THE HONG KONG POLYTECHNIC UNIVERSITY
DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

A TWO – PHASE APPROACH TO SOLVE MANPOWER SCHEDULING AND TASK ASSIGNMENT PROBLEM IN AIRCRAFT MAINTENANCE INDUSTRY

WANG Yingna

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Philosophy

February 2006

CERTIFICATE OF ORIGINALITY

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

_________________________________(Signed)

__________WANG Yingna__________(Name of student)

ABSTRACT

This research study is aimed to propose an integrated manpower scheduling and planning methodology to solve the assignment problems in the aircraft maintenance industry. The proposed methodology consists of a sequential two-phase approach to the capture various human and technical factors in aircraft maintenance organizations. In view of the scheduling difficulties facing this industry over the course of an operating day, and across a planning horizon, such that the limited resources and constraints could be satisfied, the problem thus entails a specially designed approach of finding optimal assignments of qualified manpower to a series of deterministic daily maintenance tasks and satisfying simultaneously all the relevant scheduling considerations. Previous researchers have formulated the solution into two separate approaches – rostering and assignments. In this research, we have designed and developed a sequential two-phase model which is more effective and efficient than obtaining the solution separately.

The first phase is to assess utility through a fuzzy multiple attribute decision making process, considering each task assignment problem separately. In this phase, for each engineer/task combination, a utility index is calculated from analytical hierarchy process using fuzzy triangular numbers. They are then defuzzified to a final crisp utility index. A hierarchy structure is then constructed by grouping similar

independent task assignment, as well as decision criteria, alternatives, and scoring methods.

The second phase is to conduct optimization through an integer programming model, based on the obtained utility matrix from the first phase of calculation. A mixed integer goal programming model is built, with the multiple objectives optimized in three priority levels. Various rostering and scheduling considerations are modelled in a set of linear formulas in the planning period of seven days.

The model has been tested using different data sets collected from a case study company. Computational results from this two-phase methodology indicate that the most concerned problem objectives, namely minimization of total deviation from targeted off days, minimization of total deviation from targeted shift duration (minimization of OT), and maximization of the utilities of assigning qualified engineers to tasks have been optimized within an acceptable time. The optimal solutions provide useful suggestions to the industrial scheduler from both the aspects of weekly planning and daily planning. Moreover, the efficiency of the model for large organizations with a very large manpower size as well as task size should be further investigated.

PUBLICATIONS

Linda Wang, W.H. Ip, K.L. Yung, “Fuzzy AHP for Manpower Planning System in Aircraft Maintenance Industry”, Supply Chain Seminar – An International Conference on Logistics “unlocking supply chain potential”, June 10 – 11, 2004, Brisbane, Australia.

Y.N. Wang, W.H. Ip, and K.L. Yung, “An Intelligent Approach to Service Planning in Aircraft Maintenance”, 11th World Congress of International Fuzzy Systems Association (IFSA 2005), July 28 – 31, Beijing, China.

Y.N. Wang, W.H. Ip, and K.L. Yung, “A Manpower Model for Aircraft Maintenance Using the Fuzzy Multiple Attribute Decision Making Approach”, submitted to Computers in Industry.

ACKNOWLEDGEMENT

I would like to express my gratitudes to my thesis advisor Dr. W.H. Ip for his patient guidance and his reliable support have greatly helped me in the completion of my research work. I would also like to thank Professor K. L. Yung, who is my co-supervisor, for the many valuable things he has taught me.

Thanks are also presented to Dr. Ping Ji and Dr. C.K. Kwong, for their sincere and valuable advice and recommendations during my research progress.

I would like to offer particular thanks to Mr. Sam Lam and Miss. C. L. Lai for their continuous valuable advice, information collections and recommendations throughout my research.

A scholarship provided by the Research Committee of The Hong Kong Polytechnic University is also gratefully acknowledged.

Last but not the least, my deepest appreciation goes to my friends for their support and encouragement throughout my study.

WANG Yingna

February 2006

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LM	Line Maintenance
BM	Base Maintenance
AMO	Aircraft Maintenance Organization
LP	Linear Programming
LSAP	Linear Sum Assignment Problem
GAP	Generalized Assignment Problem
SPP	Set Partitioning Problem
FJSP	Fixed Job Scheduling Problem
IS	Interval Scheduling
OFJSP	Operational Fixed Job Scheduling Problem
TFJSP	Tactical Fixed Job Scheduling Problem
MADM	Multiple Attribute Decision Making
FMADM	Fuzzy Multiple Attribute Decision Making
AHP	Analytic Hierarchy Process
NLP	Non-linear Programming
IP	Integer Programming
GP	Goal Programming
MIP	Mixed Integer Programming
PIP	Pure Integer Programming

CHAPTER 1

INTRODUCTION

1.1 Research Background

Safety management in airline industry has been receiving great attention all the time. Regulations and requirements, with maintenance as the major element, form the basis of safety management. Hence commitment to safety should be a part of the organization’s spirit in the world, including aircraft manufactures, airline operators, aviation authorities and organizations that provide aircraft maintenance service.

The first consideration to guarantee safety is to ensure that the maintenance has been performed with high quality, which very often depends on the performance of the maintenance operators. Researchers pointed out (Yang et al., 2003) that effective aircraft maintenance plans are directly related to improving flight safety, apart from reducing operating costs. One possible way to ensure high quality of maintenance is to measure various possible outcomes of different maintenance plans before task implementation. It is of special importance to the type of Line Maintenance (LM) performed on spot, for which all required resources, including manpower, equipment, spares and tooling must be taken to the parking bay of the aircraft. In addition, the maintenance work duration is bounded by the ground time of the aircraft, resulting in fixed job intervals.

During problem survey, the author found that the current planning practice in a Aircraft Maintenance Organization (AMO) is firstly constructing monthly duty rosters by considering simply some day-off constraints, and then assigning appropriate manpower to maintenance task based on on-hand rosters when the planning process goes into daily bases. These procedures are carried out by experienced experts in the organization before each planning horizon. The global optimization by considering simultaneously monthly roster construction and daily tasks assignments could thus not be obtained. Moreover, decision makers tend to make intuitive decisions involving cognitive leaps, and unlikely to consider the range of choices thoroughly. Thus the whole planning and scheduling process mainly rely on decision-maker's own experience without a systematic approach, and is apt to be biased due to limited personnel experience, knowledge and perception. In addition, this kind of expert knowledge to schedule manpower remains unstable if it is not captured by a standard systematic process. Thus there has been an eminent need to set up a systematic and effective methodology for this combined problem.

On the other hand, as far as the general personnel scheduling problem is concerned, many researchers such as Ernst et al. (2004) suggest decomposing the problem into several separate models, leaving the task assignment afterwards just before the actual operating day after carrying out days-off and shift scheduling to reduce the problem complexity. Therefore there has been a lack of studies into the combination of personnel scheduling and task assignment considerations, which has thus lead to the interests of my research.

1.2 Problem Statement

The intrinsic complexity of the combined planning problem makes it difficult to model the problems and achieve satisfactory optimization through one-step procedure. Through literature review and problem investigation, it was found that totally different characteristics of these two problems necessitate each own corresponding model. Still major issues are left to be solved, which are summarized as follows:

- In what way the weekly/monthly personnel scheduling problem and daily task assignment problem is integrated
- How to measure the efficiency of the resultant approach

1.3 Overview of Aircraft Maintenance

1.3.1 Significance of Aircraft Maintenance

Air transport has been more and more popular among the passengers in the world. The first reason is probably the convenience and speed of air transport. Another important reason is its improved safety. Despite that large transport accident rates have fallen over the last 30 years, if air traffic is as predicted, and doubles over the next 10 to 15 years, this low accident rate would also double (Thompson, 2001).

The emphasis on reducing maintenance errors when performing the maintenance and the adoption of proper maintenance management systems will lower the chance of accident enormously. The first consideration is to ensure that the maintenance has been performed with high quality, which very often depends on the performance of maintenance operators. It is therefore crucial to assign suitable operators who have the required performance level to carry out the maintenance tasks. Accordingly, Yang et al. (2003) pointed out that effective aircraft maintenance plans should not only aim at reducing operating costs, but are also directly related to improving flight safety.

1.3.2 Aircraft Line Maintenance

Aircraft maintenance has long been an important issue in scheduling flights. Maintenance requirements, which are determined by aircraft manufacturers, limit the number of trips aircraft can fly. These limitations generally have three forms: the number of hours flown, the number of cycles (landings, engine starts, and so forth), and the number of days since the last maintenance service (Martin, Jones and Keskinocak, 2003).

In the Hong Kong International Airport, there are two kinds of aircraft maintenance provided – *Base Maintenance* and *Line Maintenance*. They differ in time and severity. In order to ensure its air-worthiness, each aircraft has to go through some routine checking and defect rectification to the maximum extent during its ground time of each transit, which is called *Line Maintenance* (LM). According to Hong Kong Airport Authority (2002), Line Maintenance services include routing servicing of aircraft performed during normal turnaround periods and regularly scheduled layover periods. It contains full technical log certification and aircraft release at the Hong Kong International Airport. During Line Maintenance, these types of labor are required: airframe/engine engineer, airframe/engine mechanics, avionics engineer, avionics mechanics, male cleaner, female cleaner, and driver. It usually takes on average 1 to 2 hours according to detailed work package for each flight.

On the other hand, *Base Maintenance* covers all airframe maintenance services. It refers to heavy maintenance services such as major structural and avionics modifications, cabin refurbishment and painting services which require the use of hangers (Hong Kong Airport Authority, 2002).

Unlike Base Maintenance, Line Maintenance has to be performed on spot. All required resources, including manpower, equipment, spares and tooling must be taken to the parking bay of the aircraft. Moreover, the maintenance work duration is bounded by the ground time of the aircraft. Otherwise punctuality would be affected incurring extra penalty costs. Unlike the practice that maintenance work is mainly performed by maintenance department with the airline operators in most of the airports overseas, both Line Maintenance and Base Maintenance are franchised to Aircraft Maintenance Organization (AMO) at the New Hong Kong International

Airport. So Line Maintenance has become crucial to both relevant airline operators and AMO, especially the latter.

The aim of this research is concerned with solving engineer scheduling and task assignment problems for line maintenance within scheduled maintenance. It is applied to the case of line maintenance services within one of the AMOs, China Aircraft Service Limited (CASL) in the New Hong Kong International Airport.

1.3.3 Aircraft Maintenance Requirements

In the European practice, for example, Civil Aviation Authority (CAA) in Britain grants a license “subject to such conditions as it thinks first, upon its been satisfied that an applicant is a fit person to hold the license, and has furnished such evidence and passed such examination and tests the CAA may require for the purpose of establishing that the applicant has sufficient knowledge, experience, competence and skill in aeronautical engineering” (Friend, 1992).

In Hong Kong, the local aircraft maintenance license is available from the Flight Standards and Airworthiness Division of Hong Kong Civil Aviation Department (CAD). It is a set of requirements for the qualification of aircraft maintenance staff by the issue of an Aircraft Maintenance License.

The license is divided broadly between Mechanical and Avionic trade disciplines. Although in view of the various technologies and combinations applicable to certain aircraft, the Mechanical license category is further subdivided. In addition, there are various levels within the license, which allow the holder to be authorized to perform certain roles within the category of either Line or Base maintenance. They reflect different levels of task complexity and are supported by different standards of experience and knowledge. There is however no reason why an individual cannot hold a combination of license categories. The categories within the license are shown in the Table 1.1.

Table 1.1 Categories within Aircraft Maintenance License from the Flight Standards and Airworthiness Division of Hong Kong Civil Aviation Department (CAD)

Categories	Description
Category A	- Line Maintenance Certifying Mechanic
Category B1	- Line Maintenance Certifying Engineer (Mechanical)
Category B2	- Line Maintenance Certifying Engineer (Avionic)
Category B3	- Simple Light Aeroplane Maintenance Certifying Engineer
Category C	- Base maintenance Certifying Engineer

As Category A and C are about mechanics and Base maintenance respectively, they are not considered here. Category B license is the main license qualification for aircraft maintenance staff under the Hong Kong Aviation Requirements. Category B license is further divided into categories of mechanical and avionic with sub categories. The sub categories for Category B Line Maintenance Certifying Engineer/ Base maintenance Engineer are Category B1, Category B3 and Category B2, while the former two are mechanically orientated and the latter is avionic biased.

In this research project, subcategories within Category B are not considered separately. Rather, they are seen as a unity, that is, local license for line maintenance.

It is also assumed that an engineer could either do both the mechanical work and avionic work or none of them.

On the other hand, at times, although the line maintenance is performed in Hong Kong, line maintenance licenses authorized by the aviation authorities in other districts or countries such as mainland China or the USA are required for the aircrafts or the airline operators registered in that district or country. For example, the airline operator United Airline in the USA requires that only the engineers who possess the maintenance licenses authorized by Federal Aviation Administration (FAA) in the USA could perform line maintenance for their aircrafts. The airline operators in mainland China also require the line maintenance engineers to have the maintenance licenses authorized by Civil Aviation Administration of China (CAAC).

Maintenance requires three types of resources – manpower, material and equipment. Human resources greatly increase the cost of a service, and they are by far the most variable and most difficult to control (Paz and Leigh, 1994).

Apart from maintenance license from Hong Kong Civil Aviation Department Flight Standards and Airworthiness Directives, aircraft maintenance staffing requirements are affected by other factors for different aircrafts:

- Different types of aircraft

Apart from the license for registered districts of aircrafts, there are also licenses regarding aircraft type for maintenance of certain aircraft type, e.g. B747. These licenses are endorsed with type ratings when additional training, examination and experience requirements on certain aircraft types have been satisfied.

- Different airline regulations

In many other countries, the line maintenance is done by the airlines themselves. It is different in Hong Kong, where the line maintenance is done by several aircraft maintenance organizations in the airport. Thus, licenses are authorized by airlines to allow them to do line maintenance for their aircrafts regarding airline regulations. Maintenance engineers should also have these licenses to perform line maintenance.

- Personal experience

Although the license acquisition to carry out aircraft line maintenance already requires certain years of experience in this industries, the personal experience should also be taken into consideration in the company when evaluating the candidates' suitability for performing a certain maintenance task.

- Continuous coverage

As there are always aircraft that need line maintenance throughout the whole day, the maintenance engineers should be available all the time 24 hours a day and seven days a week. In this project, the engineers are assumed available at any time when they are needed.

- Safety laws and company policy

Because of the limited resources of maintenance engineers, these two factors should be integrated into the scheduling of maintenance engineers.

1.4 Research Objectives

The prime objective of this research is to develop an innovative approach to integrate the feasibility of solving the two different types of problems in a combined way in the context of an aircraft line maintenance environment which aims at optimizing the whole planning process. The specific objectives of this research are stated below:

- To propose a solution framework for the personnel scheduling and task assignment problem, which may consist of several methodologies;
- To develop evaluation functions to measure the planning model, by considering essential constraints and intentions of the organization; and
- To test the proposed approach by using data collected from the case study organization

1.5 Thesis Outline

This thesis is divided into the following five chapters:

Chapter 1 introduces the problem background in terms of industry practice as well as research area and the motivations of the study. Besides, an overview of the focused application industry is provided – Aircraft Maintenance in terms of its significance, especially Aircraft Line Maintenance, various requirements and limitations of existing software solutions.

Chapter 2 reviews the literature in relevant problems including General Personnel Scheduling Process, Scheduling Flexibility, Personnel Assignment Problem, Airline Crew Scheduling and Rostering, Manpower Scheduling for Aircraft Maintenance and Fixed Job Scheduling Problem, followed by a short summary.

Chapter 3 gives a detailed problem description in terms of the characteristics and time constraints associated with the scheduled jobs and personnel; and then present my proposed two-phase methodology to solve the problem, which involves Fuzzy Multiple Attribute Decision Making approach for phase 1 and Integer Goal Programming approach for phase 2.

Chapter 4 presents computational results of the methodology out from Excel computation, Mathematical Programming Language and CPLEX software, which finally evaluate and validate the proposed methodology.

Chapter 5 draws conclusions of the work undertaken. It also discusses the applicability and limitations of the methodology proposed and their contributions, together with suggestions for future study.

CHAPTER 2

LITERATURE REVIEW

Relevant background knowledge has been reviewed in this chapter, including General Personnel Scheduling Process, Scheduling Flexibility, Personnel Assignment Problem, Airline Crew Scheduling and Rostering, Manpower Scheduling for Aircraft Maintenance and Fixed Job Scheduling Problem.

2.1 General Personnel Scheduling Process

Personnel/manpower/tour scheduling or rostering is generally regarded as the process of constructing work timetables for its staff so that an organization can satisfy the demand for its goods or services (Ernst et al., 2004; Pinedo and Chao, 1999). From another perspective, personnel scheduling refers to the problem of the assigning employees to shifts and duties over a scheduling period, which is usually a week/fortnight/month, so that both organizational and personal constraints are satisfied as much as possible. Thus the output of a tour scheduling problem is usually the assignment of employees to various tours specifying the days off and the working days with daily shift start time, shift lengths, and break placements of the work day. These problems arise in both manufacturing and services industries, such as scheduling production workers, hospital nurses, airline crew, bus driver, telephone operators, call center, policeman and so on. They differ in nature for distinct application areas.

In manufacturing sectors, labor requirements rarely fluctuate so much over time because of regular operations. So schedules for production workers are more standardized. Yet, in service sectors, the operations are often prolonged and irregular, and the staff requirements fluctuate over time. The schedules are typically subject to various constraints dictated by equipment requirements, union rules, and so on (Pinedo and Chao, 1999).

Properly planned personnel and optimized schedules could reduce operation costs and thus improve benefits, but require correct models reflecting practical considerations. All industrial regulations associated with the relevant workplace agreements must be observed during the process (Ernst et al., 2004). Optimization of personnel scheduling is normally defined in terms of minimizing costs, meeting employee preferences, distributing shifts equitably among employees and satisfying all the workplace constraints.

Survey in the area of personnel rostering/scheduling has been carried out by Aggarwal (1982), Bechtold et al. (1991), Glover and McMillan (1986), and Tien and Kamiyama (1982). Three types of classification of personnel scheduling process have been proposed in the literature. Tien and Kamiyama (1982) partition the general personnel scheduling problem in a common framework into five separate but related stages, namely temporal staff requirements, total staff requirements, recreation and leave, work schedules, and shift schedules. Some general, stage-specific models then served to categorize the various algorithms from a stage-specific, problem formulation perspective (Tien and Kamiyama, 1982).

Another commonly studied general labor scheduling decomposition is days off, shift, and tour scheduling (Bailey, 1985). Ernst et al. (2004) presented a more general classification, which consisted of a number of modules associated with the processes of constructing a roster. These modules are demand modeling, days off scheduling, shift scheduling, line of work construction, task assignment and staff assignment. Actually the majority of past researches adopt the same practice: firstly, determine the sufficient number of staff and shift starting time throughout the day in order to cover the varying demand, which is recognized as workforce allocation problem (Mason et al., 1998); secondly, construct the daily stretches of work for individual staff, recognized as roster construction, which includes days-off, staff preferences and other regulation individual considerations.

Figure 2.1 outlines the rostering progress proposed by Ernst et al. (2004), consisting of a step-by-step procedure suggested by six modules. However the development of a particular roster not necessarily involves all the six modules, depending on applications. Some modules may also be joined into one procedure according to its own requirements; seen the unique characteristics of different modules, different models and solution techniques have to be adopted for each module.

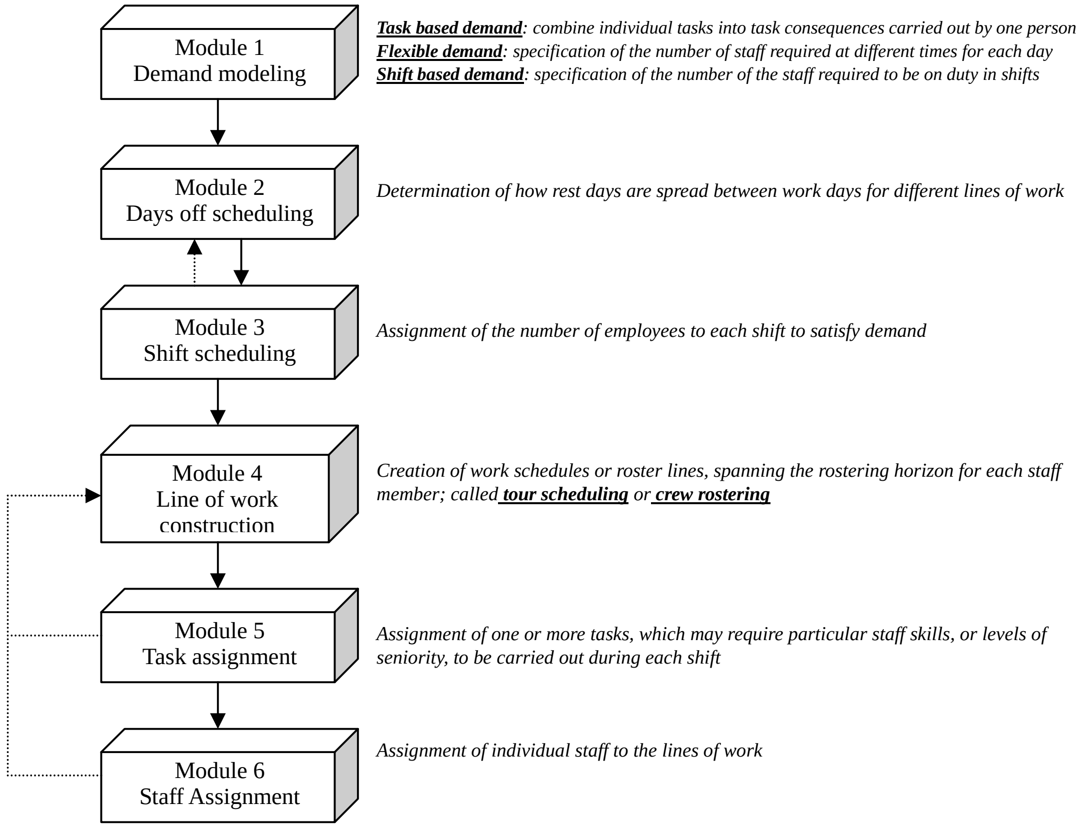


Figure 2.1 Rostering progress proposed by Ernst et al. (2004)

The directions of arrows do not necessarily reflect the logic sequences between modules. The output of a module does not necessarily become the input of the next module. Some of the processes are actually related. So, other sequences or decompositions or integrations may be applied. The dashed arrow between *Module 2* and *Module 3* implies a likely reverse sequence. Other dashed arrows from *Module 5/6* to *Module 4* mean that task/staff assignments can also be carried out when constructing lines of work.

From the computational aspect, it is usually not practical to treat simultaneously all the modules to generate a roster. So decomposing the problem into several separate modules makes it more tractable (Ernst et al., 2004). To illustrate, task assignment could be considered after the days-off and shifts scheduling have been carried out, until just before the actual operation day. This practice could be made when details of tasks to be performed including their requirements, starting time and duration are known before hand. More on relevant assignment problems are reviewed in the next subsection.

Cyclic and non-cyclic rostering are two different models to tackle the construction of lines of work based on the pattern of demand variation. Cyclic rostering involves generating a fixed roster but start with different offsets that can satisfy staff requirements for repeating demand patterns, rarely considering individual staff requests (Musliu, 2002). Howell (1966) and Frances (1966) laid down some basic principles for manual cyclic rostering while Baker (1976) reviewed some basic mathematical models for cyclical scheduling problems. Alfares (1999) had successfully developed an efficient two-phase algorithm for solving the cyclic manpower days-off scheduling problem with two consecutive off days per week.

For non-cyclic rostering models, the lines of work for individual employees are entirely independent and allocated shifts usually have different lengths and starting times (Ernst et al., 2004). A non-cyclic roster is reformulated before each rostering period, with each schedule in the roster being matched to a particular staff. This is done to accommodate individual staff preferences. The distinction between cyclic and non-cyclic rostering is mainly made based on the great distinction of required computation efforts, which has been largely improved because of high-speed computing machine. Therefore until now they have not been dealt with by designating separate specific models and algorithms.

Given the fact that the manpower scheduling problem is a combinatorial optimization problem, the associated models are mostly integer programming in formulation (Müller-Merbach, 1975; Tien and Kamiyama, 1982). Various set-covering heuristics were used then to construct schedules (Burns and Koop, 1987). Within them is Danzig set covering formulation (Danzig, 1954) and its variations with side constraints taking different practical considerations into account, such as governmental regulations, industrial regulations, and personal preferences. Similarly, set partitioning model have also been used extensively in rostering problems.

The Danzig set-covering formulation for shift scheduling was the first to model personnel scheduling problem in mathematical programming form and widely used in modeling (Bertholdi III et al., 1980; Brusco and Jacobs, 1998; Morris and Showalter, 1983; Thompson, 2001). It is typically expressed as (Danzig, 1954; Bethtold and Jacobs, 1990):

Min $\sum_{j \in J} c_j y_j$

s.t. $\sum_{j \in J} a_{ij} y_{ij} \geq b_i \qquad \forall i \in I$

$$y_j \geq 0 \text{ integer}$$

where

J – Set of indices associated with allowed shifts.

I – Set of demand periods.

y_j = the number of employees assigned to the jth shift.

c_j = the cost of an employee assigned to the jth shift.

$$a_{ij} = \begin{cases} 1 \text{ if period } i \text{ is a work period in shift } j, \\ 0 \text{ otherwise.} \end{cases}$$

b_i = the labor requirement in period i.

The set covering model can be solved using either exact or heuristic algorithms. The best known exact algorithm for linear integer programs is the branch-and-bound method. Upper bounds can be calculated using any simple or advanced heuristic. Lower bounds can be found by solving linear programming relaxations or Lagrangian relaxations. Once it is found that exploration of the branch-and-bound search tree is too time consuming, heuristic rules could be used to terminate the search. These heuristic rules include: termination after finding the first, or a good, integer solution and partially exploring the search tree by fixing fractional variables (Day and Ryan, 1997).

Bethtold et al. (1991) presented an initial study of comparatively evaluating labor tour scheduling methods. In their paper, nine labor scheduling heuristic methods, classified as either linear programming (LP) based or construction, were reviewed and compared with respect to the primary criterion of minimizing total labor hours scheduled after an integer linear programming model was formulated.

2.2 Scheduling Flexibility

Issue of scheduling flexibility was firstly brought forward systematically by Bechtold and Jacobs (1990). They indicated that the substantial reductions in model size could only be achieved by reducing the information requirements of the model, based on which a new implicit modeling approach was developed. The flexibility of break assignments for all shifts was implicitly represented in this modeling approach, by associating break variables with planning periods as opposed to shifts.

In order to provide flexibility, organizations adopt different shift start times, shift lengths, daily break windows, and days-on work patterns. As the number of allowed scheduling flexibility alternatives increases, developing tour schedules becomes more complex. Jacobs and Bechtold (1993) have summarized scheduling flexibility alternatives within the context of shift scheduling, days-off, and tour scheduling problems.

2.3 Personnel Assignment Problem

This review mainly focuses on its modeling aspects of assignment problem and much less attention is paid to its algorithmic aspects. Personnel assignment, shift assignment, roster assignment – these assignments usually occur as subproblems at various modules of the rostering process illustrated in Figure 2.1.

Assignment emerges in every area of the company: production, finance, marketing, and of course personnel management. The personnel assignment is mainly concerned with assigning individuals or groups of workforce to either post/job or tasks. The objective of the personnel assignment is generally recognized as to achieve the maximum overall benefit to the organization within the framework of established policies and procedures and the limits imposed by the composition of the incoming population (Constantopoulos, 1989).

As far as the tasks to which available personnel is assigned is concerned, there are two types of tasks, namely periodic tasks and aperiodic tasks. Periodic tasks are known tasks that are invoked at fixed time intervals, whereas aperiodic tasks are results of environmental stimuli that are often stochastic (Zolfaghari, 2004). Some organizations may deal only with either of these two types and some may experience both types simultaneously

One of the major concerns of this research is the optimal assignment of workforce to daily maintenance tasks such that maintenance quality could be maximized. This could be regarded as a personnel assignment problem: the assignment of each candidate to each job (or task) while fulfilling certain constraints in the situation where there is certain number of jobs (or tasks) to be done and certain number of candidates available to do them. It involves establishing a criterion based on optimization.

Most of the practical assignment processes were carried out by domain experts in the organizations. This manual process lacks the capability to consider systematically and effectively the relevant different decision constraints, and thus lacks planning optimization. A variety of mathematical models have been built by researchers to improve the current decision process in terms of overall optimization and resources utilization.

The personnel assignment problems firstly arose in the military, which requires the efficient distribution and utilization of skilled personnel resources (Trippi et al., 1974). This problem became difficult due to the size and frequency of personnel turnover, and also the large number of skill categories involved. Trippi (1974) pointed out that the speed and computing power of the modern digital computer make it possible to select an optimal set of assignments from an extremely large number of alternatives. Other efforts made on solving military personnel assignment problems include works by Glover, Karney and Klingman (1977); Klingman and Philips (1984); Liang and Lee (1985); Liang and Thompson (1987) and Liang and Buclatin (1988).

Constantopoulos (1989) designed a decision support system for assigning large numbers of personnel to jobs according to multiple criteria in large organization, such as the military or public service organizations. Liang and Buclatin (1988) formulated the military personnel assignment problem in U.S. Navy with enroute training as a network model.

Keen and Morton (1978) pointed out that personnel assignment was a decision of the semi structured type. The existing structure is represented as a list of objective assignment criteria and their relative importance, which is incorporated into an objective assignment procedure. On top of this structure, subjective judgment according to ad hoc, or even unarticulated criteria, finally determines an optimal assignment (Constantopoulos, 1989).

A common assignment problem, Linear Sum Assignment Problem (LSAP), belongs to the classical problems of mathematical programming. They occur mainly as subproblems in more complex situations like the traveling salesman problem, vehicle routing problems, personnel assignments and similar problems from practice (Burkard, 2002). He summarized that LSAP can be solved by only adding, subtracting and comparing the cost coefficient.

The Generalized Assignment Problem (GAP), also the general form of assignment problems which have been extensively studied in literature (Liu and Layland, 1973; Baker, 1974; Dhall and Liu, 1978; French, 1982 and Pinedo, 1995), examines the minimum cost assignment of n jobs to m agents such that each job is assigned to exactly one agent subject to capacity restrictions on the agents (Cattrysse and Wassenhove, 1992). It can be expressed as follows:

Min $\sum_i \sum_j c_{ij} x_{ij}$

Subject to:

$$\sum_j a_{ij} x_{ij} \le b_i, i \in I,$$

$$\sum_i x_{ij} = 1, j \in J,$$

$$x_{ij} = 0 or 1, i \in I, j \in J,$$

Where c_{ij} is the cost/disutility of assigning job j to agent/person i, a_{ij} is the capacity absorption when job j is assigned to agent/person i, and b_i is the available capacity of agent/person i. The assignment variable x_{ij} equals 1 if agent/person is to perform job j, 0 otherwise.

While the above formulation regards c_{ij} as the cost of each assignment, there is an alternative formulation considering c_{ij} as the value/utility of each assignment. So the minimization objective is reversed to the maximization objective, maximizing the total utilities. Besides, a_{ij} represents the qualification variable instead of capacity absorption, 1 if agent i is qualified for job j and 0 otherwise. This formulation is presented below (Caron et al., 1999), without considering capacity requirements. The two approaches are equivalent differing only in the way in which an assignment is evaluated; the former formulation uses costs to rate assignments and the later implies a measure of value (Trippi et al., 1974).

Max $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_{ij} \leq 1, i = 1,\ldots,m,$$

$$\sum_{i=1}^{m} a_{ij} x_{ij} \leq 1, \ j = 1,\ldots,n,$$

$$x_{ij} \in \{0,1\}, i = 1,2,\ldots,m; \ j = 1,\ldots,n$$

The assignment problem is a special case of the transportation problem and an NP-hard combinatorial optimization problem. Considerable research has been done in the past to find effective enumeration algorithms to solve problems of reasonable size to optimality (Cattrysse and Wassenhove, 1992). When the number of persons equals the number of jobs, it can be solved by Kuhn's Hungarian Algorithm without difficulty (Abboud et al., 1998; Bazaraa et al., 1990).

According to Cattrysse and Wassenhove (1992), most algorithms are based on branch-and-bound techniques and on relaxation of the assignment or the knapsack constraints. The methods may differ in the way the bounds are computed in the branch and bound search. Lagrangean relaxation or surrogate relaxation can be used to relax equality or inequality constraints. The different kinds of relaxation used to obtain bounds are reviewed and the strength of these bounds is compared by Cattrysse and Wassenhove (1992).

The classical assignment problem can be extended with some side constraints added into it. Caron et al. (1999) considered two types of side constraints, seniority and job priority constraints, which appear in the daily scheduling of nurses belonging to the float team and the availability list of a hospital.

Jaime (1998) pointed out that in order to be able to assign personnel, according to their capabilities, to the available activities and to a suitable location, one will have to make the transition from verbal semantics to the corresponding numerical semantics. So one will have to accept the fact that this transition from verbal semantics to numerical semantics is subjective and only safe for certain special cases where measurement is possible. There is actually nothing existing which is totally objective, for a series of reasons that would be all too tedious to detail.

Personnel assignment can also be represented by a network model according to graph theory. An example can be seen in works by Constantopoulos (1989). In his paper, a bipartite directed graph is formulated, consisting of a set of origin nodes I (personnel set), a set of destination node J (job set) and a set of directed arcs A. with each arc $(i, j) \in A$, a utility index a_{ij} is associated, as well as a decision variable x_{ij} which takes the value 0 or 1 depending on whether the assignment (i, j) is made or not. An example of assignment graph is shown in Figure 2.2.

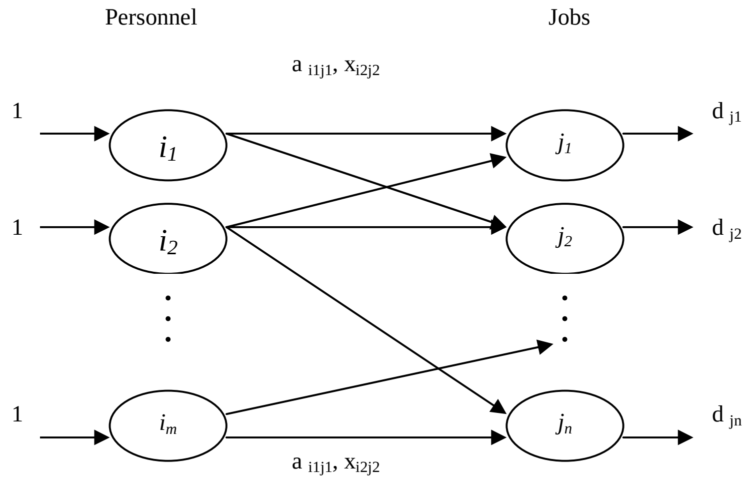


Figure 2.2 Network model of assignment (Constantopoulos, 1989)

Some other researches formulate the assignment problem as graph coloring, which is a well-known problem in graph theory. For example, a heterogeneous workforce assignment problem has been formulated as a restricted vertex coloring problem in Vall et al.'s work (1996). The objective was to minimize the number of works required to carry out a machine load plan, slightly different from the above mentioned generalized assignment problem (GAP). And a branch and bound algorithm was utilized. It was also claimed to be a pioneering study on Number of Workers Minimization Problem (NWMP) for heterogeneous workforce.

Apart from personnel assignment, various applications of generalized assignment problem and other variations could be found in other problems, such as task assignment in distributed computing system (Stone and Bokhari, 1978; Ma et al, 1982; Sinelair, 1987; Ali and El-Rewini, 1992; Peng, 1997; Sancho et al., 2006), production machine assignment, airport gate assignment (Ding et al., 2005), and project assignment among candidate students (Harper et al., 2005).

2.4 Scheduling and Assignment Problems in the Aviation Field

In this section, one popular research area, airline crew scheduling and rostering is reviewed, followed by the review of a relevant area of aircraft maintenance scheduling.

2.4.1 Airline Crew Scheduling and Rostering

The airline crew scheduling problem, attracting most operation researchers in the aviation application area, has always been an economically significant topic for airline operators. It not only affects the carrier's level of service but also its profitability (Yan et al., 2002). Crew costs contribute to one of the largest components of direct operating costs and represent hundreds of millions of dollars a year for large airline companies. However, airline crew scheduling is very difficult because of the size of the problem and the complexity of collective agreements and governmental rules that must be respected (Stojković et al., 1998).

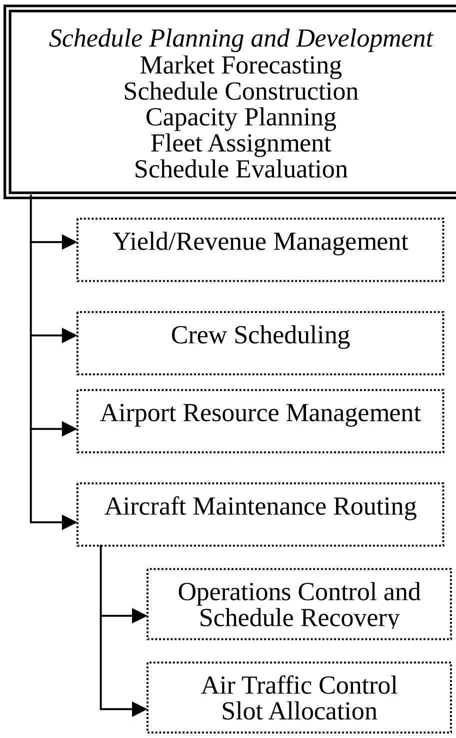


Figure 2.3 Overview of the airline planning process

In the transportation area, personnel scheduling and rostering is always recognized as crew scheduling and rostering. Airline crew scheduling is mainly concerned with flight attendant and technical personnel, pilot, who have to be on a flight. Crew costs contribute to one of the biggest components of direct operating costs and represent hundreds of millions of dollars a year for large airline companies. Due to its great economic scale and impact, airline crew scheduling and rostering has become the biggest area of staff scheduling. There are much more articles that have been devoted to the model and algorithm development in the area than to any other rostering application area (Andersson et al., 1997; Desaulniers et al., 1997; Hoffman and Padberg, 1993; Lučić and Teodorović, 1999; Marsten and Shepardson, 1981; Ryan, 1992; Stojković et al., 1998; Teodorović and Lučić, 1998; Yan and Chang, 2002; Yan et al., 2002). The problem of crew scheduling and rostering has attracted numerous attention from both airlines and the academics.

Before the detailed description and definition of the problem is given, some terms are defined. A *flight segment* represents to a nonstop flight between a pair of cities. The corresponding segment is considered to be *active* if the crew members under consideration operate a given flight. A *deadhead* flight segment is a passive segment used to transport crew members between stations, either by ground or air transportation. A *duty*, or individual work day, is a sequence of flight segments separated by connections. A *crew pairing* is a sequence of duties separated by rest periods, worked by a crew leaving and returning to the same crew home base (Stojkovic et al., 1998).

Both the temporal and spatial features are the important features for airline crew scheduling. Temporal features mean that each task, namely flight, is characterized by its starting time and finishing time. During the scheduling process, the starting time and finishing time have to be determined under certain constraints. Spatial features represent the characteristics of starting location and finishing location of each flight. The spatial features serve as fundamental constraints to the whole process.

The airline crew scheduling problem is usually decomposed into two different stages, namely crew pairing and crew rostering. Thus computational complexity could be reduced. The crew scheduling process begins with the daily crew pairing, which involves the construction of sequences of flights or sectors from the given flight timetable. The construction should ensure that the duty or trip must be “feasible” with respect to all the rules, regulations and the conditions of awards and agreement (Ryan, 1992). These pairings selected in the last stage are then sequenced into rosters, which are further allocated to individual crew. There are two ways to make allocation, the Bidding System and the Equitability System (Chang, 2002). In the Bidding System, the senior crews could bid his preferred rotation to build his personalized schedule. In the Equitability System, crew are assigned duties according to the equitability principle, trip preferences, vacation preference, crew requests, flight hours, duty days, layover days, vacation days, duty numbers, etc.

The airline crew scheduling problems could also be considered as comprising two phases: the planning and the operational phase. In spite of the fact that all the changes that occur during day-to-day operations may drastically alter the plan, the research conducted so far directs its attention only toward developing the best possible plan (Stojkovic et al., 1998). So Stojkovic et al. (1998) made the first published attempt to solve the operational airline crew scheduling problem. In the attempt, the personalized planned monthly blocks are modified during day-to-day operations to deal with considered disturbances, including sick leave of crews, rescheduled/canceled/added flights and incident interruptions.

Although most airline crew scheduling problems share common solution methods, they differ among airlines, in terms of crew categories, fleet types, network structures, rules and regulations, regularity of the flight timetables, and cost structures (Andersson et al., 1997).

Most prior and also classical approaches in the past 40 years to the crew pairing problem center around Set Partitioning problem (SPP), where each variable corresponds to a feasible pairing (Baker et al., 1985; Hoffman and Padberg, 2001; Marsten and Shepardson, 1981). Its generalized version (Chu Et al., 1997) is stated as follows:

Min $\sum_{j \in M} c_j x_j$

s.t. $\sum_{j \in M} a_{ij} x_i = 1 \qquad \forall i \in N$

$x_j = 0, 1 \qquad \forall j \in M$

where:

c_j – Cost of pairing j.

N – Set of all flights in the schedule.

M – Set of all pairings generated.

The constraint matrix *A* is defined as

$$a_{ij} = \begin{cases} 1 \text{ if flight } i \text{ participates in pairing } j, \\ 0 \text{ otherwise.} \end{cases}$$

A Set Covering formulation is used when overcovering of flight legs is allowed at the same cost as usual covering by an active crew (Desaulniers et al., 1997).

The set partitioning/covering problem and its variations have always been solved by approximation methods in the past 40 years, even for the smallest of an airline's fight. Until 1993, Hoffman and Padberg (1993) presented an exact branch-and-cut approach to solving to proven optimality large set partitioning problems arising within the airline industry.

Other important research works include genetic algorithm, plus a first-in-first-out method (Langerman and Ehlers, 1997); set-partitioning model and a column-generation approach for airline cockpit crew scheduling (Yan and Chang, 2002); several integer programming models and column-generation-based algorithms, to minimize airline crew costs and to plan the most appropriate individual pairings (Yan et al., 2002); formulating the airline crew rostering problem as a generalized set-partitioning model (Ryan, 1992); developing an aircrew rostering model that could assign approximately equal workloads to all crew members (Teodorović and Lučić, 1998); and formulating an aircrew rostering problem as a multi-objective optimization problem and then propose a two-step solution procedure to solve the problem (Lučić and Teodorović, 1999).

Other scheduling and rostering work related to aviation are customs staff scheduling at the Auckland International Airport (Mason et al., 1998), enhancement to the tour scheduling process associated with United Airlines' *Pegasys* Manpower Planning System (Brusco et al., 1995), station staffing at Pan American World Airways (Schindler, 1993), labor scheduling at an airport refueling installation (Valdes et al., 1999).

2.4.2 Manpower Scheduling for Aircraft Maintenance

Although of numerous studies on airline crew rostering, a very limited amount of research has been done with regard to the scheduling and allocation of aircraft maintenance staff. A comprehensive overview of aircraft maintenance in terms of its conditions and requirements could be referred to Chapter 2.

When scheduling maintenance manpower for inbound flights at the airport, there are some special practical considerations about this type of application. Lau (1996) considered that shift types are often associated with flights because different flights require ground crews of different skill profiles to service.

According to previous studies (Dowling et al., 1997 and Ernst et al., 2004), task allocation or assignment is carried out through another model or subsystem after a master roster has been built from the main system.

Yang et al. (2003) pointed out that short-term layover maintenance (Line Maintenance) plan has to consider the manpower demand, the aircraft type, the maintenance crew and the available time slots. It is difficult to propose a single comprehensive maintenance plan. He proposed a practical three-step procedure to determine the maintenance plan, with a focus on the second step. First, manpower demand in man-hours is estimated based on available ground holding time slots, the different aircraft types, and the tasks required. Second, a manpower supply plan is expressed in terms of the number of work shifts and the starting time for each shift to facilitate assigning the maintenance personnel. The final step is to assign maintenance personnel to meet the supply plan but still satisfy the certificate requirements, vacation schedules and other relevant regulations; See (Yang et al, 2003). This procedure shows correspondence to general personnel scheduling process, leaving individual scheduling and rostering to the last step.

Yang et al. (2003) also firstly suggested applying the concept of flexible management strategies to improve manpower supply efficiency, in particular flexible shifts, flexible squad members, and flexible working hours. They are formulated into a mixed integer program, with the objective of minimizing the total maintenance manpower supply, while satisfying all the requirements and the demands in each time slot. Still, workers are assumed to be homogenous in qualifications and abilities.

Other works include developing a Decision Support System to solve the capacity planning problem for the maintenance department of an airline company (Dijkstra, 1991); and determining number of maintenance workers on duty by means of a simple set-covering model(Alfares, 1999).

2.5 Fixed Job Scheduling Problem

A related problem, the Fixed Job Scheduling Problem (FJSP), within the study of parallel machine scheduling, is defined as scheduling jobs in non-preemptive manner that have fixed ready times and deadlines to identical parallel machines

(Eliiyi and Azizoğlu, 2004). It fits into the problem of aircraft maintenance scheduling due to the matching problem features. Sometimes each job can only be processed on certain pre-determined machines.

There are limited available studies on this problem. Published research are carried out by Dondeti and Emmons (1992), Eliiyi and Azizoğlu (2004), Fischetti et al. (1987), Fischetti et al. (1989), Gabrel (1995), Kolen and Kroon (1994), Kroon et al. (1995) and so on. FJSP initially come from the Interval Scheduling (IS) problem, which is a typical problem for the reservation systems. The difference between FJSP and IS lies on whether the job can be delayed after it has ready for processing. In FJSP, the job (with a fixed starting time r_j and a fixed ending time d_j) should be processed immediately after its starting time. Kroon et al. (1995) pointed out that FJSP has both the character of a job scheduling problem and the charter of an assignment problem.

In the earlier research, one or both of the two variants of FJSP were considered based on objective functions, namely Operational Fixed Job Scheduling Problem (OFJSP) and Tactical Fixed Job Scheduling Problem (TFJSP). Each job in OFISP is given a weight w_j, and the objective is to maximize the weighted number of processed jobs with a given number of processors is of concern (Eliiyi and Azizoğlu, 2004). In the variant of TFJSP, each machine is associated a fixed cost c_k and the objective is to minimize the total cost of the machines needed. OFISP has been used to tackle the capacity planning problem of aircraft maintenance personnel (Kroon,

1990) and scheduling earth observing satellites (Wolfe and Sorensen, 2000), while TFJS could be applied in Bus Driver Scheduling Problem (Fischetti et al., 1987) and also developing maintenance schedules for aircrafts (Kroon et al., 1995).

Almost all the FJSP variations and extensions with side constraints are proved to be NP-hard (Fischetti et al.[1], 1987; Gabrel, 1995). The FJSP is formulated as integer program and also graph theory model, and solved by a branch and bound algorithm, with designing a polynomial time algorithm to solve the preemptive version (Fischetti et al.[1], 1987). Gabrel (1995) ‘s approach deals with graph theory, modeling the FJSP as a maximum independent set problem and then proposing heuristics based on partition into cliques to solve the model. OFISP with identical machines is formulated by Kroon et al. (1995) as an integer linear program and results were obtained through an approximation algorithm based on Lagrangean relaxation and decomposition. In these papers “machine” represents either actual processing machines or available workforce.

2.6 Summary

It should be noted that underlying assumptions of almost all reviewed studies are the homogeneity of scheduled personnel. Only few consider heterogeneous worker type as categorizing workers into several types, forming a hierarchy, in which the higher-level worker can substitute lower-level worker, but not vice versa. The general employee scheduling problem extends the standard shift scheduling problem by discarding key limitations such as employee homogeneity and the absence of connections across time period blocks (Glover and McMillan, 1986). Given that they have not considered heterogeneity among individuals, one of the objectives in this research is to model this situation.

A major challenge of personnel scheduling, as stated by Ernst et al. (2004), is to obtain greater efficiency gains not by improving algorithms for solving any one aspect of the problem, but by considering more problem complexity and also integrating more of the steps outlined into a single problem. Thus a comprehensive mathematical model could be a suitable solution to this problem.

Most researches have put their focus on developing specific algorithms to a general model, not on the modeling aspect. On the other hand, modeling aspects and mathematical formulation is also my research focus.

CHAPTER 3

METHODOLOGY

3.1 Problem Analysis

After reviewing the background of aircraft maintenance and existing research of general assignment and scheduling problems, it is now turn to describe the scheduling/assignment problem to be addressed, in terms of the characteristics and time constraints associated with the scheduled jobs and personnel.

The problem faced in this research is to construct feasible rosters for individual engineer, and simultaneously optimize daily assignments of individual engineers to maintenance tasks. It consequently has the nature of both rostering and assignment problems. The traditional way, as reviewed in Chapter 2, is to divide it into two sub-problems and treat them separately by means of their own solution algorithms as well as models.

Moreover, the motivation of this research comes from incorporating these two considerations together into the problem formulation using a comprehensive mathematical programming model.

To simplify the problem, unscheduled maintenance tasks are excluded, i.e. fixing pilot-reported problems and discrepancies found during inspections, so the workload of line maintenance has become highly deterministic. Alfares (1999) also made such assumptions to schedule aircraft maintenance workforce at airport. In this study, it has been assumed a deterministic problem of periodic demand in which a set of predetermined task is to be completed in each period of the planning horizon by certain qualified engineer.

The maintenance scheduling problem can be described as follows: given a set of pre-determined maintenance tasks on each day during a planning horizon (e.g. a week) and a set of available engineers, find an optimal rostering plan for each engineer such that both scheduling requirements could be satisfied and daily task assignments could be optimized in terms of maintenance quality.

In order to provide a suitable measurement of resultant maintenance quality for each task assignment, a utility index is designated, evaluating each matching of personnel to task. This process is named *Utility Assessment*, which is actually a *Fuzzy Multiple Attribute Decision Making* process. This utility index could then serve as a guide searching for optimal assignments, added as a kind of coefficient into another mathematical programming model. Within the model, scheduling and rostering constraints are the main considerations.

3.2 Overview

In this section analysis of the problems and an overview of the integrated methodology are given, including relevant background mathematical techniques and software packages.

3.2.1 Integrated Methodology

As described in the last Chapter, a sequential methodology consisting of 2 stages (Figure 3.1) is proposed to tackle two split sub-problems. The first stage is the Fuzzy Multiple Attribute Decision Making (FMADM), and the second is the Integer Goal Programming. Fuzzy set theory and Analytic Hierarchy Process are integrated as one powerful Multiple Attribute Decision Making tool, which is utilized for utility assessment to identify the heteogenenirity among candidates at the first stage. Results from the first stage, Utility Index, is input into an integer goal programming model to find the optimal solution searching.

In the data collection the required information on aircraft maintenance processes and policies were obtained by interviewing the concerned experts, gaining insights and view on various problem areas.

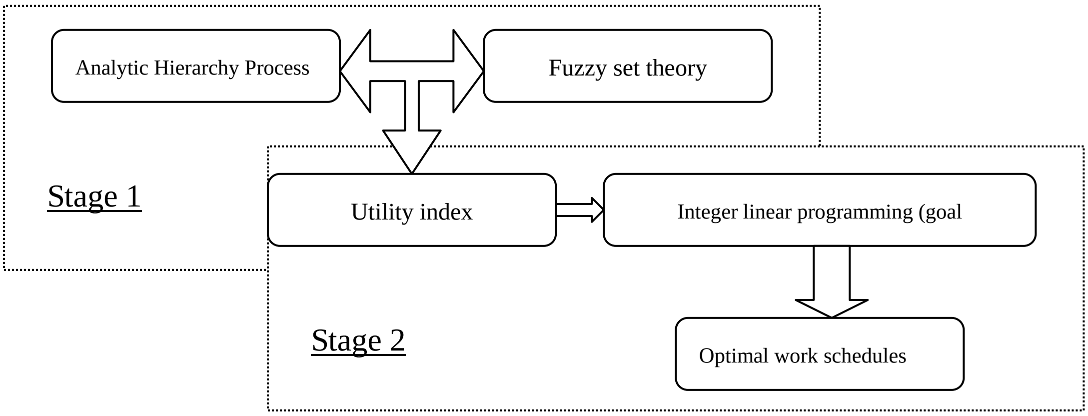


Figure 3.1 Chart to illustrate the integrated methodology

Relevant mathematical techniques and packages are briefly reviewed in the following sections, followed by the detailed methodology explanation as illustrated in Figure 3.1.

3.2.2 Analytic Hierarchy Process

Analytic hierarchy process (AHP), as proposed by Saaty (1980) has recently become increasingly popular in dealing with multi-criteria decision problems, such as selecting machines for flexible manufacturing systems (Tabucanon et al., 1994); evaluating the implementation of a maintenance system (Labib et al., 1998) and vendor selection of a telecommunication system (Tam and Rao, 2001). It enables decision makers to structure a complex problem in the form of a simple hierarchy and to evaluate a large number of quantitative and qualitative factors in a systematic manner under conflicting multiple criteria (Lee, Lau, Liu and Tam, 2001).

There are three steps in applying any decision-making technique involving numerical analysis of alternatives:

- Determine the relevant criteria and alternatives to the decision maker;
- Assign numerical measures to the importance of criteria against the goal and to the impacts of the alternatives on these criteria; and
- Process the numerical values to rank each alternative.

As one of the multiple attributes decision making methods, AHP shares the same procedure. After constructing the hierarchy structure, the importance of each criterion against the goal and weights of alternatives against criteria is determined respectively. Although there are two different procedures, with the later one following the former one, same measurement methods could apply. Saaty (1980) suggested using pair-wise comparison, while giving nine-point scale – *1, 2, 3, 4, 5, 6, 7, 8, 9*. According to Triantaphyllou (2000), the main reason for Saaty to establish *9* as the upper limit of the scale, *1* as the lower limit and a unit difference between successive scale values is based on psychology, which has shown that most individuals cannot simultaneously compare more than seven objects (plus or minus two). Decision makers thus need to give relative importance of one criterion over another criterion by means of nine-point scale proposed by Saaty (1980).

3.2.3 Fuzzy Set Theory

The fuzzy set theory, developed by Zadeh (1965), is to represent the uncertainty and fuzziness in complex real-world situations, compared to the original world of crisp numbers. Briefly the fuzzy logic principle is based on a 'superset' of Boolean logic that has been extended to handle the concept of 'partial truth' and it replaces the role of a mathematical model with another that is built from a number of rules with fuzzy variables and fuzzy terms such as very hot, fairly cold, and probably correct (Bucchanan & Shortliffe, 1984; Leung & Lam, 1988;

Orchard, 1994). Within fuzzy set theory, there are three basic concepts, namely fuzzy sets, linguistic variables, possibility distributions.

This type of problems usually occurs when there is neither definite quantitative description nor boundaries to a certain object. Unlike classical set theory, which handles with clearly defined membership to a set, fuzzy membership of an element to a set can be partial with the element belonging to a set based on a certain grade of membership (normally from *0* to *1*) (Lee et al., 2001). To represent linguistic variables, a fuzzy set is always defined in a context.

In mathematical terms, fuzzy set A is defined in a relevant universal set X by a membership function, which assigns to each element x of X a number, $A(x)$, in the closed unit interval $[0,1]$ that characterized the degrees of membership of x in A. Membership functions are thus functions of the form $A: X \longrightarrow [0,1]$ (Klir et al., 1997).

With regard to integrating fuzzy set theory and AHP, Laarhoven and Pedrycz (1983), Buckley (1985) have introduced fuzzy number operations in Saaty's AHP method by substituting crisp numbers with triangular fuzzy numbers. A fuzzy triangular number *M* (Figure 3.2), characterized by three parameters, *l, m, n,* on *R* is defined to be a fuzzy triangular number if its membership function: $R \to [0, 1]$ is equal to:

$$\mu(x) = \begin{cases} 1 & x=m \\ \dfrac{x-l}{m-l} & l<=x<=m \\ \dfrac{n-x}{n-m} & m<=x<=n \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

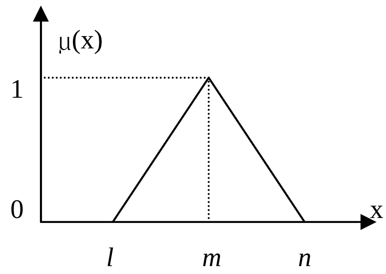


Figure 3.2 A Fuzzy Triangular Number

The basic operation rules of fuzzy triangular numbers are also defined as follows, including its addition, multiplication, division, natural logarithm, and exponential (Laarhoven and Pedrycz, 1983).

Addition: $\tilde{A} \oplus \tilde{B} = [a_1+b_1,\ a_2+b_2,\ a_3+b_3]$

Multiplication: $\tilde{A} \otimes \tilde{B} = [a_1 \times b_1,\ a_2 \times b_2,\ a_3 \times b_3]$

Division: $^1/\tilde{A} \cong [^1/a_3,\ ^1/a_2,\ ^1/a_1]$

Natural logarithm: $\ln(\tilde{A}) \cong [\ln(a_1),\ \ln(a_2),\ \ln(a_3)]$

Exponential: exp ($\tilde{A}$) $\cong$ [exp (a_1), exp (a_2), exp (a_3)]

3.2.4 Mathematical Programming

The essential feature of a mathematical model in operations research is that it involves a set of mathematical relationships (such as equations, inequalities, logical dependencies, etc.) which correspond to some more down-to-earth relationships in the real world (such as technological relationships, physical laws, marketing constraints, etc.) (Williams, 1985).

Mathematical programming model, has been the most commonly used standard type of model. The common feature which mathematical programming models have is that they all involve optimization, which could be either maximization or minimization. The quantity which is to be maximized or minimized is known as an objective function. At the same time constraints (both equations and inequalities) form as restrictions to the desire optimization process. These objectives and constraints are regarded as an abstract of the real situation or problem. The advances in computing and developments in algorithms enable the solving mathematical programming problems proceeded in tandem (Walker, 1999). One of the milestones was in 1947 when George Dantiz introduced the simplex algorithm, which promptly assumed a major role in the solution of linear problems.

Mathematical programming models are classified easily as linear programming, non-linear programming models, integer programming models, with probable further detailed categorization (Williams, 1985). Some of the essential features of linear programming (LP) were summarized in his book:

- There is a single linear expression (the *objective function*) to be maximized or minimized; and
- There is a series of *constraints* in the form of linear expressions which must not exceed (≤) some specified value, must not fall below (≥) a specified value or must exactly (=) equal a specified value.

Although linear programming has been applied to a wide range of subjects, the linearity relationships within LP are not always guaranteed in a practical problem. A non-linear programming (NLP) model is obtained when non-linear relationships are included either in the objective function expression or the constraints formula in a model. These NLP models are much more difficult to solve. Similarly, the variables may not be allowed to take fractional value, so an integer programming (IP) model is obtained when integer constraints are added into the constraints in the model. Other types of mathematical programming model are stochastic programming models and dynamic programming models.

These models are then subject to various designated tailor-made algorithms, which are a set of mathematical rules for solving the corresponding group of problems/models. They can also be presented to a standard software package for solution probing. The author will focus on one category - Integer Linear Programming.

Integer programming is a well-known, reliable and classical method in practical operations research but it must be handled carefully, since some integer programming formulations can require a large amount of computation time. Given a problem with a few dozen of variables one cannot be confident that integer problem will work until it has been tried on realistic instances (Beale, 1988). An alternative mathematical formulation of the same problem could be tried on if the prior one does not work.

Unlike Linear Programming with the simplex algorithm, there is no such universal IP algorithm which could efficiently solve all IP problems. In some IP models (usually not the binary model), noninteger solutions were allowed by solving its Liner Programming relaxation and then rounded to an integer solution which need be feasible but not be the optimal. So far, branch and bound method has found to be the most successful algorithm to solve the practical IP problems (Williams, 1985). Almost all commercial packages offering mixed integer programming adopt the branch-and-bound algorithm. Apart from branch-and-bound, others are cutting planes methods, enumerative methods, pseudo-Boolean methods.

3.2.5 Goal Programming

While faced with problems with more than one objective, they are called Multiobjective Optimization. The multiple objectives may conflict with each other and result in optimization without compromise. So one cannot expect achieving the optimal values for all objective functions simultaneously, but a 'best' value somewhere exists in between the individual optimal values (Sierksma, 2002). One promising method of determining this 'best' value is called Goal Programming, which results in a linear programming model as well finally. From its inception in the early 1950s, this method has rapidly evolved into one that now encompasses nearly all classes of multiple objective programming models since it has demonstrated its ability to serve as an efficient and effective tool for the modeling, solution, and analysis of mathematical models that involve multiple and conflicting goals and objectives(Ignizio, 1985).

Goal programming (GP), firstly the term used by Charnes and Cooper in 1961, has been an even more powerful technique than linear programming (LP) to handle such multiple objectives as well as single objective (Aouni and Kettani, 2001; Charnes and Cooper, 1961; Hannan, 1985; Ignizio, 1976; Kim and Emery, 2000; Lee, 1972; Tamiz et al., 1998). What makes GP distinguishable from LP is that GP does not optimize the desired objectives or well-defined utility function directly. Instead, it aims at minimizing the deviations from the desired goals. In other words, target goals are known and efforts are made to achieve them as closely as possible (Tamiz et al., 1998). Usually, in GP, multiple goals are tackled. For each goal, a positive deviational variable (d^+) and a negative deviational variable (d^-) are introduced, representing overachievement of the goal and underachievement of the goal respectively (Kim and Emery, 2000).

There are two ways to handle multiple objectives in GP. One is to optimize goals according to their priority hierarchy in a lexicographic sense, in which a lexicographic minimization being defined as a sequential minimization of each priority whilst maintaining the minimal value reached by other higher priority level minimizations (Tamiz et al., 1998) (preemptive priority or lexicographic). Another method is to add different weights in front of each goal and optimize the addition function onetime (non-preemptive). The general GP model can be illustrated below by using the deviational variables (Kim and Emery, 2000):

Minimize $\sum W_J P_J (d_I^+ + d_I^-)$

Subject to $\sum (A_{IJ} \cdot X_{IJ}) - d_I^+ + d_I^- = B_I$

Where W_J is the preemptive weight of each priority J, P_J is the preemptive priority of goal J, A_{IJ} is the technological coefficient linking decision variable I and constraint J, X_{IJ} is the decision variable I on constraint J, and B_I is the right-hand-side value of constraint I.

By optimizing the least deviation from pre-determined target for each goal, GP could help decision makers to deal with conflicts of interest and incompatible measurements.

3.2.6 Implementation Software Packages – CPLEX and MPL

Two important standard software packages, powerful LP solver CPLEX as well as modeling system MPL are introduced in this section before they are actually used in Chapter 4. These two software packages are widely utilized in the mathematical programming field.

In order to implement and test the prepared mathematical programming model to solve the aircraft maintenance planning problem, two sets of software are utilized, namely MPL and CPLEX. CPLEX is a powerful mathematical programming solution package widely used in academics and industry, developed by ILOG, Inc (ILOG). combining various algorithm including simplex optimizer, barrier optimizer and mixed-integer optimizer. CPLEX can be used for solving linear programming, mixed-integer programming, quadratic programming, and quadratically constrained programming problems. CPLEX Interactive Optimizer 9.0.0 is used in the project. It has a command-line interactive interface (Figure 3.3), consisting of the following basic commands in Table 3.1:

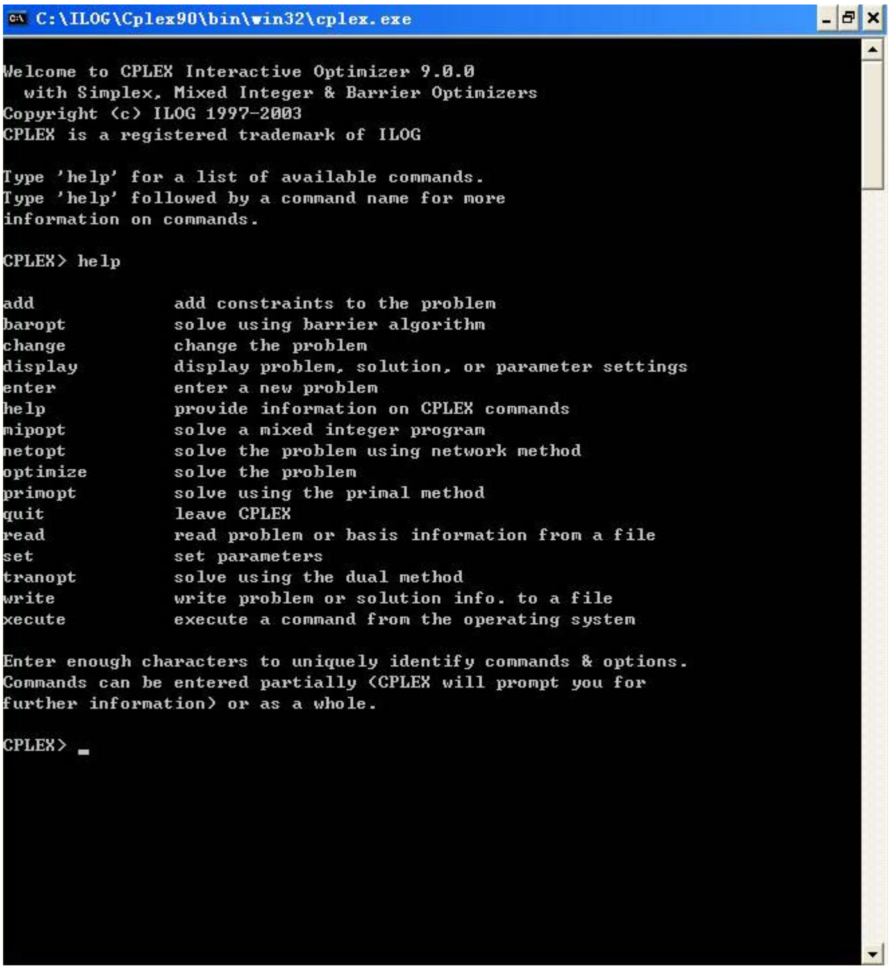


Figure 3.3 A Command-line Interactive Interface in ILOG Interactive Optimizer 9.0.0

Table 3.1 Basic Commands in ILOG Interactive Optimizer 9.0.0

add	Add constraints to the problem	**optimize**	Solve the problem
baropt	Solve using barrier algorithm	**primopt**	Solve using the primal method
change	Change the problem	**quit**	Leave CPLEX
display	Display problem, solution, or parameter settings	**read**	Read problem or basis information from a file
enter	Enter a new problem	**set**	Set parameters
help	Provide information on CPLEX commands	**tranopt**	Solve using the dual method
mipopt	Solve a mixed integer program	**write**	Write problem or solution information to a file
netopt	Solve the problem using network method	**xecute**	Execute a command from the operating system

MPL (Mathematical Programming Language) modeling system is a high-level language that translates mathematical statements that describe a mathematical program into a format readable by most optimization software packages (Maximal, Software, Inc., 2005). MPL actually acts like a user-friendly platform or complier, taking mathematical programming model itself and data instance as input, and transferring the readable model format to a pre-specified solver. The solver then finds an optimal solution to the problem and outputs the solution as an easily-viewed text file to the user. In this study, the author used MPL Modeling System Release 4.2h, in which the maximum problem size has little limitation. Besides, the MPL package offers several helpful windows indicating useful information for users (Figure 3.4).

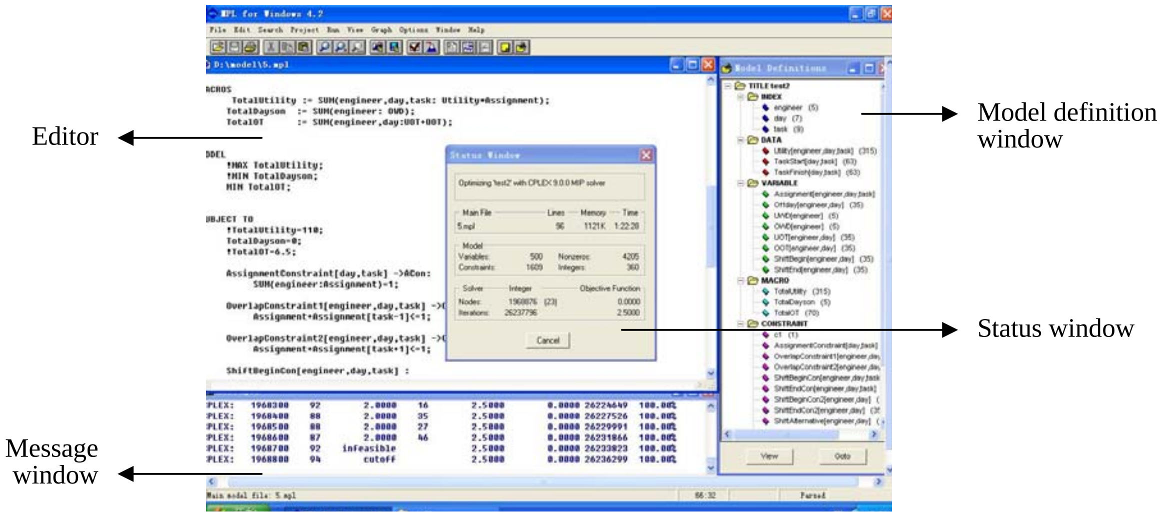


Figure 3.4 Platform Interface of MPL Modeling System Release 4.2h

Because of the easy manipulation of data input and alteration, MPL is employed for modeling and then installed solver CPLEX is recalled for solving the model within the MPL environment in the part of mathematical programming model. A diagrammatic view of the steps in using MPL is given in Figure 3.5. The word of "Solver" would be replaced by the name of the actual solver installed supported by MPL, which is CPLEX in this case.

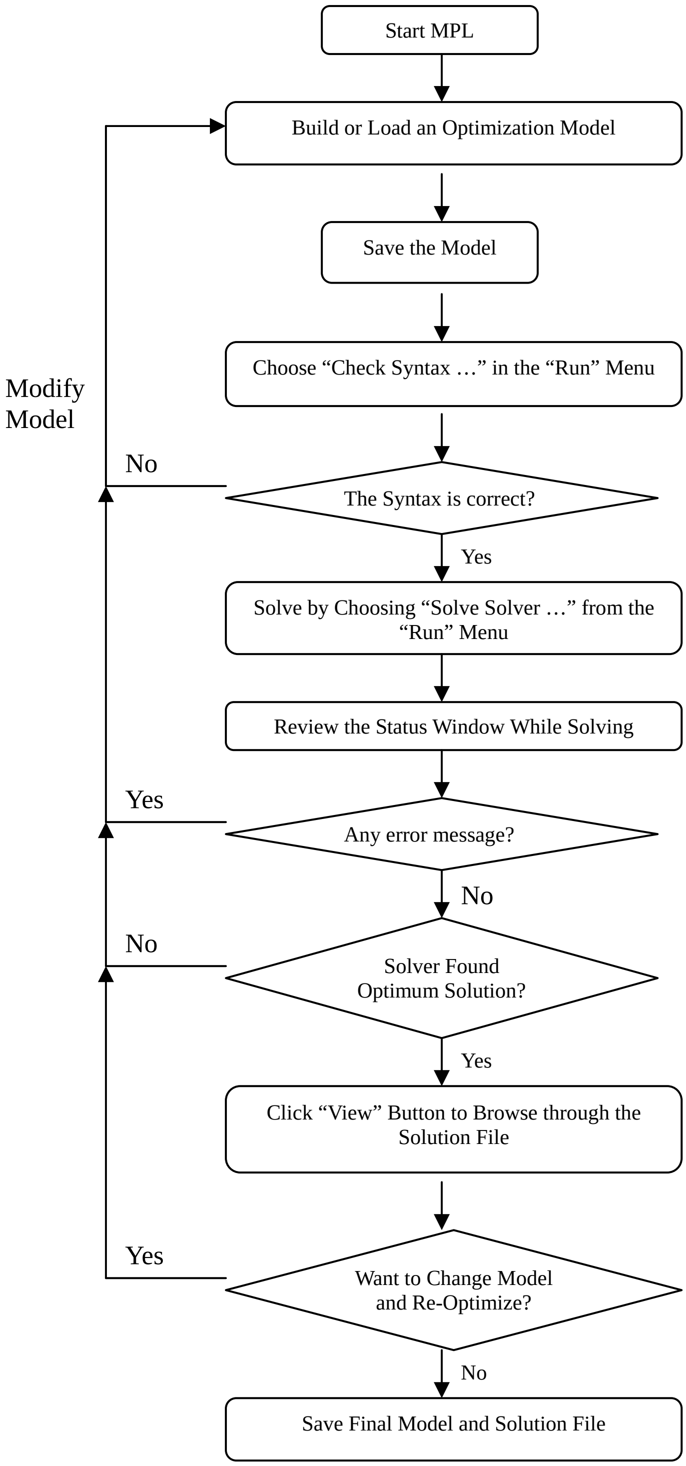


Figure 3.5 A Diagrammatic View of The Steps in Using MPL Modeling System Release 4.2h

3.3 Utility Assessment Using Fuzzy MADM Approach

As discussed in the last section, the first stage of the methodology is utility assessment using Fuzzy Multiple Attribute Decision Making. Within this stage, a utility index, for each engineer/task combination, calculated from Analytical Hierarchy Process, is developed which measures how well the engineer's qualifications as well as abilities match the requirements of the task.

The process of deriving the utility index can be considered as a multiple attribute/criteria decision making (MADM/MCDM) process, in which decisions are made based on the evaluation of decision alternatives against various designated criterion with their own weights.

The first step in any MADM problem is to define the set of alternatives and the set of decision criteria that the alternatives need to be evaluated with (Triantaphyllou, 2000). It is regarded as a critical step in solving MADM problem encountered in this study, and is identified during the first step of model construction. A further quantitative analysis is then conducted in a systematical way.

To conduct quantitative analysis, in other words, to conduct comparisons within alternatives for each task, there are two – one way is to define a priority over the set of selection criteria and then to rank the candidates for each task according to the defined, preemptive list of criteria. Another approach is to define a quantitative measure for each feasible person-to-task assignment, based on the selection criteria, which represents the utility of the particular assignment to the organization (Constantopoulos, 1989). Comparisons of candidates are then made based on these utility indices. Relevant background knowledge is briefly reviewed in section 3.2. Numerical illustration will be given in Chapter 4.

3.3.1 Constructing Hierarchy Structure

The first step, constructing hierarchy structure, consists of following sub-steps.

- Define and specify the problem;
- Identify the objectives, criteria, and factors that should be considered to accomplish the final objective; and
- Identify the alternatives.

The problem is the determination of the most suitable engineer for an aircraft maintenance task. The model needs to provide a utility index to help making the decision, which could measure how an engineer is suitable to carry out the task. Therefore a utility index is considered as the ultimate value derived from this quantitative analysis.

Experts in the industry were then interviewed on their consideration factors when making such decisions concerning daily task assignments. Then, four groups of criteria were identified and established – Licenses, Prior Experiences, Human factors, and Training. They also formed the criteria level of the hierarchy structure.

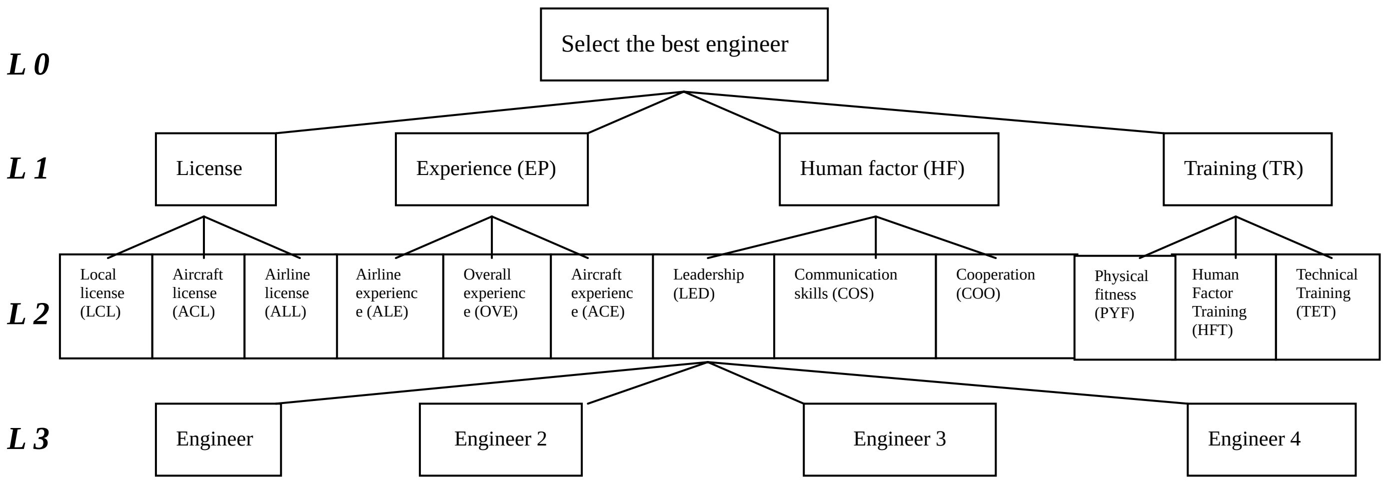


Figure 3.6 The Hierarchy Structure for A Suitability Analysis

The license requirements are to check whether the candidates possess the required relevant licenses authorized by the relevant authorities for carrying out certain maintenance tasks. This category is further divided into three specific license requirements: District license, Aircraft license, and Airline license. The district license refers to the license that allows the engineer to perform maintenance for the aircrafts registered in either Hong Kong or Britain. The aircraft license shows an engineer’s eligibility to maintain a specified group of aircraft types. The airline license is authorized by airline operators to assure the suitability of maintenance engineers to work on the aircraft owned by specific airline operators.

The group Prior Experiences criteria which includes Overall Prior Experiences, Aircraft Prior Experiences, and Airline Prior Experiences, is to record the amount of prior experiences of candidate engineers in undertaking aircraft maintenance with regard to overall career, aircraft type, and airline operators, respectively. Criteria concerning Human factors are then to consider various human factors that would affect the quality of aircraft maintenance, including Communication skills, Leadership, Cooperation capability, and Physical fitness. Training criteria consider how much company training the candidate had received, which are Technical training and Human factor training.

Also in this problem, the alternatives are clearly the maintenance engineers that are available in the company within the planning period to perform aircraft maintenance tasks.

By means of decision criteria, assumption is made on the relation of the utility of each task assignment with the scores of alternatives; the utility is defined as the combination of scores of alternatives under several criteria. Other assumptions include that criteria and their weights are the same for all tasks; and personal information for engineers is the most updated.

3.3.2 Initial Filtering

In the aircraft maintenance industry, some maintenance task requirements are clear-cut. When assessing candidates under certain requirements, the candidates are either accepted or rejected from performing maintenance tasks, because the candidates are required to obtain certain qualifications. In the problem, the license criteria should be considered to ensure that the selected engineer has the requisite licenses, namely a requisite local license, a requisite aircraft license, and a requisite airline license. Therefore, these candidate engineers should go through a conjunctive filtering process firstly – those who do not possess any of the three requisite licenses will be removed from further consideration, as illustrated in Figure 3.7. Those left for further processing are considered *eligible* for further assignment procedure.

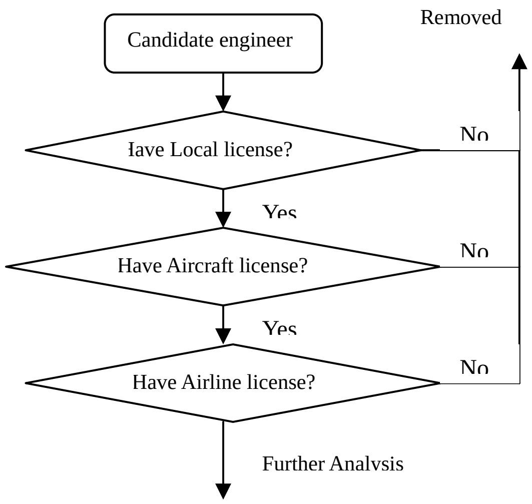


Figure 3.7 Initial Filtering Process

3.3.3 Fuzzy MADM

Detailed Fuzzy Multiple Attribute Decision Making process is proposed and discussed in this section, consisting of several steps.

3.3.3.1 Measurement of Attribute Importance

In MADM, there are two ways of weights determination – through direct assignment and through pair-wise comparison, depending on the application. Since it is neither easy nor accurate to manually assign a direct importance weight to each decision criterion, especially when decision makers are faced with too many attributes, no precise scale would exist. On the other hand, a pair-wise comparison could help by first assigning relative importance weights while comparing these attributes in pairs, and then utilizing some methods to combine these relative importance weights.

Through the pair-wise comparison, information can be obtained from each alternative. However, the major drawback in the usage of Analytical Hierarchy Process is the effort required to make all pair-wise comparisons (Millet and Harker, 1990). As the size of hierarchy increases taking in increase of number of attributes and alternatives, the number of required pair-wise comparison increases exponentially (Rezqallah, Hamad, and Salih, 1999). In other word, direct assignment would result in acceptable reliability and consistency. As shown in Figure 3.6, the number of attributes under comparison is small, which is less than 5 within each level. So pair-wise comparison is proposed to derive the attribute weights effectively.

Weight determination of level-1 attributes

A pair-wise comparison was made for every two of three level-1 criteria. The number indicating the importance of criterion *i* compared with criterion *j* was denoted by a_{ij}, namely the nine-point scale suggested by Saaty (1980) – *1, 2, 3, 4, 5, 6, 7, 8, 9*. The matrix of these numbers a_{ij} is denoted by *A*. In order to maintain consistency, let $a_{ij} = 1/a_{ij}$; thus, the matrix *A* has become reciprocal.

According to Saaty's AHP method, if *A* is the matrix of pair-wise comparison values, in order to find the priority vector, we must find the vector *w* that satisfies (Saaty, 1980) $Aw = \lambda_{max}w$. Also because it is preferable to have a normalized solution, *w* is slightly altered by setting $\alpha = \sum_{i=1}^{n} w_i$ and replacing *w* with $(1/\alpha)w$. Therefore, its uniqueness can be measured and $\sum_{i=1}^{n} w_i = 1$ (Saaty, 1980).

Determine the importance vector of level-2 factors

The same procedures were adopted to derive the importance vector of level-2 factors under each level-1 criterion as were used to derive the importance vector of level-1 criteria above. The reciprocal matrices were then obtained respectively under each of three level-1 criteria.

Determine the composite weights of level-2 factors

In the hierarchy model, the problem was decomposed into decision criteria, and each criterion further decomposed into more detailed decision factors. Therefore, after obtaining the relative weights of the criteria against the objective and the relative weights of the factors against each upper-level criterion, the composite weights should be computed by synthesizing these two kinds of weights using the following formulas:

$$f_{ij} = f'_{ij} \times c_i, \qquad (3.2)$$

where f_{ij} is the composite weight for the *jth* level-2 factor under the *ith* level-1 criterion, f'_{ij} is the importance weight for the *jth* level-2 factor under the *ith* level-1 criterion, and c_i, is the importance weight for the *ith* level-1 criterion.

3.3.3.2 Determine Scores of the Alternatives

There are several ways to determine the scores of the alternatives in terms of certain criteria. One way, the pair-wise comparison, was proposed by Saaty (1980) using the AHP method, within which the alternatives were compared to each other to obtain relative scores, and these relative scores were then combined using either the eigenvector method or a simplified algebraic mean method. Another way is to assign scores directly guided by a defined ratio scale regarding a specific criterion. These two approaches are both proposed to be used in the following situations when there exist criteria of different natures.

In addition, Fuzzy Set Theory is used to overcome the subjectivity of human judgments. Triangular fuzzy numbers as discussed in section 3.2 are employed to replace crisp numbers when the scores have been obtained either through a pair-wise comparison or a direct assignment.

Assess the prior experiences of the alternatives

When assessing the utility of alternatives in terms of prior experiences, it is more proper to directly assign the scores than to compare every two alternatives. In this study, prior experience is positively related to the actual amount of prior experience in terms of years spent on past relevant work. Thus, for every factor of Prior Experiences, a function, defined below, based on expert advice, is to transfer the actual experience years to triangular fuzzy numbers for each alternative.

$$\begin{cases} l = \frac{x-1}{10} \\ m = \frac{x}{10} \\ n = \frac{x+1}{10} \end{cases} \tag{3.3}$$

where x refers to the actual years of prior experiences of an alternative and is expressed in integers; and *l, m, n* refer to the lower, modal, and upper value of the corresponding triangular fuzzy number.

Assign the alternatives' scores against the Human Factor

For each level-2 factor under the Human Factor criterion, a pair-wise comparison was adopted to assign scores to alternatives. This is because it was difficult to give directly certain judgment to leadership or communications skills, such as "good" or "poor". Rather, they were more observable and reasonable to compare every two alternatives against these factors, by judging the extent to which one is better or poorer than the other against a criterion. Again, triangular fuzzy numbers were used to overcome the uncertainty and subjectivity of human judgment. Therefore, the fuzzy scales for all level -2 human factors are given below (Table 3.2) to assign relative scores when pair-wise comparisons are made.

Table 3.2 Fuzzy Comparison Scale for Level-2 Human Factor

Scale	Definition	Scale	Definition
(9, 9, 8)	Absolutely superior	(1/8, 1/9, 1/9)	Absolutely inferior
(8, 7, 6)	Very strongly superior	(1/6, 1/7, 1/8)	Very strongly inferior
(6, 5, 4)	Strongly superior	(1/4, 1/5, 1/6)	Strongly inferior
(4, 3, 2)	Weakly superior	(1/2, 1/3, 1/4)	Weakly inferior
(2, 1,1)	Equal performance	(1, 1, 1/2)	Equal performance

Four reciprocal matrices were then obtained after a comparison was made against four factors. The geometric mean method (Buckley,1985) was then used to derive the fuzzy importance vector, *Z*.

$$\tilde{z}_i = (\tilde{e}_{i1} \cdot \ldots \cdot \tilde{e}_{in})^{\frac{1}{n}}, \forall i \tag{3.4}$$

where $\tilde{e}_{i1},\ldots,\tilde{e}_{in}$ were the triangular fuzzy numbers assigned to *i*th alternative, which was obtained in comparison to other remaining alternatives.

The weight w_i was calculated using the following normalization formula:

$$\tilde{w}_i = \tilde{z}_i \cdot (\tilde{z}_1 + \ldots + \tilde{z}_n)^{-1}, \ \forall i . \tag{3.5}$$

Assign scores to the alternatives against Training

The alternatives were assigned scores directly against the factor of training according to the defined fuzzy scales in the following table based on how often the alternative had attended the company's technical training and human factor training sessions. The fuzzy scales are given in Table 3.3.

Table 3.3 Fuzzy Comparison Scale of The Attendance Record of The Alternatives

Scales	Training
(0.9, 1, 1)	Almost all
(0.6, 0.7, 0.8)	Majority
(0.4, 0.5, 0.6)	About half
(0.2, 0.3, 0.4)	Minority
(0, 0.1, 0.1)	Very little

3.3.3.3 Synthesize the Attribute Weights and the Fuzzy Scores of Alternatives

After the integrated weights of all consideration factors and the fuzzy alternatives scores against each factor were obtained through previous steps, next step is to synthesize them to achieve the final fuzzy index, which also took the form of triangular fuzzy numbers and were calculated using following formula:

$$\tilde{a}_i = \sum_{j=1}^{9} w_j \cdot \tilde{s}_{ij} \tag{3.6}$$

where w_j is the integrated weight of factor *j*, s_{ij} is the score of alternative *i* under factor *j*, and a_i is the fuzzy index of alternative *i*.

3.3.3.4 Obtain the Final Utility Index

Traditional MADM methods assume that all the data involved in decision making are crisp numbers, where the final index obtained can be used as the overall utility to guide decision makers. Thus it is a quite straightforward process to compare these final indexes. However, in this model, some of the data involved were fuzzy numbers and the final indexes obtained were thus fuzzy numbers, making comparisons and distinctions of them more ambiguous and indirect. Therefore, these fuzzy indexes should undergo certain fuzzy number ranking methods.

One of the many fuzzy ranking methods that can be used to compare fuzzy numbers proposed by researchers is adopted by using following defuzzition formula.

$$R(a)=\frac{2[\frac{1}{m-l}(\frac{1}{3}m^3-\frac{1}{2}m^2l+\frac{1}{6}l^3)+\frac{1}{n-m}(\frac{1}{3}m^3-\frac{1}{2}m^2n+\frac{1}{6}n^3)]}{n-l} \qquad (3.7)$$

Therefore, for each fuzzy index, a corresponding utility value could be computed. It would then be easier to compare the crisp utility value of all the alternatives.

3.4 Optimization through Integer Programming Model

This section is mainly concerned with formulating the integer goal programming model, leaving model solving to 2 standard packages discussed in section 3.3 and Chapter 4.

The usefulness of any mathematical programming model is, at least, to gain better insight and understanding of the real-world problem under consideration. It is of no exception in our study and it helps forming and solving a quantitative representation of our real-world problem.

The reason why Integer Programming is selected as the solution approach is that the integrated scheduling and assignment problem is found to have the following characteristics. These characteristics are important indicators of worth using Integer Programming (Williams[1], 1985). Firstly, the problem is concerned with discrete decisions, such as whether a unit of manpower should be assigned to certain maintenance tasks or work shift and the total number of manpower assigned. Secondly, in order to model such complex situations of scheduling requirements, it usually needs extra logical conditions on its original LP model, such as ‘If … then’. Thirdly, as far as such sequencing problems and allocation problems are concerned, they have the characteristics of a very large number of feasible solutions. And they are loosely referred to as ‘combinatorial problems’. Descriptions of our model in terms of such characteristics are given in section 3.4.8.

According to the demand profile for the rotating timetable cycle, the demand fluctuates a lot. So flexible strategies might be a good option to improve the efficiency of the maintenance management. By using flexible strategies, the author means flexible shift length, shift beginning time, shift ending time, and flexible day-off pattern.

In our problem, the planning period is 7 days a week, Monday through Sunday. We abbreviate the days to be 1, 2, 3, 4, 5, 6 and 7. Besides, each worker must receive n off-days each week, where n = 2, for 5-day, workweeks respectively.

It is generally recognized that the choice of the formulation was of crucial importance. Great care should be taken in formulating this real-world problem into precise mathematical model.

The proposed integer programming model incorporates deviational variables from introduction of goal programming formulation; preemptive goal programming has been applied. When formulating the achievement function, P_1 represents the first priority level and P_2 represents the second priority level and so on. The model is then solved by optimizing the first level priority goals initially, from which the optimized solution obtained is added as a constraint to the original sets of constraints. The model is thus solved by optimizing the second priority goals.

The model would be a mixed integer programming (MIP) model instead of pure integer programming (PIP) model because it involves both integer variables and conventional continuous variables. Some integer variables are restricted to 0-1 variables, which may represent yes/no decisions, logical conditions and dependent decisions (dependent relationships on two or more decisions). Relationships between these 0-1 variables can often be imposed using linear constraints without resorting to "=IF()" statements. So integer variables, conventional continuous variables, linear constraints are used to model the real problem. These conditions are occasionally of a logical nature which cannot be modeled by conventional LP.

By using the following model, the decision makers need not determine either the shift type with fixed beginning time and ending time or the number of required personnel to assign as model parameters.

3.4.1 Terms Definition

The following terms are essential and thus defined for model development.

A *week* – 7-day period from Monday to Sunday; the days of the week are represented by *d* {Mo, Tu, We, Th, Fr, Sa, Su}.

A *weekend* – the Sa-Su pair of consecutive days at the end of a week.

Shift – The detailed specification of working period of starting time and ending time to be allocated to each workforce.

Offday – the day on which a workforce is not assigned any shift.

Shift length – the duration between assigned starting time and ending time.

Tour – An allowable work pattern for an engineer, which consists of a set of work days for the work, as well as the shift starting and finishing times on each work day.

3.4.2 Initial Assumptions

The following initial assumptions are made before developing the new mathematical model. It is supposed, without loss of generality, that:

- The maintenance services are provided on 24-hour continuous base;
- Task starting time and finishing time are known beforehand;
- All numerical data are non-negative numbers;
- Tasks per day are already put in ascending order of their starting time;
- There are same amounts of tasks that should be performed per day.
- For each task, one qualified engineer is required according to requirements combinations;

- Time scales involved in the model are either sharp or at half of an hour, which is easy for calculating when solving the model;

The following subsection lists the notation used in the specific mathematical model to formulate the problem under consideration.

3.4.3 Notations

In order to develop the set of constraints and objectives, the following notions are defined for use in the model.

- **Indices and Sets**

i index for the engineer, 1, …, I;

j index for the day within the planning period, 1, …, J;

k index for the task throughout a day, 1, …, K;

O_k set of tasks incompatible with task k, defined by

$O_k = \{l > k: s_j < f_k\}$.

- **Parameters**

c_{ijk} utility by assigning engineer i to task k on day j, derived through previous FMADM model;

s_{jk} starting time of task *k* on day *j*;

f_{jk} finishing time of task *k* on day *j*;

SL_{min} minimum shift length;

SL_{max} maximum shift length;

SB_{ear} earliest shift beginning time;

SE_{lat} latest shift ending time;

$CscD$ maximum consecutive working days;

$SLen$ preferred shift length per day;

M_1 an upper bound of tasks assigned to each engineer per day;

M_2 a large positive number;

M_3 an upper bound of shift ending time;

M_4 minimum free time between consecutive shifts;

M_5 a large number equals maximum shift length;

ε_1 a small tolerance value;

ε_2 a small tolerance value;

m a threshold value;

- **Variables**

x_{ijk} 1 if engineer *i* is assigned task *k* on day *j* when the associated utility is more than 0, and 0 otherwise;

y_{ij} 1 if engineer *i* is day-on on day *j*, and 0 otherwise;

b_{ij} shift beginning time for engineer *i* on day *j*;

e_{ij} shift ending time for engineer *i* on day *j*;

wd^+_i positive deviation from the consecutive work days;

wd^-_i negative deviation from the consecutive work days;

ot^+_i positive deviation from the normal shift duration per day;

ot^-_i negative deviation from the normal shift duration per day;

The first 4 variables are regarded as original variables as well as decision variables in the integer programming model. On the other hand, the latter 4 variables are deviational variables to transfer soft constraints into optimization goals. Decision variable y_{ij} could also be considered as indicator variables indicating the personnel's states of either day-on or day-off. Three-dimensional variable x_{ijk} vector is used to denote the decision of either assigning engineer *i,* to task *k,* on day *j* or not.

3.4.4 Integer Goal Programming Model

The problem to be implemented in modeling packages can now be stated as:

The goal is to find an optimal rostering and daily maintenance assignment plan that satisfies various organizational constraints at minimum penalties and maximum possible utilities.

Integer Problem *P* 1

Minimize $P_1\sum_{i=1}^{I} OWD_i + P_2\sum_{i=1}^{I}\left(EC\cdot UOT_i + TC\cdot OOT_i\right)$ (3.8)

Maximize $P_3\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} c_{ijk}\cdot x_{ijk}$ (3.9)

Subject to

$\sum_{i=1}^{I} x_{ijk} = 1 \quad j=1, \ldots, J;\ k=1, \ldots, K,$ (3.10)

$x_{ijk} + x_{ijl} \le 1 \quad i=1, \ldots, I;\ j=1, \ldots, J;\ k=1, \ldots, K;\ l\in O_k,$ (3.11)

$\sum_{k=1}^{K} x_{ijk} - M_1\cdot y_{ij} \le 0 \quad i=1, \ldots, I;\ j=1, \ldots, J,$ (3.12)

$\sum_{k=1}^{K} x_{ijk} - m\cdot y_{ij} \ge 0 \quad i=1, \ldots, I;\ j=1, \ldots, J,$ (3.13)

$b_{ij} \le s_{jk}\cdot x_{ijk} + M_2\cdot(1-x_{ijk}) - \varepsilon_1 \quad i=1, \ldots, I;\ j=1, \ldots, J;\ k=1, \ldots, K,$ (3.14)

$e_{ij} + M_3\cdot(1-x_{ijk}) \ge f_{jk}\cdot x_{ijk} + \varepsilon_2 \quad i=1, \ldots, I;\ j=1, \ldots, J;\ k=1, \ldots, K,$ (3.15)

$b_{ij} \ge SB_{ear}\cdot y_{ij} \quad i=1, \ldots, I;\ j=1, \ldots, J,$ (3.16)

$e_{ij} \le SE_{lat}\cdot y_{ij} \quad i=1, \ldots, I;\ \mathrm{j}=1, \ldots, J,$ (3.17)

$e_{ij} - b_{ij} \ge SL_{\min}\cdot y_{ij} \quad i=1, \ldots, I;\ j=1, \ldots, J,$ (3.18)

$e_{ij} - b_{ij} \le SL_{\max}\cdot y_{ij} \quad i=1, \ldots, I;\ j=1, \ldots, J,$ (3.19)

$b_{i(j+1)} - e_{ij} + M_4 + M_5\cdot(1-y_{i(j+1)}) > 0 \quad i=1, \ldots, I;\ j=1, \ldots, J\text{-}1,$ (3.20)

$$\sum_{j=1}^{J} y_{ij} + wd^{-}{}_{i} - wd^{+}{}_{i} = CscD \quad i=1, \ldots, I, \tag{3.21}$$

$$e_{ij} - b_{ij} + ot^{-}{}_{i} - ot^{+}{}_{i} = SLen \cdot y_{ij} \qquad i=1, \ldots, I, \tag{3.22}$$

Detailed explanation of constraints and objectives are given below.

3.4.5 Hard Constraints

While using mathematical expressions to represent the real problem, hard constraints are necessary to denote those strict rules, regulations, requirements that had to be satisfied completely. Any violation or flexibility should not be allowed. Within them, some represent logical connections between different decisions or states by using linear constraints involving the 0-1 indicator variables. The problem of tour scheduling and task assignment studied in the research project has the following hard constraints, which are also called rigid constraint.

Constraint 1:

This constraint is to guarantee that for each task k on day j, there is always assignment to an engineer within I candidates so that the required maintenance service is always covered. So for each of the tasks an engineer should be assigned. This constraint is thus regarded as hard constraint that must be strictly satisfied without any compromise. Within hard constraints, this is the only equality constraints, which restricts the values of the decision variables in such a way that the left-hand side exactly equals the right-hand side. The following equation represents this constraint.

$$\sum_{i=1}^{I} x_{ijk} = 1 \qquad j=1, \ldots, J;\ k=1, \ldots , K, \tag{C.1}$$

Constraint 2:

This constraint, as a generalized upper bounding constraint from the aspect of computational efforts (Williams[1], 1985), is to ensure feasibility, by avoiding assigning an engineer i to overlapping task, i.e. set O_k is defined below to indicate time-crashed task l for task k. For instance, on day *2*, the work stretch of task 3 is overlapped with both task 2 and 4. So for all engineers 1… I, they cannot be assigned to task 3 and at the same time neither task 2 nor task 4. This hard constraint could be written as:

$$x_{ijk} + x_{ijl} \leq 1 \quad i=1, \ldots, I;\ j=1, \ldots, J;\ k=1, \ldots , K;\ l \in O_k, \tag{C.2}$$

Constraint 3:

This constraint represents the relation between individual task assignments on each day with day-on variables for each engineer i through the use of large positive number M_1, whose value is taken as the upper bound of task assigned to an engineer per day. M_1 should be made as small as without imposing a spurious restriction on $\sum_{k=1}^{K} x_{ijk}$ such that the size of the feasible region of the LP problem corresponding to this MIP problem could be reduced (Williams, 1985). By this constraint the decision variable y_{ij} on day-on/off is defined. It indicates that if the engineer i is assigned a day-off on day j (y_{ij}=0), he then should not be assigned any task that day ($\sum_{k=1}^{K} x_{ijk} \leq 0$), which condition is shown below.

$y_{ij}=0 \rightarrow \sum_{k=1}^{K} x_{ijk} = 0$ or $\sum_{k=1}^{K} x_{ijk} > 0 \rightarrow y_{ij}=1$

So with the help of decision variable y_{ij} , this constraint is imposed as a logical condition, implying the 'If …then…' condition. The general form of constraint 3 takes the following:

$$\sum_{k=1}^{K} x_{ijk} - M_1 \cdot y_{ij} \leq 0 \quad i=1, \ldots, I; j=1, \ldots, J, \tag{C.3}$$

Constraint 4:

This constraint represents the relation between individual task assignments on each day and day-on variables for each engineer i through the use of a threshold value m. It indicates that if the engineer i is assigned a day-on on day j (y_{ij}=1), he then should be assigned at least one task that day ($\sum_{k=1}^{K} x_{ijk} \geq m$), as a logical condition below usually states.

$$y_{ij}=1 \rightarrow \sum_{k=1}^{K} x_{ijk} > 0 \text{ or } \sum_{k=1}^{K} x_{ijk} = 0 \rightarrow y_{ij}=0$$

Constraint 3 and 4 thus together present a sufficient link between x_{ijk} and y_{ij}. Here y_{ij} also serves as an indicator variable to distinguish between the state x_{ijk} =0 and the state where x_{ijk} >0.

$$\sum_{k=1}^{K} x_{ijk} - m \cdot y_{ij} \geq 0 \qquad i=1, \ldots, I; j=1, \ldots, J, \qquad \text{(C.4)}$$

Constraint 5:

This constraint is to define the individual shift beginning time for each engineer i on day j, by associating it with the starting time of the first assigned task during the day. A large number M_2 and a small tolerance value ε_1 are introduced considering engineer i is assigned none of the tasks, in which case a day-off is assigned to engineer i. $M2$ will take the possible largest hourly value to throw the loosest boundary on the shift beginning time b_{ij} when some tasks assignments are not made for engineer i on day j. The general form of constraint 5 can be written as:

$$b_{ij} \leq s_{jk} \cdot x_{ijk} + M_2 \cdot (1 - x_{ijk}) - \varepsilon_1 \qquad i=1, \ldots, I; j=1, \ldots, J; k=1, \ldots, K, \qquad \text{(C.5)}$$

Constraint 6:

Similar to C.5, this constraint associates the individual shift ending time for each engineer i on day j with the finishing time of the last assigned task during the day. A large number M_3 and a small tolerance value ε_2 are introduced considering engineer i is assigned none of the tasks, in which case a day-off is assigned to engineer i. The general form of constraint 6 can be written as:

$$e_{ij} + M_3 \cdot (1 - x_{ijk}) \geq f_{jk} \cdot x_{ijk} + \varepsilon_2 \quad i=1, \ldots, I;\ j=1, \ldots, J;\ k=1, \ldots, K, \tag{C.6}$$

Constraint 7:

The purpose of constraint 7 is to give a bound (SB_{ear}) on the earliest shift beginning time for each engineer i on day j, prohibiting that the solution of shift beginning time would be far earlier than the earliest assigned task, under the condition that engineer i is day-on on day j. This constraint is also called simple bounding constraint from the aspect of computational efforts, incorporating an indicator variable y_{ij} in the right-hand side.

$$b_{ij} \geq SB_{ear} \cdot y_{ij} \qquad i=1, \ldots, I;\ j=1, \ldots, J, \tag{C.7}$$

Constraint 8:

Similarly, constraint 8 tries to give a bound (SE_{lat}) on the latest shift ending time for each engineer *i* on day *j*, prohibiting that the solution of shift ending time would be far later than the latest assigned task, under the condition that engineer *i* is day-on on day *j*. This constraint is also called simple bounding constraint, incorporating an indicator variable y_{ij} in the right-hand side.

$$e_{ij} \le SE_{lat} \cdot y_{ij} \qquad i=1, \ldots, I;\ j=1, \ldots, J, \qquad \text{(C.8)}$$

Constraint 9:

This constraint provides the identical lower bound (SL_{min}) on shift length for engineer *i* on day *j*. Besides, it associates with day-on variables y_{ij}. Although penalty costs are allowed for both shorter and longer shift length, there is still hard constraint on limiting the possible shift length within certain ranges. The general form of constraint 9 can be written as:

$$e_{ij} - b_{ij} \ge SL_{\min} \cdot y_{ij} \qquad i=1, \ldots, I;\ j=1, \ldots, J, \qquad \text{(C.9)}$$

Constraint 10:

Apart from constraint 9, constraint 10 presents an upped bound (SL_{max}) on shift length, identical for engineer *i* on day *j*. It also associates with day-on variables y_{ij}. Together with constraint 9, the unbroken range of shift length is specified. If the engineer *i* is scheduled day off on day *j*, constraint 7, 8, 9 and10 would make sure that $e_{ij}=b_{ij}=0$ accordingly. Constraints 9 and 10 jointly belong to ranged constraints from the aspect of computational efforts. The general form of constraint 10 can be written as:

$$e_{ij} - b_{ij} \leq SL_{\max} \cdot y_{ij} \qquad i=1, \ldots, I; j=1, \ldots, J, \qquad \text{(C.10)}$$

Constraint 11:

This constraint guarantees that there is sufficient rest time (M_4) between consecutive shifts on consecutive days, j and $(j+1)$. The cases of day-off assignment on either or both consecutives days are included by introducing M_5, a large number equal to maximum shift length. It models several sequentially dependent decisions in which day-on decisions will affect decisions of shift beginning time on the next day.

$$b_{i(j+1)} - e_{ij} + M_4 + M_5 \cdot (1 - y_{i(j+1)}) > 0 \quad i=1, \ldots, I; j=1, \ldots, J\text{-}1, \qquad \text{(C.11)}$$

3.4.6 Soft Constraints

Sometimes, it is neither possible nor practical to satisfy all specified constraints in the model, by considering all constraints as hard constraints. Therefore soft constraints are needed which can be broken at a certain cost such that feasible solutions may exist in the end. While specifying soft constraints, an amount of surplus or/and slack is allowed to offset the target goal value. The amount of surplus or/and slack is represented by surplus or/and slack variables respectively, which are then incorporated into the objective functions. The problem of tour scheduling and task assignment investigated in this research has the following soft constraints.

Constraint 12:

Two vectors of deviational variables (wd^{-}_{i} & wd^{+}_{i}) are introduced in this constraint such that the original constraint $\sum_{j=1}^{J} y_{ij} \le CscD$ could be transformed into optimized goals, by minimizing the associated deviational variables.

$$\sum_{j=1}^{J} y_{ij} + wd^{-}_{i} - wd^{+}_{i} = CscD \quad i=1, \ldots, I, \qquad \text{(C.12)}$$

Constraint 13:

In constraint 13, two vectors of deviational variables (ot^{-}_{i} & ot^{+}_{i}) are introduced so that the original constraint $e_{ij} - b_{ij} \le SLen$ could transform into optimization goals, by minimizing the associated deviational variables. The indicator variable y_{ij} is linked also because the original constraint need not hold if the engineer is day off.

$$e_{ij} - b_{ij} + ot^{-}_{i} - ot^{+}_{i} = SLen \cdot y_{ij} \qquad i=1, \ldots, I, \qquad \text{(C.13)}$$

3.4.7 Multiple Objectives

The objectives in our problem are multiple. Therefore, lexicographic GP, preemptive approach, is utilized to seek an minimum/maximum of an ordered vector of the unwanted goal deviation. There are altogether 3 objectives to be optimized. To deal with these multiple objectives in the model, a goal associated with each soft constraint is introduced.

Each of them represents a soft constraint that could be formulated as a goal programming model. The following is the mathematical formulation of each objective.

Minimize $P_1 \sum_{i=1}^{I} OWD_i + P_2 \sum_{i=1}^{I} \left(EC \cdot UOT_i + TC \cdot OOT_i \right)$

Maximize $P_3 \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} c_{ijk} \cdot x_{ijk}$

The priorities of the goals are specified as follows:

Priority 1: minimization of total deviation of off-days from the target goal

Priority 2: linear combination of two objective functions – minimization of the total weighted earliness and tardiness

Priority 3: maximization of total utilities of task assignments

3.4.8 Summary of the Important Considerations in the Models

The followings are major considerations of this model far side.

- This is an optimization model incorporating dichotomous decisions (yes/no decisions) and scheduling problems.
- This is a Mixed Integer Linear Program, a model in which only some of the decision variables are required to take integral values and others can assume any nonnegative numbers.
- It has properties of *assignment problem.*
- Notice that the constraint functions and the objective functions are linear functions of the decision variables; there are no products or quotients of variables, no exponents (other than 1), no logarithmic, exponential or trigonometric terms, and no IF() statements within these functions.
- As can be seen, integer programming can be obtained by adding an "integrality condition" to the LP model, which forces some decision variables and other deviational variables to take on only integer values; then the model becomes an Integer Optimization Model.
- Situations with unacceptable assignments: in terms of uncertain for various reasons, *zero* has been assigned to that unacceptable assignment to eliminate the use of that assignment; the utility of making that assignment would

be much larger than that of any other feasible alternative.

- This mixed-integer model does not attach to any specialized or simplified model including the General Assignment Problem and Shortest-path Problem.
- Use deterministic model instead of probabilistic (stochastic) model (models in which some inputs to the model are not known with certainty; uncertainty is incorporated via probabilities on these “random” variables) because probabilistic models can often used for strategic decision making involving an organization’s relationship to its environment; probabilistic models are very useful when there are only a few uncertain model inputs and few or no constraints.
- While formulating models, cautions are taken applying equality constraints, because the resultant model may be over-constrained or low payoff decisions may be produced when it is optimized, or in the worst case, there may not be feasible solutions at all.
- This problem turns out to have the characteristic of a very large number of feasible solutions, arising from many combinations of allocating personnel to daily maintenance tasks and corresponding working days. So the problem would fall into a category of *combinatorial problems*, which can be further divided into *sequencing problems* and *allocation problems*.

- Also a constrained optimization model, which is described as a constrained optimization model representing the problem of allocating scarce resources (resources that are subject to constraints) in such a way as to optimize an objective of interest; a problem in which you want to maximize or minimize some performance measure subject to a set of constraints.

- Other nonnegativity inequalities constraints (e.g. $x_{111} \geq 0$), although not specified here, should also be added into hard constraints. But since while using standard software, nonnegativity constraint is usually set default.

- The above mathematical model will become a solver-ready model after numerical values are inserted into it.

- No specialist IP algorithm is used to exploit its structure because it is of no particular types of model. Instead, standard software package with built-in IP general methods, i.e. branch-and-bound, is utilized to solve the model, discussed in the following chapter.

CHAPTER 4

COMPUTATIONAL RESULTS

Having described the detailed methodology in Chapter 3, the purpose of this Chapter is to use a numerical example to illustrate the proposed 2-stage methodology of the utility assessment using fuzzy MADM approach and optimization through integer goal programming model. The author will demonstrate how task assignment and personnel scheduling/rostering are integrated and solved through computational results. Results are reported and discussions and implications are given based on these model results, through which the proposed model is validated.

Computational results are presented in the following sections using the modeling language MPL and the LP-IP solver CPLEX. In the first stage, utility assessment is carried out through Microsoft Excel spreadsheet calculation. After a set of utility indexes for all feasible assignments are obtained, an integer programming model with sequential goal optimization is formulated in MPL. As discussed in section 3.2.6 of Chapter 3, MPL is a modeling system that allows us to set up complicated linear programming models, involving thousands of variables and constraint, in a clear, concise, and efficient way. The variables of the linear programs may be continuous or integer. In addition, CPLEX is a solver which computes an optimal solution for the integer linear programming formulation.

Due to the limitations on computational efforts in solving integer programming model, the author is not going to input the whole set of real-world data into the methodology, whose size is too large to handle efficiently. Instead, a simplified version of data is input for easy of illustration. Hence, in this example, it is assumed that there are 5 engineers, 7 days, and 8 tasks per day altogether. The input data are shown below.

4.1 Description of Input Data

Since there are two sequential stages to implement the methodology with different algorithms, there are one set of input data for each stage.

In the first stage – Utility Assessment through Fuzzy Multiple Attribute Decision Making, input data includes personnel profile with their information in terms of various requirements, and also the description of each maintenance task on each day. They are listed in the tables 4.1 and 4.2.

Table 4.1 Personnel Profile in Terms of Licenses, Prior Experience and Training Requirements

	License		*Prior experience (in years)*			*Training*		
	ACL	ALL	OVE	ATE	ALE	PYF	HFT	TET
E1	c2; c3	a1; a2; a3	8	7[c2]; 2[c3]	8[a1]; 4[a2]; 5[a3]	Very little	Almost all	About half
E2	c1; c3	a1; a2; a3	7	6[c1]; 3[c3]	7[a1]; 2[a2]; 3[a3]	Minority	Very little	Minority
E3	c1; c2; c3	a2; a3	13	11[c1]; 7[c2]; 3[c3]	13[a2]; 2[a3]	Minority	Majority	Almost all
E4	c1; c2; c3	a1; a3	4	3[c1]; 4[c2]; 1[c3]	4[a1]; 3[a3]	Majority	About half	Majority
E5	c1; c2; c3	a1; a2	10	5[c1]; 1[c2]; 7[c3]	4[a1]; 3[a2]	Almost all	Minority	Minority

Table 4.2 Task Composition in Terms of License Requirements

	D1	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*
T1	a1; c1	a1; c1	a1; c2	a2, c3	a1, c1	a1, c2	a1, c1
T2	a1; c2	a3; c2	a2; c1	a3, c3	a1, c3	a2, c3	a2, c2
T3	a2; c1	a2; c1	a2; c1	a2, c1	a2, c2	a2, c2	a2, c3
T4	a3; c2	a1; c3	a1; c3	a1, c1	a3, c3	a1, c1	a3, c1
T5	a2; c2	a3; c2	a2; c2	a2, c2	a2, c2	a1, c1	a2, c2
T6	a1; c3	a1; c3	a1; c3	a1, c3	a1, c3	a3, c3	a1, c3
T7	a3; c3	a3; c3	a2; c3	a3, c3	a3, c3	a3, c3	a3, c3
T8	a2; c1	a2; c1	a3; c1	a1, c2	a2, c1	a2, c1	a2, c1

4.2 Computational Results of Stage 1- FMADM Process

Calculations of stage 1 are made in Microsoft Excel spreadsheet according to the proposed methodology in section 3.3.3 of Chapter 3.

The planning horizon is one week, involving seven days, since the demand based on flight schedules is replicated on a weekly basis for each month period.

4.2.1 Determining Weights of the Attributes

According to the suggested FMADM approach, the first step is to measure attribute importance through pair-wise comparisons for each level of attributes and they are demonstrated in Table 4.3. The comparisons then lead to the attribute weights in Table 4.4.

Table 4.3 Pair-wise Comparisons of Decision Attributes*

	EP	HF	TR	Weight	HF	LED	COS	COO	Weight
EP	1	3	2	*0.55*	**LED**	1	2	3	*0.54*
HF	1/3	1	2	*0.26*	**COS**	1/2	1	2	*0.3*
TR	1/2	1/2	1	*0.19*	**COO**	1/3	1/2	1	*0.16*
EP	OVE	ALE	ACE	Weight	**TR**	HFT	TET	PYF	Weight
OVE	1	5	1	*0.45*	**HFT**	1	2	3	*0.54*
ALE	1/5	1	1/5	*0.11*	**TET**	1/2	1	1	*0.30*
ACE	1	5	1	*0.45*	**PYF**	1/4	1	1	*0.16*

Table 4.4 Attribute Weights Derived from the Pair-wise Comparisons*

Attribute	***EP***			***HF***			***TR***		
	OVE	**ALE**	**ACE**	**LED**	**COS**	**COO**	**PYF**	**HFT**	**TET**
Weight	0.2475	0.0605	0.2475	0.1400	0.0800	0.0400	0.0361	0.1064	0.0456

* These abbreviations are referred to Figure 3.6.

4.2.2 Determining Scores of the Alternatives

After a vector of attributes weights has been obtained through pair-wise comparison, the next step is to determine scores of alternatives under each of the attributes: the various methods are grouped by level-1 attributes proposed in section 3.3.3.2 in Chapter 3. Accordingly, same method is utilized for each level-1 attribute.

4.2.2.1 Assess the Prior Experiences of the Alternatives

Under the level-1 attribute, Prior experience, for every factor of Prior Experiences, a function, defined below, based on expert advice, is to transfer the actual prior experience in years to triangular fuzzy numbers for each alternative. Results are shown in Table 4.5. For those who do not possess required licenses, the scores would thus be zero.

$$\begin{cases} l = \dfrac{x-1}{10} \\ m = \dfrac{x}{10} \\ n = \dfrac{x+1}{10} \end{cases}$$

Table 4.5 Engineers' Scores against the Criteria of Prior Experience

	OVE	***ALE***			***ACE***		
		a1	**a2**	**a3**	**c1**	**c2**	**c3**
E1	(0.7,0.8,0.9)	(0.7,0.8,0.9)	(0.3,0.4,0.5)	(0.4,0.5,0.6)	0	(0.6,0.7,0.8)	(0.1,0.2,0.3)
E2	(0.6,0.7,0.8)	(0.6,0.7,0.8)	(0.1,0.2,0.3)	(0.2,0.3,0.4)	(0.5,0.6,0.7)	0	(0.2,0.3,0.4)
E3	(1.2,1.3,1.4)	0	(1.2,1.3,1.4)	(0.1,0.2,0.3)	(1,1.1,1.2)	(0.6,0.7,0.8)	(0.2,0.3,0.4)
E4	(0.3,0.4,0.5)	(0.3,0.4,0.5)	0	(0.2,0.3,0.4)	(0.2,0.3,0.4)	(0.3,0.4,0.5)	(0.9,1,1.1)
E5	(0.6,0.7,0.8)	(0.3,0.4,0.5)	(0.2,0.3,0.4)	0	(0.4,0.5,0.6)	(0.9,1,1.1)	(0.6,0.7,0.8)

4.2.2.2 Assign the Alternatives' Scores against the Human Factor

While assessing alternatives in terms of various human factors, pair-wise comparisons in triangular fuzzy numbers are adopted for all of the three human factors, yielding the following results from Table 4.6 to 4.8.

Table 4.6 Pair-wise Comparison under Subcriteria of Human Factor – Leadership

	E1	***E2***	***E3***	***E4***	***E5***	***Fuzzy score***
E1	(1/2, 1, 1)	(2, 3, 4)	(1/8, 1/7, 1/6)	(1/2, 1, 1)	(1/2, 1, 1)	***(1, 1.5, 1.74)***
E2	(1/4, 1/3, 1/2)	(1/2, 1, 1)	(1/4, 1/3, 1/2)	(2, 3, 4)	(2, 3, 4)	***(0.71, 0.9, 1.15)***
E3	(6, 7, 8)	(2, 3, 4)	(1/2, 1, 1)	(8, 9, 9)	(8, 9, 9)	***(4.21, 5.38, 5.93)***
E4	(1/2, 1, 1)	(1/4, 1/3, 1/2)	(1/9, 1/9, 1/8)	(1/2, 1, 1)	(8, 9, 9)	***(0.71, 1.12, 1.25)***
E5	(1/2, 1, 1)	(1/4, 1/3, 1/2)	(1/9, 1/9, 1/8)	(1/9, 1/9, 1/8)	(1/2, 1, 1)	***(0.4, 0.64, 0.71)***

Table 4.7 Pair-wise Comparison under Subcriteria of Human Factor – Communication

	E1	***E2***	***E3***	***E4***	***E5***	***Fuzzy score***
E1	(1/2, 1, 1)	(1/4, 1/3, 1/2)	(1/6, 1/5, 1/4)	(1/4, 1/3, 1/2)	(2, 3, 4)	***(0.57, 0.76, 0.96)***
E2	(2, 3, 4)	(1/2, 1, 1)	(1/4, 1/3, 1/2)	(1/2, 1, 1)	(4, 5, 6)	***(1.15, 1.72, 2.05)***
E3	(4, 5, 6)	(2, 3, 4)	(1/2, 1, 1)	(2, 3, 4)	(1/8, 1/7, 1/6)	***(1.07, 1.45, 1.76)***
E4	(2, 3, 4)	(1/2, 1, 1)	(1/4, 1/3, 1/2)	(1/2, 1, 1)	(4, 5, 6)	***(1.1, 1.63, 1.91)***
E5	(1/4, 1/3, 1/2/)	(1/6, 1/5, 1/4)	(1/8, 1/7, 1/6)	(1/6, 1/5, 1/4)	(1/2, 1, 1)	***(0.26, 0.34, 0.42)***

Table 4.8 Pair-wise Comparison under Subcriteria of Human Factor – Cooperation

	E1	***E2***	***E3***	***E4***	***E5***	*Fuzzy score*
E1	(1/2, 1, 1)	(2, 3, 4)	(1/2, 1, 1)	(1/6, 1/5, 1/4)	(1/4, 1/3, 1/2)	*(0.7, 0.95, 1.12)*
E2	(1/4, 1/3, 1/2)	(1/2, 1, 1)	(1/4, 1/3, 1/2)	(1/8, 1/7, 1/6)	(1/6, 1/5, 1/4)	*(0.36, 0.48, 1.12)*
E3	(1/2, 1, 1)	(2, 3, 4)	(1/2, 1, 1)	(1/6, 1/5, 1/4)	(1/4, 1/3, 1/2)	*(0.7, 0.95, 1.12)*
E4	(4, 5, 6)	(6, 7, 8)	(4, 5, 6)	(1/2, 1, 1)	(2, 3, 4)	*(2.08, 2.69, 3.14)*
E5	(2, 3, 4)	(4, 5, 6)	(2, 3, 4)	(1/4, 1/3, 1/2)	(1/2, 1, 1)	*(1.41, 1.89, 2.29)*

4.2.2.3 Assign Scores to the Alternatives against Training

The alternatives were assigned scores directly against the factor of training according to the defined fuzzy scales in Table 3.3 in Chapter 3 and their own profile information, which yields resultant scores in Table 4.9.

Table 4.9 Engineers' Scores against The Criteria of Training

	Training		
	PYT	**HFT**	**HFT**
E1	(0, 0.1, 0.1)	(0.9, 1, 1)	(0.4, 0.5, 0.6)
E2	(0.2, 0.3, 0.4)	(0, 0.1, 0.1)	(0, 0.1, 0.1)
E3	(0.2, 0.3, 0.4)	(0.6, 0.7, 0.8)	(0.9, 1, 1)
E4	(0.6, 0.7, 0.8)	(0.4, 0.5, 0.6)	(0.6, 0.7, 0.8)
E5	(0.9, 1, 1)	(0, 0.1, 0.1)	(0, 0.1, 0.1)

It should be noted that for each different task, experience criteria are different; but human factor and training remain the same. Thus, only alternatives' scores against experience criteria will vary dependent on each task composition.

4.2.3 Synthesize the Attribute Weights and the Fuzzy Scores of Alternatives

Attribute weights and scores of alternatives in triangular fuzzy numbers obtained from the above steps were input to a synthesizing process based on the formula below. Utility indexes in triangular fuzzy numbers are shown in Table 4.10.

$$\tilde{a}_i = \sum_{j=1}^{9} w_j \cdot \tilde{s}_{ij}$$

Table 4.10 Fuzzy Utility Indexes Obtained from Utility Assessment

		T1	*T2*	*T3*	*T4*	*T5*	*T6*	*T7*	*T8*
D1	**E1**	0	(0.5045,0.5817,0.6426)	0	(0.4863,0.5635,0.6244)	(0.4803,0.5575,0.6184)	(0.3807,0.4579,0.5188)	(0.3626,0.4398,0.5007)	0
	E2	(0.3559,0.4324,0.4951)	0	(0.3196,0.4021,0.4649)	0	0	(0.2817,0.3581,0.4209)	(0.2448,0.3213,0.3840)	(0.3257,0.4021,0.4649)
	E3	0	0	(0.7641,0.9142,0.9891)	(0.6712,0.7487,0.8235)	(0.7377,0.8152,0.8901)	0	(0.5722,0.6497,0.7245)	(0.8367,0.9142,0.9891)
	E4	(0.2887,0.3643,0.4399)	(0.3143,0.3890,0.4647)	0	(0.3569,0.3830,0.4586)	0	(0.2513,0.3148,0.3904)	(0.2331,0.3087,0.3844)	0
	E5	(0.3928,0.4674,0.5241)	(0.2938,0.3684,0.4251)	(0.3867,0.4614,0.5180)	0	(0.2877,0.3624,0.4190)	(0.4544,0.5169,0.5736)	0	(0.3867,0.4614,0.5180)
D2	**E1**	0	(0.4863,0.5635,0.6244)	0	(0.3807,0.4579,0.5188)	(0.4863,0.5635,0.6244)	(0.3807,0.4579,0.5188)	(0.3626,0.4398,0.5007)	0
	E2	(0.3559,0.4324,0.4951)	0	(0.3257,0.4021,0.4649)	(0.2817,0.3581,0.4209)	0	(0.2817,0.3581,0.4209)	(0.2575,0.3339,0.3967)	(0.3257,0.4021,0.4649)
	E3	0	(0.8367,0.9142,0.9891)	(0.8367,0.9142,0.9891)	0	(0.6712,0.7487,0.8235)	0	(0.5722,0.6497,0.7245)	(0.8367,0.9142,0.9891)
	E4	(0.2887,0.3643,0.4399)	(0.3134,0.3890,0.4647)	0	(0.2392,0.3148,0.3904)	(0.3074,0.3830,0.4586)	(0.2392,0.3148,0.3904)	(0.2826,0.3582,0.7245)	0
	E5	(0.3928,0.4674,0.5241)	0	(0.3867,0.4614,0.5180)	(0.4423,0.5169,0.5736)	0	(0.4423,0.5169,0.5736)	0	(0.3867,0.4614,0.5180)
D3	**E1**	(0.5045,0.5817,0.6426)	0	0	(0.3807,0.4579,0.5188)	(0.4803,0.5575,0.6184)	(0.3807,0.4579,0.5188)	(0.3565,0.4337,0.4946)	0
	E2	0	(0.3257,0.4021,0.4649)	(0.3257,0.4021,0.4649)	(0.2817,0.3581,0.4209)	0	(0.2817,0.3581,0.4209)	(0.2514,0.3279,0.3906)	(0.3317,0.4082,0.4709)
	E3	0	(0.8367,0.9142,0.9891)	(0.8367,0.9142,0.9891)	0	(0.7377,0.8152,0.8901)	0	(0.6024,0.6799,0.7548)	(0.7702,0.8477,0.9225)
	E4	(0.3143,0.3890,0.4647)	0	0	(0.2392,0.3148,0.3904)	0	(0.2392,0.3148,0.3904)	0	(0.2826,0.3582,0.4339)
	E5	(0.2938,0.3684,0.4251)	(0.3867,0.4614,0.5180)	(0.3867,0.4614,0.5180)	(0.4423,0.5169,0.5736)	(0.2877,0.3624,0.4190)	(0.4423,0.5169,0.5736)	0	0
D4	**E1**	(0.4308,0.5080,0.5689)	(0.3626,0.4398,0.5007)	0	0	(0.4803,0.5575,0.6184)	(0.3807,0.4579,0.5188)	(0.3626,0.4398,0.5007)	(0.5045,0.5817,0.6426)
	E2	(0.2514,0.3279,0.3906)	(0.2575,0.3339,0.3967)	(0.3257,0.4021,0.4696)	(0.3559,0.4324,0.4951)	0	(0.2817,0.3581,0.4209)	(0.2575,0.3339,0.3967)	0
	E3	(0.6387,0.7162,0.7911)	(0.5722,0.6497,0.7245)	(0.8367,0.9142,0.9891)	0	(0.7377,0.8152,0.8901)	0	(0.5722,0.6497,0.7245)	0
	E4	0	(0.2331,0.3087,0.3844)	0	(0.2887,0.3643,0.4399)	0	(0.2392,0.3148,0.3904)	(0.2331,0.3087,0.3844)	(0.3134,0.3890,0.4647)
	E5	(0.4362,0.5109,0.5675)	0	(0.3867,0.4614,0.5180)	(0.3928,0.4674,0.5241)	(0.2877,0.3624,0.4190)	(0.4423,0.5169,0.5736)	0	(0.2938,0.3684,0.4251)
D5	**E1**	0	(0.3807,0.4579,0.5188)	(0.4803,0.5575,0.6184)	(0.3626,0.4398,0.5007)	(0.4803,0.5575,0.6184)	(0.3807,0.4579,0.5188)	(0.3636,0.4398,0.5007)	0
	E2	(0.3559,0.4324,0.4951)	(0.2817,0.3581,0.4209)	0	(0.2575,0.3339,0.3967)	0	(0.2817,0.3581,0.4209)	(0.2575,0.3339,0.3967)	(0.3257,0.4021,0.4649)
	E3	0	0	(0.7377,0.8152,0.8901)	(0.5722,0.6497,0.7245)	(0.7377,0.8152,0.8901)	0	(0.5722,0.6497,0.7245)	(0.8367,0.9142,0.9891)
	E4	(0.2887,0.3643,0.4399)	(0.2392,0.3148,0.3904)	0	(0.2331,0.3087,0.3844)	0	(0.2392,0.3148,0.3904)	(0.2331,0.3087,0.3844)	0
	E5	(0.3928,0.4674, .5241)	(0.4423,0.5169,0.5736)	(0.2877,0.3624,0.4190)	0	(0.2877,0.3624,0.4190)	(0.4423,0.5169,0.5736)	0	(0.3867,0.4614,0.5180)
D6	**E1**	(0.4302,0.5074,0.5683)	(0.3565,0.4337,0.4946)	(0.4803,0.5575,0.6184)	0	0	(0.3626,0.4398,0.5007)	(0.3626,0.4398,0.5007)	0
	E2	(0.2569,0.3334, 0.3961)	(0.2514,0.3279,0.3906)	0	(0.3559,0.4324,0.4951)	(0.3559,0.4324,0.4951)	(0.2575,0.3339,0.3967)	(0.2575,0.3339,0.3967)	(0.3378,0.4142,0.4770)
	E3	0	(0.6387,0.7162,0.7911)	(0.7377,0.8152,0.8901)	0	0	(0.5722,0.6497,0.7245)	(0.5722,0.6497,0.7245)	(0.7702,0.8477,0.9225)
	E4	0	0	0	(0.2887,0.3643,0.4399)	(0.2887,0.3643,0.4399)	(0.2331,0.3087,0.3844)	(0.2331,0.3087,0.3844)	0
	E5	(0.2938,0.3684,0.4251)	(0.4362,0.5109,0.5675)	(0.2877,0.3624,0.4190)	(0.3928,0.4674,0.5241)	(0.3928,0.4674,0.5241)	0	0	(0.3867,0.4614,0.5180)
D7	**E1**	0	(0.4803,0.5575,0.6184)	(0.3565,0.4337,0.4946)	0	(0.4803,0.5575,0.6184)	(0.3807,0.4579,0.5188)	(0.3626,0.4398,0.5007)	0
	E2	(0.2569,0.3334,0.3961)	0	(0.2514,0.3279,0.3906)	(0.3317,0.4082,0.4709)	0	(0.2817,0.3581,0.4209)	(0.2575,0.3339,0.3967)	(0.3257,0.4021,0.4649)
	E3	0	(0.7377,0.8152,0.8901)	(0.6387,0.7162,0.7911)	(0.7702,0.8477,0.9225)	(0.7377,0.8152,0.4190)	0	(0.5722,0.6497,0.7245)	(0.8367,0.9142,0.9891)
	E4	0	0	0	(0.2826,0.3582,0.4339)		(0.2392,0.3148,0.3904)	(0.2331,0.3087,0.3844)	0
	E5	(0.3928,0.4674,0.5241)	(0.2877,0.3624,0.4190)	(0.4362,0.5109,0.5675)	0	(0.2877,0.3624,0.4190)	(0.4423,0.5169,0.5736)	0	(0.3867,0.4614,0.5180)

4.2.4 Obtain the Final Utility Index

Fuzzy utility indexes were given a defuzzition process, transferring triangular fuzzy numbers into crisp numbers by using the following formula for each triangular fuzzy number $\widetilde{A}(l, m, n)$. See Table 4.11 for numerical results.

$$R(a) = \frac{2[\frac{1}{m-l}(\frac{1}{3}m^3 - \frac{1}{2}m^2 l + \frac{1}{6}l^3) + \frac{1}{n-m}(\frac{1}{3}m^3 - \frac{1}{2}m^2 n + \frac{1}{6}n^3)]}{n-l}$$

Table 4.11 Final Utility Indexes in Crisp Number Obtained from Stage 1 – FMADM

		T1	*T2*	*T3*	*T4*	*T5*	*T6*	*T7*	*T8*
D1	**E1**	0	0.4279	0	0.4141	0.4096	0.3346	0.3209	0
	E2	0.3156	0	0.2917	0	0	0.2597	0.232	0.2928
	E3	0	0	0.6829	0.5545	0.6044	0	0.4802	0.6786
	E4	0.2669	0.2858	0	0.3216	0	0.2345	0.2253	0
	E5	0.3421	0.2674	0.3376	0	0.2629	0.3813	0	0.3376
D2	**E1**	0	0.4141	0	0.3346	0.4141	0.3346	0.3209	0
	E2	0.3156	0	0.2928	0.2597	0	0.2597	0.2415	0.2928
	E3	0	0.6786	0.6786	0	0.5545	0	0.4802	0.6786
	E4	0.2669	0.2855	0	0.2298	0.281	0.2298	0.387	0
	E5	0.3421	0	0.3376	0.3795	0	0.3795	0	0.3376
D3	**E1**	0.4279	0	0	0.3346	0.4096	0.3346	0.3163	0
	E2	0	0.2928	0.2928	0.2597	0	0.2597	0.237	0.2974
	E3	0	0.6786	0.6786	0	0.6044	0	0.5029	0.6287
	E4	0.2858	0	0	0.2298	0	0.2298	0	0.2624
	E5	0.2674	0.3376	0.3376	0.3795	0.2629	0.3795	0	0
D4	**E1**	0.3723	0.3209	0	0	0.4096	0.3346	0.3209	0.4279
	E2	0.237	0.2415	0.2934	0.3156	0	0.2597	0.2415	0
	E3	0.5301	0.4802	0.6786	0	0.6044	0	0.4802	0
	E4	0	0.2253	0	0.2669	0	0.2298	0.2253	0.2855
	E5	0.3749	0	0.3376	0.3421	0.2629	0.3795	0	0.2674
D5	**E1**	0	0.3346	0.4096	0.3209	0.4096	0.3346	0.3209	0
	E2	0.3156	0.2597	0	0.2415	0	0.2597	0.2415	0.2928
	E3	0	0	0.6044	0.4802	0.6044	0	0.4802	0.6786
	E4	0.2669	0.2298	0	0.2253	0	0.2298	0.2253	0
	E5	0.3421	0.3795	0.2629	0	0.2629	0.3795	0	0.3376
D6	**E1**	0.3719	0.3163	0.4096	0	0	0.3209	0.3209	0
	E2	0.2411	0.237	0	0.3156	0.3156	0.2415	0.2415	0.3019
	E3	0	0.5301	0.6044	0	0	0.4802	0.4802	0.6287
	E4	0	0	0	0.2669	0.2669	0.2253	0.2253	0
	E5	0.2674	0.3749	0.2629	0.3421	0.3421	0	0	0.3376
D7	**E1**	0	0.4096	0.3163	0	0.4096	0.3346	0.3209	0
	E2	0.2411	0	0.237	0.2974	0	0.2597	0.2415	0.2928
	E3	0	0.6044	0.5301	0.6287	0.8638	0	0.4802	0.6786
	E4	0	0	0	0.2624	0	0.2298	0.2253	0
	E5	0.3421	0.2629	0.3749	0	0.2629	0.3795	0	0.3376

4.2.5 Results Analysis

In the above result table, for each entry corresponding to the assignment variable, value zero represent that the related engineer does not possess all the requisite licenses for performing that maintenance task.

The number which is larger than zero in each entry, denotes the consequential utility value for each match between certain engineer and certain task. They are in crisp number, making the subsequent model building and solution finding easier.

4.3 Results of Stage 2 – Integer Goal Programming Model

Before implementing on a large-scale data set, it is often a very good idea to build a small version of the model first to test the integer programming model. Experimentation with different solution strategies and possibly with reformulation may be required.

Numerical results from stage 1, which is a 5*7*8 matrix of utility index for each assignment as crisp entries, were entered into the integer goal programming model of stage-2 methodology, being the coefficient of the task assignment variables. The integer programming model is written in MPL and solved through CPLEX on a Dell PC Dimension 4600 under the Windows XP operating system.

4.3.1 Parameter Setting

Apart from task compositions containing license requirements input into the stage 1 model, other compositions, such as task starting time and finishing time are presented in Table 4.13. In order to make the illustration simple, the work duration of each maintenance task has been assumed to be 30 minutes, 1 hour or 90 minutes. Besides, their starting time and finishing time are also assumed to be at either first or half an hour.

Other parameter values used in model P1 are specified in Table 4.12. Minimum (SL_{min}) and maximum shift length (SL_{max}) is set as 6h and 12h respectively. The earliest shift beginning time (SB_{ear}) and latest shift ending time (SE_{lat}) is 7 and 24 respectively. The target work days are 5 and target shift length is 9. Penalty costs for both earliness and tardiness are set to be the same – 1 because these two deviations are regarded equivalently. Other useful parameters in model building (M_1, M_2, M_3, M_4, M_5, ε_1, ε_2, m) are also given numerical values as shown in Table 6.12.

Table 4.12 Parameter Settings for Integer Programming Model *P1*

Parameters	I	J	K	SL_{min}	SL_{max}	SB_{ear}	SE_{lat}	$CscD$	$SLen$	M_1
Value	5	7	8	6	12	7	24	5	9	9
Parameters	EC		TC	M_2	M_3	M_4	M_5	ε_1	ε_2	m
Value	1		1	24	26	12	12	0.5	0.5	0.1

Table 4.13 Task Starting Time and Task Finishing Time

		D1	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*
T1	s_{jk}	9:00	9:30	9:00	10:00	10:00	9:30	9:00
	f_{jk}	10:30	10:30	10:00	11:00	10:30	10:30	9:30
T2	s_{jk}	10:00	10:00	9:30	10:30	11:30	10:30	10:30
	f_{jk}	11:00	10:30	10:30	12:00	13:00	12:30	11:30
T3	s_{jk}	10:30	10:30	10:30	11:00	11:30	12:30	12:30
	f_{jk}	11:30	11:30	12:30	12:00	13:00	14:30	14:30
T4	s_{jk}	11:00	12:00	11:00	12:00	13:30	13:30	14:00
	f_{jk}	12:00	13:30	12:30	13:30	14:30	15:30	15:00
T5	s_{jk}	13:00	14:00	12:30	12:00	14:00	15:30	16:30
	f_{jk}	13:30	14:30	13:30	14:30	15:30	17:30	18:00
T6	s_{jk}	15:00	17:30	14:00	14:30	15:00	19:00	19:00
	f_{jk}	17:00	19:30	15:00	15:30	16:00	21:30	19:30
T7	s_{jk}	17:00	19:00	15:00	16:30	16:00	20:30	19:00
	f_{jk}	18:00	20:00	15:30	18:00	18:00	21:30	20:30
T8	s_{jk}	19:30	22:00	17:30	18:00	19:00	21:30	21:00
	f_{jk}	20:30	23:30	19:30	19:30	21:00	23:00	22:30

4.3.2 Model Instance

Having set the above parameter values, the solver-ready model described in Chapter 4 is thus specified, with specified values for all parameters for the model P1. Due to space limitations, only one equation is written down for each constraint for illustration. The following demonstration is thus just one of the modeling parts of the whole model programming file.

Priority 1: Minimize

$$(OWD_1 + OWD_2 + OWD_3 + OWD_4 + OWD_5) \quad (4.1)$$

Priority 2: Minimize

$$[UOT_1 + UOT_2 + UOT_3 + UOT_4 + UOT_5 + 3*(OOT_1 + OOT_2 + OOT_3 + OOT_4 + OOT_5)] \quad (4.2)$$

Priority 3: Maximize

$$\{[(c_{111}*x_{111}+c_{112}*x_{112}+\ldots)+(c_{121}*x_{121}+c_{122}*x_{122}+\ldots)+\ldots]+[(c_{211}*x_{211}+c_{212}*x_{212}+\ldots)+(c_{221}*x_{221}+c_{222}*x_{222}+\ldots)+\ldots]+[(c_{311}*x_{311}+c_{312}*x_{312}+\ldots)+(c_{321}*x_{321}+c_{322}*x_{322}+\ldots)+\ldots]+[(c_{411}*x_{411}+c_{412}*x_{412}+\ldots)+(c_{421}*x_{421}+c_{422}*x_{422}+\ldots)+\ldots]+[(c_{511}*x_{511}+c_{512}*x_{512}+\ldots)+(c_{521}*x_{521}+c_{522}*x_{522}+\ldots)+\ldots]\} \quad (4.3)$$

Subject to:

C.1:

$x_{168}+x_{268}+x_{368}+x_{468}+x_{568}=1$, for day 6, task 8, (4.4)

C.2:

$x_{267}+x_{266}\le 1$

$x_{267}+x_{268}\le 1$, for engineer 2, day 6, task 7, (4.5)

C.3:

$\sum_{k=1}^{K} x_{ijk} - M_5 \cdot y_{ij} \le 0$ for engineer 2, day 6, (4.6)

C.4:

$\sum_{k=1}^{K} x_{ijk} - 0.1 \cdot y_{ij} \le 0$ for engineer 2, day 6, (4.7)

C.5:

$b_{26}\le s_{61}\cdot x_{261}+24\cdot(1-x_{261})-0.5$ (4.8)

$b_{26}\le s_{62}\cdot x_{262}+24\cdot(1-x_{262})-0.5$ (4.9)

…

$b_{26}\le s_{68}\cdot x_{268}+24\cdot(1-x_{268})-0.5$ for engineer 2, day 6, (4.10)

C.6:

$e_{26}+M_3\cdot(1-x_{261})\ge f_{61}\cdot x_{261}+0.5$ for engineer 2, day 6, , (4.11)

$e_{27}+M_3\cdot(1-x_{271})\ge f_{71}\cdot x_{271}+0.5$ (4.12)

…

$e_{27}+M_3\cdot(1-x_{271})\ge f_{71}\cdot x_{271}+0.5$ (4.13)

C.7:

$b_{26}\ge 7\cdot y_{26}$ for engineer 2, day 6, (4.14)

C.8:

$e_{26}\le 24\cdot y_{26}$ for engineer 2, day 6, (4.15)

C.9:

$e_{26}-b_{26}\ge 6\cdot y_{26}$ for engineer 2, day 6, (4.16)

C.10:
$e_{26} - b_{26} \leq 12 \cdot y_{26}$ for engineer 2, day 6, (4.17)
C.11:
$b_{27} - e_{26} + 12 + 12 \cdot (1 - y_{27}) > 0$ for engineer 2, day 6, (4.18)
C.12:
$\sum_{j=1}^{7} y_{2j} + wd^{-}{}_{2} - wd^{+}{}_{2} = 5$ for engineer 2, (4.19)

C.13:
$e_{26} - b_{26} + ot^{-}{}_{2} - ot^{+}{}_{2} = 9 \cdot y_{26}$ for engineer 2, day 6, (4.20)

4.3.3 Computational Results and Discussions– IP Model

The integer goal programming approach was able to obtain the optimal values of the decision variables for all the goals. Table 4.14 summarizes the optimal objective values, iterations, solution times, number of constraints, number of decision variables and number of integer variables for at each priority level. I use the search strategies embedded in the mixed-integer-programming module of CPLEX shown in Figure 4.1, which could have a great effect on the solution time. The optimal solutions have been calculated in acceptable times. The maximum solution times, 2:1:14 CPUs, as well as the maximum number of iterations 23,499,928 have been found for priority 2 optimization.

Table 4.14 Model Statistics Summary

	Priority 1 Optimization	Priority 2 Optimization	Priority 3 Optimization
Objective value	0	0.5	22.1140
Iterations	511	23,499,928	98,351
Integer nodes	0	609,642	2,337
CPU Solution time	0.36s	2: 1:14	20.06s
Result code	101	101	102
Constraints	1402	1403	1404
Variables	374	374	374
Integers	234	234	234
Nonzeros	2955	2960	3030
Density	0.6%	0.6%	0.6%

Accordingly we could change the default values of these strategies and evaluate the different results to reduce the solution time. By default, the branching direction and the variable selection for the search tree are determined automatically and the best-bound search is applied for node selection.

Another combination of search strategies is also tried – minimum infeasibility for variable selection, branch up for branch direction, and best estimate for node selection. However, optimal solutions could not be found within solution time 6:26:26 in iteration 89449000. Other trial combinations include strong branching (variable selection), branch down (branch direction), best estimate (node selection), adopted at solution time 2:28:11 with 2864532 iterations and maximum infeasibility (variable selection), branch up (branch direction), best estimate (node selection) at solution time 2:16:18 with 21427424 iterations. For this reason, I applied the default search strategies, namely automatically determining branch direction and variable selection for the search tree, and best-bound search for node selection.

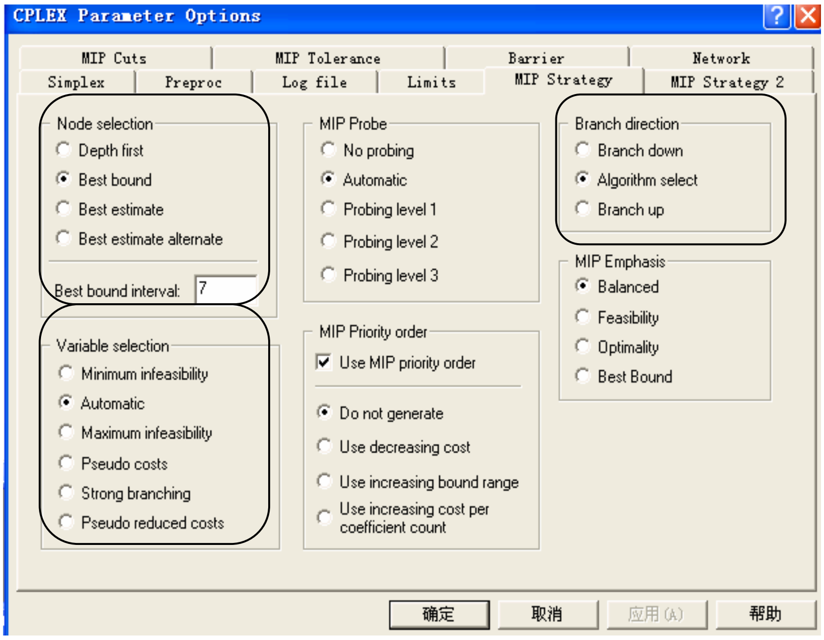


Figure 4.1 Default MIP Search Strategy within CPLEX Parameter Options Setting

Optimal solutions obtained using default search strategies in Figure 4.1, are then summarized in Table 4.14 by specifying the allocated engineer for each task, whose assignment decision variable x_{ijk} is 1, and in Table 4.15 by indicating the optimal value of objective functions of different priorities.

Table 4.15 Task assignment solution summary for the Model P1

	Day 1	*Day 2*	*Day 3*	*Day 4*	*Day 5*	*Day 6*	*Day 7*
Task 1	E2	E5	E5	E3	E5	E1	E2
Task 2	E1	E1	E3	E2	E1	E2	E1
Task 3	E5	E5	E5	E3	E5	E1	E3
Task 4	E1	E4	E4	E2	E1	E2	E2
Task 5	E5	E1	E3	E3	E3	E4	E3
Task 6	E4	E2	E5	E2	E5	E3	E5
Task 7	E5	E4	E3	E3	E1	E4	E3
Task 8	E4	E2	E4	E4	E3	E3	E5

Given two of the decision variables in model P1 are shift beginning time (b_{ij}) and ending time (e_{ij}) for each one of the five engineers, their corresponding solutions are also summarized in Table 4.16 – 4.22. Individual tour for each engineer is afterwards constructed in Table 4.23 – 4.27.

Table 4.16 Shift Allocation of Engineers in Day 1

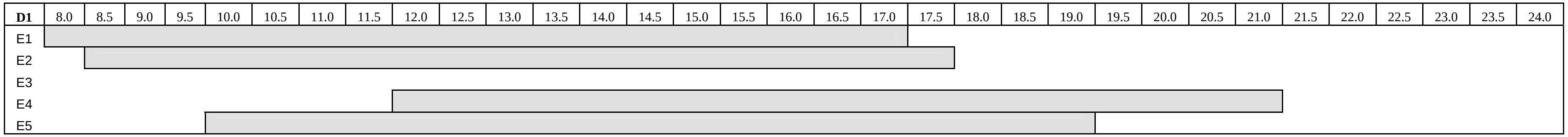


Table 4.17 Shift Allocation of Engineers in Day 2

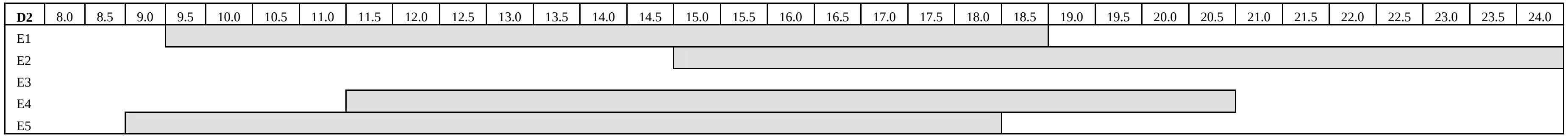


Table 4.18 Shift Allocation of Engineers in Day 3

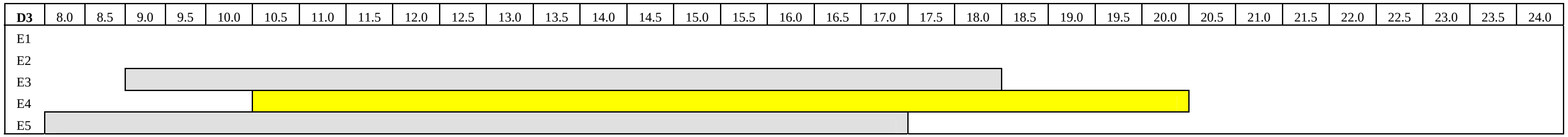


Table 4.19 Shift Allocation of Engineers in Day 4

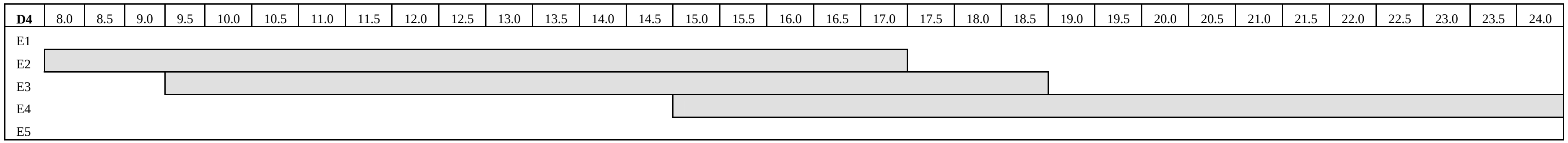


Table 4.20 Shift Allocation of Engineers in Day 5

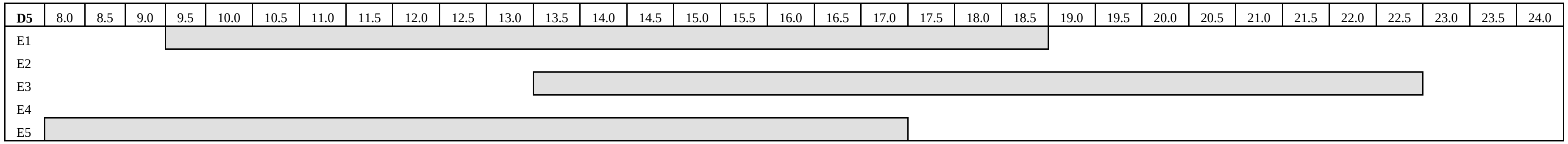


Table 4.21 Shift Allocation of Engineers in Day 6

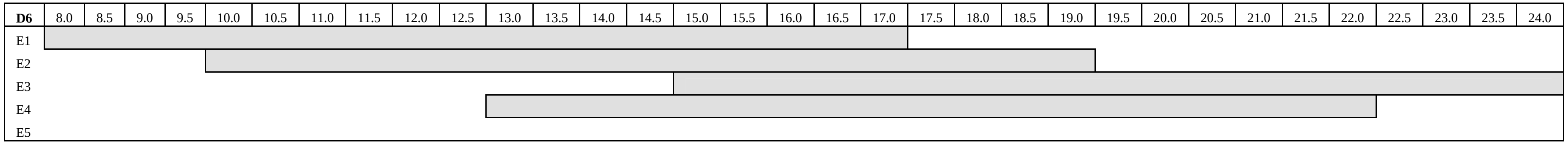


Table 4.22 Shift Allocation of Engineers in Day 7

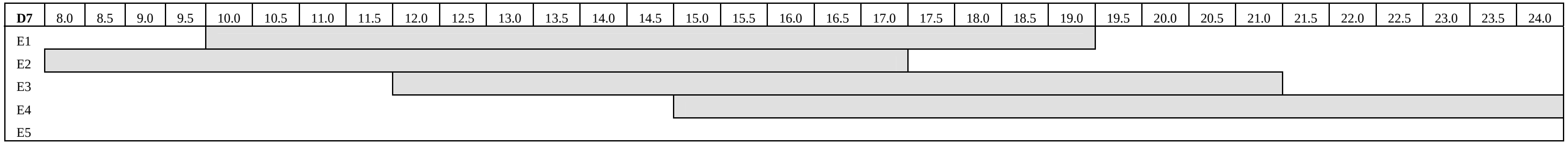


Table 4.23 Individual Tour Schedule for Engineer 1

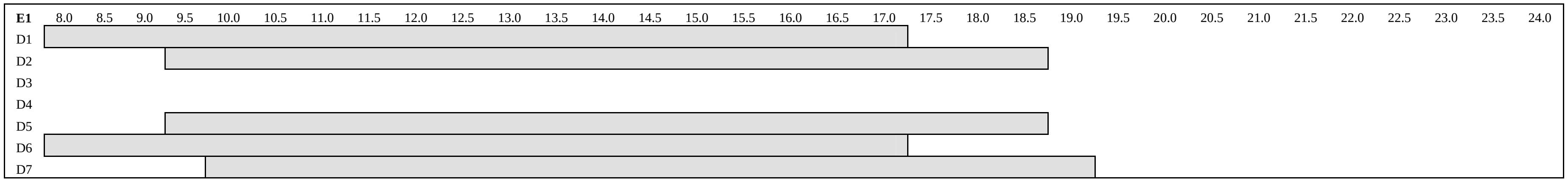


Table 4.24 Individual Tour Schedule for Engineer 2

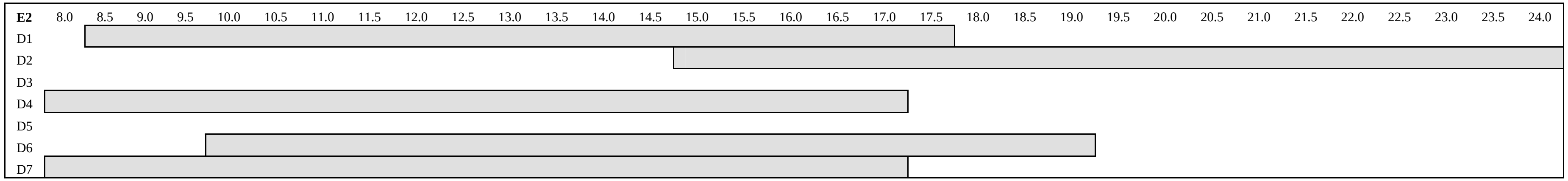


Table 4.25 Individual Tour Schedule for Engineer 3

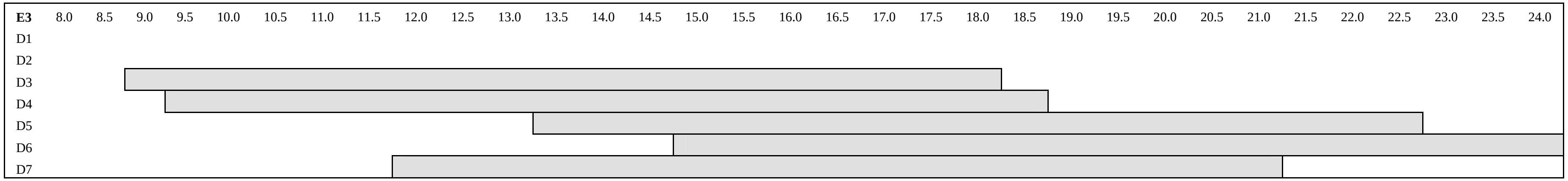


Table 4.26 Individual Tour Schedule for Engineer 4

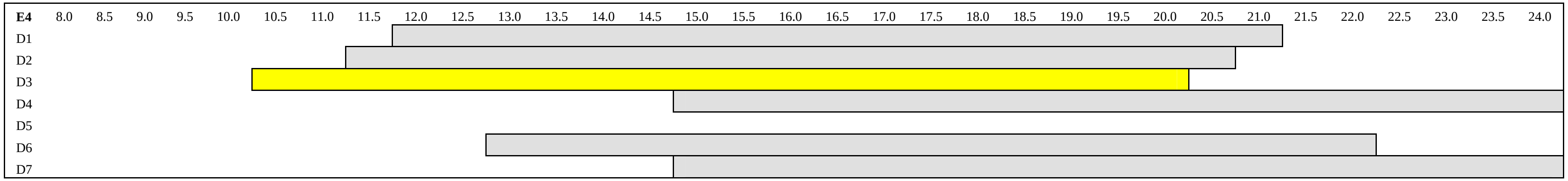
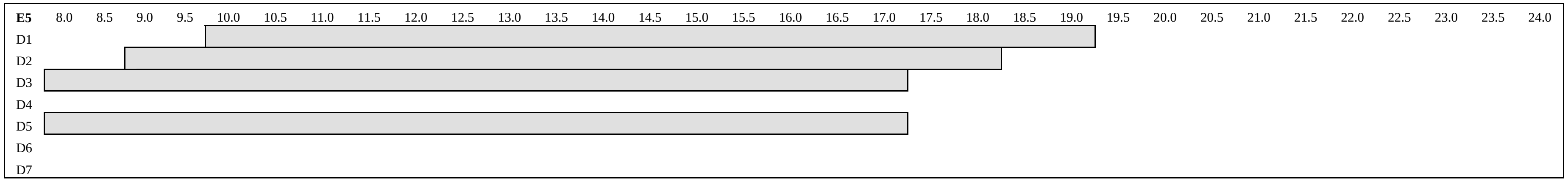


Table 4.27 Individual Tour schedule for Engineer 5



CHAPTER 5

CONCLUSIONS

In light of the eminent needs to model the hard operational manpower planning problem in a systematic as well as comprehensive way for the aircraft maintenance industry, this research proposed an optimization approach to improve the current manual maintenance scheduling process.

Since rostering and assignment are regarded as two different categories of problems studied in the literature, different mathematical techniques should be adopted to cater for their own problem characteristics. Thus my proposed approach is realized in a sequential way, with the result of stage-1 calculation entered into stage-2 mathematical model. In the first stage, for each maintenance task with predetermined starting time and finishing time, Fuzzy Analytic Hierarchy Process – one of the Multiple Attribute Decision Making methods, is applied to analyze each assignment candidate's utility in terms of qualifications, experience considerations, human factors and training requirements of the organization. During problem survey, these are regarded as essential concerns when experts make daily task assignment. Major scheduling and rostering constraints are modeled in an Integer Goal Programming model in the second stage of the proposed methodology. Apart from hard constraints that can not be violated at any circumstances, three objectives are figured out for optimization in three priorities, which are minimization of target days-off violation, minimization of target shift length violation (overtime), and lastly maximization of the total utilities for all feasible task assignments.

The first stage – FMADM computation is carried out in Excel spreadsheet. The second stage the Integer Goal Programming model is built in a modeling language MPL and solved through powerful package – CPLEX. The model is then validated to be capable of achieving the set objectives and demonstrating the possibility of achieving acceptable savings using illustrative example. The scheduling and assignments results are considered satisfactory when they are evaluated by the industry scheduling experts.

In my study, one major contribution to personnel scheduling is to provide a novel integer formation for a combined scheduling and assignment consideration, which has little been tackled in the literature. Compared to the manual process, this systematic approach captures major manual decision making process and assist in more global optimization of organizational objectives, which could not be handled by manual intuitive scheduling process.

Limitations of the model include that the model is still a simplified version of the real situation, leaving some constraints unconsidered for the sake of large computational efforts. Yet this small-size simplified model needs more than two hour's computation time.

In terms of further research, there are several potential areas for investigation. One focus is on improving, refining and extending the model for longer term planning. A monthly trial period is suggested to cover more practical problems. Work should be also be done to integrate the proposed two-phase approach into the maintenance decision support system in order to ensure the effective planning and scheduling of maintenance manpower.

REFERENCES

Abboud, N., M. Inuiguchi, M. Sakawa, Y. Uemura (1998), Manpower Allocation Using Genetic Annealing, *European Journal of Operational Research*, 111, pp. 405 – 420.

Aggarwal, S. (1982), A Focused Review of Scheduling in Services, *European Journal of Operational Research*, 9(2), pp. 114 – 121.

Alfares, H. (1999), Aircraft Maintenance Workforce Scheduling: A Case Study, *Journal of Quality in Maintenance Engineering*, 5(2), pp. 78–88.

Ali, H. H. and H. El-Rewini (1992), A Graph Theoretic Approach for Task Allocation, *Proceedings of the 25th Hawaii International Conference on System Sciences*, Kauai. Hawaii, 11, pp. 577 – 684.

Andersson, E., E. Housos, N. Kohl, D. Wedelin (1997), Crew Pairing Optimization, in: G. Yu (Ed.), *OR in Airline Industry*, Kluwer Academic Publishers, Boston, pp. 1 – 31.

Aouni, Belaid and Ossama Kettani (2001), Goal Programming Model: A Glorious History and a Promising Future, *European Journal of Operational Research*, 135, pp. 225 – 231.

Bailey, J. (1985), Integrated Days Off and Shift Personnel Scheduling, *Computers and Industrial Engineering,* 9(4), pp. 395 – 404.

Baker, E. K., L. D. Bodin, M. Fisher (1985), The Development of A Heuristic Set Covering Based System for Aircrew Scheduling, *Transportation Policy Decision Making,* 3, pp. 95 – 110.

Baker, K. (1976), Workforce Allocation in Cyclical Scheduling Problems: A Survey, *Operations Research Quarterly,* 27(1), pp. 155–167.

Baker, K. R. (1974), *Introduction to Sequencing and Scheduling,* Wiley and Sons, New York.

Bazaraa, M., J. Jarvis and H. Sherali (1990), *Linear Programming and Network Flows*, Wiley, New York.

Beale, E. M. L. (1988), *Introduction to Optimization,* Wiley, New York.

Bechtold, S. E. and Jacobs, L. W. (1990), Implicit Modeling of Flexible Break Assignments in Optimal Shift Scheduling, *Management Science,* Vol. 36, No. 11, pp. 1339 – 1351.

Bechtold, S. E., Michael J. Brusco, Michael J. Showalter (1991), A Comparative Evaluation of Labor Tour Scheduling Methods, *Decision Sciences,* 22, pp. 683 – 699.

Bertholdi III, J. J., J. B. Orlin, H. D. Ratliff (1980), Cyclic Scheduling via Integer Programs with Circular Ones, *Operations Research,* Vol. 28, No. 5, pp. 1074 – 1085.

Buckley, J. J. (1985), Fuzzy Hierarchical Analysis, *Fuzzy Sets and Systems,* Vol. 17, pp. 233 – 247.

Brusco, M. J., L. W. Jacobs (1998), Personnel Tour Scheduling When Starting-Time Restrictions Are Present, *Management Science*, Vol. 44, No. 4, pp. 534 – 547.

Brusco, M. J., L. W. Jacobs, R. J. Bongiorno, D. V. Lyons and B. Tang (1995), Improving Personnel Scheduling at Airline Stations, *Operations Research*, Vol. 43, No. 5, pp. 741 – 751.

Buchanan, B.G. and E.H. Shortliffe (1984), *Rule-based Expert Systems*, Reading, MA: Addison-Wesley.

Burkard, R. E. (2002), Selected Topics on Assignment Problems, *Discrete Applied Mathematics,* 123, pp. 257 – 302.

Burns, R. N. and G. J. Koop (1987), A Modular Approach to Optimal Multiple-Shift Manpower Scheduling, *Operations Research*, Vol. 35, No. 1, pp. 100 – 110.

Caron, G., P. Hansen, B. Jaumard (1999), The Assignment Problem with Seniority and Job Priority Constraints, *Operations Research,* Vol. 47, No. 3, pp. 449 – 453.

Cattrysse, D. G. and L. N. V. Wassenhove (1992), A Survey of Algorithms for The Generalized Assignment Problem, *European Journal of Operational Research,* 60, pp. 260 – 272.

Chang, S. C. (2002), A New Aircrew-scheduling Model for Short-haul Routes, *Journal of Air Transport Management,* 8, pp. 249 – 260.

Charnes A. and Cooper W. (1961), *Management Models and Industrial Applications of Linear Programming*, Wiley, New York.

Chu, H. D., E. Gelman, E. L. Johnson (1997), Solving Large Scale Crew Scheduling Problems, *European Journal of Operational Research,* 97, pp. 260 – 268.

Constantopoulos, P. (1989), Decision Support for Massive Personnel Assignment, *Decision Support System,* 5, pp. 355 – 363.

Danzig, G. (1949), Maximization of a Linear Form Whose Variables are Subject to a System of Linear Inequalities, Headquarters, USAAF.

Danzig, G. (1954), A Comment on Edie's Traffic Delay at Toll Booths, *Operations Research*, 2, pp. 339 – 341.

Day, P. and D. Ryan (1997), Flight Attendant Rostering for Short-haul Airline Operations, *Operations Research*, 45 (5), pp. 649 – 661.

Desaulniers, D., J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M. M. Solomon, F. Soumis (1997), Crew Pairing at Air France, *European Journal of Operational Research*, 97, pp. 245 – 259.

Dhall, S. K. and C. L. Liu, On a Real-time Scheduling Problem, *Operations Research*, Vol. 26, No. 1, pp. 127 – 140.

Dijkstra, M. C., L. G. Kroon, Nunen, J. and M. Saromon (1991), A DSS for Capacity Planning of Aircraft Maintenance Personnel, *International Journal of Production Economics*, 23, pp. 69 – 78.

Ding H., A. Lim, B. Rodrigues and Y. Zhu (2005), The Over-constrained Airport Gate Assignment Problem, *Computers and Operations Research*, 32, pp. 1867 – 1880.

Dondeti, V. R. and H. Emmons (1992), Fixed Job Scheduling with Two Types of Processors, *Operations Research*, Vol. 40, Supp. No. 1, pp. 76 – 85.

Dowling, D., M. Krishnamoorthy, H. Machenzie, and D. Sier (1997), Staff Rostering At A Large International Airport, *Annals of Operations Research*, 72, pp. 125 – 147.

Duffuaa S.O. and Al-Sultan K.S. (1999), A Stochastic Programming Model for Scheduling Maintenance Personnel, *Applied Mathematical Modeling*, Vol. 23, No. 5, pp. 385 – 397 (13).

Eliiyi, D. T. and M. Azizoğlu (2004), The Fixed Job Scheduling Problem: A Literature Review, *Yőneylem Araştirmasi/Endűstri Műhendisliği*, Gaziantep – Adana.

Ernst, A.T., H. Jiang, M. Krishnamoorthy, and D. Sier (2004), Staff Scheduling and Rostering: A Review of Applications, Methods and Models, *European Journal of Operational Research*, Vol. 153, Issue 1, pp. 3 – 27.

Fischetti, M., S. Martello and Paolo Toth[1] (1987), The Fixed Job Schedule Problem with Spread-Time Constraints, *Operations Research*, Vol. 35, No. 6, pp. 849 – 858.

Fischetti, M., S. Martello and Paolo Toth[2] (1989), The Fixed Job Schedule Problem with Working-Time Constraints, *Operations Research*, Vol. 37, No. 3, pp. 359 – 403.

Frances, M.A. (1966), Implementing a Program of Cyclical Scheduling of Nursing Personnel, *Hospitals, J.A.H.A.* 40, pp. 108-125.

French, S. (1982), *Sequencing and Scheduling*, Halsted Press.

Franz, L. S. and J. L. Miller (1993), Scheduling Medical Residents to Rotations: Solving the Large-Scale Multiperiod Staff Assignment Problem, *Operations Research*, Vol. 41, No. 2, pp. 269 – 279.

Friend, C.H. (1992), *Aircraft Maintenance Management*, Longman, England.

Gaballa, A. and Pearce W. (1979), Telephone Sales Manpower Planning at Qantas, *Interfaces*, Vol. 9, No. 3, pp. 1 – 9.

Gabrel, V. (1995), Scheduling Jobs within Time Windows on Identical Parallel Machines: New Model and Algorithms, *European Journal of Operational Research*, 83, pp. 320 – 329.

Glover, F. and C. McMillan (1986), The General Employee Scheduling Problem: An Integration of MS and AI, *Computers and Operations Research*, Vol. 13, No. 5, pp. 563 – 573.

Glover, F., D. Karney, and D. Klingman (1977), A Policy Evaluation Model and Prototype Computer-Assisted Policy Evaluation System for Navy Personnel Management, *Technical Report*, Navy Personnel Research and Development Center, San Diego.

Hannan, Edward L. (1985), An Assessment of Some Criticisms of Goal Programming, *Computers and Operations Research*, Vol. 12, No. 6, pp. 525 – 541.

Harper, P. M., V. D. Senna, I. T. Vieira, A. K. Shahani, A Genetic Algorithm for The Project Assignment Problem, *Computers and Operations Research,* Vol. 32, Iss. 5.

Hoffman. K. L. and M. Padberg (1993), Solving Airline Crew Scheduling Problems by Branch-and-Cut, *Management Science,* Vol. 39, No.6, pp. 657 – 682.

Howell, J.P. (1966), Cyclical Scheduling of Nursing Personnel, *Hospitals, J.A.H.A.*, 40, pp 77-85.

Hong Kong Airport Authority (2002), *Aircraft Maintenance,* Available from: http://www.hkairport.com/eng/afacilities/airmaintenance.html.

Ignizio, J.P. (1976), *Goal Programming and Extensions,* Lexington Books, Lexington, Massachusetts.

Ignizio, J.P. (1985), *Introduction to Linear Goal Programming*, Sage Publications, USA.

ILOG CPLEX Optimization Engine, 9.0 for Windows NT/2000, http://www.ilog.com/products/cplex/, ILOG, USA.

Jacobs, L.W. and Bechtold, S.E. (1993), Labor Utilization Effects of Workforce Scheduling Flexibility Alternatives in A Tour Scheduling Environment, *Decision Science*, Vol. 24, No. 1, pp. 148 – 166.

Jaime, G. A. (1998), *The Interactive Management of Human Resources in Uncertainty,* Kluwer Academic Publishers, The Netherlands.

Keen, P. G. W. and M. S. S. Morton (1978), *Decision Support Systems: An Organizational Perspective,* Addison – Wesley.

Kim, Gyu C. and John Emery (2000), An Application of Zero-One Goal Programming in Project Selection and Resource Planning – A Case Study from the Woodward Governor Company, *Computers and Operations Research*, 27, pp. 1389 – 1408.

Klingman, D. and N. V. Philips (1984), Topological and Computational Aspects of Preemptive Multicriteria Military Personnel Assignment Problems, *Management Science,* Vol. 30, No. 11, pp. 1362 – 1375.

Klir, George J., Clair, Ute H. ST. and Yuan Bo (1997), *Fuzzy Set Theory Foundations and Applications*, USA, Prentice Hall.

Kolen A. W. J. and L. G. Kroon (1994), An Analysis of Shift Class Design Problems, *European Journal of Operational Research*, 79, pp. 417 – 430.

Kroon, L. G. (1990), “Job Scheduling and Capacity Planning in Aircraft Maintenance”, PhD Thesis, Rotterdam School of Management, Erasmus University, The Netherlands.

Kroon, L. G., M. Salomon, and L. N. V. Wassenhove (1995), Exact and Approximation Algorithms for the Operational Fixed Interval Scheduling Problem, *European Journal of Operational Research*, 82, pp. 190 – 205.

Kuhn, H. W. (1955), The Hungarian Method for The Assignment Problem, *Naval Research Logistics Quarterly,* 2, pp. 83 – 97.

Laarhoven, P.J.M., and W. Pedrycz (1983), A Fuzzy Extension of Saaty’s Priority Theory, *Fuzzy Sets and Systems,* Vol. 11, pp. 229 – 241.

Labib, A.W., R.F. O’Connor and G.B. Williams (1998), Effective Maintenance System Using the Analytic Hierarchy Process, *Integrated Manufacturing System*, 9(2), pp. 87 – 98.

Langerman, J. J., E. M. Ehlers (1997), Agent-based Airline Scheduling, *Computers and Industrial Engineering,* 33, pp. 849 – 852.

Lau, H. C. (1996), On the Complexity of Manpower Shift Scheduling, *Computers and Operations Research*, Vol. 23, No. 1, pp. 93 – 102.

Lee S. M. (1972), *Goal Programming for Decision Analysis,* Auerbach Publishers, Philadelphia.

Leung, K.S. and W. Lam (1988), Fuzzy Concepts in Expert Systems, *Computer*, 21(9), pp. 43 – 56.

Lee, W.B, Lau, H. Liu, Z.Z & Tam, S., (2001), A Fuzzy Analytic Hierarchy Process Approach in Modular Product Design, *Expert System*, Vol. 18, No. 1, pp. 32 – 41.

Liu, C. L. and J. W. Layland (1973), Scheduling Algorithms for Multiprogramming in A Hard – real – time Environment, *Journal ACM,* Vol. 20, No. 1, pp. 46 – 61.

Liang, T. T. and B. B. Buclatin (1988), Improving the Utilization of Training Resources through Optimal Personnel Assignment in the U.S. Navy, *European Journal of Operational Research*, 33, pp. 183 – 190.

Liang, T. T., and S. J. Lee (1985), A Systems Approach to Integrate Manpower Planning and Operation, *Socio-Economic Planning Sciences*, Vol. 19, No. 6, pp. 371 – 377.

Liang, T. T., and T. J. Thompson (1987), A Large-scale Personnel Assignment Model for the Navy, *Decision Sciences*, Vol. 18, No. 2, pp. 235 – 250.

Lučić, P. and D. Teodorović (1999), Simulated Annealing for the Multi-objective Aircrew Rostering Problem, *Transportation Research A,* 33, PP. 19 – 45.

Ma. P. R., E. Y. S. Lee and M. Tsuchiya (1982), A Task Allocation Model for Distributed Computing Systems, *IEEE Transactions and Computers*, C-31(1), pp. 41 – 47.

Marsten, R. E., F. Shepardson (1981), Exact Solution of Crew Scheduling Problems Using Set Partitioning Model: Recent Successful Applications, *Networks*, 11, pp. 165 – 177.

Martin, C., D. Jones and P. Keskinocak (2003), Optimizing On-Demand Aircraft Schedules for Fractional Aircraft Operators, *Interfaces*, Vol. 33, No. 5, pp. 22 – 35.

Mason A.J., D. M. Ryan, and D. M. Panton (1998), Integrated Simulation, Heuristic and Optimization Approaches to Staff Scheduling, *Operations Research*, Vol. 46, No. 2, pp. 161 – 175.

McClean, Sally (1991), Manpower Planning Models and Their Estimation, *European Journal of Operational Research*, 51, pp. 179 – 187.

Millet, I., and Harker, P.T., (1990), Globally Effective Questioning in the Analytic Hierarchy Process, *European Journal of Operational Research*, 48, pp. 88 – 98.

Morris, J. G., and M. J. Showalter (1983), Simple Approaches to Shift, Days-off And Tour Scheduling Problems, *Management Science*, Vol. 29, No. 8, pp. 942 – 950.

Müller-Merbach, H. (1975), Modeling Techniques and Heuristics for Combinatorial Problems, *Combinatorial Programming: Methods and Applications*, B. Roy, ed., D. Reidel, Dordrecht, pp. 3 – 27.

Musliu N., Gärtner J. and Slany W. (2002), Efficient generation of rotating workforce schedules, *Discrete Applied Mathematics*, 118 (1-2), pp. 85 – 98.

MPL Modeling System (2002), Maximal Software, Release 4.2, USA.

Orchard, A. (1994), *FuzzyCLIPS Version 6.02A User's Guide*, National Research Council of Canada.

Paz, N. M. and Leigh, W. (1994), Maintenance Scheduling: Issues, Results and Research Needs, *International Journal of Operations & Production Management*, Vol. 14, No. 8, pp. 47 – 69.

Peng, D. T., K. G. Shin and T. E. Abdelzaher, Assignment and Scheduling of Communicating Periodic Tasks in Distributed Real-time Systems, *IEEE Transactions on Software Engineering*, Vol. 23, No. 12, pp. 745 – 758.

Pinedo, M. (1995), *Scheduling: Theory, Algorithms and Systems,* Prentice Hall.

Pinedo, Michael and Xiaoli Chao (1999), *Operations Scheduling – with Applications in Manufacturing and Services,* McGraw-Hill, Singapore.

Rezqallah, H. Ramadhan, Hamad, I. Al-abdul Wahhab and Salih, O. Duffuaa (1999), The Use of An Analytical Hierarchy Process in Pavement Maintenance Priority Ranking, *Journal of Quality in Maintenance Engineering,* Vol. 5, No. 1, pp. 25 – 39.

Ryan, D.M. (1992), The Solution of Massive Generalized Set Partitioning Problems in Aircrew Rostering, *Journal of Operational Research Society,* Vol. 43, No.5, pp. 459 – 467.

Sancho, S. S., Y. Xu, and X. Yao, (2006), Hybrid Meta-heuristics Algorithms for Task Assignment in Heterogeneous Computing Systems, *Computers and Operations Research,* 33, pp. 820 – 835.

Saaty, T.L. (1980), *The Analytic Hierarchy Process,* McGraw-Hill, New York.

Schindler, S. and T. Semmel (1993), Station staffing at Pan American World Airways, *Interfaces,* 23, pp. 91 – 98.

Sierksma, Gerard (2002), *Linear and Integer Programming: Theory and Practice,* 2nd, Pure and Applied Mathematics – A Series of Monographs and Textbooks, Marcel Dekker, Inc., USA.

Sinelair, J. B., Efficient Computation of Optimal Assignments for Distributed Tasks, *Journal of Parallel and Distributed Computing*, Vol. 4, No. 4, pp. 342 – 362.

Stojković, M., F. Soumis and J. Desrosiers (1998), The Operational Airline Crew Scheduling Problem, *Transportation Science,* Vol. 32, No.3, pp. 232 – 245.

Stone H. S. and S. H. Bokhari (1978), Control of Distributed Processes, *Computers,* Vol. 11, No. 7, pp. 97 – 106.

Tabucanon, M.T., D.N. Batanov and D.K. Verma (1994), Decision Support System for Multi-Criteria Machine Selection for Flexible Manufacturing Systems, *Computers in Industry,* Vol. 25, No. 2, pp. 131 – 143.

Tam, C.Y., Rao, V.M. (2001), An Application of the AHP in Vendor Selection of A Telecommunications System, *The International Journal of Management Science*, 29, pp. 171 – 182.

Tamiz, M., Dylan Jones and Carlos Romero (1998), Goal Programming for Decision Making: An Overview of the Current State-of-art, *European Journal of Operational Research,* Vol. 111, pp. 569 – 581.

Teodorović, D. and P. Lučić (1998), A Fuzzy Set Theory Approach to the Aircrew Rostering Problem, *Fuzzy Set and System,* 95, pp. 261 – 271.

Triantaphyllou, Evangelos (2000), *Multi-criteria Decision Making Methods: A Comparative Study*, Netherlands, Kluwer Academic Publishers.

Thompson, G. M. (1995), Improved Implicit Optimal Modeling of the Labor Shift Scheduling Problem, *Management Science*, Vol. 41, No. 4, pp. 595 – 607.

Thompson, M. (2001), "Maintenance error reduction, Aircraft Engineering and Aerospace Technology", *An International Journal*, 73(5).

Tien, J. and A. Kamiyama (1982), On Manpower Scheduling Algorithms, *SIAM Review*, 24 (3), pp. 275 – 287.

Triantaphyllou, E. (2000), *Multi-criteria Decision Making Methods: A Comparative Study*, Kluwer Academic Publishers, Netherlands.

Trippi, R. R., A. W. Ash, and J. V. Ravenis II (1974), A Mathematical Approach to Large Scale Personnel Assignment, *Computers and Operations Research*, Vol. 1, pp. 111 – 117.

Valdes, R. A., E. Crespo and J. M. Tamarit (1999), Labor Scheduling at an Airport Refueling Installation, *Journal of the Operational Research Society*, 50, pp. 211 – 218.

Valls, V., Perez, A., and Quintanilla, S. (1996), A Graph Coloring Model for Assigning A Heterogeneous Workforce to A Given Schedule, *European Journal of Operations Research*, 90, pp. 285 – 302.

Venkataraman, R. and MJ Brusco (1996), An Integrated Analysis of Nurse Staffing and Scheduling Policies, *Omega*, Vol. 24, No. 1, pp. 57 – 71.

Walker, Russel C. (1999), *Introduction to Mathematical Programming*, Prentice Hall, USA.

Williams, H. P. (1985), *Model Building in Mathematical Programming*, 2nd ed., John Wiley & Sons, England.

Williams, H. P. (1990), *Model Building in Mathematical Programming*, 3rd ed., John Wiley & Sons, England.

Williams, H. P. (1993), *Model Solving in Mathematical Programming*, John Wiley & Sons, England.

Wolfe, W. J. and Sorensen S. E. (2000), Three Scheduling Algorithm Applied to the Earth Observing Systems Domain, *Management Science*, 46, pp. 148 – 168.

Yan, S., J. C. Chang (2002), Airline Cockpit Crew Scheduling, *European Journal of Operational Research*, Vol. 136, pp. 501 – 511.

Yan, S., T. T. Tang, Y. P. Tu (2002), Optimal Construction of Airline Individual Crew Pairing, *Computers and Operations Research*, 29, pp. 341 – 363.

Yang, T. H., S. Y. Yan, H. H. Chen (2003), An Airline Maintenance Manpower Planning Model with Flexible Strategies, *Journal of Air Transport Management*, 9, PP. 233 – 239.

Zaheh, L.A. (1965), Fuzzy Sets, *Information and Control*, 8, pp. 338 – 353.

Zimmermann, H.J. (1996), *Fuzzy Set Theory and Its Applications*, 3rd ed, Boston, USA, Kluwer Academic Publishers.

Zolfaghari S., M. Y. Jaber and H. Hamoudi (2004), A Goal Programming Approach for A Multi-period Task Assignment Problem, *INFOR*, Vol. 42, No. 4, pp. 299 – 309.

APPENDIX A
MODEL FILE of PRIORITY 1 OPTIMIZATION

```
{pr1.mpl}

TITLE
     Pri1;

INDEX
     engineer   :=(1..5);
     day        :=(1..7);
     task       :=(1..8);

DATA
     Utility[engineer,day,task]  :=DATAFILE("utility.dat");
     TaskStart[day,task]         :=DATAFILE("taskstart.dat");
     TaskFinish[day,task]        :=DATAFILE("taskfinish.dat");


BINARY VARIABLES
     Assignment[engineer,day,task]
          WHERE (Utility[engineer,day,task]>0);
     Offday[engineer,day]           ;


INTEGER VARIABLES
     UWD[engineer]    ;
     OWD[engineer]    ;


DECISION VARIABLES
     UOT[engineer,day];
     OOT[engineer,day];
     ShiftBegin[engineer,day]       ;
     ShiftEnd[engineer,day]         ;



MACROS
     TotalUtility := SUM(engineer,day,task: Utility*Assignment);
     TotalDayson  := SUM(engineer: OWD);
     TotalOT      := SUM(engineer,day:UOT+OOT);


MODEL
     MIN TotalDayson;


SUBJECT TO

     AssignmentConstraint[day,task] ->ACon:
          SUM(engineer:Assignment)=1;

     OverlapConstraint1[engineer,day,task] ->OCon1:
          Assignment+Assignment[task-1]<=1;
```

```
    OverlapConstraint2[engineer,day,task] ->OCon1:
         Assignment+Assignment[task+1]<=1;

    ShiftBeginCon[engineer,day,task] :
         ShiftBegin<=Assignment*TaskStart+24*(1-Assignment)-0.5;

    ShiftEndCon[engineer,day,task] -> SEnd:
         ShiftEnd+(1-Assignment)>=Assignment*TaskFinish+0.5;

    ShiftBeginCon2[engineer,day] :
          ShiftBegin>=8*Offday;

    ShiftEndCon2[engineer,day]  :
          ShiftEnd<=24*Offday;

    ShiftAlternative[engineer,day] ->SAlt:
          ShiftBegin[day+1] - ShiftEnd+12+12*(1-Offday[day+1])>0;

    ShiftLength1[engineer,day] -> SLen1:
          ShiftEnd - ShiftBegin >=6*Offday;

    ShiftLength2[engineer,day] -> SLen2:
          ShiftEnd - ShiftBegin <=12*Offday;

    ShiftDefinition1[engineer,day] -> SDef1:
          SUM(task: Assignment) - 9*Offday <=0;

    ShiftDefinition2[engineer,day] -> SDef2:
          SUM(task: Assignment) - 0.1*Offday >=0;

    OffdayConstraint[engineer] -> OfCon:
          SUM(day:Offday)+ UWD - OWD =5;

    ShiftLengPenalty[engineer,day] -> SPen:
          ShiftEnd - ShiftBegin + UOT - OOT =9*Offday;


END
```

APPENDIX B
MODEL FILE of PRIORITY 2 OPTIMIZATION

```
{pr2.mpl}

TITLE
     Pri2;

INDEX
     engineer   :=(1..5);
     day        :=(1..7);
     task       :=(1..8);

DATA
     Utility[engineer,day,task]  :=DATAFILE("utility.dat");
     TaskStart[day,task]         :=DATAFILE("taskstart.dat");
     TaskFinish[day,task]        :=DATAFILE("taskfinish.dat");


BINARY VARIABLES
     Assignment[engineer,day,task]
          WHERE (Utility[engineer,day,task]>0);
     Offday[engineer,day]           ;


INTEGER VARIABLES
     UWD[engineer]    ;
     OWD[engineer]    ;


DECISION VARIABLES
     UOT[engineer,day];
     OOT[engineer,day];
     ShiftBegin[engineer,day]       ;
     ShiftEnd[engineer,day]         ;



MACROS
     TotalUtility := SUM(engineer,day,task: Utility*Assignment);
     TotalDayson  := SUM(engineer: OWD);
     TotalOT      := SUM(engineer,day:UOT+OOT);


MODEL
     MIN TotalOT;


SUBJECT TO
     TotalDayson=0;

     AssignmentConstraint[day,task] ->ACon:
          SUM(engineer:Assignment)=1;

     OverlapConstraint1[engineer,day,task] ->OCon1:
```

```
          Assignment+Assignment[task-1]<=1;

     OverlapConstraint2[engineer,day,task] ->OCon1:
          Assignment+Assignment[task+1]<=1;

     ShiftBeginCon[engineer,day,task] :
          ShiftBegin<=Assignment*TaskStart+24*(1-Assignment)-0.5;

     ShiftEndCon[engineer,day,task] -> SEnd:
          ShiftEnd+(1-Assignment)>=Assignment*TaskFinish+0.5;

     ShiftBeginCon2[engineer,day] :
           ShiftBegin>=8*Offday;

     ShiftEndCon2[engineer,day]  :
           ShiftEnd<=24*Offday;

     ShiftAlternative[engineer,day] ->SAlt:
           ShiftBegin[day+1] - ShiftEnd+12+12*(1-Offday[day+1])>0;

     ShiftLength1[engineer,day] -> SLen1:
           ShiftEnd - ShiftBegin >=6*Offday;

     ShiftLength2[engineer,day] -> SLen2:
           ShiftEnd - ShiftBegin <=12*Offday;

     ShiftDefinition1[engineer,day] -> SDef1:
           SUM(task: Assignment) - 9*Offday <=0;

     ShiftDefinition2[engineer,day] -> SDef2:
           SUM(task: Assignment) - 0.1*Offday >=0;

     OffdayConstraint[engineer] -> OfCon:
           SUM(day:Offday)+ UWD - OWD =5;

     ShiftLengPenalty[engineer,day] -> SPen:
           ShiftEnd - ShiftBegin + UOT - OOT =9*Offday;


END
```

APPENDIX C
MODEL FILE of PRIORITY 3 OPTIMIZATION

```
{pr3.mpl}

TITLE
     Pri1;

INDEX
     engineer   :=(1..5);
     day        :=(1..7);
     task       :=(1..8);

DATA
     Utility[engineer,day,task]  :=DATAFILE("utility.dat");
     TaskStart[day,task]         :=DATAFILE("taskstart.dat");
     TaskFinish[day,task]        :=DATAFILE("taskfinish.dat");


BINARY VARIABLES
     Assignment[engineer,day,task]
          WHERE (Utility[engineer,day,task]>0);
     Offday[engineer,day]           ;


INTEGER VARIABLES
     UWD[engineer]    ;
     OWD[engineer]    ;


DECISION VARIABLES
     UOT[engineer,day];
     OOT[engineer,day];
     ShiftBegin[engineer,day]       ;
     ShiftEnd[engineer,day]         ;



MACROS
     TotalUtility := SUM(engineer,day,task: Utility*Assignment);
     TotalDayson  := SUM(engineer: OWD);
     TotalOT      := SUM(engineer,day:UOT+OOT);


MODEL
     MAX TotalUtility;


SUBJECT TO
     TotalDayson=0;
     TotalOT=0.5;

     AssignmentConstraint[day,task] ->ACon:
          SUM(engineer:Assignment)=1;
```

```
     OverlapConstraint1[engineer,day,task] ->OCon1:
          Assignment+Assignment[task-1]<=1;

     OverlapConstraint2[engineer,day,task] ->OCon1:
          Assignment+Assignment[task+1]<=1;

     ShiftBeginCon[engineer,day,task] :
          ShiftBegin<=Assignment*TaskStart+24*(1-Assignment)-0.5;

     ShiftEndCon[engineer,day,task] -> SEnd:
          ShiftEnd+(1-Assignment)>=Assignment*TaskFinish+0.5;

     ShiftBeginCon2[engineer,day] :
           ShiftBegin>=8*Offday;

     ShiftEndCon2[engineer,day]  :
           ShiftEnd<=24*Offday;

     ShiftAlternative[engineer,day] ->SAlt:
           ShiftBegin[day+1] - ShiftEnd+12+12*(1-Offday[day+1])>0;

     ShiftLength1[engineer,day] -> SLen1:
           ShiftEnd - ShiftBegin >=6*Offday;

     ShiftLength2[engineer,day] -> SLen2:
           ShiftEnd - ShiftBegin <=12*Offday;

     ShiftDefinition1[engineer,day] -> SDef1:
           SUM(task: Assignment) - 9*Offday <=0;

     ShiftDefinition2[engineer,day] -> SDef2:
           SUM(task: Assignment) - 0.1*Offday >=0;

     OffdayConstraint[engineer] -> OfCon:
           SUM(day:Offday)+ UWD - OWD =5;

     ShiftLengPenalty[engineer,day] -> SPen:
           ShiftEnd - ShiftBegin + UOT - OOT =9*Offday;


END
```

APPENDIX D
DATA FILE – utility.dat

```
!utility.dat
!utility[engineer,day,task]:
!
0       0.4279  0       0.4141  0.4096  0.3346  0.3209  0
0       0.4141  0       0.3346  0.4141  0.3346  0.3209  0
0.4279  0       0       0.3346  0.4096  0.3346  0.3163  0
0.3723  0.3209  0       0       0.4096  0.3346  0.3209  0.4279
0       0.3346  0.4096  0.3209  0.4096  0.3346  0.3209  0
0.3719  0.3163  0.4096  0       0       0.3209  0.3209  0
0       0.4096  0.3163  0       0.4096  0.3346  0.3209  0
0.3156  0       0.2917  0       0       0.2597  0.232   0.2928
0.3156  0       0.2928  0.2597  0       0.2597  0.2415  0.2928
0       0.2928  0.2928  0.2597  0       0.2597  0.237   0.2974
0.237   0.2415  0.2934  0.3156  0       0.2597  0.2415  0
0.3156  0.2597  0       0.2415  0       0.2597  0.2415  0.2928
0.2411  0.237   0       0.3156  0.3156  0.2415  0.2415  0.3019
0.2411  0       0.237   0.2974  0       0.2597  0.2415  0.2928
0.4279  0       0       0.3346  0.4096  0.3346  0.3163  0
0       0.6786  0.6786  0       0.5545  0       0.4802  0.6786
0       0.6786  0.6786  0       0.6044  0       0.5029  0.6287
0.5301  0.4802  0.6786  0       0.6044  0       0.4802  0
0       0       0.6044  0.4802  0.6044  0       0.4802  0.6786
0       0.5301  0.6044  0       0       0.4802  0.4802  0.6287
0       0.6044  0.5301  0.6287  0.8638  0       0.4802  0.6786
0.3723  0.3209  0       0       0.4096  0.3346  0.3209  0.4279
0.2669  0.2855  0       0.2298  0.281   0.2298  0.387   0
0.2858  0       0       0.2298  0       0.2298  0       0.2624
0       0.2253  0       0.2669  0       0.2298  0.2253  0.2855
0.2669  0.2298  0       0.2253  0       0.2298  0.2253  0
0       0       0       0.2669  0.2669  0.2253  0.2253  0
0       0       0       0.2624  0       0.2298  0.2253  0
0       0.3346  0.4096  0.3209  0.4096  0.3346  0.3209  0
0.3421  0       0.3376  0.3795  0       0.3795  0       0.3376
0.2674  0.3376  0.3376  0.3795  0.2629  0.3795  0       0
0.3749  0       0.3376  0.3421  0.2629  0.3795  0       0.2674
0.3421  0.3795  0.2629  0       0.2629  0.3795  0       0.3376
0.2674  0.3749  0.2629  0.3421  0.3421  0       0       0.3376
0.3421  0.2629  0.3749  0       0.2629  0.3795  0       0.3376
```

APPENDIX E
DATA FILE – taskstart.dat

```
! taskstart.dat
! taskstart[day,task]


!         T1    T2      T3      T4      T5      T6      T7      T8
!
!day 1:
          9,    10,     10.5,   11,     13,     15,     17,     19.5
!day 2:
          9.5,  10,     10.5,   12,     14,     17.5,   19,     22,
!day 3:
          9,    9.5,    10.5,   11,     12.5,   14,     15,     17.5,
!day 4:
          10,   10.5,   11,     12,     12,     14.5,   16.5,   18,
!day 5:
          10,   11.5,   11.5,   13.5,   14,     15,     16,     19,
!day 6:
          9.5,  10.5,   12.5,   13.5,   15.5,   19,     20.5,   21.5,
!day 7:
          9,    10.5,   12.5,   14,     16.5,   19,     19,     21
```

APPENDIX F
DATA FILE – taskfinish.dat

```
!taskfinish.dat
!taskfinish[day,task]


!         T1    T2      T3      T4      T5      T6      T7      T8
!
!day 1:
          10.5, 11,     11.5,   12,     13.5,   17,     18,     20.5,
!day 2:
          10.5, 10.5,   11.5,   13.5,   14.5,   19.5,   20,     23.5,
!day 3:
          10,   10.5,   12.5,   12.5,   13.5,   15,     15.5,   19.5,
!day 4:
          11,   12,     12,     13.5,   14.5,   15.5,   18,     19.5,
!day 5:
          10.5, 13,     13,     14.5,   15.5,   16,     18,     21,
!day 6:
          10.5, 12.5,   14.5,   15.5,   17.5,   21.5,   21.5,   23,
!day 7:
          9.5,  11.5,   14.5,   15,     18,     19.5,   20.5,   22.5
```

APPENDIX G
SOLUTION FILE AFTER PRIORITY 3 OPTIMIZATION

```
MPL Modeling System   -   Copyright (c) 1988-2005, Maximal Software, Inc.
--------------------------------------------------------------------


MODEL STATISTICS

Problem name:       test8

Filename:           8.mpl
Date:               February 20, 2006
Time:               17:36
Parsing time:       0.05 sec

Solver:             CPLEX
Objective value:    22.1140000000
Iterations:         98351
Integer nodes:      2337
Solution time:      20.06 sec
Result code:        102

Constraints:        1404
Variables:          374
Integers:           234
Nonzeros:           3030
Density:            0.6 %


SOLUTION RESULT


  Optimal solution within tolerance found

    MAX Z        =         22.1140


MACROS


    Macro Name                              Values
  -----------------------------------------------
    TotalUtility                           22.1140
    TotalDayson                             0.0000
    TotalOT                                 0.5000
  -----------------------------------------------


DECISION VARIABLES


VARIABLE Assignment[engineer,day,task] :

    engineer  day  task          Activity     Reduced Cost
  ---------------------------------------------------------
         1    1     2           1.0000           0.4279
         1    1     4           1.0000           0.4141
         1    1     5           0.0000           0.4096
```

1	1	6	0.0000	0.3346
1	1	7	0.0000	0.3209
1	2	2	1.0000	0.4141
1	2	4	0.0000	0.3346
1	2	5	1.0000	0.4141
1	2	6	0.0000	0.3346
1	2	7	0.0000	0.3209
1	3	1	0.0000	0.4279
1	3	4	0.0000	0.3346
1	3	5	0.0000	0.4096
1	3	6	0.0000	0.3346
1	3	7	0.0000	0.3163
1	4	1	0.0000	0.3723
1	4	2	0.0000	0.3209
1	4	5	0.0000	0.4096
1	4	6	0.0000	0.3346
1	4	7	0.0000	0.3209
1	4	8	0.0000	0.4279
1	5	2	1.0000	0.3346
1	5	3	0.0000	0.4096
1	5	4	1.0000	0.3209
1	5	5	0.0000	0.4096
1	5	6	0.0000	0.3346
1	5	7	1.0000	0.3209
1	6	1	1.0000	0.3719
1	6	2	0.0000	0.3163
1	6	3	1.0000	0.4096
1	6	6	0.0000	0.3209
1	6	7	0.0000	0.3209
1	7	2	1.0000	0.4096
1	7	3	0.0000	0.3163
1	7	5	0.0000	0.4096
1	7	6	0.0000	0.3346
1	7	7	0.0000	0.3209
2	1	1	1.0000	0.3156
2	1	3	0.0000	0.2917
2	1	6	0.0000	0.2597
2	1	7	0.0000	0.2320
2	1	8	0.0000	0.2928
2	2	1	0.0000	0.3156
2	2	3	0.0000	0.2928
2	2	4	0.0000	0.2597
2	2	6	1.0000	0.2597
2	2	7	0.0000	0.2415
2	2	8	1.0000	0.2928
2	3	2	0.0000	0.2928
2	3	3	0.0000	0.2928
2	3	4	0.0000	0.2597
2	3	6	0.0000	0.2597
2	3	7	0.0000	0.2370
2	3	8	0.0000	0.2974
2	4	1	0.0000	0.2370
2	4	2	1.0000	0.2415
2	4	3	0.0000	0.2934
2	4	4	1.0000	0.3156
2	4	6	1.0000	0.2597
2	4	7	0.0000	0.2415
2	5	1	0.0000	0.3156
2	5	2	0.0000	0.2597
2	5	4	0.0000	0.2415
2	5	6	0.0000	0.2597
2	5	7	0.0000	0.2415
2	5	8	0.0000	0.2928

2	6	1	0.0000	0.2411
2	6	2	1.0000	0.2370
2	6	4	1.0000	0.3156
2	6	5	0.0000	0.3156
2	6	6	0.0000	0.2415
2	6	7	0.0000	0.2415
2	6	8	0.0000	0.3019
2	7	1	1.0000	0.2411
2	7	3	0.0000	0.2370
2	7	4	1.0000	0.2974
2	7	6	0.0000	0.2597
2	7	7	0.0000	0.2415
2	7	8	0.0000	0.2928
3	1	1	0.0000	0.4279
3	1	4	0.0000	0.3346
3	1	5	0.0000	0.4096
3	1	6	0.0000	0.3346
3	1	7	0.0000	0.3163
3	2	2	0.0000	0.6786
3	2	3	0.0000	0.6786
3	2	5	0.0000	0.5545
3	2	7	0.0000	0.4802
3	2	8	0.0000	0.6786
3	3	2	1.0000	0.6786
3	3	3	0.0000	0.6786
3	3	5	1.0000	0.6044
3	3	7	1.0000	0.5029
3	3	8	0.0000	0.6287
3	4	1	1.0000	0.5301
3	4	2	0.0000	0.4802
3	4	3	1.0000	0.6786
3	4	5	1.0000	0.6044
3	4	7	1.0000	0.4802
3	5	3	0.0000	0.6044
3	5	4	0.0000	0.4802
3	5	5	1.0000	0.6044
3	5	7	0.0000	0.4802
3	5	8	1.0000	0.6786
3	6	2	0.0000	0.5301
3	6	3	0.0000	0.6044
3	6	6	1.0000	0.4802
3	6	7	0.0000	0.4802
3	6	8	1.0000	0.6287
3	7	2	0.0000	0.6044
3	7	3	1.0000	0.5301
3	7	4	0.0000	0.6287
3	7	5	1.0000	0.8638
3	7	7	1.0000	0.4802
3	7	8	0.0000	0.6786
4	1	1	0.0000	0.3723
4	1	2	0.0000	0.3209
4	1	5	0.0000	0.4096
4	1	6	1.0000	0.3346
4	1	7	0.0000	0.3209
4	1	8	1.0000	0.4279
4	2	1	0.0000	0.2669
4	2	2	0.0000	0.2855
4	2	4	1.0000	0.2298
4	2	5	0.0000	0.2810
4	2	6	0.0000	0.2298
4	2	7	1.0000	0.3870
4	3	1	0.0000	0.2858
4	3	4	1.0000	0.2298

4	3	6	0.0000	0.2298
4	3	8	1.0000	0.2624
4	4	2	0.0000	0.2253
4	4	4	0.0000	0.2669
4	4	6	0.0000	0.2298
4	4	7	0.0000	0.2253
4	4	8	1.0000	0.2855
4	5	1	0.0000	0.2669
4	5	2	0.0000	0.2298
4	5	4	0.0000	0.2253
4	5	6	0.0000	0.2298
4	5	7	0.0000	0.2253
4	6	4	0.0000	0.2669
4	6	5	1.0000	0.2669
4	6	6	0.0000	0.2253
4	6	7	1.0000	0.2253
4	7	4	0.0000	0.2624
4	7	6	0.0000	0.2298
4	7	7	0.0000	0.2253
5	1	2	0.0000	0.3346
5	1	3	1.0000	0.4096
5	1	4	0.0000	0.3209
5	1	5	1.0000	0.4096
5	1	6	0.0000	0.3346
5	1	7	1.0000	0.3209
5	2	1	1.0000	0.3421
5	2	3	1.0000	0.3376
5	2	4	0.0000	0.3795
5	2	6	0.0000	0.3795
5	2	8	0.0000	0.3376
5	3	1	1.0000	0.2674
5	3	2	0.0000	0.3376
5	3	3	1.0000	0.3376
5	3	4	0.0000	0.3795
5	3	5	0.0000	0.2629
5	3	6	1.0000	0.3795
5	4	1	0.0000	0.3749
5	4	3	0.0000	0.3376
5	4	4	0.0000	0.3421
5	4	5	0.0000	0.2629
5	4	6	0.0000	0.3795
5	4	8	0.0000	0.2674
5	5	1	1.0000	0.3421
5	5	2	0.0000	0.3795
5	5	3	1.0000	0.2629
5	5	5	0.0000	0.2629
5	5	6	1.0000	0.3795
5	5	8	0.0000	0.3376
5	6	1	0.0000	0.2674
5	6	2	0.0000	0.3749
5	6	3	0.0000	0.2629
5	6	4	0.0000	0.3421
5	6	5	0.0000	0.3421
5	6	8	0.0000	0.3376
5	7	1	0.0000	0.3421
5	7	2	0.0000	0.2629
5	7	3	0.0000	0.3749
5	7	5	0.0000	0.2629
5	7	6	1.0000	0.3795
5	7	8	1.0000	0.3376

VARIABLE Offday[engineer,day] :

engineer	day	Activity	Reduced Cost
1	1	1.0000	0.0000
1	2	1.0000	0.0000
1	3	0.0000	0.0000
1	4	0.0000	0.0000
1	5	1.0000	0.0000
1	6	1.0000	0.0000
1	7	1.0000	0.0000
2	1	1.0000	0.0000
2	2	1.0000	0.0000
2	3	0.0000	0.0000
2	4	1.0000	0.0000
2	5	0.0000	0.0000
2	6	1.0000	0.0000
2	7	1.0000	0.0000
3	1	0.0000	0.0000
3	2	0.0000	0.0000
3	3	1.0000	0.0000
3	4	1.0000	0.0000
3	5	1.0000	0.0000
3	6	1.0000	0.0000
3	7	1.0000	0.0000
4	1	1.0000	0.0000
4	2	1.0000	0.0000
4	3	1.0000	0.0000
4	4	1.0000	0.0000
4	5	0.0000	0.0000
4	6	1.0000	0.0000
4	7	0.0000	0.0000
5	1	1.0000	0.0000
5	2	1.0000	0.0000
5	3	1.0000	0.0000
5	4	0.0000	0.0000
5	5	1.0000	0.0000
5	6	0.0000	0.0000
5	7	1.0000	0.0000

VARIABLE UWD[engineer] :

engineer	Activity	Reduced Cost
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000
5	0.0000	0.0000

VARIABLE OWD[engineer] :

engineer	Activity	Reduced Cost
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000

```
        5           0.0000          0.0000
  ---------------------------------------------


VARIABLE UOT[engineer,day] :

    engineer  day          Activity     Reduced Cost
  --------------------------------------------------
         1     1             0.0000          0.0000
         1     2             0.0000          0.0000
         1     3             0.0000          0.0000
         1     4             0.0000          0.0000
         1     5             0.0000          0.0000
         1     6             0.0000          0.0000
         1     7             0.0000          0.0000
         2     1             0.0000          0.0000
         2     2             0.0000          0.0000
         2     3             0.0000          0.0000
         2     4             0.0000          0.0000
         2     5             0.0000          0.0000
         2     6             0.0000          0.0000
         2     7             0.0000          0.0000
         3     1             0.0000          0.0000
         3     2             0.0000          0.0000
         3     3             0.0000          0.0000
         3     4             0.0000          0.0000
         3     5             0.0000          0.0000
         3     6             0.0000          0.0000
         3     7             0.0000          0.0000
         4     1             0.0000          0.0000
         4     2             0.0000          0.0000
         4     3             0.0000          0.0000
         4     4             0.0000          0.0000
         4     5             0.0000          0.0000
         4     6             0.0000          0.0000
         4     7             0.0000          0.0000
         5     1             0.0000          0.0000
         5     2             0.0000          0.0000
         5     3             0.0000          0.0000
         5     4             0.0000          0.0000
         5     5             0.0000          0.0000
         5     6             0.0000          0.0000
         5     7             0.0000          0.0000
  --------------------------------------------------


VARIABLE OOT[engineer,day] :

    engineer  day          Activity     Reduced Cost
  --------------------------------------------------
         1     1             0.0000          0.0000
         1     2             0.0000          0.0000
         1     3             0.0000          0.0000
         1     4             0.0000          0.0000
         1     5             0.0000          0.0000
         1     6             0.0000          0.0000
         1     7             0.0000          0.0000
         2     1             0.0000          0.0000
         2     2             0.0000          0.0000
         2     3             0.0000          0.0000
         2     4             0.0000          0.0000
         2     5             0.0000          0.0000
         2     6             0.0000          0.0000
```

```
        2    7            0.0000           0.0000
        3    1            0.0000           0.0000
        3    2            0.0000           0.0000
        3    3            0.0000           0.0000
        3    4            0.0000           0.0000
        3    5            0.0000           0.0000
        3    6            0.0000           0.0000
        3    7            0.0000           0.0000
        4    1            0.0000           0.0000
        4    2            0.0000           0.0000
        4    3            0.5000           0.0000
        4    4            0.0000           0.0000
        4    5            0.0000           0.0000
        4    6            0.0000           0.0000
        4    7            0.0000           0.0000
        5    1            0.0000           0.0000
        5    2            0.0000           0.0000
        5    3            0.0000           0.0000
        5    4            0.0000           0.0000
        5    5            0.0000           0.0000
        5    6            0.0000           0.0000
        5    7            0.0000           0.0000
  ---------------------------------------------------

VARIABLE ShiftBegin[engineer,day] :

    engineer  day          Activity     Reduced Cost
  ---------------------------------------------------
        1    1            8.0000           0.0000
        1    2            9.5000           0.0000
        1    3            0.0000           0.0000
        1    4            0.0000           0.0000
        1    5            9.5000           0.0000
        1    6            8.0000           0.0000
        1    7           10.0000           0.0000
        2    1            8.5000           0.0000
        2    2           15.0000           0.0000
        2    3            0.0000           0.0000
        2    4            8.0000           0.0000
        2    5            0.0000           0.0000
        2    6           10.0000           0.0000
        2    7            8.0000           0.0000
        3    1            0.0000           0.0000
        3    2            0.0000           0.0000
        3    3            9.0000           0.0000
        3    4            9.5000           0.0000
        3    5           13.5000           0.0000
        3    6           15.0000           0.0000
        3    7           12.0000           0.0000
        4    1           12.0000           0.0000
        4    2           11.5000           0.0000
        4    3           10.5000           0.0000
        4    4           15.0000           0.0000
        4    5            0.0000           0.0000
        4    6           13.0000           0.0000
        4    7            0.0000           0.0000
        5    1           10.0000           0.0000
        5    2            9.0000           0.0000
        5    3            8.0000           0.0000
        5    4            0.0000           0.0000
        5    5            8.0000           0.0000
        5    6            0.0000           0.0000
```

5	7	15.0000	0.0000

VARIABLE ShiftEnd[engineer,day] :

engineer	day	Activity	Reduced Cost
1	1	17.0000	0.0000
1	2	18.5000	0.0000
1	3	0.0000	0.0000
1	4	0.0000	0.0000
1	5	18.5000	0.0000
1	6	17.0000	0.0000
1	7	19.0000	0.0000
2	1	17.5000	0.0000
2	2	24.0000	0.0000
2	3	0.0000	0.0000
2	4	17.0000	0.0000
2	5	0.0000	0.0000
2	6	19.0000	0.0000
2	7	17.0000	0.0000
3	1	0.0000	0.0000
3	2	0.0000	0.0000
3	3	18.0000	0.0000
3	4	18.5000	0.0000
3	5	22.5000	0.0000
3	6	24.0000	0.0000
3	7	21.0000	0.0000
4	1	21.0000	0.0000
4	2	20.5000	0.0000
4	3	20.0000	0.0000
4	4	24.0000	0.0000
4	5	0.0000	0.0000
4	6	22.0000	0.0000
4	7	0.0000	0.0000
5	1	19.0000	0.0000
5	2	18.0000	0.0000
5	3	17.0000	0.0000
5	4	0.0000	0.0000
5	5	17.0000	0.0000
5	6	0.0000	0.0000
5	7	24.0000	0.0000