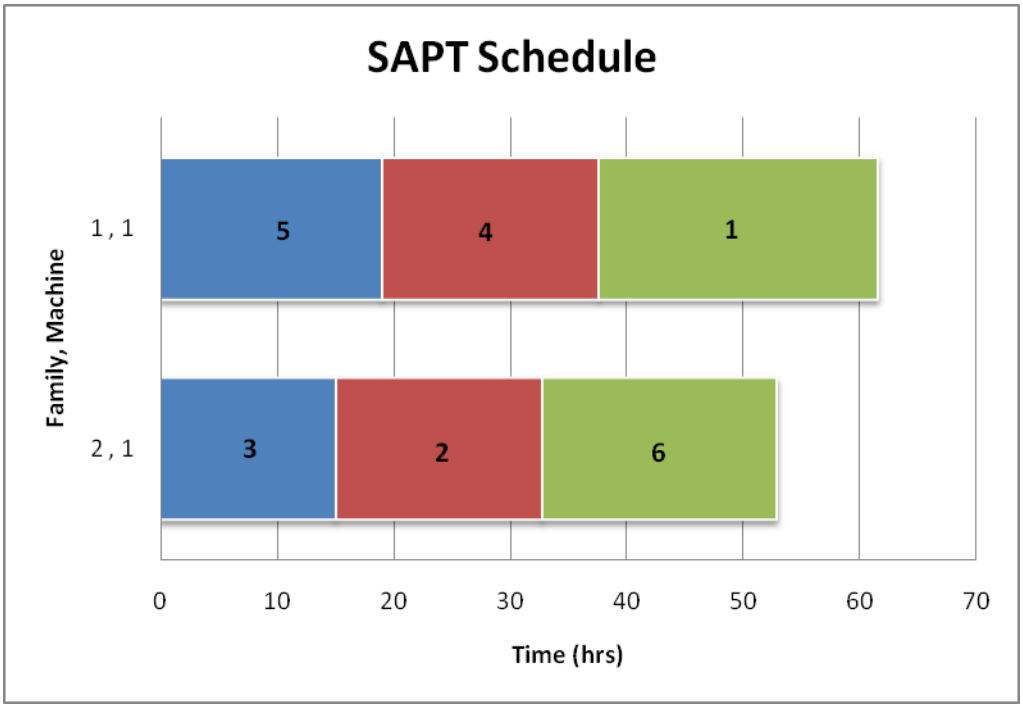
Case 1, Trial 1



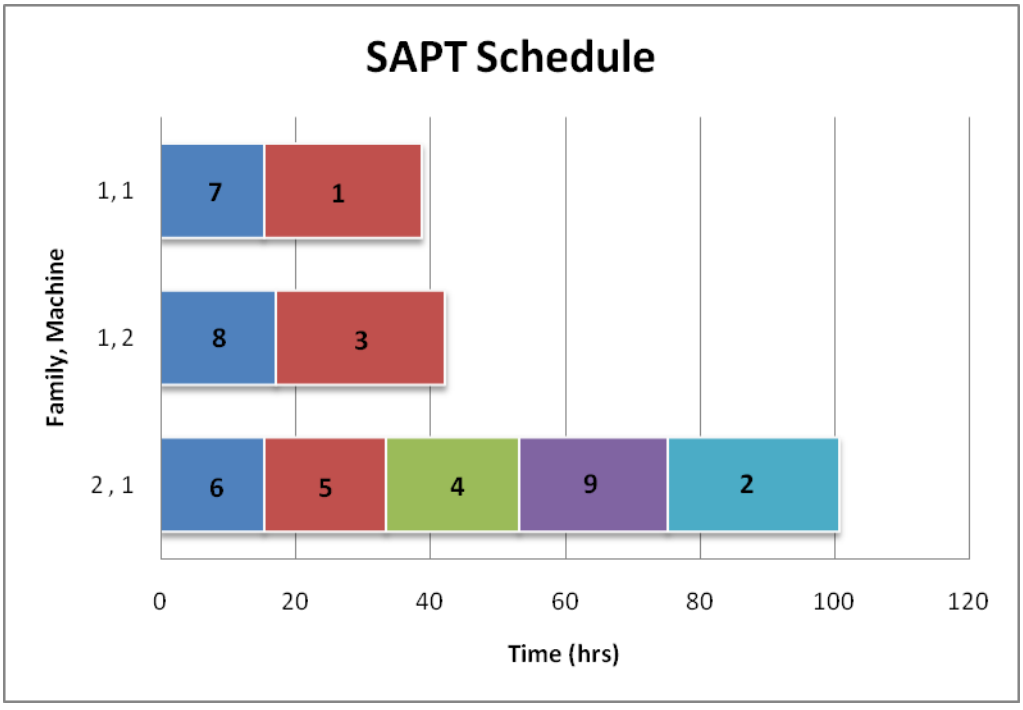
Case 1, Trial 2



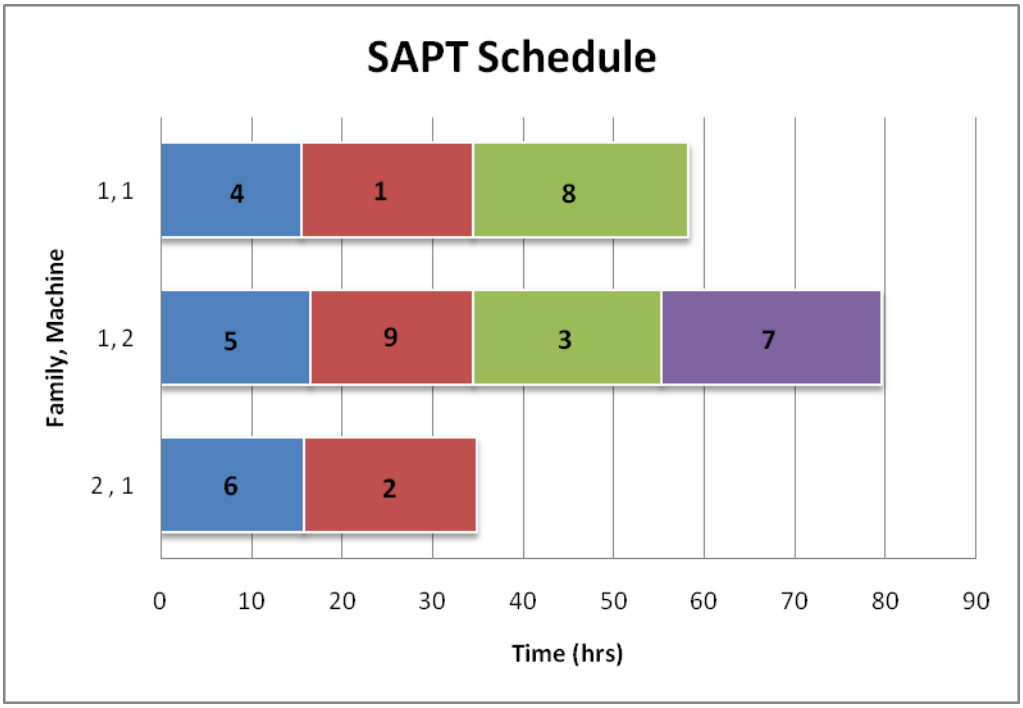
Case 1, Trial 3



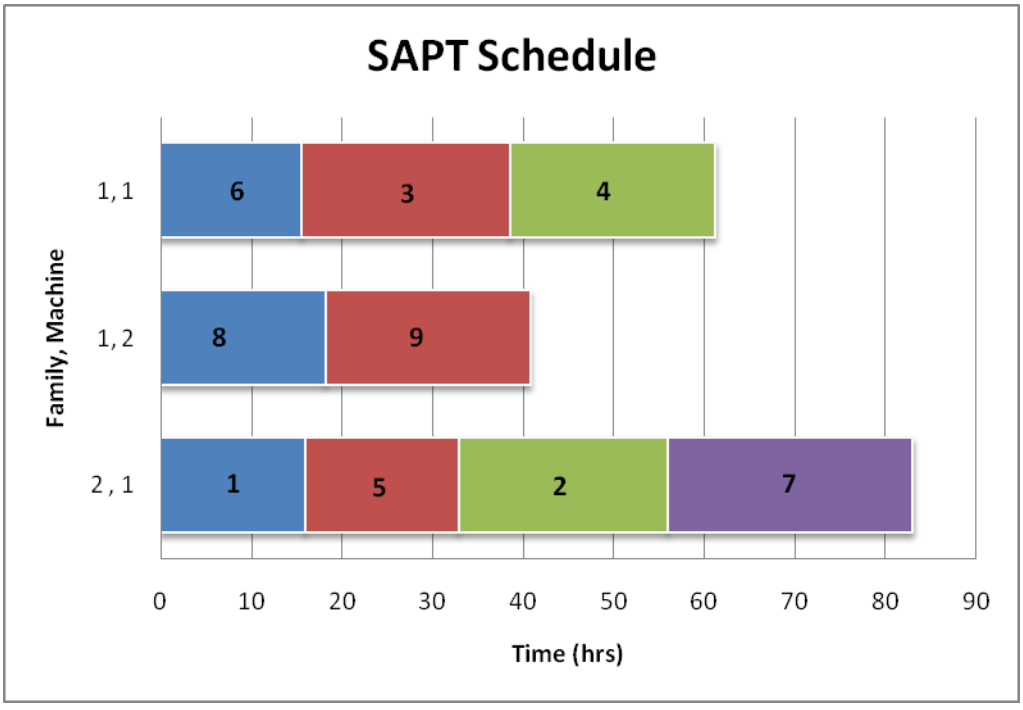
Case 2, Trial 1



Case 2, Trial 2



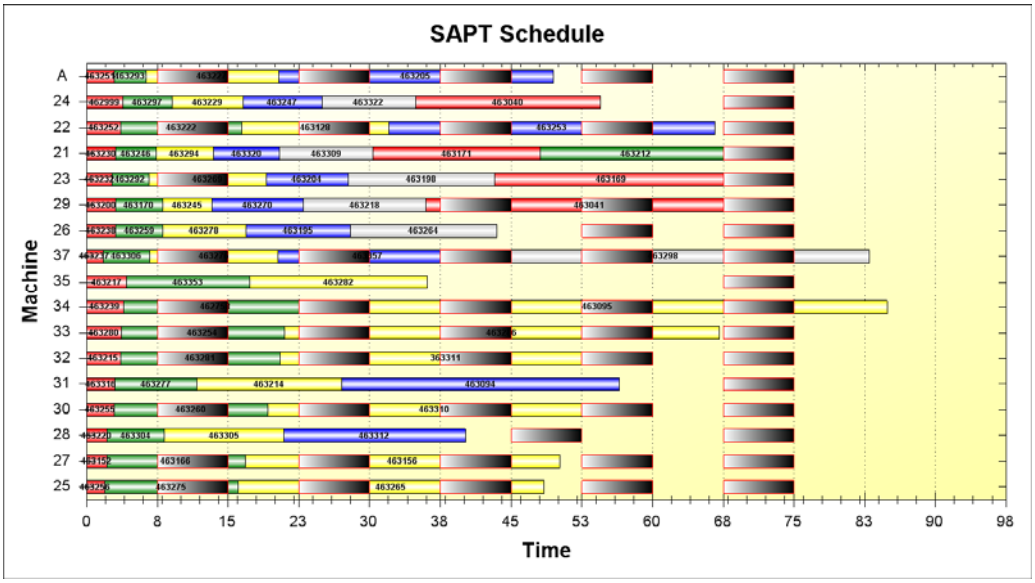
Case 2, Trial 3



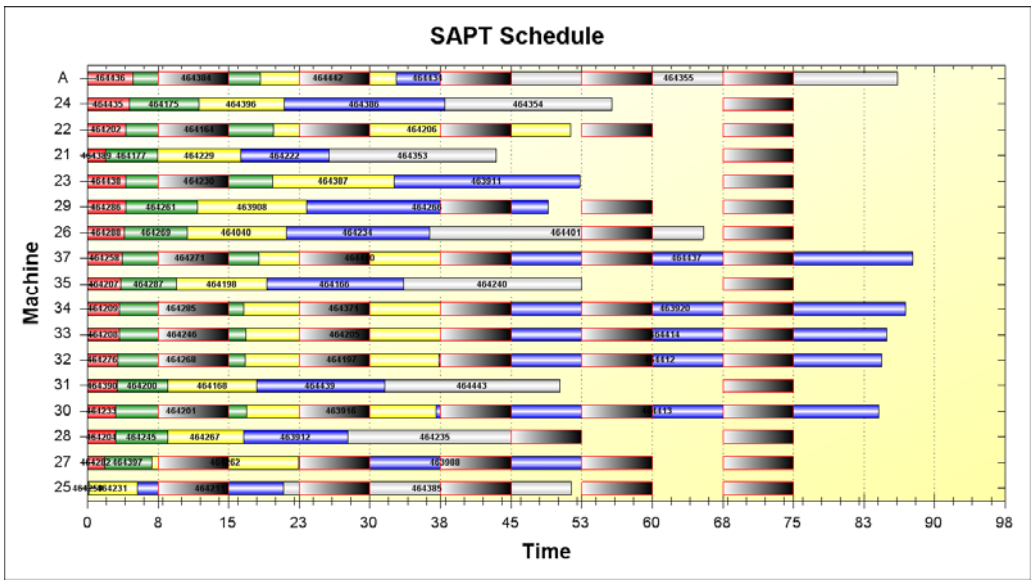
APPENDIX D: SCHEDULE CHARTS, EXPERIMENTATION PHASE III

SAPT Algorithm

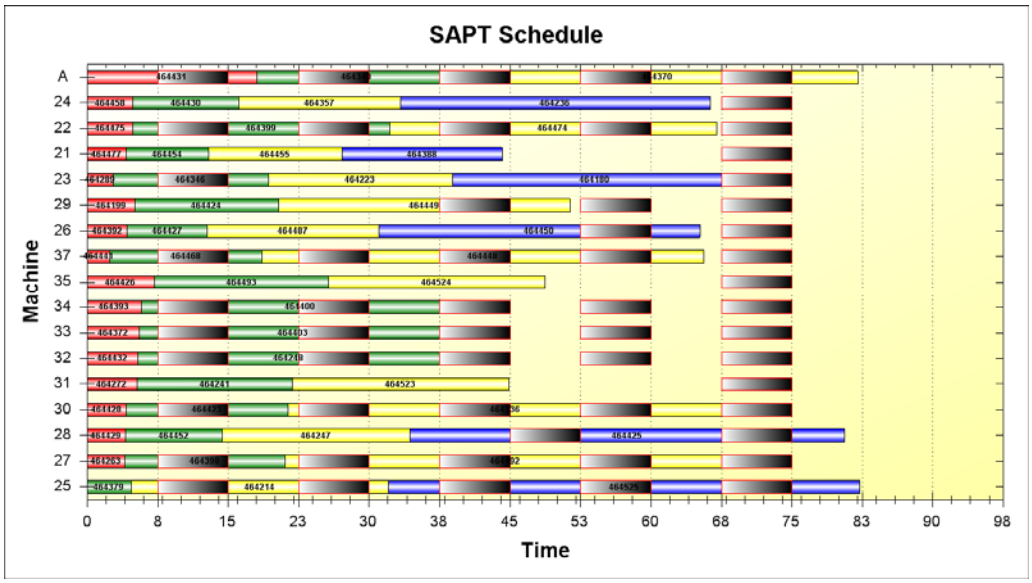
Bucket Set # 2



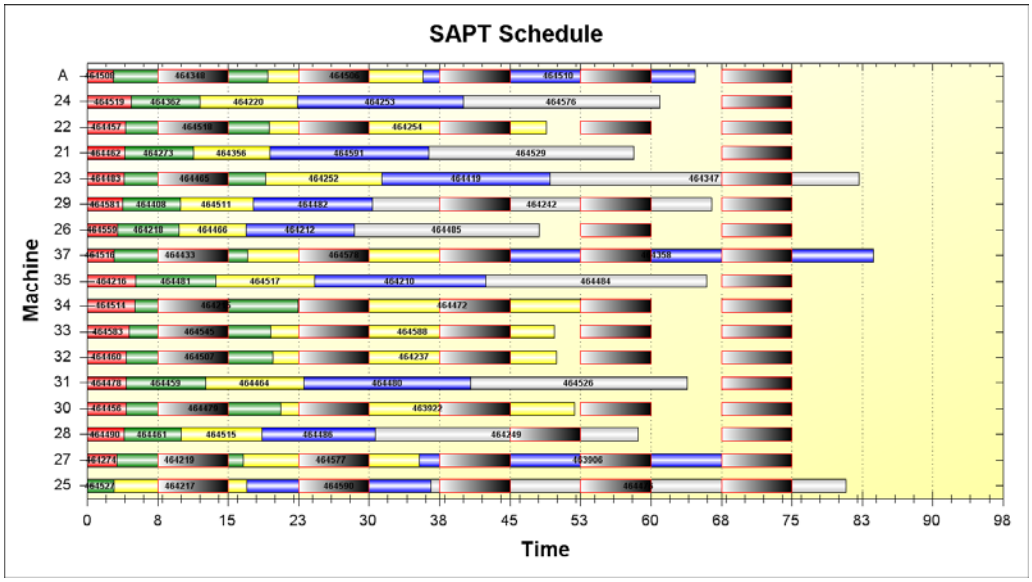
Bucket Set # 3



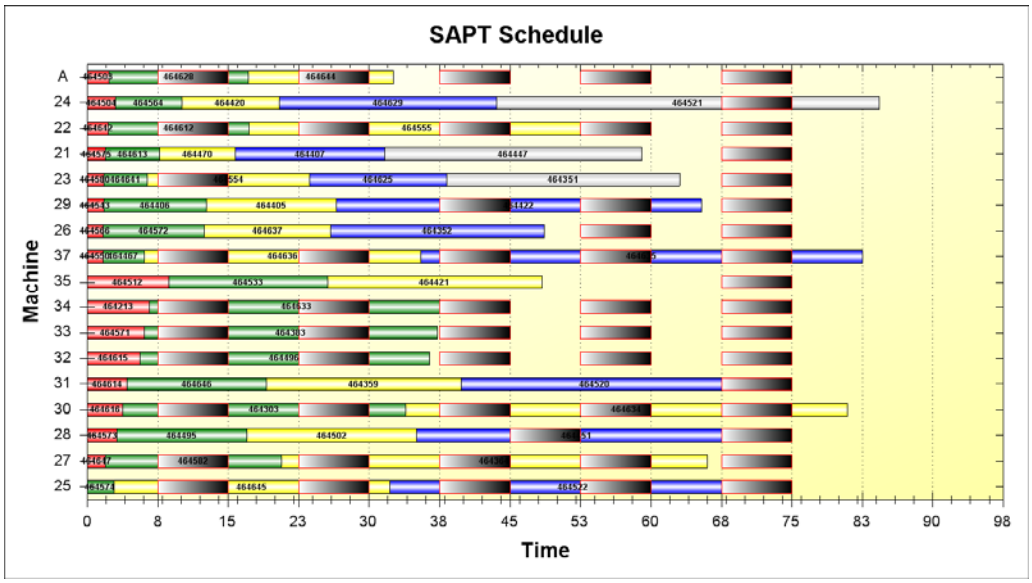
Bucket Set # 4



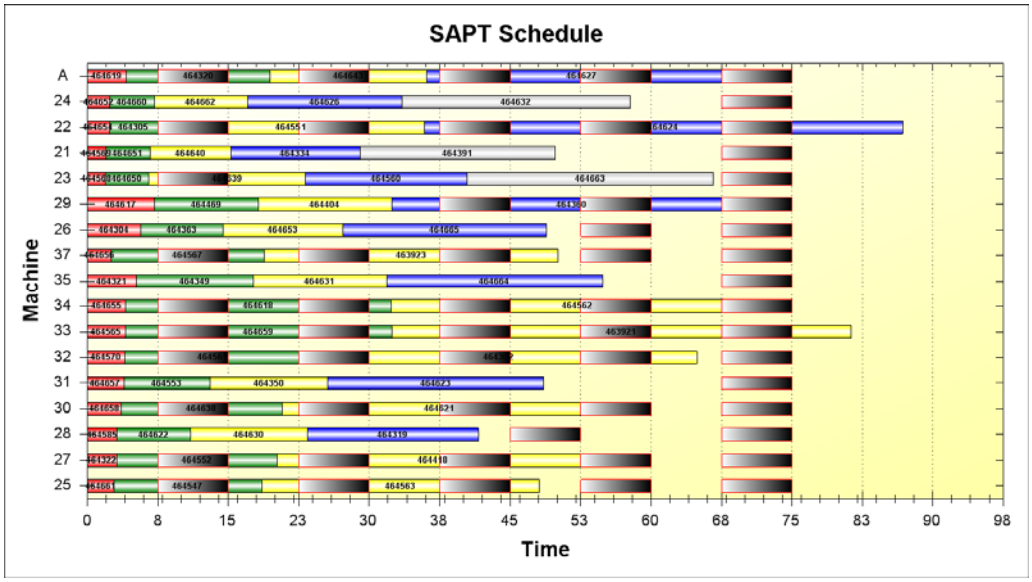
Bucket Set # 5



Bucket Set # 6

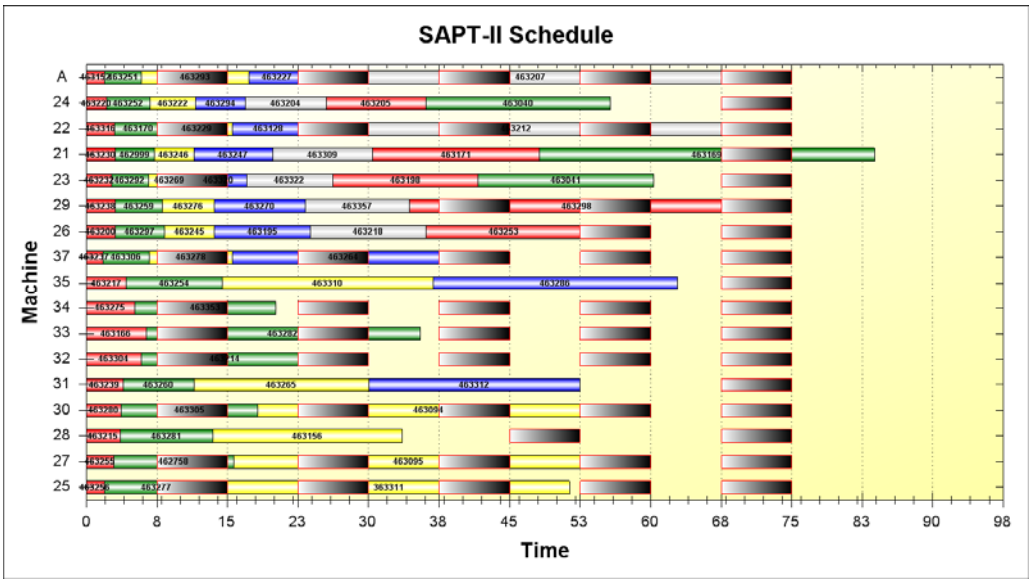


Bucket Set # 7

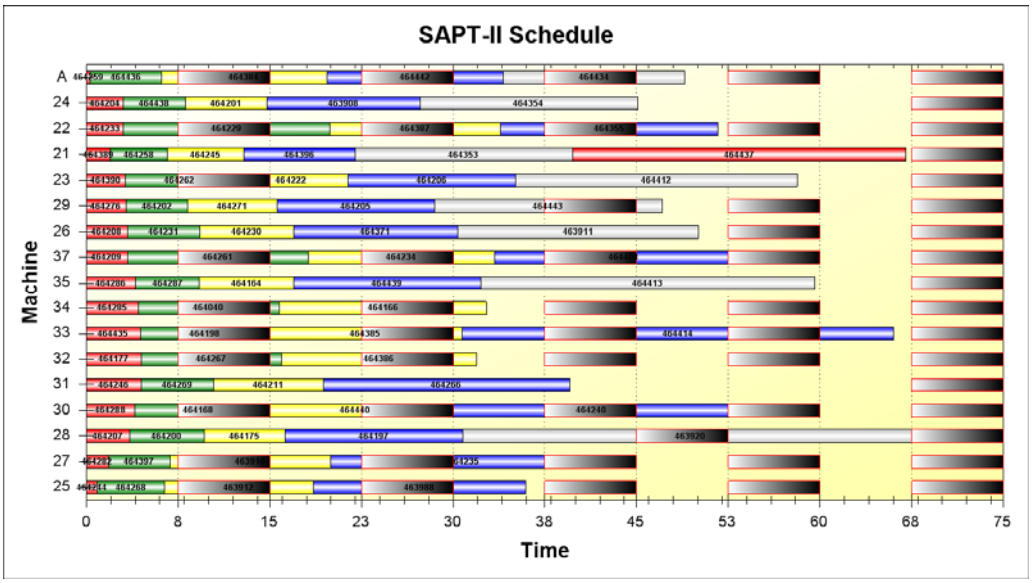


SAPT-II Algorithm

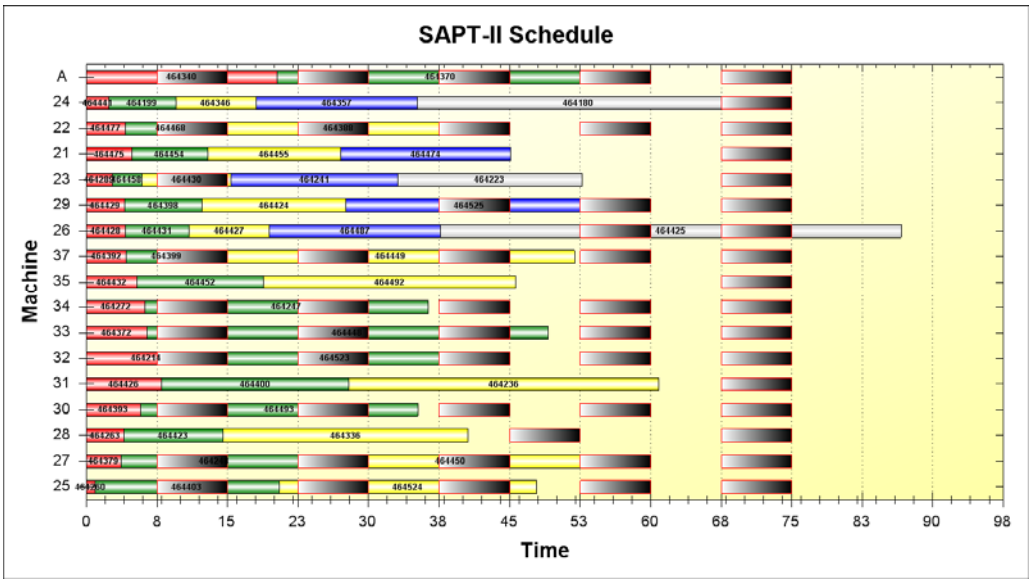
Bucket Set # 2



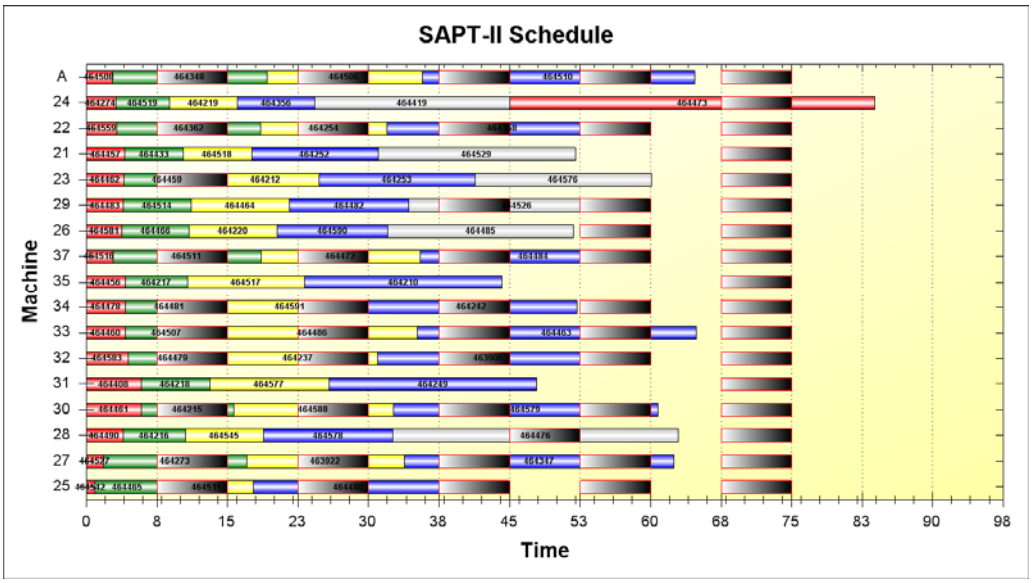
Bucket Set # 3



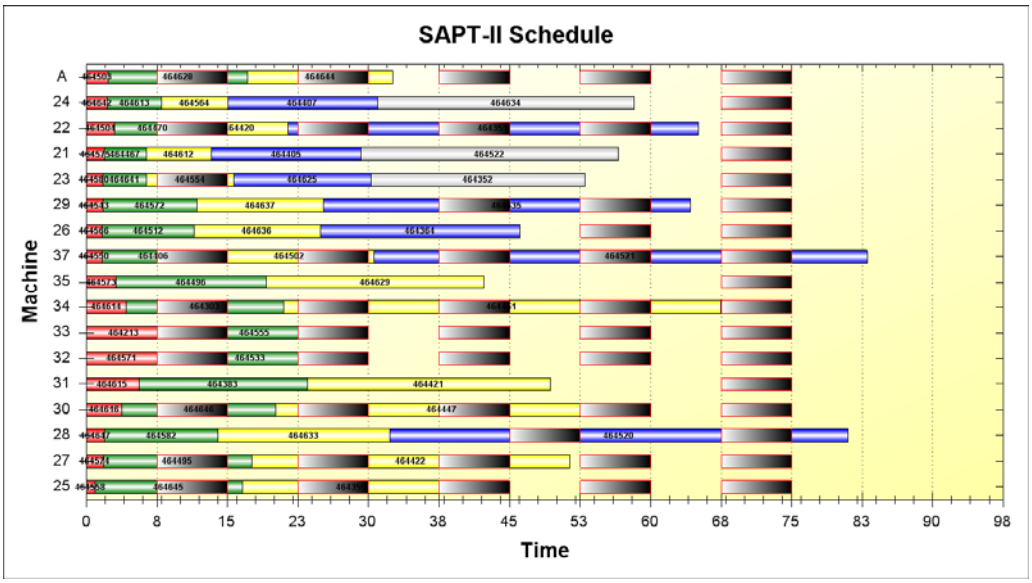
Bucket Set # 4



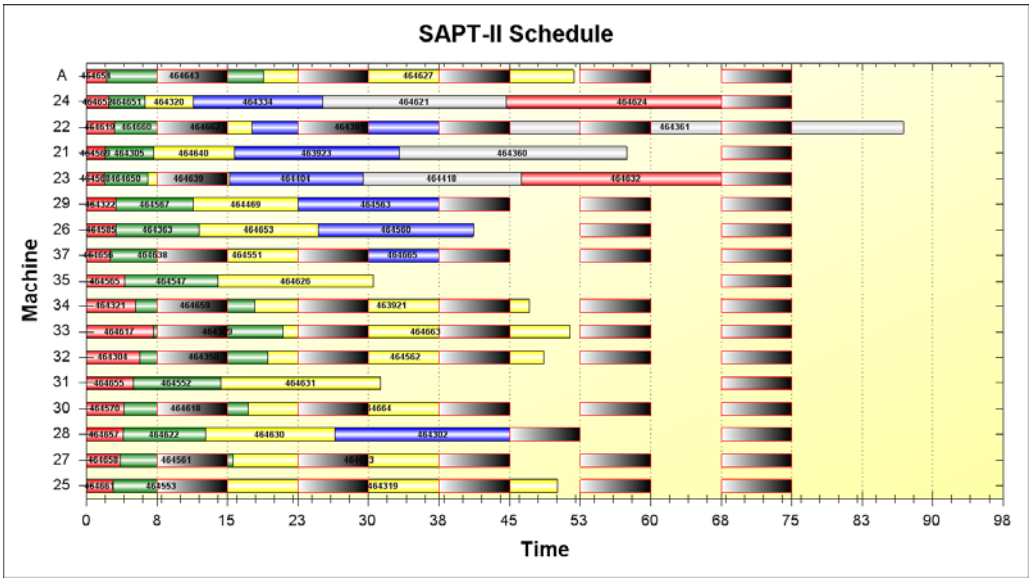
Bucket Set # 5



Bucket Set # 6



Bucket Set # 7



REFERENCES

Allahverdi, A., C.T. Ng, T.C.E. Cheng, and M. Y. Kovalyov. "A survey of scheduling problems with setup times or costs." *European Journal of Operational Research* 187, no. 3 (June 2008): 985-1032.

Allahverdia, A., J. N. D. Gupta, and T. Aldowaisan. "A review of scheduling research involving setup considerations ." *Omega* 27, no. 2 (April 1999): 219-239.

Al-Salem, A. "Scheduling to minimize makespan on unrelated parallel machines with sequence dependent setup times." *Engineering Journal of the University of Qatar* 17 (2004): 177–187.

Arnaout, J.-P.M., and G. Rabadi. "Minimizing the total weighted completion time on unrelated parallel machines with stochastic times." *Winter Simulation Conference.* 2005. 2141-2147.

Chang, P.-C., S.-H. Chen, and K.-L. Lin. "Two-phase sub population genetic algorithm for parallel machine-scheduling problem." *Expert Systems with Applications* 29, no. 3 (October 2005): 705-712.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6, no. 2 (April 2002): 182-197.

Gao, L., C. Wang, D. Wang, Z. Yin, and S. Wang. "A production scheduling system for parallel machines in an electrical appliance plant." *Computers and Industrial Engineering* 35, no. 1-2 (October 1998): 105-108.

Garey, M.R., and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness.* San Francisco, CA.: Newman Press, 1979.

He, Y., and C.-W. Hui. "Genetic algorithm based on heuristic rules for high-constrained large-size single-stage multi-product scheduling with parallel units." *Chemical Engineering and Processing: Process Intensification* 46, no. 11 (November 2007): 1175-1191.

Kim, D.-W., K.-H. Kim, W. Jang, and F. F. Chen. "Unrelated parallel machine scheduling with setup times using simulated annealing." *Robotics and Computer Integrated Manufacturing* 18 (2002): 223–231.

Kurz, M. E., and R. G. Askin. "Heuristic scheduling of parallel machines with sequence-dependent set-up times." *International Journal of Production Research* 39, no. 16 (2001): 3747 - 3769.

Lam, K., and W. Xing. "New trends in parallel machine scheduling." *International Journal of Operations & Production Management* 17, no. 3 (1997): 326-338.

Leung, J. Y.-T., and C.-L. Li. "Scheduling with processing set restrictions: A survey." *nternational Journal of Production Economics* 116, no. 2 (December 2008): 251-262.

Li, K., and S.-L. Yang. "Non-identical parallel-machine scheduling research with minimizing total weighted completion times: Models, relaxations and algorithms." *Applied Mathematical Modelling* 3 (2009): 2145–2158.

Lin, C.-H., and C.-J. Liao. "Makespan minimization for multiple uniform machines." *Computers & Industrial Engineering* 54, no. 4 (2008): 983-992.

Mendes, A. S., F. M. Müller, P. M. França, and P. Moscato. "Comparing meta-heuristic approaches for parallel machine scheduling problems." *Production Planning and Control* 13, no. 2 (March 2002): 143-154.

Mokotoff, E. "Parallel machine scheduling: A survey." *Asia-Pacific Journal of Operational Research* 18 (November 2001): 193-242.

Mokotoff, E., and P. Chretienne. "A cutting plane algorithm for the unrelated parallel machine scheduling problem." *European Journal of Operational Research* 141 (2002): 515-525.

Muruta, T., and H. Ishibuchi. "MOGA: Multi-objective genetic algorithm." *Proceedings of second IEEE international conference on evolutionary computation*, 1996: 170-175.

Nessah, R., C. Chu, and F. Yalaoui. "An exact method for Pm / sds, ri / ∑i = 1,n Ci problem." *Computers and Operations Research* 34, no. 9 (September 2007): 2840-2848.

Piersma, N., and W. Van Dijk. "A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search." *Mathematical and Computer Modelling* 24, no. 9 (November 1996): 11-19.

Rabadi, G., R. J. Moraga, and A. Al-Salem. "Heuristic for the unrelated parallel machine scheduling problem with setup times." *Journal of Intelligent Manufacturing* 17 (2006): 85-97.

Suresh, V., and D. Chaudhuri. "Bicriteria scheduling problem for unrelated parallel machines." *Computers and Industrial Engineering* 30, no. 1 (1996): 77-82.

Suresh, V., and D. Chaudhuri. "Scheduling of unrelated parallel machines when machine availability is specified." *Production Planning and Control* 7, no. 4 (1996): 393-400.

Weng, M. X., J. Lu, and H. Ren. "Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective." *International Journal of Production Economics* 70 (2001): 215-226.

Yalaoui, F., and C. Chu. "An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times." *IIE Transactions* 35, no. 2 (February 2003): 183-190.

Yu, L., Shih, Pfund, M. H. M., W. M. Carlyle, and J. W. Fowler. "Scheduling of unrelated parallel machines: an application to PWB manufacturing." *IIE Transactions* 34 (2002): 921-931.

Zhu, X., and W. E. Wilhem. "Scheduling and lot sizing with sequence-dependent setup: A literature review." *IIE Transactions* 38 (2006): 987-1007.

VITA

Mohannad Abdelrahman Shuaib

Date and Place of Birth

November 19, 1980; Khartoum, Sudan

Education

Bachelors of Science in Electrical Engineering
King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, 2003

Professional Experience

Laundry Packing Operations Manager (July 2006 – August 2007)
Procter & Gamble, Dammam, Saudi Arabia

Laundry Packing Process & Reliability Manager (June 2004 – July 2006)
Procter & Gamble, Dammam, Saudi Arabia

Awards and Honors

1. Kentucky Graduate Scholarship
   University of Kentucky, Lexington, KY, Fall 2007 – Spring 2009

2. Outstanding performance for Band I management level
   Procter & Gamble, Dammam, Saudi Arabia, 2006

3. Institute of Electrical and Electronics Engineers (IEEE) academic excellence award
   King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, 2003

4. Saudi Electronics Company (SEC) academic excellence
   King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, 2003

Publication

Mohannad Shuaib, Fazleena Badurdeen, "Job Scheduling in Associative Parallel Machines to Minimize Average Flow Time", ISDSI 2009 Conference, Mumbai, India, January 2009.