Job Scheduling considering both Mental Fatigue and Boredom

Sina Jahandideh

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE

In Mechanical Engineering
Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

University of Ottawa

March 2012

Abstract

Numerous aspects of job scheduling in manufacturing systems have been the focus of several studies in the past decades. However, human factors in manufacturing systems such as workers' mental conditions are still neglected issues and have not received adequate attentions. Job boredom and mental fatigue are both aspects of workers' mental condition. They affect work performances by increasing sick leave duration and decreasing work productivity. On the other hand, job rotation could be an alternative strategy to cope with such human issues at work. The benefits of job rotation for both employees and firms have been widely recognized in the literature. Although some studies found job rotation as a means to reduce workers' physical work-related traumas, they did not consider the effect of variable mental conditions on workers. Despite the proven importance of boredom and mental fatigue at the workplace, they have not been a combined precise objective of any job rotation problem in current literature.

The study of mental conditions proposed in this paper attempts to extend the previous works by addressing new methods and developing a feasible solution to increase manufacturing productivity. A new job scheduling program has been designed specifically which combines a new job rotation model and a job assignment method.

Acknowledgements

I would like to thank my supervisor Dr. Ming Liang for his direction, assistance, support and guidance. In addition, I offer my regards to all of those who supported me in any respect during the completion of the Thesis.

Table of Contents

List of Figures

List of Tables

Nomenclature

Parameters:

Cm	Task's mental difficulty
Cp	Task's physical difficulty
ap	Human physical ability
am	Human mental ability
O	Mental agility factor
β	Percentage of influence of worker's physical condition on mental condition

Variables:

PF	Physical fatigue factor
MF	Mental fatigue factor
Time	Production time for a specific task
MC	Mental condition factor

Abbreviation

MC	Mental condition
MCDM	Multi-criteria decision making method
AHP	Analytical hierarchy process
MSD	Musculoskeletal disorder
DRC	Dual-resource constrained
IDE	Integrated development environment
OR	Operational research

Chapter 1

Introduction

Manufacturing systems are often highly complex and their behaviour is dynamic and stochastic in nature. These systems consist of extensive interactions between people, information, materials, and machines. The study of the interface between human factors and manufacturing is not a new subject, but managers and production planners have minimized the significance of human factors to reduce complexity in modeling. There are many human factors that have been studied but some human factors are measurable and related specifically to performance productivity. They include:

1. Education
2. Experience
3. Motivation
4. Mental Health
5. Mental Fatigue
6. Job Boredom

In the following chapter, the gaps between theory and practice in job scheduling - as it pertains to human factors are discussed. The job scheduling problem specifies the allocation of resources to perform a set of activities in a period of time (Baker, 1974). As job shop scheduling is a very complex task in scheduling problems, job planners attempting to design a schedule for workers in manufacturing systems have to consider human factors very accurately. Nevertheless, they are still difficult to quantify and as a result, unpredictable factors. The purpose of this study is to define and develop a model for two important human factors: mental fatigue and boredom, and then use those in a model which allocates *n* workers to *m* tasks and increases productivity. The purported benefits of job rotation in this research are reduced employee mental fatigue and boredom, which leads to increased productivity in manufacturing systems, especially with night shift jobs. If proper quantitative models are available to help managers develop good rotation plans then the efficiency of job rotation can be significantly increased. The need for an effective job rotation model is further necessitated by the fact that many current management practices want to have multi-skilled employees which are cross-trained and rotated in different jobs in a predetermined fashion (Dulin, 1998, Thompson, 1999 and Anselmi, 2000). In the fourth chapter, the rotation model which is proposed in third chapter has been applied to a job-scheduling program in Visual Basic. Manufacturing companies can establish a strategic competitive advantage by placing a greater importance on the human element in the design and manufacturing process (Jordan, 1997).

1.1 Outline of the work

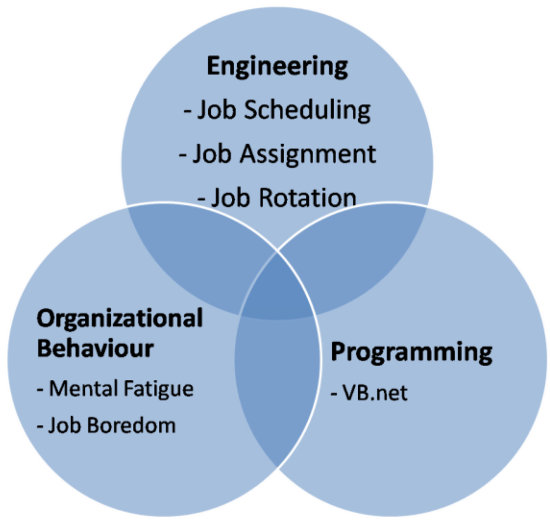

Figure 1 Interdisciplinary subject of the thesis

This research attempts to fill gaps in the area of human issues in manufacturing systems. The Human Mental Condition: Self-reporting scales, which are the typical way to measure job boredom and mental fatigue in past decades. Quantitative measurement methods such as Operational Research (OR) are very limited for these human conditions in the accessible literature. The OR methods, which have their own pros and cons, are not used in this research. Instead, there is a new method presented which is able to measure both boredom and mental fatigue and is then used for job rotation and job scheduling.

Job Rotation: Job rotation is mainly used to increase workers' skills and reduce exposure to risk factors such as injuries. However, there is no available research that introduces a model to deal with both job boredom and mental fatigue and then implement it.

Job Assignments: Although there are different methods for assigning jobs, this study adopts an analytical hierarchical process (AHP) method because it is an easy-to-implement multi-criteria decision making (MCDM) method to prioritize jobs based on multiple factors related to: difficulty level of tasks, capabilities of workers, and environmental constraints such as a production line capacity.

Job Scheduling: A job scheduling method generally optimizes the makespan and/or the sum of completion times at work. This research presents a new model to increase job productivity when dealing with workers mental conditions, particularly with boredom and mental fatigue.

1.2 Organization of thesis

This research is organized according to the following issues:

Worker's mental conditions**:** This combines job boredom and mental fatigue, which are two important human factors. The formulas are developed to quantify job boredom and mental fatigue and then two models are proposed for job rotation and job assignment.

Job Rotation: The formulas will be used to provide input to the rotation of jobs so as to reduce the mental burdens.

Job Assignment**:** An analytical hierarchical process (AHP) is proposed to prioritize jobs based on multiple factors related to:

1- Tasks

2- Workers

3- Environmental Constraints

Job Scheduling: The purpose of the job scheduling program is to assign a best possible task to each worker and increase job productivity**.**

Chapter 2

Literature Review

The goal of this thesis is to provide new job rotation, job assignment and job scheduling models, which help to maintain workers' optimal mental condition during their work and decrease productivity losses. Some related literature is briefly reviewed as follows.

2.1 Review of Job Boredom

Boredom at work is a common complaint among employees. There is a significant and positive relationship between repetitive task and boredom. Today workplaces are increasingly automated, with technology being the interface through which many tasks are completed. Many jobs in the past that involved skill use, decision-making, and contact with people, can now be achieved with the press of a few simple buttons.

2.1.1 Boredom definitions

Although boredom is considered a universal experience "there is no agreed definition or well-developed instrument for measuring it, and there is no comprehensive theory of its causes" (Fisher, 1993). There are different definitions for boredom in the literature:

Boredom is defined as an unwanted temporary state in which individuals feel an extreme lack of interest in their current activity(Dyer-Smith, 1997).

On the other hand, several researchers such as Hamilton (1981) have suggested that attention difficulties such as mind-wandering, are central to the experience of boredom. When people feel bored, they report having difficulty putting enough effort in to keep their attention focused on the task. Damrad-Frye (1989) stated that, "the essential behavioural component of boredom is the struggle to maintain attention." In addition, Sawin and Scerbo (1995) have defined boredom as a lack of motivation.

In this research, we believe that boredom is introduced by lack of mental agility in work places to address lost functional mental condition. Mental agility is an ability to work with changes and to react positively in unexpected situations. The word agility has the following two definitions:

"The power of moving quickly and easily; nimbleness: exercises demanding agility." and "the ability to think and draw conclusions quickly; intellectual acuity" (Dyer-Smith, 1997).

Dyer-Smith and Wesson (1997) developed a computer task to test the predictive ability of the Inertial Resource Allocation Model's (IRAM) measure of boredom . The procedure involves simple mental puzzles solved against a background of a continuous vigilance task. They also generated a tool (SWAT) which can discriminate when certain individuals with mostly low mental agility (or high inertia) are temperamentally unsuited for boring or repetitive work. In contrast, there are other individuals whose mental agility is high and are particularly well suited for repetitive tasks.

2.1.2 Boredom Causes

Traditionally, factors causing boredom are assumed to be external, such as repetitive or unstimulating tasks. However, according to recent studies there are possible internal causes of boredom relating to individual personality. Moreover, studies have shown that the degrees of boredom reported by different individuals performing in the same repetitive work environment may vary significantly. It has been also shown that boredom could be highly situation-specific and reversible, if the situation changes to some extent.

2.1.3 Boredom effects (outcomes)

Several researchers have studied boredom effects (Drory 1982, and Fisher 1993), and according to their research job boredom has the following effects on workers:

a) Employee absenteeism

b) Lack of job satisfaction

c) Productivity losses and performance variation

d) Accidents

In addition, it usually coincides with feelings of:

a) Restlessness

b) Moodiness

c) Desire to escape or change the situation to a more interesting activity

There are some studies concerning effects of boredom on manufacturing systems. Kass (2001) has used psychometrically-sound assessment devices to investigate the impact of job boredom, boredom proneness on job satisfaction, absenteeism, and organizational tenure, using a sample of employees at a manufacturing plant. His data from 292 workers in a manufacturing plant in the southeast United States showed that workers who experienced a high level of job boredom possessed much greater absenteeism and longer organizational tenure.

2.1.4 Boredom Measurement

For many years, self-reporting was the only method to measure boredom at work. Researchers Lee (1986) and Grubb (1975) measured boredom by similar self-reporting methods but with different items. Azizi (2010) used a Bayesian Networks method to measure boredom.

Bhadury (2006) developed several formulas to quantify boredom. In his formulas, boredom is measured by two methods: first, the boredom is equal to the total number of times that a worker is assigned to any task and second, the boredom is equal to the total number of times that a worker is assigned to the same task over a period of time.

2.2 Review of Mental Fatigue

2.2.1 Mental health

Mental health of workers has two impacts on a company:

1- During work hour: Mental fatigue increases on-the-job productivity losses.

2- After work hour: Workers who have mental fatigue or boredom long term are high risk for mental health issues and this increases a company's health care costs. There is some evidence showing that workers' mental health issues may have greatest impact on productivity losses after work hours, including increased absenteeism, short-term disability, and higher turnover. According to the Washington Business Group on Health, the most serious employee benefit problem facing employers today is the unrelenting increase in health care costs—up 50% in the past 5 years—with no apparent end in sight. With the cost of benefits rising so fast, why should employers remain engaged in the provision of health care benefits to their employees? Aside from attention to employee morale and union pressures, one answer may be found in the following observation regarding emerging corporate priorities: employers are becoming increasingly aware that workers' health influences workers' productivity, and that productivity has a positive impact on organizational performance and competitiveness. Employers are recognizing the importance of their role in better promoting the health and productivity of their workers (Goetzel, 2002).

2.2.2 Occupational Fatigue

Fatigue is defined as lassitude, or exhaustion of mental and physical strength resulting from bodily labour or mental exertion. It is a concern of workers in many occupations throughout the world. While occupational safety and health has improved in recent decades, fatigue remains a common problem in developed countries (Lewis, 1992). Fatigue is a serious threat to quality of life and severely compromises work performance when it becomes chronic or excessive (Piper, 1989 and Okogbaa, 1994). Unfortunately, its complex and dynamic nature makes fatigue difficult to define, observe and measure.

Fatigue is a very common experience at work especially for workers who do shift work or extended work shifts, as they are at higher risk of being affected. Many individual factors, such as age, health, and living conditions, influence a worker's ability to adapt to shift work or changes in shift sequences, timing or duration. Likewise, extended work shifts and overtime introduce unique fatigue issues. Shift design strategies are essential to preventing both mental and localized physical fatigue. The effect of extended work shifts is highlighted because long working hours can negatively affect workers' health and well-being (Sparks, 1997 and Hulst, 2003). There are multiple individual characteristics, such as strong physical and mental ability, or better work conditions which allow workers to better adjust to extended work shifts, as not all workers are affected by extended work shifts.

According to fundamental social psychology literature summarized by Reason (1990): accidents, rework, conflicts, and decreases in productivity are caused by deficiencies in mental function that are accelerated as mental and physical fatigue increase.

Immediate effects of fatigue on humans include:

•Reduction in physical capacity (weakness)

•Reduction in work

• Human errors (Mistakes, Slips, Lapses)

•Weariness

•Memory loss

•Sleepiness

•Discomfort

•Weakened motivation

•Irritability

•Illness

Presenteeism is another affect of fatigue on workers and companies. According to Biron (2006), presenteeism is the action of employees coming to work despite having a sickness that justifies an absence and consequently they are performing their work under sub-optimal conditions.

Three different fatigues have been recognized at work environments: physical fatigue, mental fatigue, visual fatigue.

I. Physical Fatigue

Physical fatigue in industrial activities has been defined as potential musculoskeletal disorder (MSD) risks, but modeling approaches of physical fatigue have not been explored to prevent MSD risks in worker assignment planning.

II. Mental Fatigue

Mental fatigue is a psychobiological state caused by prolonged periods of demanding cognitive activity and characterized by subjective feelings of “tiredness” and “lack of energy” (Boksem, 2008). The effects of mental fatigue on cognitive performance and skilled performance of drivers and air pilots have been widely studied. Mental fatigue can also affect physical performance in humans and vice versa.

The main finding of Marcora's (2009) experimental study was that mental fatigue impairs physical performance, which was measured as time to exhaustion during high-intensity cycling exercise. This type of physical performance (short-term endurance in thermo neutral conditions) is traditionally thought to be limited by cardio-respiratory and muscle-energetic factors.

III. Visual Fatigue:

Visual fatigue is defined as a feeling of tiredness resulting from a visual task or the visual environment in which the task is performed. Nowadays, more people are suffering from visual fatigue because of stress, pollution, computer work, television, video games, etc. For some people, visual fatigue can also lead to reduction in job productivity and diminished ability to concentrate on tasks.

Human errors can be found in judgment, decision-making and physical actions, and they result in loss of productivity and the need for improvement in industrial operations and occupational injuries. In order to prevent human error, organizations conduct training, provide feedback to workers, and perform inspections (Hinze, 2006). But training is only one option to decrease human errors caused by human fatigue. Another option is finding a proper job schedule for workers to control and decrease the workers fatigue.

Fatigue has been shown to cause significant and frequent health impacts on quality of life such as coronary disease, hypertension (Dembe, 2005), diabetes (Harma, 2006), insomnia (Tari, 2007) and injuries (Sparks, 1997) among workers in multiple industries.

Boksem (2008) defines mental fatigue as a feeling of cognitive activity experienced after or during prolonged periods. Mental fatigue is very common in everyday modern life and generally involves tiredness or exhaustion, an aversion to continue with the present activity, and a decrease in the level of commitment to the task (Holding, 1983, Meijman, 2000).

The effects of fatigue on workers during and after work hours are:

1- Effects of fatigue on performance during work hours:

- decreased quality of job
- increased amount of injuries during work hour
- decreased job productivity

2- after work hours fatigue impacts:

- stress

- injury at home
- slow recovery from illness
- smoking
- suicide due to overwork
- Cardiovascular disease
- Lack of exercise
- Heart disease

2.2.3 Influence of physical fatigue on mental fatigue

Despite the effects of physical fatigue, relatively few studies have investigated both physical and mental fatigue in industry. Most studies observed the impact of mental activities to physical and mental fatigue, but few observed the impact of physical activities to physical and mental fatigue. The studies conducted by Boksem (2005), Murata (2005) and Kurimori (1995) have shown that there is a relationship between physical and mental fatigue during repetitive tasks that leads to mental stress and physical tension.

2.2.4 Fatigue and lost production time

Ricci (2007) conducted a survey estimating fatigue prevalence and associated health-related lost productive time (LPT) in U.S. workers. The results gathered from 2746 employees showed that:

- The estimated prevalence of fatigue in the U.S. workforce for a two-week period was 37.9%.

- Fatigue was more prevalent in women, workers under age 50, white workers and workers earning more than $30,000 per year in “high control” positions—that is, jobs with a high degree of latitude in decision-making.
- Overall, 9.2% of U.S. workers with fatigue reported LPT specifically due to fatigue in the previous two weeks. Such workers lost an average of 4.1 productive work hours per week, most of which was reflected in reduced performance at work rather than absence from work, i.e., presenteeism rather than absenteeism. For these workers, fatigue affected their work performance primarily by impairing their concentration and increasing the time required for them to complete tasks. In addition, distracted workers were naturally more likely to have safety incidents. The researchers estimated that workers with fatigue cost U.S. employers $136.4 billion per year in health-related LPT—$101 billion more than workers without fatigue.

2.2.5 Fatigue measurement and solutions

There are psychological and physiological methods to measure mental fatigue. Boksem (2005), Murata (2005) and Kurimori (1995) used Electroencephalography (EEG) and Event-related Brain Potentials (ERP) to detect and evaluate mental fatigue. The EEGs and ERPs simultaneous measurements can discern the effects of mental fatigue on ongoing brain processes and reveal the brain’s responses to particular events.

In contrast, there is very little research available which measures or uses a mathematical model of fatigue . Jaber (2010) investigated the effects of human fatigue and recovery on the performance of dual-resource constrained (DRC) systems. He developed a mixed-integer linear programming model to formulate and solve the workforce flexibility issues when fatigue and recovery are taken into consideration in

DRC system design. The results showed that short rest breaks after each task, short cycle times, and faster recovery rates improve workers performance.

Konz (2004) presented a working rest method to deal with fatigue. The purpose of working rest is to rotate the tasks with different demands, which provides recovery time to the newly unloaded musculature.

2.3 Health care and productivity loss

Recently, awareness of quantifying the relationship between workers' health statuses and their lost productivity has been an important subject for employers and researchers. According to Boles (2004) and Mattke (2007), the high cost of health care, including health-related productivity loss, is recognized as a serious threat to the competitiveness of US employers. The first national Workforce Health and Productivity Summit 2008 in New Mexico, US confirmed that employers are spending an estimated $13,000 per employee per year in total direct and indirect (i.e. productivity- related) health costs. Keeping people healthy and productive at work is one of the factors that can enhance productivity in a manufacturing system.

2.4 Review of Job Rotation

Job rotation has been described as a replacement method for moving workers from one task to another, which increases an employee's capacity and value to an organization. Job rotation can also be defined as a job design technique in which workers are transferred among two or more workstations in a planned manner. Job rotation has recently attracted much attention in both industry and academia (Osterman, 1994, Gittleman, 1998, Krajewski, 1999 and Ortega, 2001). The benefits of job rotation mentioned in the literature include fostering employee learning, reducing employee boredom, and leading to increased motivation and human capital accumulation.

Azizi (2010) proposed a model, which was aimed at balancing the positive effects of job rotation intervals on worker's boredom recovery/motivation and the unavoidable cost of worker's skill variations. The objective of his model was to minimize the total delay caused by the lack of skill and/or motivation during production horizon.

2.5 Review of Job Assignment

Traditionally the purpose of job assignment is to assign a given number of workers to a given set of tasks at the same time, in order to optimize some criteria such as the total cost, or profit generated by the assignment. Job planners also have to consider other factors when they assign workers to each task, such as machines' capacities. An important role of job assignment is to properly place workers in jobs to ensure compatibility between the workers and their required tasks. Appropriate job assignment ideally benefits both employers and workers. However, this approach has one major drawback: when the same job-assigning problem is repetitively solved, the worker–task assignments may turn out to be the same (i.e. the same jobs will need to be repeatedly performed by the same workers without any rotation). This may lead to boredom on the job due to repetition (Pinedo, 1995, Shtub, 1996 and Al-Subhi, 1997).

There are different methods for assigning jobs in manufacturing systems depending on several factors such as tasks' conditions or production plans. Bhadury (2006) have proposed models to assign tasks to workers by restricting the maximum number of assignments of a worker to any job and the maximum number of consecutive assignments of a worker to the same job.

In this research, the multi criteria decision analysis (MCDA) method is used to assign tasks to workers. Multi Criteria Analysis is a process that utilizes weighted criteria to rank alternatives. MCDA methods develop a decision matrix to present a systematic analytical approach for integrating uncertainty, risk levels, and valuation. According to Howard (1991), the MCDA methods consist of procedures and mathematical algorithms used in a decision-making process when there are multiple criteria to analyze. MCDA methods are usually divided by discrete and continuous methods (Janssen, 1992). Discrete methods try to identify the most desirable alternative from a finite set of alternatives. In contrast, continuous methods attempt to identify an optimal alternative from an infinite number of feasible alternatives (Hajkowicz, 2000).

The general classification of MCDA methods (Table 1) adopted in this paper is suggested by Hajkowicz (2000).

<table>
<tr><td rowspan="7">Multi Criteria Decision Analysis Methods</td><td>Continuous Methods</td><td colspan="3">Linear Programming Method
Goal Programming Method
Aspiration-based Models</td></tr>
<tr><td rowspan="6">Discrete Methods</td><td>Transforming Methods</td><td colspan="2">Methods for standardizing data into common units
EV Method
Naive Method</td></tr>
<tr><td rowspan="2">Weighting criteria Methods</td><td>Qualitative Methods</td><td>Raking Methods
Rating Methods</td></tr>
<tr><td>Quantitative Methods</td><td>Fixed point Scoring Method
Graphical Methods
Analytical Hierarchy Process
Judgement Analysis
Trade-Off Method</td></tr>
<tr><td rowspan="3">Ranking alternative Methods</td><td>Qualitative Methods</td><td>Concordance Analysis Method
Lexicographic Ordering Method
Linear Assignment Method
Frequency Method</td></tr>
<tr><td>Quantitative Methods</td><td>Multi Attribute Value Theory
Multi Attribute Utility Theory
Discordance Analysis
Weighted Summation
ELECTRE Method
PROMETHEE Method</td></tr>
<tr><td>Mixed Methods</td><td>Evamix
Cardinalisation of Evaluation Matrix
Value Focused Thinking
Analytical Hierarchy Process</td></tr>
</table>

Table 1 MCDA methods classified by Hajkowicz (2000)

Therefore, choosing the proper method can be important to achieve better results and conclusions. The analytical hierarchy process, which is a discrete method used to weight criteria and rank alternatives, is a preferred method for assigning tasks to workers.

2.5.1 Analytical Hierarchy Process (AHP)

The Analytical Hierarchy Process (AHP) analysis is one of the most widely used methods for addressing decision-making problems with multiple criteria and a mixture of qualitative and quantitative data inputs, developed by Saaty in 1980. This method is well suited for situations where criteria can be organized into a hierarchy by dividing the problem characteristics into sub-criteria. AHP helps to deconstruct a complex problem into its sub-components.

It was widely used in various industries and problem settings such as software selection (Lai, 2001), higher education (Liberatore, 1997), and environmental conflict analysis (Malczewski, 1997). It has a pair-wise comparison feature, which facilitates the weighing process of components of the hierarchy.

Most multi-criterion decision analysis methods require some measurement of relative importance to be added to the criteria. Assigning weights to the criteria is probably the most important part of MCDA as it allows different views and their impact on the ranking of alternatives to be explicitly expressed (Royal Assessment Commission, 1992).

2.5.1.1 Weighting the Hierarchy

AHP method provides decision-makers with a structured weighting system (Saaty, 1980). The Analytical Hierarchy Process method involves the following five main steps:

Step 1: Defining the problem and objective of the analysis.

Step 2: Categorizing the problem into more comprehensive hierarchy sub-problems. In this step the hierarchy from the top level (criteria) through the intermediate level (sub-criteria) to the lowest level (alternatives) is constructed.

Step 3: Conducting pair-wise comparisons among levelling problem hierarchy characteristics and problem alternatives into collecting trade-off data inputs. In this step, comparison matrices are created by determining the relative importance of each element (alternative) with respect to each higher-level using the fundamental scale of absolute numbers from 1 to 9 (or 1 to 100).

Step 4: Estimating the hierarchy criteria's relative weights. In this step, the priority weight (score) of each element (alternative) is determined and then the consistency for the current level is evaluated.

Step 5: Collecting criteria by combining relative weights (criteria and sub-criteria) for performing an overall evaluation of alternatives. In this step, overall priorities are determined and the overall consistencies are evaluated.

In figure 2 below, the general form for such a problem hierarchy as used in AHP, is illustrated. The lines between levels indicate relationships between criteria, alternatives and objectives.

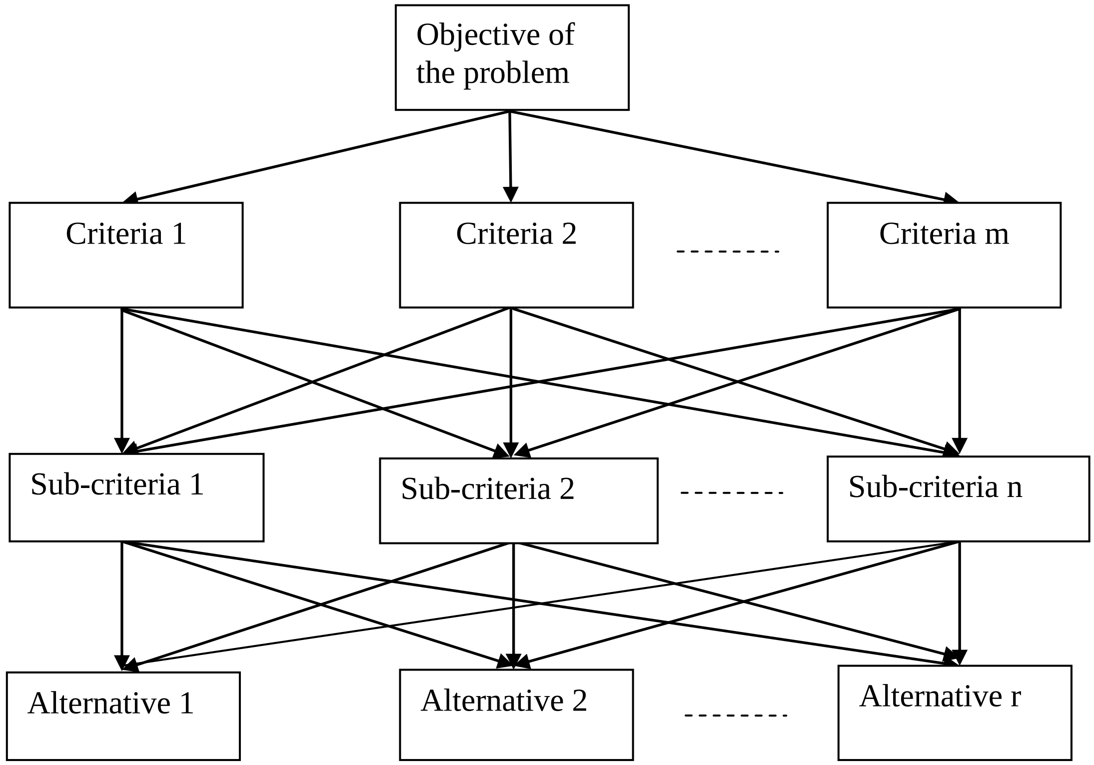

Figure 2 Hierarchy of components of job assignment

Next, two methods are presented to solve the objective of this thesis. The first is a pair-wise comparison matrix method for both alternatives and criteria and the second is a utility function with pair-wise comparison method.

2.5.1.1.1 Pair-wise comparison matrixes

This method compares both criteria and alternatives determined by pair wise comparisons matrixes. To determine the priority weight of each element, first the general case needs to be examined. The criteria and sub-criteria are indicated by $A1, \ldots, An$ and their weights by $w1, \ldots, wn$. The pair wise comparison matrixes take the following form:

$$A=\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{2n} \end{bmatrix} = (a_{ij})_{nxn} \tag{1}$$

where $a_{ij} = w_i/w_j = 1/a_{ji}$ and $a_{ii}=1$, i ,j=1,2,...,n. a_{ij} represent the pair wise comparison rating for criteria *i* to *j*. The number of comparisons is a combination of the number of sub-criteria to be compared. $n(n - 1)/2$ pair-wise comparisons are needed for a nxn matrix. Saaty (1977) developed the eigenvector method for the purpose of constructing the vector of priority weights and for testing inconsistency. In the case of perfect consistency:

$$AW = nW \tag{2}$$

A is the $n \times n$ comparison matrix and $W = (w_1, w_2, \ldots, w_n)^T$ is the preference weightings of each criteria. By multiply matrix A by the column vector W the vector nW can be determined.

$$AW = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{2n} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} nw_1 \\ \vdots \\ nw_n \end{bmatrix} = nW \tag{3}$$

In other words, the best alternative is indicated by the following relationship:

$$A_{BestAlternative} = \max_i \sum_{j=1}^{n} a_{ij} w_j \tag{4}$$

where i ,j=1,2,...,n.

Saaty suggested nine scales for pair-wise comparison. The details of these scales are explained in Table 2.

<u>**Value**</u>	<u>**Definition**</u>	<u>**Explanation**</u>
1	Equal importance	Two activities contribute equally to the objective.
3	Moderate importance	Experience and judgement moderately favour one activity over another.
5	Essential or strong importance	Experience and judgment strongly favour one activity over another.
7	Very strong importance	An activity is strongly favoured and its dominance demonstrated in practice.
9	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation.
2,4,6,8	Intermediate values between	These numerical values are used when the two adjacent judgments compromise is needed between the odd numbered relative values.

Table 2 The Saaty Rating Scale (1980)

2.5.1.1.2 Utility Function

In this method, after constructing the hierarchy, the next step is to create the utility functions. In this process, value functions are used to determine utilities. There are different value function techniques and the next graph illustrates the increasing continuous value function used in this research (Figure 3).The continuous value functions are categorized as increasing and decreasing functions. In increasing functions, the minimum score has value of 0 and maximum score has value of 1. Alternatively, in the decreasing functions, the lowest score has value of 1 and the highest score has value of 0.

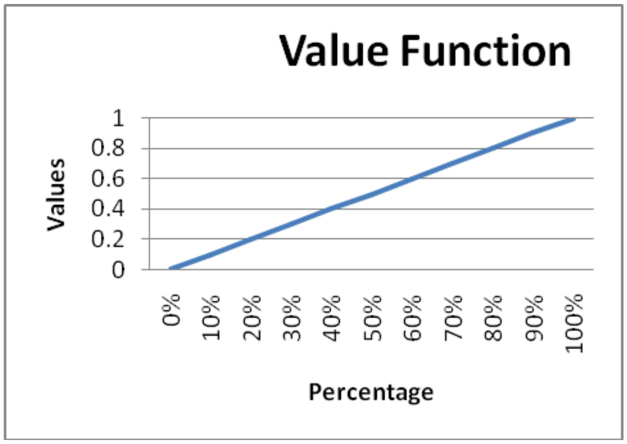

Figure 3 The continuous value function used in this research

With this method, the relative importance of the alternative with respect to each sub-criteria is not constructed. Instead, the utilities of all alternatives with respect to each sub-criteria are determined by experts or decision makers. Then weights of criteria and sub-criteria are multiplied by weights of alternative. This method will be used in this research to find ideal match tasks for each worker.

In general, AHP has a number of advantages that are relevant to this research. It is a useful method to describe the group of pair wise comparison data to decision makers, especially in subjective cases.

2.6 Review of Job Scheduling

According to Sabuncuoglu (2003) scheduling is a “decision making process that concerns the allocation of limited resources (machines, material handling equipment, operators, tools, fixtures, etc) to competing tasks (operations of jobs) over time, with the goal of optimizing one or more objectives”. Alternatively, Voudouris in 2008 explained that the job scheduling consists of allocating jobs to a worker in order to deliver services to customers. It means assigning a job to the right sources in the right place, at the correct time.

The ideal condition for researchers is to create sequences of jobs that can be followed without human involvement, and to have the shop floor execute the plans as directed without deviation. This goal has been set in some manufacturing systems such as automated assembly lines.

Nowadays job scheduling is an automated process as most medium and large-sized organizations are negatively affected when they undertake manual job-scheduling while trying to uphold the company's rules and goals. This leads to sub-optimal work allocation over the scheduling horizon, done in a reactive way without looking beyond the next jobs to be dispatched (Voudouris, 2008).

2.6.1 Job scheduling and human factors

A common discussion of job scheduling in recent literature concerns gaps between theory and practice in scheduling. There are many possible reasons for these gaps, such as difficulties associated with improving the effectiveness and efficiency of production in various manufacturing systems. One of these gaps is a lack of fit between the human factors and the formal or systemized factors found in the scheduling methods and systems, which can cause failure in scheduling (McKay, 1999).

Although human performance, safety, and health have traditionally been the subject of study in engineering and industrial psychology, very few job schedulers considered other human factors in their models and problems. In this way, job schedulers avoid the complexity of mathematical analyses. However, a complicated mathematical analysis method, such a mixed integer nonlinear mathematical programming method, can lead to infeasible answers.

The purpose of this research is to introduce a new method to schedule jobs that is reliable, adaptable with any manufacturing systems, and easy to operate by managers.

2.7 Limitations of previous work

Job boredom and mental fatigue are measured by two proposed formulas in this research. In previous research on job boredom, two of three formulas were used based on calculating production time in repetitive tasks and only one of them had used a human factor (Motivation). After investigating formulas, methods such as the mixed integer linear programming were developed to measure job boredom and mental fatigue. These methods were either too simple, or too complicated and impractical such that they could not been used for other purposes. There is only one formula for mental fatigue in available research, which was also calculated by production time.

The job rotation model proposed in this research is based on studies on psychology, human behaviour, and also on real world work experiences obtained from large and very complex companies such as automotive companies. The following solutions were proposed in the literature for workers' job boredom and fatigue:

1- Short rest breaks after each task to deal with fatigue
2- Learning new skills to deal with job boredom

Short rest breaks decrease the periods of cognitive activity of workers. However, short cycle rests are not always an option for mangers, especially in complex manufacturing systems. Having short rest breaks in a production line increases the setup time. Setup time is time to get a machine ready for production. Longer setup time increases production costs and reduces production capacity (idle machines during setup). On the other hand, learning new skills increases the mentally engagement of workers. But learning new skills is not always a preferred option because there may not be many skills to learn and workers can forget what they have learned. Moreover, training multi-skilled workers is costly and requires continuous attention (Azizi, 2010).

This research first considers primary states of mind and then proposes a basic and a viable option for job rotation. Although having breaks and learning new skills helps to deal with mental fatigue and job boredom, working is a fundamental task for all workers. Consequently, assigning different tasks with different conditions to workers could be more complex but a better option for managers and job planners.

Another limitation in previous research is the complexity of implementing job rotation. Azizi (2010) proposed linear and nonlinear mixed integer mathematical programming models. The complexity of this operational research method may result in unrealistic solutions. For that reason, going further and conducting job scheduling based on a job rotation model can be more complex.

Chapter 3

Analysis and Modeling

The purpose of this new job scheduling method is to assign a best possible match task to each worker and increase job productivity when it deals with job boredom and mental fatigue. Learning new skills has been introduced by some researchers as a way to deal with workers' boredom but the problem is that this causes mental fatigue, especially when learning while working, and during difficult conditions like night shifts. On the other hand, mental fatigue can be caused by frequent mental attempts and constant attention on a task, as well as a high level of stress or emotion. The over-activity of the brain creates a condition where the brain cells become exhausted, similar to when human bodies are physically exerted.

As a result, mental fatigue and job boredom decrease on-the-job productivity and increase negative after-job issues. This research reveals that these two human mental conditions can be solved with a solution for job rotation. The entire decision process, including mental condition quantification, job rotation, AHP, and scheduling has been implemented using Visual Basic.Net to design a new job scheduling program, which is interactive and easy to implement by managers. This program is capable of measuring two important human factors and providing a unified solution.

This new job scheduling program helps manufacturing systems cope with workers' job boredom and mental fatigue. There are additional benefits including:

- Increased job satisfaction
- Decreased job boredom
- Increased manufacturing system flexibility with more skilled workers

- Decreased absenteeism
- Decreased accidents
- Decreased mental fatigue

When a worker becomes bored, to prevent the lost productivity, he/she will move to a new task with higher level of task condition in regards to his/her mental agility. Then the worker stays there until he/she gradually becomes mentally fatigued and again, to prevent the lost productivity, he/she will be moved to a new task with a lower level of mental condition. Then after obtaining mental rest in an easier job, the process of becoming bored will commence and he/she should be moved to a new task with higher level of mental condition. The overall program structure is shown in figure 4.

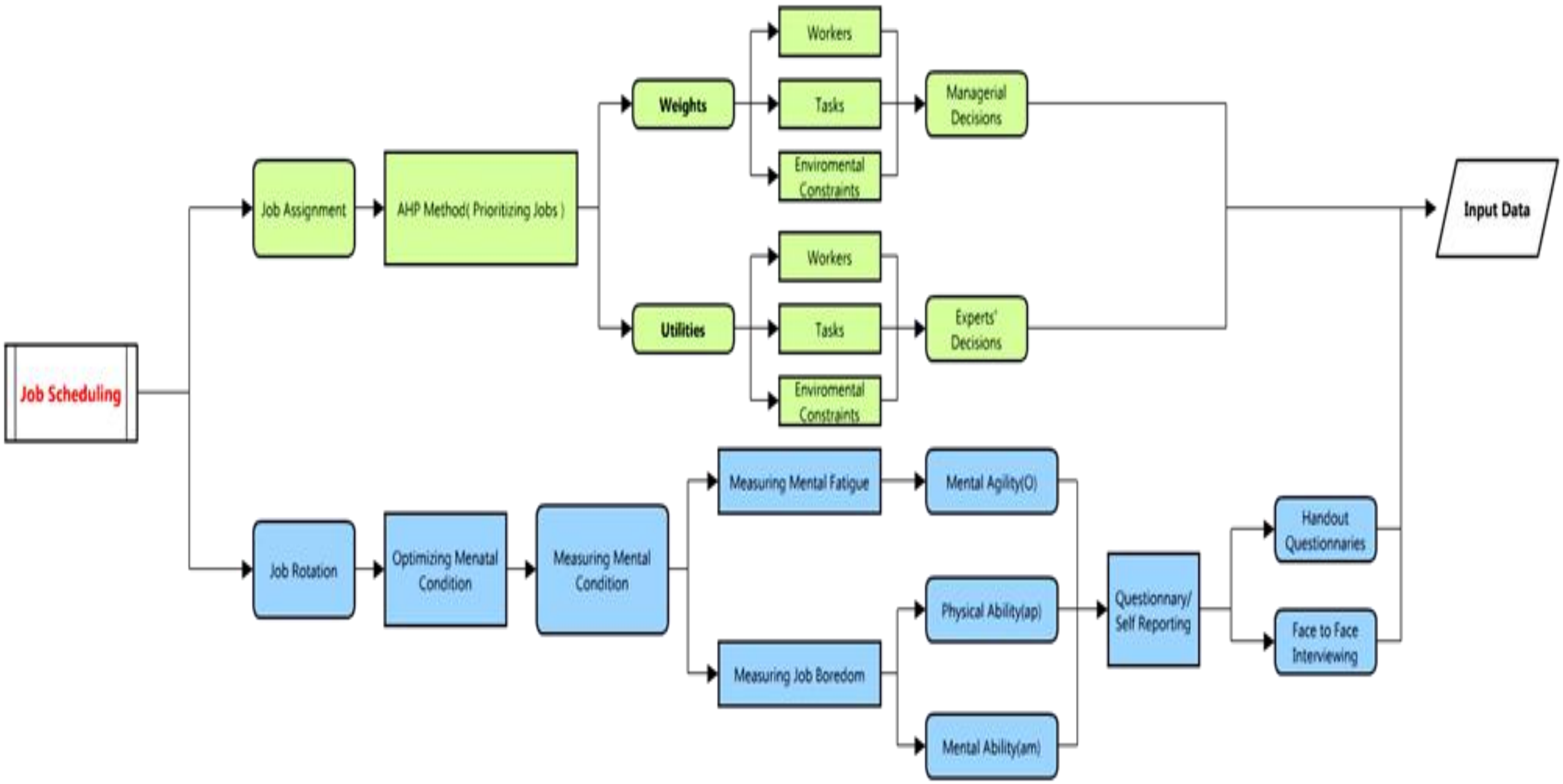

Figure 4 Overall Structure of the Program

3.1 Problem statement

Our purpose is to maximize manufacturing system productivity and minimize boredom and mental fatigue. These two objectives are often conflicting states of humans' brains and can be defined as factors which are connected together contrariwise to each other (i.e. optimizing one will impair the other). When workers are bored it means that their minds (with different mental abilities) have lost mental agility (engagements). By doing another task with different work conditions, they need to reengage or relearn/learn new skills, and their mental agility will be decreased but their mental fatigue will be increased. Therefore, it is desirable to find a preferred trade-off between the two.

Two task conditions are considered in this research:

1- Rule-based task (need more physical effort and not mental effort): Low complex condition task.

2- Skill and knowledge-based task (need more mental effort and not physical effort): high complex condition task.

3.2 Sub-problem

The program is divided into three sub-problems, which are: mental fatigue, job boredom and mental condition measurements.

3.2.1 Mental fatigue measurement

The human mental ability and agility of each worker are described by *am, O* values, which range from zero to one. The condition of each task is described by *cm, cp, CRT,* and *SR*. An *am* or *O* value of zero means that worker *i* is fully qualified to work in all tasks with all mental conditions, and a value of one means that he/she has mental difficulties working there. A fractional *am* or *O* value applies when a worker can be assigned to a particular task, but he/she is not fully qualified. The *cm* value of zero means that the task has a condition, which does not need a great deal of mental effort. When *cm* equals one, it means the task fully engages workers' mentality. The mental fatigue and physical fatigue can be calculated by the following proposed equations:

$$\text{Mental Fatigue (MF)} = \text{Mental Task Condition (cm)} \cdot \text{Human Mental Ability (am)} \cdot \text{Production Time} + \text{Physical Fatigue} \qquad (5)$$

$$\text{Physical Fatigue (PF)} = \text{Physical Task Condition (cp)} \cdot \text{Human Physical Ability (ap)} \cdot \text{Production Time} \qquad (6)$$

The following formula is the sum of the mental fatigue that all workers in a manufacturing system experience:

$$\text{Mental fatigue index} = (am \cdot cm + \beta \cdot ap \cdot cp) \qquad (7)$$

$$MF = (am \cdot cm + \beta \cdot ap \cdot cp)\, Time \qquad (8)$$

The *Time* variable is defined as production time, which is dependent on the amount of time in a specific task and must be equal to length of the work-shift. *cm* and *cp* depend on task conditions. *am and ap* depend on workers abilities. The parameter β shows how strongly physical fatigue effects mental fatigue.

Figure 5 shows a relationship between production time and the mental fatigue index. When a task is assigned to a worker, the start point is located depending on the worker and initial task conditions. However, when production time is increased, the mental fatigue felt by workers increases. Once a worker is assigned to a task, his/her mental fatigue increases as long as he/she continues to work on a same task. Worker's mental fatigue index for each task depends on worker and task conditions. Hence, the mental fatigue diagram, which shows the amount of mental fatigue before and after task rotation, is presented in Figure 5.

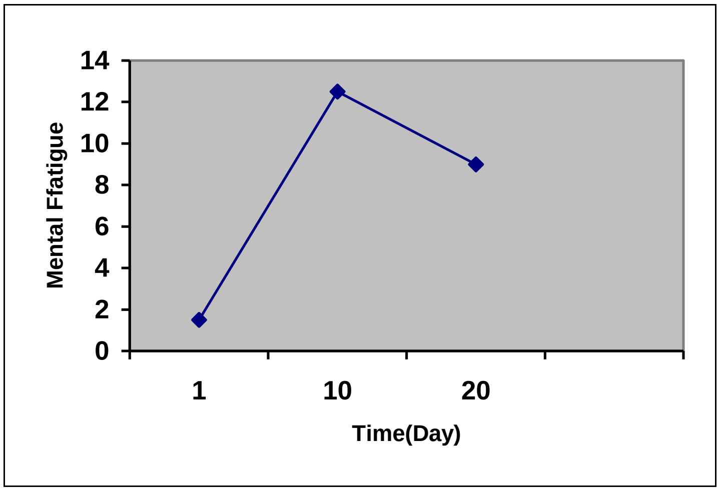

Figure 5 Mental Fatigue Chart

The slope of mental fatigue specifies if the mental fatigue is in increasing or decreasing mode.

3.2.2 Job boredom measurement

The Mental Agility Factor (MAF) of each worker is described by *O* value, which ranges from 0 to 1. The value 1 means worker *i* experiences mental disengagement with his/her task *j*. Zero value means worker *i* desires to engage completely to his/her task *j*.

$$Boredom = Mental\ Agility\ Factor\ (O) \cdot Mental\ Task\ Condition\ (cm) \cdot Production\ Time\ (Time) \quad (9)$$

The following proposed formula is the sum of the Boredom that all workers in a manufacturing system experience:

$$Boredom = O \cdot cm \cdot Time \quad (10)$$

Figure 6 shows a relationship between production time and the job boredom index. When a task is assigned to a worker, the start point is located depending on the workers and initial task conditions. However, when production time is increased the job boredom felt by workers grows. Once a worker is assigned to a task, his/her job boredom increases if he/she continues to work on the same task. Worker's job boredom index for each task depends on the workers and task conditions. Therefore, the job boredom diagram, which shows the amount of job boredom before and after task rotation, can be presented in figure 6.

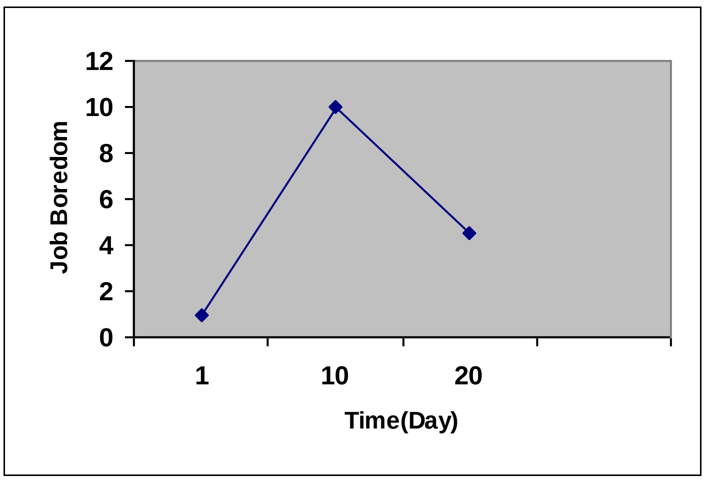

Figure 6 Job Boredom Chart

The slope of job boredom specifies if the boredom is in increasing or decreasing mode.

3.3 Mental Condition Measurement

The purpose of this formula is to present a job sequence schedule in a manufacturing system considering job boredom and mental fatigue.

$$MC = MF + Boredom \quad (11)$$

$$MC = [(am \cdot cm + \beta \cdot ap \cdot cp) + O \cdot cm] Time \quad (12)$$

Mental condition index = [(am·cm+ β·ap·cp) + O·cm]

Figure 7 shows a relationship between production time and the mental condition index. When a task is assigned to a worker, the start point is located depending on the worker and initial task conditions. However, when production time is increased, the mental condition increases. Once a worker is assigned to a task, his/her mental condition deteriorates if he/she continues to work on the same task. Workers' mental condition index for each task depends on the worker and task conditions. Therefore, the mental condition diagram, which shows the amount of mental condition before and after task rotation, can be illustrated in Figure 7.

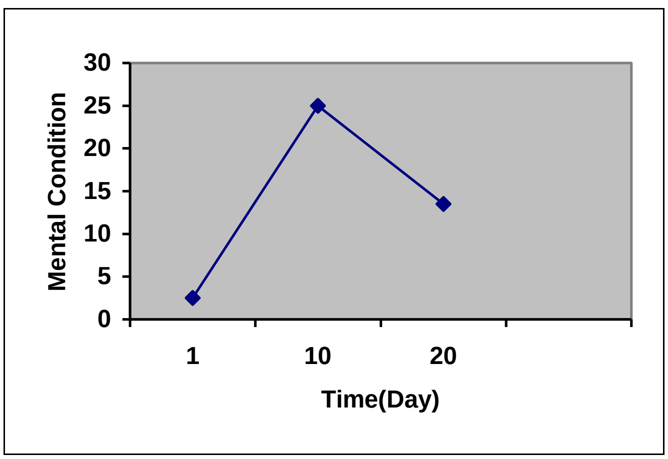

Figure 7 Mental Condition Chart

The slope of mental condition specifies if the mental condition is in increasing or decreasing mode.

3.4 Job rotation

The job rotation model proposed in this research considers an integrated solution for mental fatigue and job boredom.

1- If a worker with low level of mental fatigue (MF) transfers to another task, then his/her MF at the new task regularly increases until it reaches the upper bound level of MF. However, if the worker continues to perform the same task, his/her productivity declines because of the lost mental ability to work. In such cases, when the worker's MF reaches the upper bound, he/she will be transferred to a new task with lower work condition (needs less mental activity).

2- If a worker with high level of mental fatigue (MF) transfers to a new task with lower work condition (needs less mental activity), then his/her MF at the new task regularly decreases until it reaches the lower bound level of MF and at the same time the level of boredom will begin to increase.

3- If the worker continues to perform the same task, his/her productivity declines because of boredom (low level of mental agility). In such cases, when the worker's boredom reaches the higher bound, he/she will be transferred to a new task with higher work condition, which will increase his/her mental agility (MA).

4- If a worker with high level of boredom (less level of MA) transfers to a new task with higher work condition (needs more mental activity), then his/her boredom at the new task regularly decreases until it reaches the lower bound level of boredom and at the same time the level of MF will begin to increase, as mentioned in part 1.

3.4.1 Job rotation example

To demonstrate how job rotation works, an example is presented. Table 3 illustrates a job rotation model for worker A. To create a job rotation model, first the mental condition of the worker needs to be calculated for each day. The parameters are shown in column no. 2- 9 are specified by managers and experts in the chapter 4. These parameters use the range of numbers from zero to one.

Time(Day)	Mental ability	Task's mentality difficulty	Influence coefficient	Worker's physical ability	Task's physicality difficulty	Mental agility	Task's mentality difficulty	Mental condition	Task group
1	0.6	0.8	0.5	0.5	0.5	0.6	0.8	2.835	Harder Task
2	0.6	0.8	0.5	0.5	0.5	0.6	0.8	3.92	
3	0.6	0.8	0.5	0.5	0.5	0.6	0.8	5.005	
4	0.5	0.4	0.5	0.4	0.7	0.7	0.4	2.37	Comfort Task
5	0.5	0.4	0.5	0.4	0.7	0.7	0.4	2.99	
6	0.5	0.4	0.5	0.4	0.7	0.7	0.4	3.61	
7	0.5	0.4	0.5	0.4	0.7	0.7	0.4	4.23	
8	0.5	0.4	0.5	0.4	0.7	0.7	0.4	4.85	
9	0.5	0.4	0.5	0.4	0.7	0.7	0.4	5.47	
10	0.6	0.9	0.5	0.5	0.3	0.6	0.9	2.905	Harder Task
11	0.6	0.9	0.5	0.5	0.3	0.6	0.9	4.06	
12	0.6	0.9	0.5	0.5	0.3	0.6	0.9	5.215	
13	0.7	0.5	0.5	0.5	0.4	0.7	0.5	2.55	Comfort Task
14	0.7	0.5	0.5	0.5	0.4	0.7	0.5	3.35	
15	0.7	0.5	0.5	0.5	0.4	0.7	0.5	4.15	
16	0.7	0.5	0.5	0.5	0.4	0.7	0.5	4.95	
17	0.7	0.5	0.5	0.5	0.4	0.7	0.5	5.75	
18	0.6	0.8	0.5	0.5	0.9	0.5	0.3	2.605	Harder Task
19	0.6	0.8	0.5	0.5	0.9	0.5	0.3	3.46	
20	0.6	0.8	0.5	0.5	0.9	0.5	0.3	4.315	
21	0.6	0.8	0.5	0.5	0.9	0.5	0.3	5.17	

Table 3 Job rotation model for worker A for 21 days

To calculate mental condition, seven parameters mentioned above need to be described. Parameters are expressed by different methods:

- Collect questionnaire or self report by workers
- Estimate task's descriptions by experts or decision makers

The job rotation problem is calculated for 21 work days and the results are shown in the last two columns in table 3. When mental condition of the worker A reaches the upper bound of optimum area which is 5, the job rotation takes place. But to which task group the worker A will be transferred depends on the current task group. This means, if the current task group is mentally harder, then the next group will be more mentally able to deal with worker's mental fatigue. However, if the current task group is mentally more comfortable, then the next task will be chosen from mentally harder task group to deal with worker's job boredom. In the table 3, four job rotations can be

3.5 Job assignment

The analytical hierarchical process (AHP) method, which is a multi-criterion decision making (MCDM) method, has been chosen for the job assignment in this research. There are three criteria and twelve sub-criteria for finding the ideal match task for each worker. As figure 8 shows there are three criteria and each one has some sub criteria.

I. Workers: The first criterion is workers. In this research, five sub-criteria are characterized to specify each worker.

- Success History: the indication of worker's job history is important for decision makers in manufacturing systems. With the records of all worker's job history, managers can find out whether each worker was often absent from work when he/she was scheduled, arrived late when he/she did show up or caused a great deal of problems with supervisors and co-workers. Therefore, rating the success of worker's job history can increase the accuracy of job assignment.

- Mental Ability: the ability to learn new skill or retain knowledge indicates how a worker can react in different workstations.
- Physical Ability: the worker's ability to work is one of main concerns for job planners.
- Mental Agility: this is another worker's ability, which indicates the ability of worker to engage the mental activities.
- Training Requirement: this sub-criterion indicates how much training a worker had in the past and also helps managers to specify how much training the worker requires for the future jobs.

II. Tasks: to describe tasks in a manufacturing environment, there are four features in this research.

- Task's Mentality Difficulty: describes how mentally difficult a task is.
- Task's Physicality Difficulty: this sub-criterion describes how physically difficult a task is.
- Cost Required for Training: managers always want to know the cost for required training of workers.
- Safety Risk: based on a task's description and history of accidents happened in that task, managers can take some action to prevent further health risks.

III. Environmental Constraints: in this research, three main environmental constraints are considered.

- Production Line Capacity: this sub-criterion describes the capacity of a production line, which is very important as in some scenarios, there are sequences for production. For example, to make a part, a capacity of a lathe is zero when the part needs to be drilled first.
- Production Scheduling: Production Scheduling is the process of generating to-do lists for a manufacturing system and it needs to be considered to show how much on-time delivery is important for the system.
- Production Setup time: the time required to prepare a machine, process or a system to be ready for production. By indicating this factor, decision makers can consider the time needed before production starts in a system.

The hierarchy for the planning problem is illustrated as follows (Figure 8):

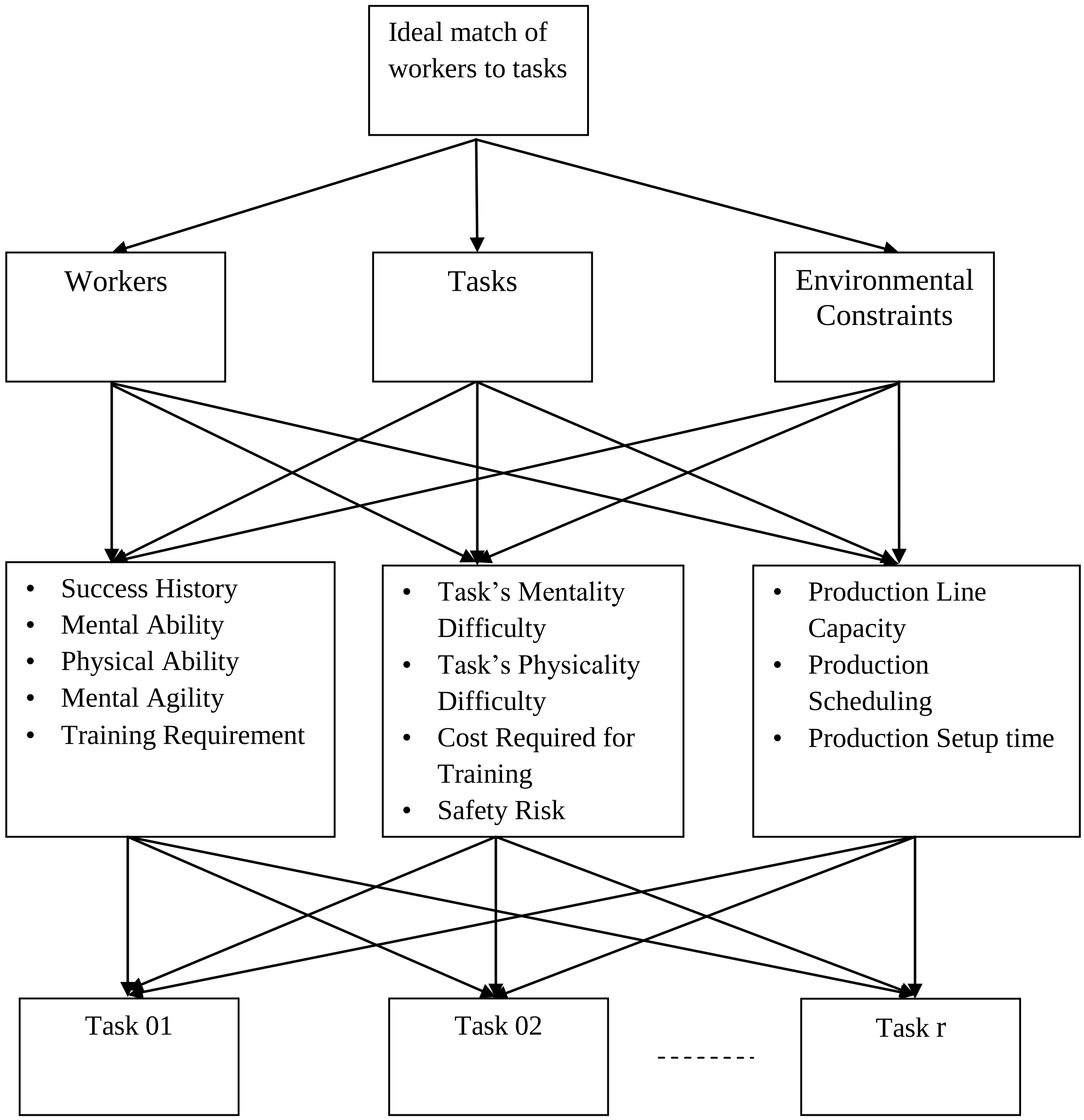

Figure 8 Hierarchy of job assignment components

Table 4 shows the numerical values used in the pair-wise comparisons which are modified from the 9-point scale table (suggested by Saaty).

Value	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective.
2	Moderate importance	Experience and judgement moderately favour one activity over another.
3	Essential or strong importance	Experience and judgment strongly favour one activity over another.
4	Very strong importance	An activity is strongly favoured and its dominance demonstrated in practice.
5	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation.

Table 4 Pair-wise comparison scale modified based on Saaty's 9 point scales

3.5.1 AHP example

To demonstrate the process for overall evaluation of alternatives, one example with ten alternatives (divided into two different tasks groups) and twelve criteria has been selected. First, the comparison matrix is constructed to determine the weights of sub-criteria. Tables 5 to 9 present the criteria comparison data for the job assignment. In these tables, the evaluations of each criterion compared with other criteria are used to calculate weights. To normalize, the total of the column is calculated and then each cell value is divided by this total. This will give the normalized matrix. A simple average will give the row average.

Table 5 shows the pair-wise comparison results for job assignment.

Criteria	Workers	Tasks	Environmental Constraints	Row Average
Workers	1	1.666	1	0.3846
Tasks	0.6	1	0.6	0.2307
Environmental Constraints	1	1.666	1	0.3846
Total	2.6	4.332	2.6	

Table 5 Comparing each criterion with other criteria

Table 6 shows the pair-wise comparison results of the adaptation options for job assignment with respect to the Workers weights.

Workers	Success History	Mental Ability	Physical Ability	Mental Agility	Training Requirement	**Row Average**
Success History	1	0.333	0.333	0.333	1.5	0.0939
Mental Ability	3	1	1	1	4	0.2750
Physical Ability	3	1	1	1	4	0.2750
Mental Agility	3	1	1	1	4	0.2750
Training Requirement	0.666	0.333	0.333	0.333	1	0.0807
Total	10.666	3.666	3.666	3.666	14.5	

Table 6 Evaluating adaptation options with respect to workers criteria

Tables 7 and 8 show the pair-wise comparison results of the adaptation options for job assignment with respect to the Tasks dimensions. The first one demonstrates harder tasks group weights (Table 7).

Tasks	Task's Mentality Difficulty	Task's Mentality Difficulty	Cost Required for Training	Safety Risk	Row Average
Task's Mentality Difficulty	1	4	3	3	0.4794
Task's Mentality Difficulty	0.5	1	3	3	0.2893
Cost Required for Training	0.333	0.333	1	1	0.1156
Safety Risk	0.333	0.333	1	1	0.1156
Total	2.166	4.332	8	8	

Table 7 Evaluating adaptation options with respect to tasks (harder tasks group) criteria

Table 8 summarizes the comfortable task group weights.

Tasks	Task's Mentality Difficulty	Task's Physically Difficulty	Cost Required for Training	Safety Risk	Row Average
Task's Mentality Difficulty	1	4	3	3	0.4794
Task's Physically Difficulty	0.5	1	3	3	0.2893
Cost Required for Training	0.333	0.333	1	1	0.1156
Safety Risk	0.333	0.333	1	1	0.1156
Total	2.166	4.332	8	8	

Table 8 Evaluating adaptation options with respect to tasks (comfortable task group) criteria

Table 9 shows the pair-wise comparison results of the adaptation options for job assignment with respect to the Environmental Constraints weights.

Environmental Constraints	Production Line Capacity	Production Scheduling	Production Line Setup time	Row Average
Production Line Capacity	1	1	2	0.4
Production Scheduling	1	1	2	0.4
Production Line Setup time	1	0.5	1	0.2
Total	2.5	2.5	5	

Table 9 Evaluating adaptation options with respect to environmental constraints

To calculate the overall weight to assign tasks to workers, multiplying the sub-criterion weights by the criterion weights provides overall weights for all twelve criteria. For example, the overall weight of "Production line capacity" is calculated as follows:

Environmental Constraints' Row Average × Production Line Capacity's Row Average:

(03846)(0.4) = **0.1538**

Repeating the same calculation for the criteria gives the following results (Table 10):

Criteria	**Overall Weights**
Production Line Capacity	0.1538
Production Scheduling	0.1538
Production Line Setup time	0.0769
Task's Mentality Difficulty	0.1106
Task's Physically Difficulty	0.0667
Cost Required for Training	0.0266
Safety Risk	0.0266
Success History	0.0361
Mental Ability	0.1058
Physical Ability	0.1058
Mental Agility	0.1058
Training Requirement	0.0310

Table 10 Overall weights of all sub criteria

After setting up the job assignment evaluation problem and determining the priority vectors for the comparison criteria, the next step in the analytic hierarchy process is to find utilities of each task based on each sub-criterion. There are different methods to rank alternatives (tasks) such as making a new comparison matrix or using direct utility collected from experts' decisions. The process to create new comparison matrixes for alternatives is the same as creating comparison matrixes for criteria and sub-criteria. Based on the definition of the problem and decreasing complexity in programming with VB, utilities obtained by experts' decisions are taking place. It means an expert can specify the utilities directly from the program interface. The next table (Table 11) shows the weights for one specific worker (worker A) which has specific characters in a manufacturing system.

Criteria	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**
Production Line Capacity	0.6	1	0.9	0	1
Production Scheduling	0.4	1	0.8	0	0.2
Production Line Setup time	0	1	0.5	0	0.3
Task's Mentality Difficulty	0.9	0.9	0.8	0.7	0.8
Task's Physically Difficulty	0.6	0.5	0.5	0.5	0.6
Cost Required for Training	0.5	0.4	0.3	0.5	0.5
Safety Risk	0.8	0.3	0.5	1	0.7
Success History	0.9	0.9	0.9	0.9	0.9
Mental Ability	0.5	0.5	0.5	0.5	0.5
Physical Ability	0.8	0.8	0.8	0.8	0.8
Mental Agility	0.5	0.5	0.5	0.5	0.5
Training Requirement	0.6	0.6	0.0310	0.6	0.6

Table 11 Utilities of worker A for comfortable task group with five tasks (Alternatives)

By multiplying weights calculated by pair-wise comparison matrixes and utility functions, and then normalizing them, the ideal match tasks for workers can be determined. For example to find the score of task 1, the calculation is as follows:

The score of task 1= Overall Weight of sub-criterion x *utility of task 1* (14)

The score of task 1=

0.1538x0.6+1538x0.4+0.0769x0+0.1106x0.9+0.0668x0.6+0.0266x0.5+0.0266x0.8+0.0361x0.9+0.1058x0.5+0.1058x0.8+0.1058x0.5+0.0310x0.6=0.5697

In table 12, the priorities of the example are illustrated.

Criteria	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**
Score	0.5697	0.7778	0.6848	0.3924	0. 6098
Normalizing the score	0.1877	0.2563	0.2256	0.1293	0.2009
Priorities	4	1	2	5	3

Table 12 Final Results

The final results show that the ideal match for worker A is task 2.

Generally, AHP has a number of advantages that are relevant for this research. It is a very useful method to describe to managers and job planners the collection of pair-wise comparison data, especially in subjective cases.

In the next chapter, the job scheduling program will be discussed. The program has been coded by visual basic language (VB.Net) due to the following considerations:

1- The software is free for students and for developers.

2- VB.Net has a rich user interface features available in the Windows operating system.

3- Compared to other available software, VB.Net is easy to learn.

3.6 Job scheduling

The goal for the job scheduling problem in this research is to simultaneously find the ideal task match for workers, and maintain workers' mental conditions at optimum level with the proposed job rotation model. This problem consists of two models: job rotation model and job assignment model.

To create the job rotation model, two formulas have been presented to measure job boredom and mental fatigue. Based on solutions proposed by psychologists and human behaviour researchers, the model is presented in order to restrict worker's mental conditions from unproductive mental states. The rotation model transfers workers when their mental fatigue and boredom exceed the optimum level of workers' mental condition, to new tasks.

The job assignment problem is formulated with the AHP method. The model first finds the best possible match task for each worker when he/she is ready to be transferred to the new task according to the job rotation model.

The job scheduling method proposed in this research is programmed by VB.net which helps to provide a verity of decisions by managers without knowledge of complicated methods.

3.6.1 Job scheduling example

To schedule a job for worker A, tasks and time when he/she will be transferred need to be specified (Figure 9).

	Mentally Harder Task Group					Mentally Comforter Task Group					
	Options(Task No.)					Options(Task No.)					Working days on a task
	H1	H2	H3	H4	H5	C1	C2	C3	C4	C5	
1st Job Assignment	4	1	2	5	4	2	1	4	3	5	3
1st Job Rotation	2	5	3	1	5	1	5	3	4	2	6
2nd Job Rotation	1	5	2	4	3	3	5	4	1	2	3
3rd Job Rotation	2	1	4	3	6	3	5	4	1	2	5

Figure 9 Job scheduling example for worker A with ten different tasks

Figure 9 shows a job schedule for worker A in 17 days. He/she has enough skills to work in ten different tasks. Ten tasks are divided into two groups: mentally harder (H1, .., H5) and easier task (C1, ...,C5) groups. The program first assigns the worker to his/her ideal match task (H2). Then after three days when the worker's mental condition exceeds the previously intended boundaries, he/she is transferred to new task (C1). After six and then three days, the worker will be transferred to tasks H1 and then C4. The last task is C4 and the worker will work there for five days after he/she is transferred to a new task. In the real world, other factors needed to be consider during job scheduling. For example, tasks may not be available each time worker A is required to be transferred to a new task. Or, in the case of some production plans, some tasks need to be done consecutively. One of the reasons that VB.net has been chosen to program this job scheduling model is that all features mentioned above can be added. The ability to be modified helps researchers to decrease the gap between theory and practice in scheduling.

Chapter 4

Job Scheduling Programming for Manufacturing Systems using Visual Basic.Net

Job scheduling is a time consuming and complex process. As there is no job scheduling program in industries which help to deal with job boredom and mental fatigue, this program can support the decision making in job scheduling process while considering job boredom and mental fatigue. Furthermore, this program integrates the AHP method with the job rotation model in a single environment.

4.1 Problem Description

The purpose of this new job scheduling program is to assign a best possible task to each worker and increase job productivity when dealing with the following mental conditions:

- Boredom
- Mental Fatigue

This program combines a job rotation model with a job assignment model for job scheduling in manufacturing systems.

Job Rotation: The formulas are developed to quantify the job boredom and mental fatigue. These formulas will be used to provide input to rotate jobs and mitigate the mental burdens.

$$MC = [(am \cdot cm + \beta \cdot \mathrm{ap} \cdot cp) + O \cdot cm]Time \qquad (15)$$

In addition, the AHP prioritized results will be used to assign jobs to workers.

4.2.1 Visual Basic

According to Visual Basic Developer Center (web site), Visual Basic 2010 is an evolution of the Visual Basic language that is engineered for object-oriented applications. Visual Basic enables developers to target Windows, Web, and mobile devices.

4.2.2 Visual Studio and the .Net Framework

Visual Studio is the Integrated Development Environment (IDE) in which developers create programs in one of many languages, including Visual Basic for the .Net Framework. The .Net Framework is a development and execution environment that allows different programming languages and libraries to work together seamlessly to create Windows, Web, Mobile, and Office applications. Microsoft implementation of Visual Basic .Net is called “Microsoft Visual Basic”.

There are two versions of Visual Basic:

1- Microsoft Visual Studio, which is commercial software, and

2- Microsoft Visual Studio Express, which is free of charge.

In this research, Microsoft Visual Studio Express has been used to write a program for job scheduling.

There are some benefits for this version of VB:

1- More inviting start page: Develop an application more quickly with friendlier and more useful start options.

2- Streamlined user experience: Focus on the most common commands by choosing to hide more advanced menus and toolbars.

3- A new WPF-based IDE: The new code editor makes it easy to zoom in on the code, highlight method references, and overlay powerful features.

4.2.3 System and Hardware Requirements

4.2.3.1 System

Visual Studio 2010 can be installed on the following operating systems:

- Windows XP (x86) with Service Pack 3 - all editions except Starter Edition
- Windows Vista (x86 & x64) with Service Pack 2 - all editions except Starter Edition
- Windows 7 (x86 & x64)
- Windows Server 2003 (x86 & x64) with Service Pack 2
- Windows Server 2003 R2 (x86 & x64)
- Windows Server 2008 (x86 & x64) with Service Pack 2
- Windows Server 2008 R2 (x64)

Supported Architectures:

- 32-Bit (x86)

- 64-Bit (x64)

4.2.3.2 Hardware

- Computer that has a 1.6GHz or faster processor
- 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine)
- 3GB of available hard disk space
- 5400 RPM hard disk drive
- DirectX 9 capable video card running at 1024 × 768 or higher-resolution display

4.3 Programming

The program starts with asking to input data. In the first page, it asks for total number of workers, the task period for job scheduling, duration of scheduling, the percentage of influence of worker's physical condition on mental condition, and at the end: maximum and minimum level of mental conditions, which will specify the acceptable boundaries of workers' mental conditions. In the second page it asks for all weights that the AHP model will need, in order to prioritize tasks for workers.

The next six pages ask questions about workers, tasks, and environmental constraints. These data will be used in both job rotation and job assignment models. The next step is focused on dividing all tasks into two mentally harder and mentally easier task groups. After that, the program allocates all workers into those two tasks groups. Then the program allocates the best possible task match to each worker using the AHP method.

In the next step, each worker's mental condition (MC) is calculated for every day. Then it checks to see if these MC parameters are between $MC\ min \leq MC \leq MC\ max$ limits. If the answer is yes, this checking continues for the next day until the answer is no. With the no answer, the program checks if there is an available task for the next rotation.

If there is an available task for the next rotation, the program transfers the worker with out of rang mental condition into an available task but in a different task group. For example, if the worker's previous task was in the easier task group according to the proposed rotation model, the worker's current task will be from the harder task group. After the worker has been transferred to the new task, the program continues the same process for each worker during the scheduled time.

If there is no task available for the worker for his/her next move, the worker waits for the next day to see if a task is available for him/her. This waiting process can be done only for three days. After that, if there is still no available task, the program transfers the worker with low mental condition and productivity to the best possible, already occupied task (different task group), in order to prevent further productivity loss during his/her previous task (with low mental condition).

This process continues for each worker until the program has scheduled all workers. Then the result can be shown separately or collectively. The result page contains worker's number, the amount of rotation scheduled for the worker and the task number specified for the worker. In addition, the diagram illustrates how and when the workers are scheduled. The out-of-range mental conditions that causes production loss are revealed by red dots.

This program is able to include or remove tasks for specific periods of time. This means that if there is a sequence for a group of tasks, the program can assign tasks whenever they are ready to be executed, one by one. The flowchart of the job scheduling program is shown in figure 10.

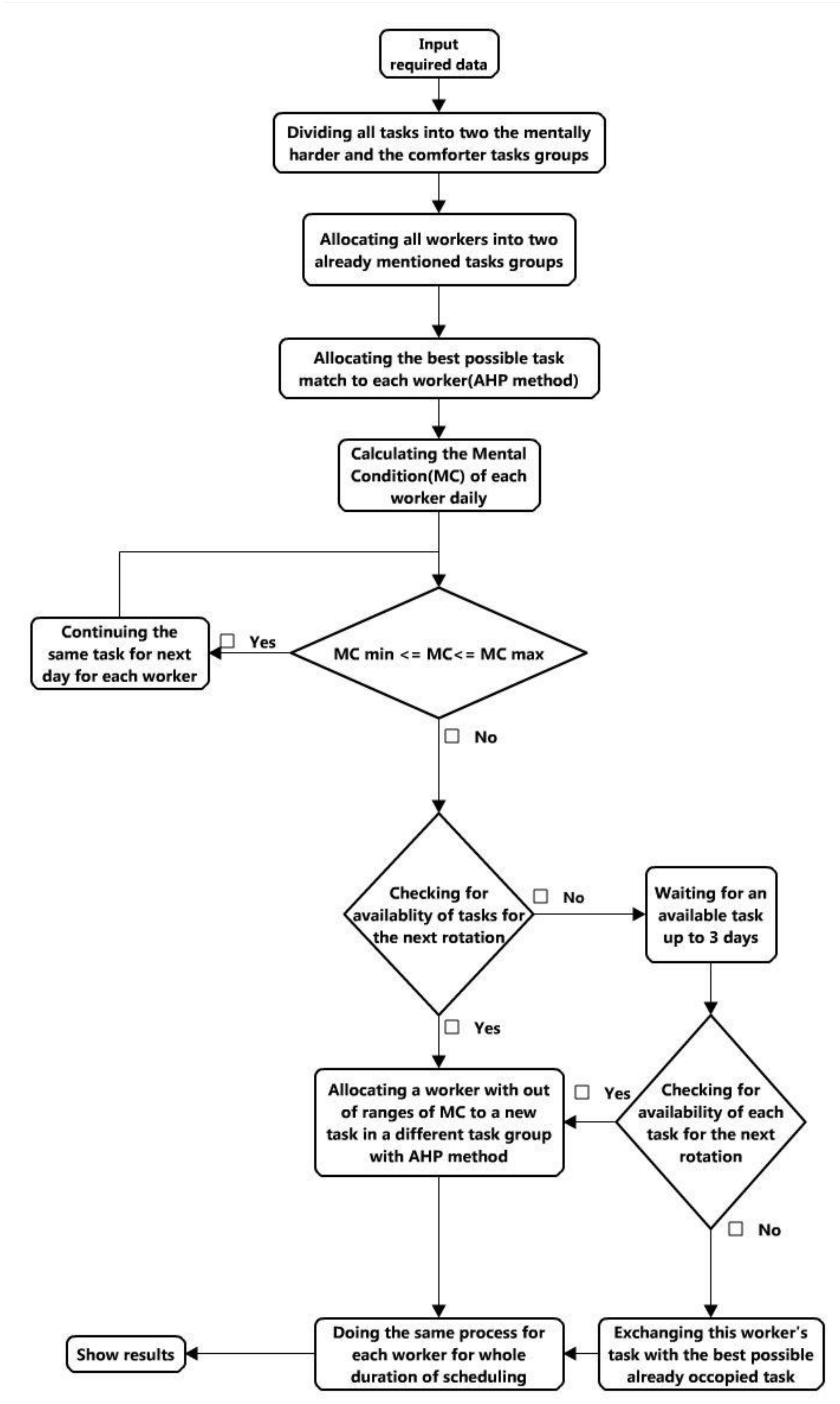

Figure 10 The job scheduling program flowchart

4.4 Assumptions

I. In the real world, the mental ability (*am*) and physical ability (*ap*) of workers vary and depend on some other factors such as physical, mental health and time. As the studies of these abilities have not been done in this research, *am* and *ap* are assumed to be nearly fixed but slightly decreased by doing easier tasks and increased by performing harder tasks.

II. All workers, before assigning tasks, must meet the same task skill requirements or otherwise the manager will ask them to acquire the skills.

III. As the causes of mental fatigue and job boredom in this research are assumed to be dependant only on tasks and workers' conditions, mental fatigue and job boredom increase linearly over time if their conditions do not change.

4.5 Impetus for applications

There following reasons are impetus for application of the program to job scheduling:

- The accessible literature suggests that there is no job scheduling program in industries that deals with job boredom and mental fatigue.
- Job scheduling is a time consuming and complex process. This program can help to support the decision making in the job scheduling process.
- The entire decision process, including mental condition quantification, job rotation, AHP, and scheduling has been implemented using this program in a single environment.

A case study will be presented in this thesis to illustrate the applicability of the methodology to a real world situation.

Chapter 5

Case Study

5.1 Problem description

The case study is based on a car manufacturing company. The KIA metal body assembly production line has a 200,000ft^2 manufacturing area (Figure 11) with three round shifts and about 160 workers per shift. From the beginning of production, one shift started its daily production with 75% direct labour performance efficiency, 14% rework and a high rate of accidents (major and minor injuries). After six months of improvement in all aspects, the situation changed significantly to 95% direct labour performance efficiency , 4% rework rate, and low rate of accident. Most of the reworks are activities that are caused by human errors. There were several reasons for human errors but a key reason was workers' mental condition during working hours, especially in night shifts. Job boredom and mental fatigue are most commonly reported as reasons for errors, caused by workers' mental condition. Boredom and fatigue can decrease direct labour performance efficiency, and increase workers' errors and accidents.

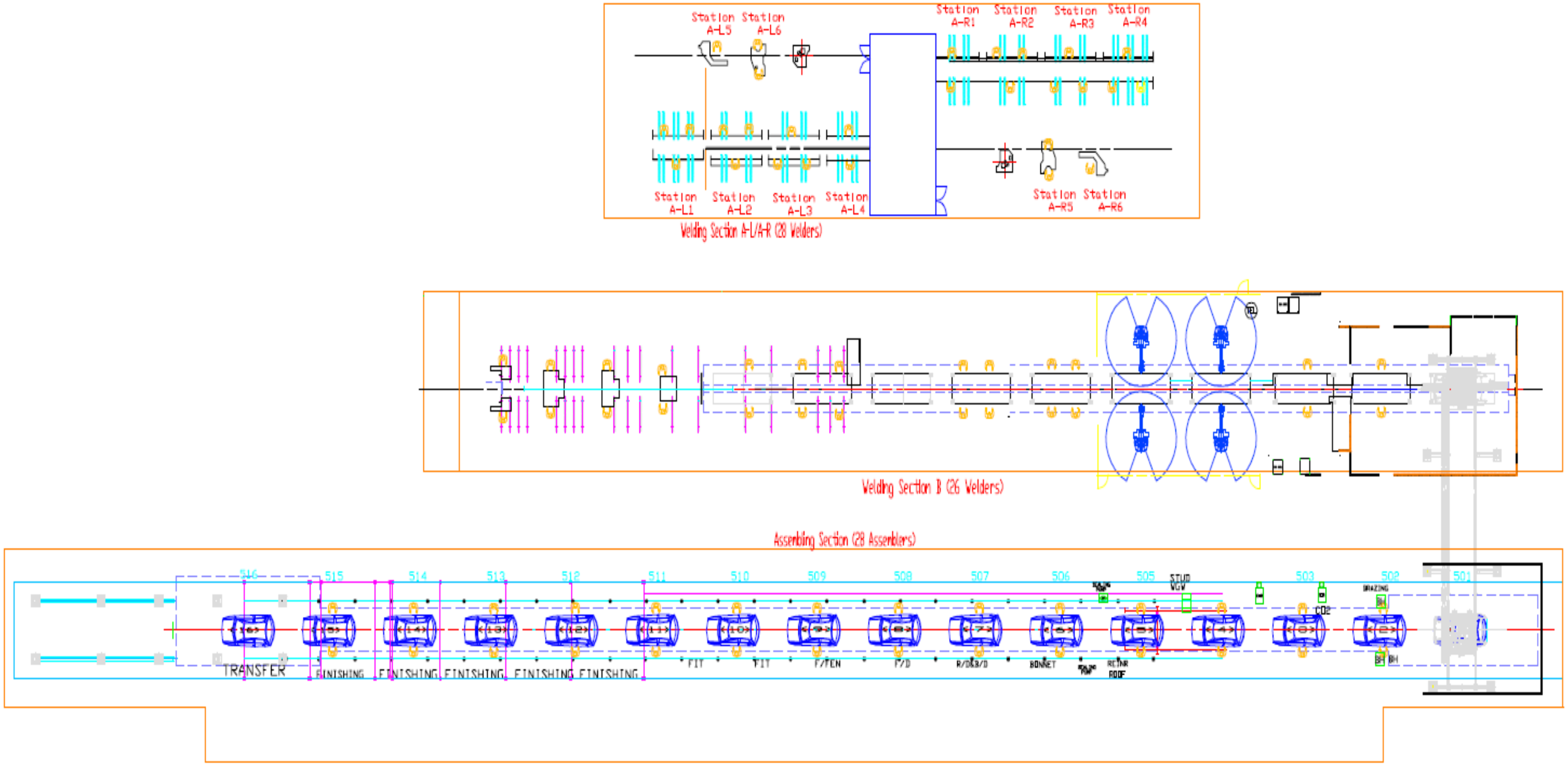

Figure 11 The car assembly line-Metal part

5.2 Solution

The scheduling program presents a job schedule for a section of the assembly line and shows how it affects some important outputs of the production line. Figure 12 illustrates the Sides assembly line that consists of welding machines, fixtures and workers.

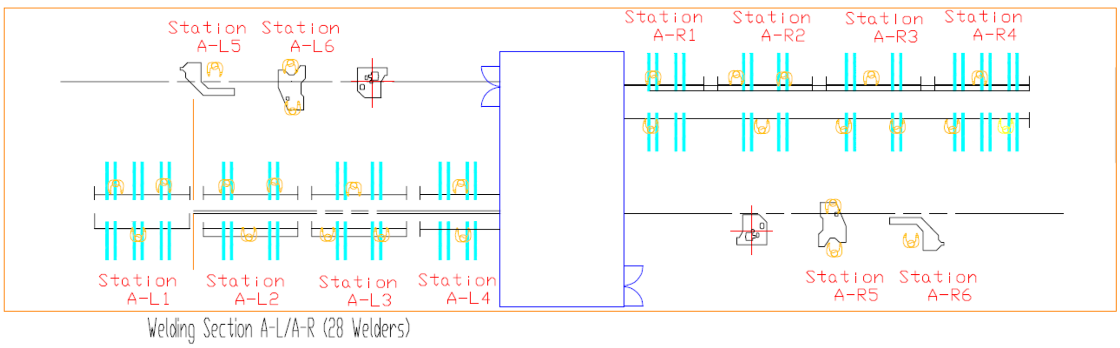

Figure 12 Welding Section A-L/A-R, Sides Assembly Line

There are four steps to input data to the program:

Management: in the assembly line, 28 welders work in 12 workstations and there are 28 different tasks. The duration of job scheduling is 60 days in three periods. It means there are twelve weeks divided by three periods and workers come to work five days a week. The maximum and minimum levels of mental condition specify the upper and lower levels of optimal mental condition for each worker. These two numbers have been created to demonstrate the proper graphs for daily scheduling. They need to be revised if the job schedule needs to be calculated hourly or in other time units. The last data specify how much managers recognize the influence of worker's physical condition on their mental conditions.

Table 13 shows the number of workers in each station for the related tasks.

Station No.	*Number of required workers*	*Task No.*
A-L1	3	1,2,3
A-L2	3	4,5,6,
A-L3	3	7,8,9
A-L4	2	10,11
A-L5	1	12
A-L6	2	13,14
A-R1	3	15,16,17
A-R2	3	18,19,20
A-R3	3	21,22,23
A-R4	2	24,25
A-R5	1	26
A-R6	2	27,28

Table 13 Tasks descriptions

Figure 13 demonstrates the first page of the program. In this page, seven data needed to be specified.

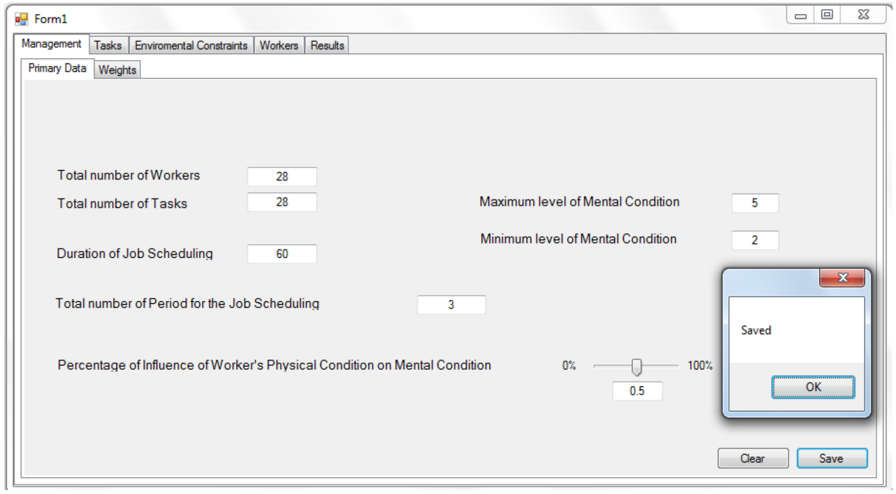

Figure 13 Graphical user interface "Primary data"

The managers then decide how priorities need to be considered. In this study a production manager decides that workers and environmental constraint are important equally and more important than task. The user interface for assigning weights is shown in Figure 14.

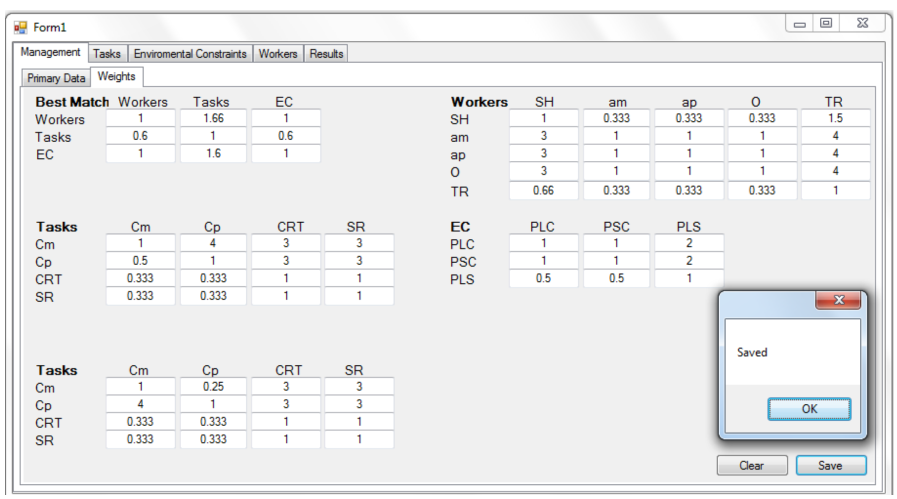

Figure 14 Graphical user interface "weights"

Tasks (Figure 15): The welding process seems to be the same as a car production line. However, the descriptions and the difficulty of tasks depend on the location of welding spot, the welding machine, number of welding spots and accuracy needed for welding. The workers are well trained in car manufacturing companies and training is very costly. Moreover, each task has its own safety risk. Next, two data specify the cost of training and the safety risk of each task.

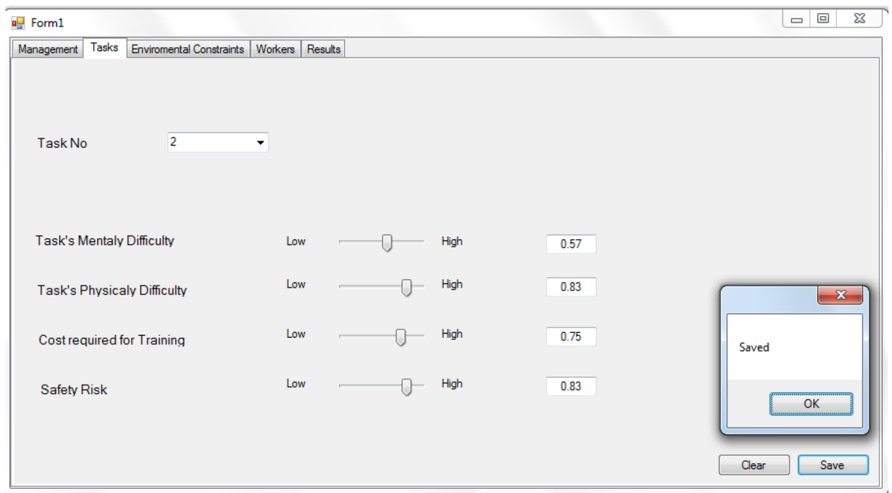

Figure 15 Graphical user interface "task"

Environmental constraints (Figure 16): to schedule jobs in a production plant, some constraints must be considered for tasks in each period of scheduling. For this case study, the production line capacity is the highest. In addition, there is no conflict in production scheduling and there is no need for setup time as the automated production line does not stop for shift changing.

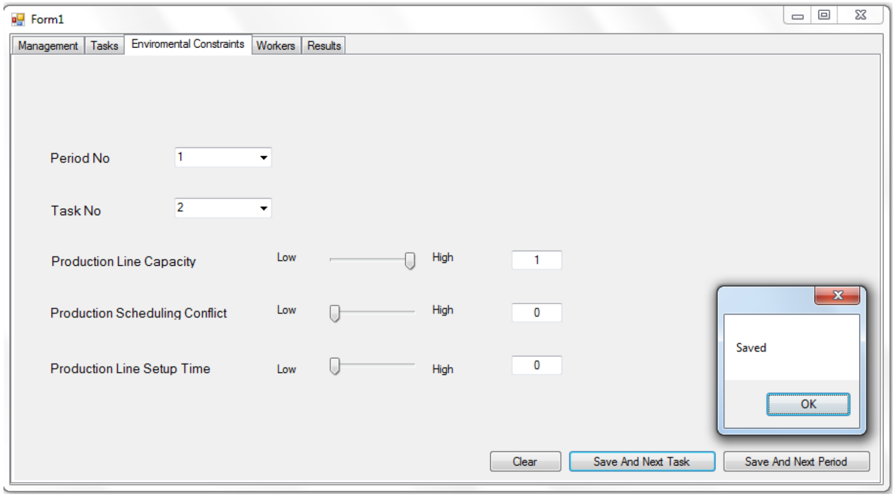

Figure 16 Graphical user interface "environmental contraints"

Workers (Figure 17, 18, 19, 20): There are some important factors such as training history, mental ability, physical ability and mental agility that are obtained by questionnaires for each worker. As accessing to workers for interviewing or filling questionnaires was not possible for this research, these human factors were assumed. The figure 17 collects success history and training required for each worker.

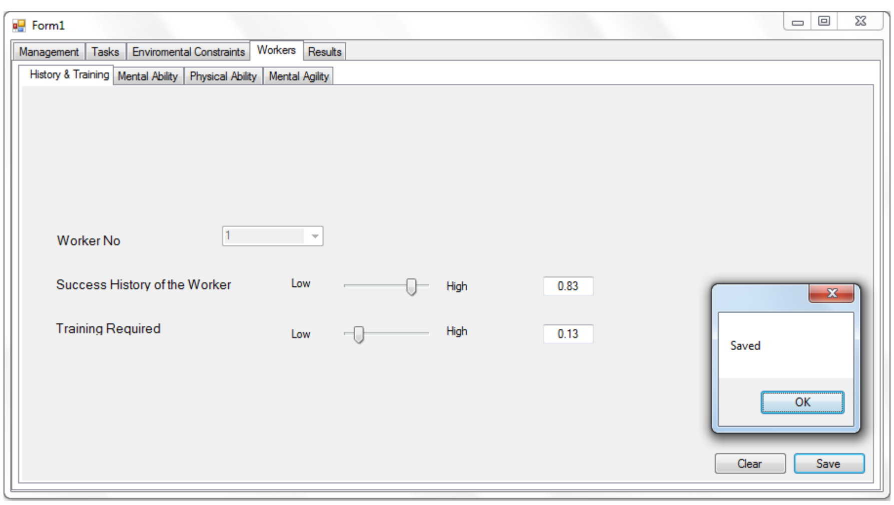

Figure 17 Graphical user interface "worker-History & Training"

In Figure 18, the mental ability of each worker can be specified based on the answers to the five questions.

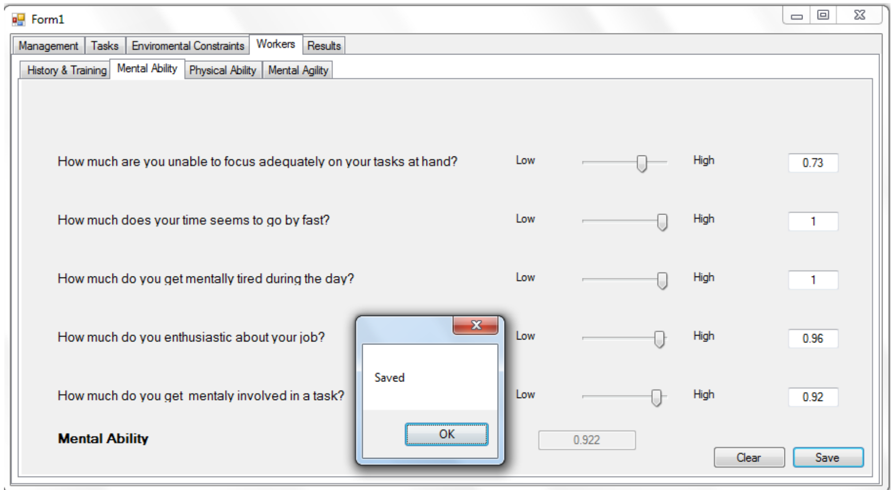

Figure 18 Graphical user interface "worker-Mental ability"

In Figure 19, the physical ability of each worker can be specified based on the answers to another five questions.

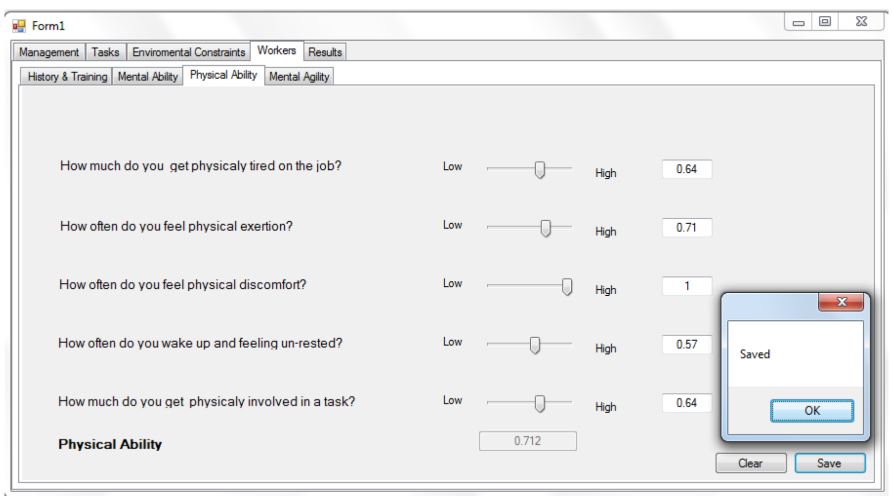

Figure 19 Graphical user interface "worker-Physical ability"

The interface shown in Figure 20 illustrates that how the mental agility of each worker is specified by the answers to the five questions.

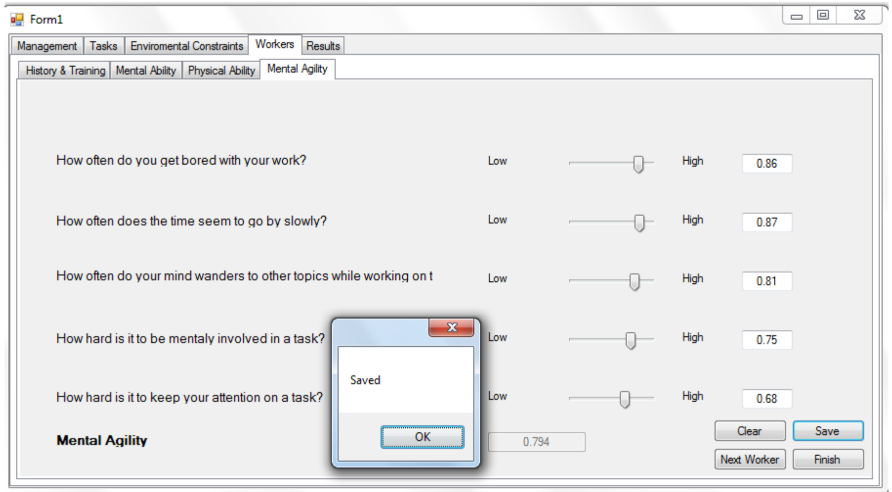

Figure 20 Graphical user interface "worker-Mental agilty"

Results: in this final page (Figures 21 and 22) the result is calculated and presented two diffrent ways. First, as it's shown in Figure 21, the result provides the amount of rotation, names of tasks and numnbers of each tasks that are allocated to each worker.

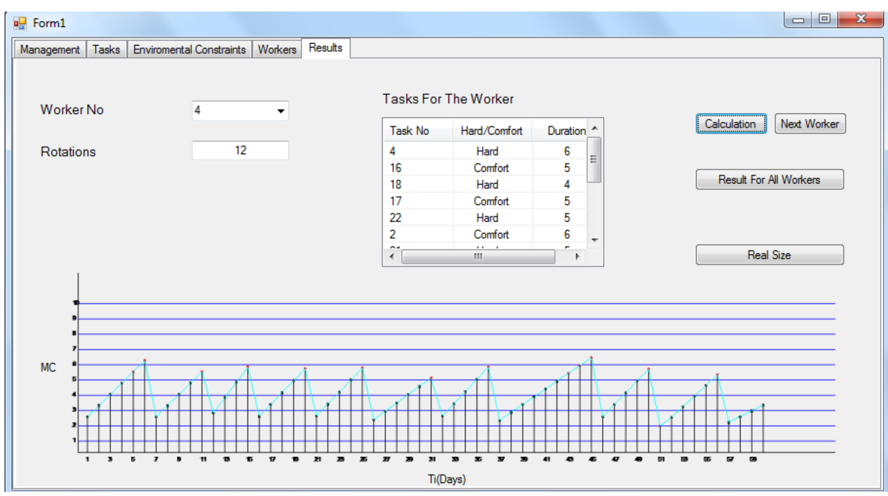

Figure 21 Graphical user interface "Results"

Figure 22 shows the graph (MC, Days) in a seprate page and is easy to restore.

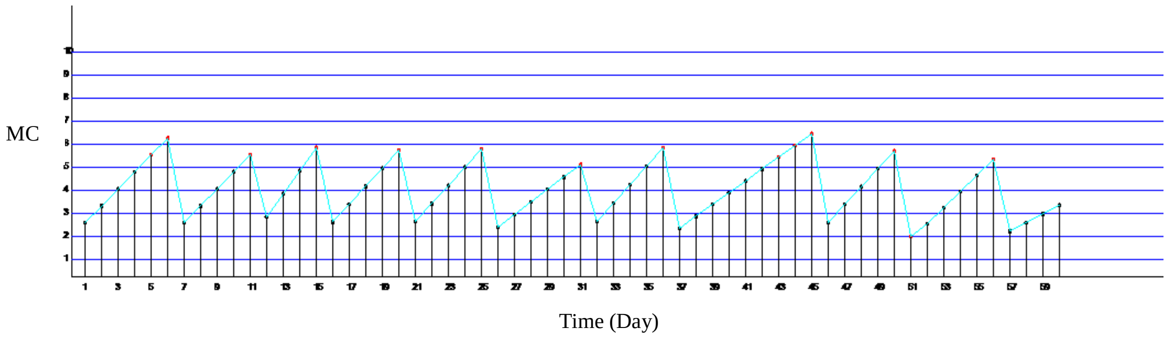

Figure 22 Graphical user interface "Results/Graph"

Table 14 summarizes all results for all workers.

Worker ID	*Rotation No*	*Task [Rotation]*
1	14	1 [5], 18 [2], 24 [5], 3 [5], 15 [3], 4 [3], 23 [7], 22 [3], 27 [9], 21 [3], 16 [4], 9 [4], 16 [4], 18 [3],
2	11	2 [5], 19 [5], 26 [9], 19 [3], 2 [5], 13 [4], 23 [9], 21 [4], 28 [6], 21 [4], 2 [6],
3	12	3 [6], 6 [13], 7 [4], 17 [4], 18 [3], 1 [6], 21 [4], 23 [8], 4 [6], 24 [6],
4	12	4 [6], 16 [5], 18 [4], 17 [5], 22 [5], 2 [6], 21 [5], 1 [9], 7 [5], 5 [1], 22 [5], 17 [4],
5	12	5 [5], 20 [5], 17 [4], 10 [4], 16 [5], 9 [4], 14 [5], 9 [4], 14 [6], 4 [5], 2 [7], 21 [6],
6	12	6 [6], 3 [6], 24 [6], 18 [5], 28 [5], 19 [4], 26 [13], 20 [4], 6 [2], 3 [7], 25 [1], 4 [1],
7	11	7 [6], 17 [4], 20 [3], 23 [10], 4 [6], 16 [6], 3 [4], 16 [6], 10 [4], 14 [6], 20 [5],
8	11	8 [9], 2 [5], 13 [5], 2 [6], 22 [5], 23 [1], 13 [5], 17 [6], 9 [7], 27 [4], 7 [7],
9	14	9 [6], 14 [4], 10 [4], 17 [1], 18 [3], 24 [9], 10 [3], 17 [6], 7 [4], 16 [5], 21 [4], 24 [5], 4 [5], 25 [1],
10	11	10 [6], 15 [6], 4 [4], 16 [4], 20 [3], 25 [11], 18 [4], 28 [6], 21 [6], 17 [5], 9 [5],
11	13	11 [6], 16 [6], 7 [4], 17 [7], 7 [3], 17 [4], 20 [3], 28 [5], 18 [4], 17 [5], 22 [4], 5 [4], 20 [5],
12	12	12 [6], 17 [6], 21 [4], 28 [7], 18 [4], 17 [3], 10 [4], 16 [5], 3 [5], 17 [7], 18 [8], 5 [1],
13	11	13 [6], 28 [6], 22 [4], 5 [3], 21 [5], 15 [7], 21 [7], 6 [11], 20 [6], 5 [1], 4 [4],
14	12	14 [6], 9 [3], 27 [10], 9 [4], 16 [4], 3 [4], 6 [7], 21 [4], 27 [7], 9 [6], 27 [1], 21 [4],
15	10	15 [6], 10 [4], 14 [4], 19 [5], 26 [13], 19 [6], 2 [8], 18 [5], 1 [6], 3 [3],
16	12	16 [6], 4 [6], 15 [5], 3 [3], 1 [5], 21 [6], 2 [7], 4 [4], 5 [8], 7 [3], 27 [2], 9 [5],
17	13	17 [6], 7 [6], 16 [4], 4 [4], 15 [4], 21 [1], 1 [5], 18 [4], 25 [4], 4 [4], 14 [7], 10 [6], 23 [5],
18	12	18 [5], 1 [3], 21 [4], 17 [4], 7 [3], 6 [12], 3 [4], 16 [5], 7 [5], 1 [6], 18 [6], 16 [3],
19	12	19 [5], 2 [4], 22 [3], 28 [4], 21 [3], 5 [12], 7 [5], 17 [6], 18 [4], 2 [5], 3 [6], 1 [3],
20	10	20 [5], 5 [11], 22 [4], 17 [6], 7 [5], 5 [11], 4 [6], 23 [7], 10 [4], 27 [1],
21	9	21 [8], 1 [5], 9 [6], 27 [14], 22 [5], 2 [9], 4 [9], 5 [1], 21 [3],
22	7	22 [9], 2 [10], 13 [8], 2 [11], 22 [9], 17 [9], 22 [4],
23	6	23 [13], 20 [7], 16 [9], 4 [9], 25 [20], 3 [2],
24	12	24 [7], 18 [4], 16 [7], 10 [3], 14 [6], 9 [5], 14 [4], 9 [6], 17 [8], 21 [7], 5 [2], 18 [1],
25	10	25 [9], 13 [5], 2 [8], 19 [6], 28 [5], 20 [5], 24 [11], 21 [7], 27 [3], 10 [1],
26	7	26 [10], 19 [4], 14 [7], 10 [6], 24 [11], 20 [7], 26 [15],
27	9	27 [9], 9 [4], 1 [7], 3 [7], 16 [7], 10 [4], 17 [6], 3 [7], 6 [9],
28	8	28 [6], 13 [3], 25 [14], 20 [7], 17 [8], 10 [7], 16 [8], 9 [7],

Table 14 Graphical user interface "All Results"

For the future research, it is suggested that the workers' productivity be measured before and after the job scheduling takes place and compare them to find how efficient this program is. The proposed job rotation method implemented by author in a car manufacturing company for some workers with bad outputs in a very limited situation. The results showed that significant improvements were made in terms of workers' productivity and fewer injuries in the production line. With implementing this job-scheduling program, job rotations (to reduce boredom and mental fatigue) and job assignment (to find ideal match task for each worker) can be done in a single environment.

Finally, it must be mentioned that all of the figures, drawings and the data regarding this case study have been designed or obtained by the author of the thesis during his work as a chief of assembly line and head of improving productivity committee at the Pars Khodro car manufacturing company, Tehran, Iran from 2005 to 2007.

Chapter 6.

Conclusion and future work

6.1 Conclusion

The economic reality is motivating many companies to consider workers' health and productivity management as an integral part of their business strategy. It means employers are becoming increasingly aware that a worker's health condition influences his/her productivity, and that productivity has a positive impact on organizational performance and competitiveness. Human mental conditions such as job boredom and mental fatigue are important to productivity and safety. However, they have often been overlooked in manufacturing planning and job scheduling because of the complexity in analytical analysis. Therefore, previous planning and/or scheduling results may not be implemented in real world situations. In addition, the planning and scheduling decisions obtained in such a way may further worsen job boredom and mental fatigue as their scheduling has limitations and assumptions that cannot address human conditions properly. For the above reasons, schedule jobs in consideration of human mental conditions have been proposed in this research.

Three different methods have been used in literatures to measure mental fatigue and job boredom: psychological and physiological methods, questionnaire and mathematical methods. In this research, the mathematical formulas are developed to quantify the job boredom and mental fatigue. These formulas have been used to provide input to rotate jobs to moderate the mental burdens. Worker's Mental Condition (MC) has been introduced in this research by combination of two most important and affective human factors, i.e., job boredom and mental fatigue factors. After calculating each worker's mental condition every day, if the MC of a worker is out of boundaries the worker needs to move to a new task. The purpose of this rotation is to keep MC at optimum level and hence maintain the productivity of the worker at the high level.

Corresponding to this, an analytical hierarchical process (AHP) is proposed to prioritize jobs based on multiple factors related to tasks, workers and environmental constraints. The AHP prioritized results will be used to assign jobs to workers. The AHP method is preferred as it is interactive, easy and has a potential to use by managers in real world.

Then the scheduling is done based on the job rotation and assignment results. The entire decision process including mental condition quantification, job rotation, AHP and scheduling has been implemented using visual basic.Net.

6.2 Future work

- The worker's productivity can be calculated in the diagram for each worker and compare it with other methods.
- Use more criteria for AHP method in job assignment.
- Other rotation models can be added to the proposed job rotation model for more comprehensive job scheduling.
- Job flow scheduling and/or production scheduling can be added to the program.
- The main purpose of this research is to increase the quality of workers' performances during work hours and to increase workers' mental health, which causes their life quality after work hours. In addition, minimizing the production time (i.e. makespan) can be considered for job scheduling.

References

1. American College of Occupational and Environmental Medicine Integrated Benefits Institute (ACOEM/IBI). Improving the Health and Productivity of the American Workforce: A Blueprint for Action, 2008, Santa Ana Pueblo, NM: Workforce Health and Productivity Summit.
2. Al-Subhi, A.K., Al-Sinan, M. and Shokri, Z.S., 1997, A multi-objective linear program for scheduling repetitive projects, International Journal of Industrial Ergonomics, 60/61, 17–26.
3. Anselmi, F. and Sundarrajan S., 2000, Aligning workers and workloads, International Journal of Industrial Ergonomics, 28-32.
4. Ash P. and Goldstein S.I., 1995, Predictors of returning to work. Bulletin of the American Academy of Psychiatry and Law, 23, 205–210.
5. Baker, K.R., 1974, Introduction to Sequencing and Scheduling, John Wiley, New York: Wiley and Sons.
6. Bhadury, J. and Radovilsky, Z., 2006, Job rotation using the multi-period assignment model, International Journal of Production Research, 44, 4431-4444.
7. Biron C., 2006, At work but ill: psychosocial work environment and well-being determinants of presenteeism propensity, Journal of Public Mental Health, 5, 26-38.
8. Boksem, M., Meijman, T.F. and Lorist, M.M., 2005, Effects of mental fatigue on attention: An ERP study, Cognitive Brain Research, 25, 107–116.
9. Boksem, M.A. and Tops, M., 2008, Mental fatigue: costs and benefits, Brain Research Reviews, 59, 125–139.

10. Boles M., Pelletier B. and Lynch W., 2004, The relationship between health risks and work productivity, Journal of Occupational and Environmental Medicine, 46, 737–745.
11. Cunningham, B.J., and Eberle T., 1990, A Guide to job enrichment and redesign, Personnel, February, 56–61.
12. Damrad-Frye, R. and Laird J.D., 1989, The experience of boredom: The role of the self-perception of attention. Journal of Personality and Social Psychology, 57, 315-320.
13. Davis, L.E., Canter, R.R. and Hoffman, J., 1955, Current job design criteria, International Journal of Industrial Ergonomics, 16, 1-7.
14. Dubois, D., Fargier, H. and Prade, H., 1995, Fuzzy constraint in job shop scheduling, Journal of Intelligent Manufacturing, 6, 215–234.
15. Dembe, A., Erickson, J. and Delbos, R., 2005, The impact of overtime and long work hours on occupational injuries and illnesses: New evidence from the United States. Occupational and Environmental Medicine, 62(9), 588-597.
16. Dullin, E., 1998, Enterprise and plant-centric scheduling, (Cover story), Institute of Industrial Engineers Solutions, 30(2), 16-18.
17. Dyer-Smith M.B.A. and Wesson, D.A., 1997, Resource allocation efficiency as an indicator of boredom, work performance and absence, Ergonomics, 40(5), 515-521(7).
18. Farmer, R., and Sundberg, N. D., 1986. Boredom proneness-the development and correlates of a new scale. Journal of Personality Assessment, 50, 4-17.

19. Fisher, C.D., 1993, Boredom at work: A neglected concept. Human Relations, 46, 395-417.

20. Gittleman, M., Horrigan, M. and Joyce, M., 1998, Flexible workplace practices: evidence from a nationally representative survey. Industrial and Labour Relations Review, 52, 99–115.

21. Goetzel, R.Z., Ozminkowski, R.J. and Sederer, L.I., 2002, The Business Case for Quality Mental Health Services: Why Employers Should Care About the Mental Health and Well-Being of Their Employees, Journal of Occupational and Environmental Medicine, 44(4):320-330.

22. Hajkowicz, S., Young, M., Wheeler, S., MacDonald, D.H. and Young, D., 2000, Supporting Decisions: Understanding Natural Resource Management Assessment Techniques, Technical report, Land and Water Resources Research and Development Corporation.

23. Harma, M., 2006, Work hours in relation to work stress, recovery and health. Scandinavian Journal of Work, Environment and Health, 32(6), 502-514.

24. Hamilton, J., 1981, Attention, personality, and self-regulation of mood: Absorbing attention and boredom, In B.A. Maher (Ed.) Progress in experimental personality research, 10, 281-315.

25. Hinze, J., 2006, Construction safety, Upper Saddle River, New Jersey: Prentice Hall, 43-57.

26. Holding, D., 1983, Fatigue. In: Hockey, G.R.J. (Ed.), Stress and fatigue in human performance, John Wiley and Sons, Durnham, 145–164.

27. Howard, A.F., 1991, A critical look at multiple criteria decision making techniques with reference to forestry applications, Canadian Journal of Forest Research, 21, 1649-1659.

28. Sabuncuoglu, I. and Kizilisik, O. B., 2003, Reactive scheduling in a dynamic and stochastic FMS environment, International Journal of Production Research, 41(17), 4211-4231.

29. Jaber, M.Y. and Neumann, W.P, 2010, Modelling worker fatigue and recovery in dual-resource constrained systems, Computers and Industrial Engineering, 59(1), 75-84.

30. Janssen, R., 1992, Multi objective Decision Support for Environmental Management. Nor- well, MA: Kluwer Academic Publishers, Dordrecht, The Netherlands, 232-238.

31. Kass, S.J., Vodanovich, S.J. and Callander, A., 2001, State-trait boredom: The relationship to absenteeism, tenure and job satisfaction, Journal of Business and Psychology, 16, 317-327.

32. Konz, S., and Johnson, S., 2004, Work design-occupational ergonomics (6^{th} ed.).Scottsdale, Arizona: Holcomb Hathaway.

33. Kurimori, S. and Kakizaki, T., 1995, Evaluation of Work Stress Using Psychological and Physiological Measures of Mental Activity in a Paced Calculating Task, Industrial Health, 33, 7-22.

34. Lewis, G. and Wessely, S., 1992, The epidemiology of fatigue: More questions than answers, Journal of Epidemiology and Community Health, 46(2), 92-97.

35. Marcora, S.M., Staiano, W. and Manning, V., 2009, Mental fatigue impairs physical performance in humans, School of Sport, Health and Exercise Sciences, Bangor

University, Bangor, Wales, United Kingdom, Journal of Applied Physiology, 106, 857-864.

36. Mattke, S., Balakrishnan A., Bergamo G. and Newberry S.J., 2007, A review of methods to measure health-related productivity loss, American Journal of Managed Care, 13, 211–217.
37. McKay K.N. and Wiers, V.C.S., 1999, Unifying the Theory and Practice of Production Scheduling, Journal of Manufacturing Systems, 18(4), 241-255.
38. Meijman, T.F., 2000, The theory of the stop-emotion: on the functionality of fatigue, In: Pogorski, D., Karwowski, W. (Eds.), Ergonomics and safety for global business quality and production. CIOP, Warschaw, 45–50.
39. Murata, A., Uetake, A. and Takasawa, Y., 2005, Evaluation of mental fatigue using feature parameter extracted from event-related potential, International Journal of Industrial Ergonomics, 35, 761–770.
40. Okogbaa, O.G., Shell, R.L. and Filipusic, D., 1994, On the investigation of the neurophysiologic correlates of knowledge worker mental fatigue using the EEG signal, Applied Ergonomics, 25(6), 355-365.
41. Ortega, J., 2001, Job rotation as a learning mechanism. Management Science, 47, 1361–1370.
42. Osterman, P., 1994, How common is workplace transformation and who adopts it? Industrial and Labour Relations Review, 47, 173–188.
43. Pinedo, M., 1995, Scheduling: Theory, Algorithms and Systems, Prentice Hall: Englewood Cliffs, New Jersey.
44. Reason, J., 1990, Human error, New York: Cambridge University Press, 26-35.

45. Ricci, J.A., Chee, E., Lorandeau, A. and Berger, J., 2007, Fatigue in the U.S. Workforce: Prevalence and Implications for Lost Productive Work Time, Journal of Occupational and Environmental Medicine, 49, 1, 1-10.

46. Saaty T. L., 1980, The Analytic Hierarchy Process, New York: McGraw Hill International.

47. Sawin, D.A. and Scerbo, M.W., 1995, Effects of instruction type and boredom proneness in vigilance: Implications for boredom and workload, Human Factors, 37, 752-765.

48. Shtub, A., Leblanc, L.J. and Cai, Z., 1996, Scheduling problems with repetitive projects: a comparison of a simulated annealing, a genetic and pair-wise swap algorithm, European Journal of Operational Research, 88, 124–138.

49. Sparks, K., Cooper, C. and Fried, Y., 1997, The effects of hours of work on health: Ameta-analytic review, Journal of Occupational and Organizational Psychology, 70, 391-408.

50. Thompson, G.M., 1999, Controlling Workforce Schedules in Real-time Cornell Hotel and Restaurant Administration Quarterly Labour scheduling, Part 4, 40, 85-96.

51. Taris, T., Beckers, D. and Dahlgren, A., 2007, Overtime work and well-being: Prevalence, conceptualization and effects of working overtime, In S. McIntyre and J. Houdmont (Eds.), Occupational health psychology: European perspectives on research, education and practice, Volume 2, Maia, Portugal: ISMAI, 21-41.

52. Van der Hulst,M., 2003, Long work hours and health. Scandinavian Journal of Work, Environment and Health, 29(3), 171-188.

53. Visual Basic Developer Center, http://msdn.microsoft.com/en-us/vbasic/bb466159.

54. Voudouris, C., Owusu, G., Dorne, R. and Lesaint,D., 2008, Service Chain Management: Technology Innovation for the Service Business, DOI: 10.1007/978-3-540-75504-3.

Appendix A - Data of program codes:

I. Form1.vb

```
Public Class Form1

    Private globalInfo As New GlobalInfo()


    Private Sub Save1_1Button_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Save1_1Button.Click

        globalInfo.It = Convert.ToSingle(ItTextBox.Text)
        globalInfo.MCmax = Convert.ToSingle(MCmaxTextBox.Text)
        globalInfo.MCmin = Convert.ToSingle(MCminTextBox.Text)
        globalInfo.Ti = Convert.ToSingle(TiTextBox.Text)
        globalInfo.TNT = Convert.ToSingle(TNTTextBox.Text)
        globalInfo.PeriodNo =
Convert.ToSingle(PeriodTextBox.Text)
        globalInfo.PeriodLength = Math.Ceiling(globalInfo.Ti /
globalInfo.PeriodNo)
        globalInfo.B = Convert.ToSingle(BTrackBar.Value) / 100


'/////////////////////////////////////////////////////////////////
////////////////

        Dim periods(globalInfo.PeriodNo - 1) As Integer
        Dim tasks(globalInfo.TNT - 1) As Integer
        Dim workers(globalInfo.It - 1) As Integer

        For index = 0 To globalInfo.PeriodNo - 1
            periods(index) = index + 1
        Next

        For index = 0 To globalInfo.TNT - 1
            tasks(index) = index + 1
        Next

        For index = 0 To globalInfo.It - 1
            workers(index) = index + 1
        Next
```

```
        ComboBox1.DataSource = periods
        ComboBox2.DataSource = tasks
        ComboBox3.DataSource = tasks
        ComboBox4.DataSource = workers
        ComboBox5.DataSource = workers


'/////////////////////////////////////////////////////////////////
////////////////////

        MessageBox.Show(Me, "Saved")

        TabControl2.SelectTab(1)

    End Sub

    Private Sub Save1_2Button_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Save1_2Button.Click

        globalInfo.a1 =
Convert.ToSingle(Worker_WorkerTextBox.Text)
        globalInfo.a2 =
Convert.ToSingle(Worker_TasksTextBox.Text)
        globalInfo.a3 = Convert.ToSingle(Worker_ECTextBox.Text)

        globalInfo.a4 =
Convert.ToSingle(Tasks_WorkerTextBox.Text)
        globalInfo.a5 = Convert.ToSingle(Tasks_TasksTextBox.Text)
        globalInfo.a6 = Convert.ToSingle(Tasks_ECTextBox.Text)

        globalInfo.a7 = Convert.ToSingle(EC_WorkerTextBox.Text)
        globalInfo.a8 = Convert.ToSingle(EC_TasksTextBox.Text)
        globalInfo.a9 = Convert.ToSingle(EC_ECTextBox.Text)


'/////////////////////////////////////////////////////////////////
/////

        globalInfo.b1 = Convert.ToSingle(CM1_CM1TextBox.Text)
        globalInfo.b2 = Convert.ToSingle(CM1_CM2TextBox.Text)
        globalInfo.b3 = Convert.ToSingle(CM1_CRTTextBox.Text)
        globalInfo.b4 = Convert.ToSingle(CM1_SRTextBox.Text)

        globalInfo.b5 = Convert.ToSingle(CM2_CM1TextBox.Text)
        globalInfo.b6 = Convert.ToSingle(CM2_CM2TextBox.Text)
```

```
        globalInfo.b7 = Convert.ToSingle(CM2_CRTTextBox.Text)
        globalInfo.b8 = Convert.ToSingle(CM2_SRTextBox.Text)

        globalInfo.b9 = Convert.ToSingle(CRT_CM1TextBox.Text)
        globalInfo.b10 = Convert.ToSingle(CRT_CM2TextBox.Text)
        globalInfo.b11 = Convert.ToSingle(CRT_CRTTextBox.Text)
        globalInfo.b12 = Convert.ToSingle(CRT_SRTextBox.Text)

        globalInfo.b13 = Convert.ToSingle(SR_CM1TextBox.Text)
        globalInfo.b14 = Convert.ToSingle(SR_CM2TextBox.Text)
        globalInfo.b15 = Convert.ToSingle(SR_CRTTextBox.Text)
        globalInfo.b16 = Convert.ToSingle(SR_SRTextBox.Text)


'/////////////////////////////////////////////////////////////////
//////////


        globalInfo.c1 = Convert.ToSingle(B_CM1_CM1TextBox.Text)
        globalInfo.c2 = Convert.ToSingle(B_CM1_CM2TextBox.Text)
        globalInfo.c3 = Convert.ToSingle(B_CM1_CRTTextBox.Text)
        globalInfo.c4 = Convert.ToSingle(B_CM1_SRTextBox.Text)

        globalInfo.c5 = Convert.ToSingle(B_CM2_CM1TextBox.Text)
        globalInfo.c6 = Convert.ToSingle(B_CM2_CM2TextBox.Text)
        globalInfo.c7 = Convert.ToSingle(B_CM2_CRTTextBox.Text)
        globalInfo.c8 = Convert.ToSingle(B_CM2_SRTextBox.Text)

        globalInfo.c9 = Convert.ToSingle(B_CRT_CM1TextBox.Text)
        globalInfo.c10 = Convert.ToSingle(B_CRT_CM2TextBox.Text)
        globalInfo.c11 = Convert.ToSingle(B_CRT_CRTTextBox.Text)
        globalInfo.c12 = Convert.ToSingle(B_CRT_SRTextBox.Text)

        globalInfo.c13 = Convert.ToSingle(B_SR_CM1TextBox.Text)
        globalInfo.c14 = Convert.ToSingle(B_SR_CM2TextBox.Text)
        globalInfo.c15 = Convert.ToSingle(B_SR_CRTTextBox.Text)
        globalInfo.c16 = Convert.ToSingle(B_SR_SRTextBox.Text)


'/////////////////////////////////////////////////////////////////
//////////

        globalInfo.d1 = Convert.ToSingle(SH_SHTextBox.Text)
        globalInfo.d2 = Convert.ToSingle(SH_amTextBox.Text)
        globalInfo.d3 = Convert.ToSingle(SH_apTextBox.Text)
        globalInfo.d4 = Convert.ToSingle(SH_OTextBox.Text)
```

```
        globalInfo.d5 = Convert.ToSingle(SH_TRTextBox.Text)

        globalInfo.d6 = Convert.ToSingle(am_SHTextBox.Text)
        globalInfo.d7 = Convert.ToSingle(am_amTextBox.Text)
        globalInfo.d8 = Convert.ToSingle(am_apTextBox.Text)
        globalInfo.d9 = Convert.ToSingle(am_OTextBox.Text)
        globalInfo.d10 = Convert.ToSingle(am_TRTextBox.Text)

        globalInfo.d11 = Convert.ToSingle(ap_SHTextBox.Text)
        globalInfo.d12 = Convert.ToSingle(ap_amTextBox.Text)
        globalInfo.d13 = Convert.ToSingle(ap_amTextBox.Text)
        globalInfo.d14 = Convert.ToSingle(ap_OTextBox.Text)
        globalInfo.d15 = Convert.ToSingle(ap_TRTextBox.Text)

        globalInfo.d16 = Convert.ToSingle(O_SHTextBox.Text)
        globalInfo.d17 = Convert.ToSingle(O_amTextBox.Text)
        globalInfo.d18 = Convert.ToSingle(O_apTextBox.Text)
        globalInfo.d19 = Convert.ToSingle(O_OTextBox.Text)
        globalInfo.d20 = Convert.ToSingle(O_TRTextBox.Text)

        globalInfo.d21 = Convert.ToSingle(TR_SHTextBox.Text)
        globalInfo.d22 = Convert.ToSingle(TR_amTextBox.Text)
        globalInfo.d23 = Convert.ToSingle(TR_apTextBox.Text)
        globalInfo.d24 = Convert.ToSingle(TR_OTextBox.Text)
        globalInfo.d25 = Convert.ToSingle(TR_TRTextBox.Text)


'/////////////////////////////////////////////////////////////////
//////////

        globalInfo.e1 = Convert.ToSingle(PLC_PLCTextBox.Text)
        globalInfo.e2 = Convert.ToSingle(PLC_PSCTextBox.Text)
        globalInfo.e3 = Convert.ToSingle(PLC_PLSTextBox.Text)

        globalInfo.e4 = Convert.ToSingle(PSC_PLCTextBox.Text)
        globalInfo.e5 = Convert.ToSingle(PSC_PSCTextBox.Text)
        globalInfo.e6 = Convert.ToSingle(PSC_PLSTextBox.Text)

        globalInfo.e7 = Convert.ToSingle(PLS_PLCTextBox.Text)
        globalInfo.e8 = Convert.ToSingle(PLS_PSCTextBox.Text)
        globalInfo.e9 = Convert.ToSingle(PLS_PLSTextBox.Text)

        globalInfo.Process()

        MessageBox.Show(Me, "Saved")
```

```
        TabControl1.SelectTab(1)

    End Sub

    Private Sub Save2Button_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Save2Button.Click

        Dim taskID As Integer
        Dim CM1, CM2, CRT, SR As Single

        taskID = ComboBox3.SelectedValue
        CM1 = Convert.ToSingle(CM1TrackBar.Value) / 100
        CM2 = Convert.ToSingle(CM2TrackBar.Value) / 100
        CRT = Convert.ToSingle(CRTTrackBar.Value) / 100
        SR = Convert.ToSingle(SRTrackBar.Value) / 100

        Dim newTask As New Task(taskID, CM1, CM2, CRT, SR, False)

        globalInfo.TasksList.Add(newTask)

        ComboBox3.SelectedIndex = (ComboBox3.SelectedIndex + 1)
Mod ComboBox3.Items.Count

        MessageBox.Show(Me, "Saved")

    End Sub

    Private Sub SaveAndNextTaskButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SaveAndNextTaskButton.Click

        Dim taskNo, periodNo As Integer

        periodNo = ComboBox1.SelectedValue
        taskNo = ComboBox2.SelectedValue

        Dim PLC As Single = Convert.ToSingle(PLCTrackBar.Value) /
100
        Dim PSC As Single = Convert.ToSingle(PSCTrackBar.Value) /
100
        Dim PLS As Single = Convert.ToSingle(PLSTrackBar.Value) /
100

        Dim NewEC As New EC(PLC, PSC, PLS)

        globalInfo.GetTask(taskNo).AddEC(NewEC)
```

```
        ComboBox2.SelectedIndex = (ComboBox2.SelectedIndex + 1)
Mod ComboBox2.Items.Count

        MessageBox.Show(Me, "Saved")

    End Sub

    Private Sub SaveAndNextPeriodButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SaveAndNextPeriodButton.Click

        Dim taskNo, periodNo As Integer

        periodNo = ComboBox1.SelectedValue
        taskNo = ComboBox2.SelectedValue

        Dim PLC As Single = Convert.ToSingle(PLCTrackBar.Value) /
100
        Dim PSC As Single = Convert.ToSingle(PSCTrackBar.Value) /
100
        Dim PLS As Single = Convert.ToSingle(PLSTrackBar.Value) /
100

        Dim NewEC As New EC(PLC, PSC, PLS)

        globalInfo.GetTask(taskNo).AddEC(NewEC)

        ComboBox1.SelectedIndex = (ComboBox1.SelectedIndex + 1)
Mod ComboBox1.Items.Count

        MessageBox.Show(Me, "Saved")

    End Sub

    Private Sub Save4_1Button_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Save4_1Button.Click

        Dim workerNo As Integer = ComboBox4.SelectedValue
        Dim SH As Single = Convert.ToSingle(SHTrackBar.Value) /
100
        Dim TR As Single = Convert.ToSingle(TRTrackBar.Value) /
100
```

```
        Dim newWorker As New Worker(workerNo, SH, TR, 0, 0, 0,
Nothing)

        globalInfo.WorkersList.Add(newWorker)

        ComboBox4.Enabled = False

        MessageBox.Show(Me, "Saved")

        TabControl3.SelectTab(1)

    End Sub

    Private Sub Save4_2Button_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Save4_2Button.Click

        Dim am As Single = (TrackBar3.Value + TrackBar4.Value +
TrackBar5.Value + TrackBar6.Value + TrackBar7.Value) / 500

        globalInfo.GetWorker(ComboBox4.SelectedValue).am = am

        TextBox44.Text = am

        MessageBox.Show(Me, "Saved")

        TabControl3.SelectTab(2)

    End Sub

    Private Sub Save4_3Button_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Save4_3Button.Click

        Dim ap As Single = (TrackBar1.Value + TrackBar2.Value +
TrackBar8.Value + TrackBar9.Value + TrackBar10.Value) / 500

        globalInfo.GetWorker(ComboBox4.SelectedValue).ap = ap

        TextBox35.Text = ap

        MessageBox.Show(Me, "Saved")

        TabControl3.SelectTab(3)

    End Sub
```

```
    Private Sub Save4_4Button_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Save4_4Button.Click

        Dim O As Single = (TrackBar11.Value + TrackBar12.Value +
TrackBar13.Value + TrackBar14.Value + TrackBar15.Value) / 500

        globalInfo.GetWorker(ComboBox4.SelectedValue).O = O

        TextBox53.Text = O

        MessageBox.Show(Me, "Saved")

    End Sub

    Private Sub NextWorkerButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NextWorkerButton.Click

        TextBox44.Text = String.Empty
        TextBox35.Text = String.Empty
        TextBox53.Text = String.Empty

        ComboBox4.Enabled = True
        ComboBox4.SelectedIndex = (ComboBox4.SelectedIndex + 1)
Mod ComboBox4.Items.Count

        TabControl3.SelectTab(0)

    End Sub

    Private Sub FinishButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles FinishButton.Click

        FinishButton.Enabled = False

        globalInfo.Initialization()

        globalInfo.TaskAssigning()

        MessageBox.Show(Me, "Complete")

    End Sub
```

```
    Private Sub CalculationButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CalculationButton.Click

        Dim workerID As Integer = ComboBox5.SelectedValue

        Dim Worker = globalInfo.GetWorker(workerID)


'/////////////////////////////////////////////////////////////////
//////////////////////////////

        RotationsTextBox.Text = Worker.GetRotationCount()

        ListView1.Items.Clear()

        For index = 1 To Worker.GetRotationCount()
            Dim rotation As Rotation = Worker.GetRotation(index)
            Dim newLVI As ListViewItem = New ListViewItem(New
String() {rotation.Task.taskID, rotation.Task.Status,
rotation.Duration})
            ListView1.Items.Add(newLVI)
        Next

        DrawChart(Worker)

    End Sub

    Private Sub DrawChart(ByVal Worker As Worker)

        Dim Bitmap As New Bitmap(Convert.ToInt32(globalInfo.Ti *
15 + 100), Convert.ToInt32(globalInfo.MCmax * 2 * 20) + 50)

        Dim g As Graphics = Graphics.FromImage(Bitmap)

        '////////////////'rasme mehvarha
        Dim font As New Font("Microsoft Sans Serif", 6)

        g.DrawLine(Pens.Black, 8, Bitmap.Height - 15,
Bitmap.Width, Bitmap.Height - 15) 'rasme mehvare ofoghi
        g.DrawLine(Pens.Black, 8, 0, 8, Bitmap.Height - 15)
'rasme mehvare amoodi

        'rasme adade zire mehvare ofoghi
        For Ti = 1 To globalInfo.Ti Step 2
```

```
            g.DrawString(Ti, font, Brushes.Black, (Ti * 15) + 2,
Bitmap.Height - 10)
        Next

        'rasme adade kenare mehvare amoodi va khotoote ofoghi
        For mc = 1 To globalInfo.MCmax * 2
            g.DrawString(mc, font, Brushes.Black, 0,
(Bitmap.Height - 15) - mc * 20)
            g.DrawLine(New Pen(Brushes.Blue, 0.01), 8,
(Bitmap.Height - 10) - mc * 20, Bitmap.Width, (Bitmap.Height -
10) - mc * 20)
        Next
        '////////////////


        Dim MCPoints As New List(Of Point)

        For index = 1 To globalInfo.Ti

            Dim mc As Single = Worker.GetMC(index)

            MCPoints.Add(New Point(index * 15 + 5, Bitmap.Height
- 12 - mc * 20))


            If mc <= globalInfo.MCmax And mc >= globalInfo.MCmin
Then
                g.FillEllipse(Brushes.Black, index * 15 + 3,
Bitmap.Height - 14 - mc * 20, 4, 4)
            Else
                g.FillEllipse(Brushes.Red, index * 15 + 3,
Bitmap.Height - 14 - mc * 20, 4, 4)
            End If

            g.DrawLine(Pens.Black, index * 15 + 5, Bitmap.Height
- 12 - mc * 20, index * 15 + 5, Bitmap.Height - 15)

        Next

        g.DrawLines(Pens.Aqua, MCPoints.ToArray())

        PictureBox1.Image = Bitmap

    End Sub
```

```
    Private Sub NextWorker5Button_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NextWorker5Button.Click

        ComboBox5.SelectedIndex = (ComboBox5.SelectedIndex + 1)
Mod ComboBox5.Items.Count

        RotationsTextBox.Text = String.Empty

        ListView1.Items.Clear()

        PictureBox1.Image = Nothing

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        Dim form2 As New Form2(globalInfo.WorkersList)
        form2.ShowDialog()

    End Sub



'////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
/////


    Private Sub BTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BTrackBar.Scroll

        TextBox71.Text = BTrackBar.Value / 100

    End Sub

    Private Sub CM1TrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CM1TrackBar.Scroll
        TextBox75.Text = CM1TrackBar.Value / 100
    End Sub

    Private Sub CM2TrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CM2TrackBar.Scroll
        TextBox74.Text = CM2TrackBar.Value / 100
    End Sub
```

```
    Private Sub CRTTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CRTTrackBar.Scroll
        TextBox73.Text = CRTTrackBar.Value / 100
    End Sub

    Private Sub SRTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SRTrackBar.Scroll
        TextBox72.Text = SRTrackBar.Value / 100
    End Sub

    Private Sub PLCTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PLCTrackBar.Scroll
        TextBox83.Text = PLCTrackBar.Value / 100
    End Sub

    Private Sub PSCTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PSCTrackBar.Scroll
        TextBox84.Text = PSCTrackBar.Value / 100
    End Sub

    Private Sub PLSTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PLSTrackBar.Scroll
        TextBox85.Text = PLSTrackBar.Value / 100
    End Sub

    Private Sub SHTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SHTrackBar.Scroll
        TextBox86.Text = SHTrackBar.Value / 100
    End Sub

    Private Sub TRTrackBar_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TRTrackBar.Scroll
        TextBox88.Text = TRTrackBar.Value / 100
    End Sub

    Private Sub TrackBar6_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar6.Scroll
        TextBox94.Text = TrackBar6.Value / 100
    End Sub

    Private Sub TrackBar5_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar5.Scroll
        TextBox93.Text = TrackBar5.Value / 100
    End Sub
```

```
    Private Sub TrackBar4_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar4.Scroll
        TextBox92.Text = TrackBar4.Value / 100
    End Sub

    Private Sub TrackBar3_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar3.Scroll
        TextBox90.Text = TrackBar3.Value / 100
    End Sub

    Private Sub TrackBar7_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar7.Scroll
        TextBox89.Text = TrackBar7.Value / 100
    End Sub

    Private Sub TrackBar9_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar9.Scroll
        TextBox95.Text = TrackBar9.Value / 100
    End Sub

    Private Sub TrackBar8_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar8.Scroll
        TextBox96.Text = TrackBar8.Value / 100
    End Sub

    Private Sub TrackBar1_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar1.Scroll
        TextBox97.Text = TrackBar1.Value / 100
    End Sub

    Private Sub TrackBar10_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar10.Scroll
        TextBox98.Text = TrackBar10.Value / 100
    End Sub

    Private Sub TrackBar2_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar2.Scroll
        TextBox99.Text = TrackBar2.Value / 100
    End Sub

    Private Sub TrackBar14_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar14.Scroll
        TextBox100.Text = TrackBar14.Value / 100
    End Sub
```

```
    Private Sub TrackBar13_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar13.Scroll
        TextBox108.Text = TrackBar13.Value / 100
    End Sub

    Private Sub TrackBar11_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar11.Scroll
        TextBox109.Text = TrackBar11.Value / 100
    End Sub

    Private Sub TrackBar15_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar15.Scroll
        TextBox110.Text = TrackBar15.Value / 100
    End Sub

    Private Sub TrackBar12_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar12.Scroll
        TextBox111.Text = TrackBar12.Value / 100
    End Sub

    Private Sub TabPage4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TabPage4.Click

    End Sub

    Private Sub SH_OTextBox_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SH_OTextBox.TextChanged

    End Sub

    Private Sub Label1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Label1.Click

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
        PictureBox1.Image.Save("result.bmp")
        System.Diagnostics.Process.Start("result.bmp")
    End Sub

    Private Sub TextBox49_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox49.TextChanged
```

```
    End Sub

    Private Sub TextBox40_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox40.TextChanged

    End Sub
End Class
```

II. Form2.vb

```
Public Class Form2

    Private workersList As List(Of Worker)

    Public Sub New(ByRef workersList As List(Of Worker))

        InitializeComponent()

        Me.workersList = workersList

        For Each Worker As Worker In workersList

            Dim str As String = String.Empty

            For index = 1 To Worker.GetRotationCount()
                Dim rotation As Rotation =
Worker.GetRotation(index)
                str += rotation.Task.taskID.ToString() + " [" +
rotation.Duration.ToString() + "] " + "-- "
            Next

            ListView1.Items.Add(New ListViewItem(New String()
{Worker.workerID, Worker.GetRotationCount(), str}))

        Next

        ListView1.AutoResizeColumn(2,
ColumnHeaderAutoResizeStyle.ColumnContent)

    End Sub
```

```
    Private Sub PlainTextToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
PlainTextToolStripMenuItem.Click

        If SaveFileDialog1.ShowDialog(Me) <> DialogResult.Cancel
Then

            Dim result As System.IO.StreamWriter = New
IO.StreamWriter(SaveFileDialog1.FileName)
'SaveFileDialog1.OpenFile()
            result.AutoFlush = True


result.WriteLine(String.Format("{0,10}{1,5}{2,11}{3,5}{4,15}",
"Worker ID ", " ", "Rotation No", " ", "Task [Rotation]"))
            result.WriteLine()

            For Each Worker As Worker In workersList


result.Write(String.Format("{0,5}{1,9}{2,18}{3,16}",
Worker.workerID, " ", Worker.GetRotationCount(), " "))

                For index = 1 To Worker.GetRotationCount()
                    Dim rotation As Rotation =
Worker.GetRotation(index)
                    result.Write("{0} [{1}],    ",
rotation.Task.taskID, rotation.Duration)
                Next

                result.WriteLine()

            Next

            result.Close()
            Dim answer As DialogResult = MessageBox.Show(Me,
"Would You Like To View Result", "", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)

            If answer = DialogResult.Yes Then
                System.Diagnostics.Process.Start("notepad",
SaveFileDialog1.FileName)
            End If

        End If
```

```
        End Sub
    End Class
```

III. GlobalInfo.vb

```
    Public Class GlobalInfo

        Public WorkersList As New Collections.Generic.List(Of Worker)
        Public TasksList As New Collections.Generic.List(Of Task)

        Public It As Single
        Public MCmax As Single
        Public MCmin As Single
        Public Ti As Single
        Public TNT As Single
        Public PeriodNo As Single
        Public PeriodLength As Integer
        Public B As Single

        Public a1, a2, a3, a4, a5, a6, a7, a8, a9 As Single
        Public b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12,
    b13, b14, b15, b16 As Single
        Public c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12,
    c13, c14, c15, c16 As Single
        Public d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12,
    d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25
    As Single
        Public e1, e2, e3, e4, e5, e6, e7, e8, e9 As Single


        Dim BestMatchWieght1 As Single
        Dim BestMatchWieght2 As Single
        Dim BestMatchWieght3 As Single

        Dim HardTaskWeight1 As Single
        Dim HardTaskWeight2 As Single
        Dim HardTaskWeight3 As Single
        Dim HardTaskWeight4 As Single

        Dim ComfortTaskWeight1 As Single
        Dim ComfortTaskWeight2 As Single
        Dim ComfortTaskWeight3 As Single
        Dim ComfortTaskWeight4 As Single
```

```
Dim WorkerWeight1 As Single
Dim WorkerWeight2 As Single
Dim WorkerWeight3 As Single
Dim WorkerWeight4 As Single
Dim WorkerWeight5 As Single

Dim ECWeight1 As Single
Dim ECWeight2 As Single
Dim ECWeight3 As Single

Dim cw1 As Single
Dim cw2 As Single
Dim cw3 As Single
Dim cw4 As Single
Dim cw5 As Single
Dim cw6 As Single
Dim cw7 As Single
Dim cw8 As Single
Dim cw9 As Single
Dim cw10 As Single
Dim cw11 As Single
Dim cw12 As Single

Dim hw1 As Single
Dim hw2 As Single
Dim hw3 As Single
Dim hw4 As Single
Dim hw5 As Single
Dim hw6 As Single
Dim hw7 As Single
Dim hw8 As Single
Dim hw9 As Single
Dim hw10 As Single
Dim hw11 As Single
Dim hw12 As Single


Public Sub Process()

    Dim t1 As Single = a1 + a4 + a7
    Dim t2 As Single = a2 + a5 + a8
    Dim t3 As Single = a3 + a6 + a9

    BestMatchWieght1 = (a1 / t1 + a2 / t2 + a3 / t3) / 3
    BestMatchWieght2 = (a4 / t1 + a5 / t2 + a6 / t3) / 3
    BestMatchWieght3 = (a7 / t1 + a8 / t2 + a9 / t3) / 3
```

```
        '//////////////////////////////////////////////////

        Dim tt1 As Single = b1 + b5 + b9 + b13
        Dim tt2 As Single = b2 + b6 + b10 + b14
        Dim tt3 As Single = b3 + b7 + b11 + b15
        Dim tt4 As Single = b4 + b8 + b12 + b16

        HardTaskWeight1 = (b1 / tt1 + b2 / tt2 + b3 / tt3 + b4 /
tt4) / 4
        HardTaskWeight2 = (b5 / tt1 + b6 / tt2 + b7 / tt3 + b8 /
tt4) / 4
        HardTaskWeight3 = (b9 / tt1 + b10 / tt2 + b11 / tt3 + b12
/ tt4) / 4
        HardTaskWeight4 = (b13 / tt1 + b14 / tt2 + b15 / tt3 +
b16 / tt4) / 4
        '///////////////////////////////////////////////////////

        Dim ttt1 As Single = c1 + c5 + c9 + c13
        Dim ttt2 As Single = c2 + c6 + c10 + c14
        Dim ttt3 As Single = c3 + c7 + c11 + c15
        Dim ttt4 As Single = c4 + c8 + c12 + c16

        ComfortTaskWeight1 = (c1 / ttt1 + c2 / ttt2 + c3 / ttt3 +
c4 / ttt4) / 4
        ComfortTaskWeight2 = (c5 / ttt1 + c6 / ttt2 + c7 / ttt3 +
c8 / ttt4) / 4
        ComfortTaskWeight3 = (c9 / ttt1 + c10 / ttt2 + c11 / ttt3
+ c12 / ttt4) / 4
        ComfortTaskWeight4 = (c13 / ttt1 + c14 / ttt2 + c15 /
ttt3 + c16 / ttt4) / 4
        '///////////////////////////////////////////////////////

        Dim tttt1 As Single = d1 + d6 + d11 + d16 + d21
        Dim tttt2 As Single = d2 + d7 + d12 + d17 + d22
        Dim tttt3 As Single = d3 + d8 + d13 + d18 + d23
        Dim tttt4 As Single = d4 + d9 + d14 + d19 + d24
        Dim tttt5 As Single = d5 + d10 + d15 + d20 + d25

        WorkerWeight1 = (d1 / tttt1 + d2 / tttt2 + d3 / tttt3 +
d4 / tttt4 + d5 / tttt5) / 5
        WorkerWeight2 = (d6 / tttt1 + d7 / tttt2 + d8 / tttt3 +
d9 / tttt4 + d10 / tttt5) / 5
        WorkerWeight3 = (d11 / tttt1 + d12 / tttt2 + d13 / tttt3
+ d14 / tttt4 + d15 / tttt5) / 5
        WorkerWeight4 = (d16 / tttt1 + d17 / tttt2 + d18 / tttt3
+ d19 / tttt4 + d20 / tttt5) / 5
```

```
        WorkerWeight5 = (d21 / tttt1 + d22 / tttt2 + d23 / tttt3
+ d24 / tttt4 + d25 / tttt5) / 5
        '//////////////////////////////////////////////////////

        Dim ttttt1 As Single = e1 + e4 + e7
        Dim ttttt2 As Single = e2 + e5 + e8
        Dim ttttt3 As Single = e3 + e6 + e9

        ECWeight1 = (e1 / ttttt1 + e2 / ttttt2 + e3 / ttttt3) / 3
        ECWeight2 = (e4 / ttttt1 + e5 / ttttt2 + e6 / ttttt3) / 3
        ECWeight3 = (e7 / ttttt1 + e8 / ttttt2 + e9 / ttttt3) / 3

'/////////////////////////////////////////////////////////////////
///////

        cw1 = BestMatchWieght3 * ECWeight1
        cw2 = BestMatchWieght3 * ECWeight2
        cw3 = BestMatchWieght3 * ECWeight3

        cw4 = BestMatchWieght2 * ComfortTaskWeight1
        cw5 = BestMatchWieght2 * ComfortTaskWeight2
        cw6 = BestMatchWieght2 * ComfortTaskWeight3
        cw7 = BestMatchWieght2 * ComfortTaskWeight4

        cw8 = BestMatchWieght1 * WorkerWeight1
        cw9 = BestMatchWieght1 * WorkerWeight2
        cw10 = BestMatchWieght1 * WorkerWeight3
        cw11 = BestMatchWieght1 * WorkerWeight4
        cw12 = BestMatchWieght1 * WorkerWeight5

'/////////////////////////////////////////////////////////////////
//////

        hw1 = BestMatchWieght3 * ECWeight1
        hw2 = BestMatchWieght3 * ECWeight2
        hw3 = BestMatchWieght3 * ECWeight3

        hw4 = BestMatchWieght2 * HardTaskWeight1
        hw5 = BestMatchWieght2 * HardTaskWeight2
        hw6 = BestMatchWieght2 * HardTaskWeight3
        hw7 = BestMatchWieght2 * HardTaskWeight4

        hw8 = BestMatchWieght1 * WorkerWeight1
        hw9 = BestMatchWieght1 * WorkerWeight2
        hw10 = BestMatchWieght1 * WorkerWeight3
        hw11 = BestMatchWieght1 * WorkerWeight4
```

```
        hw12 = BestMatchWieght1 * WorkerWeight5

    End Sub

    Public Function GetTask(ByVal taskID As Integer) As Task

        Return TasksList.Item(taskID - 1)

    End Function

    Public Function GetWorker(ByVal workerID As Integer) As
Worker

        Return WorkersList.Single(Function(worker)
worker.workerID = workerID)

    End Function

    Public Sub Initialization()

        Dim query As IEnumerable(Of Task) =
TasksList.OrderByDescending(Function(Task)
Task.cm1).Take(TasksList.Count / 2)

        For Each Task As Task In query
            Task.IsHard = True
        Next


'/////////////////////////////////////////////////////////////////
//////////////////////////////////////

        For Each Worker As Worker In WorkersList

            Dim task = TasksList.Where(Function(t)
t.GetECVlaues(1).PLC <> 0).First()

            Worker.AddRotation(New Rotation(0, task, 0))
            TasksList.Remove(task)

        Next

    End Sub

    Sub TaskAssigning()
```

```
        For t = 1 To Me.Ti

            UpdateWorkers()

            For Each Worker As Worker In WorkersList

                Dim newTask As Task = Nothing
                Dim TodayRotation = Worker.CurrentRotation
                TodayRotation.Duration += 1
                Dim TodayTask = Worker.CurrentTask


                Dim TodayMC As Single = (Worker.am *
TodayTask.cm1 + B * Worker.ap * TodayTask.cm2 + Worker.O *
TodayTask.cm1) _
                                  * (t - TodayRotation.StartDay)
+ (MCmax + MCmin) / 4


                Worker.AddMC(TodayMC)

                If TodayMC < MCmin Or TodayMC > MCmax Then

                    If TodayTask.IsHard Then
                        newTask = NextTask(Worker, Math.Ceiling(t
/ PeriodLength), False)
                    Else
                        newTask = NextTask(Worker, Math.Ceiling(t
/ PeriodLength), True)
                    End If

                    If IsNothing(newTask) Then

                        Worker.IsVolunteer = True
                        Worker.Violation += 1

                        If Worker.Violation > 3 Then

                            Dim query =
WorkersList.Where(Function(w) w.IsVolunteer = True And
w.CurrentTask.IsHard = Not TodayTask.IsHard)
                            Try
                                Dim volWorker =
query.OrderByDescending(Function(w)
w.CurrentTask.Normalization).First()
```

```
                                Worker.AddRotation(New
Rotation(t, volWorker.CurrentTask, 0))
                                volWorker.AddRotation(New
Rotation(t, TodayTask, 0))

                                volWorker.IsVolunteer = False
                                Worker.Violation = 0
                                Worker.IsVolunteer = False

                                If TodayTask.IsHard Then
                                    Worker.am -= 0.1
                                    Worker.ap -= 0.1
                                    Worker.O += 0.1

                                    volWorker.am += 0.1
                                    volWorker.ap -= 0.1
                                    volWorker.O -= 0.1
                                Else
                                    Worker.am += 0.1
                                    Worker.ap -= 0.1
                                    Worker.O -= 0.1

                                    volWorker.am -= 0.1
                                    volWorker.ap -= 0.1
                                    volWorker.O += 0.1
                                End If

                            Catch ex As Exception
                            End Try

                        End If 'Worker.Violation > 3 Then

                    Else 'new task is not null
                        Worker.IsVolunteer = False

                        TasksList.Add(TodayTask)

                        Worker.AddRotation(New Rotation(t,
newTask, 0))
                        TasksList.Remove(newTask)

                        If TodayTask.IsHard Then
                            Worker.am -= 0.1
                            Worker.ap -= 0.1
                            Worker.O += 0.1
                        Else
```

```
                            Worker.am += 0.1
                            Worker.ap -= 0.1
                            Worker.O -= 0.1
                        End If

                    End If 'IsNothing(newTask)
                End If 'TodayMC < MCmin Or TodayMC > MCmax
            Next
        Next

        For Each w As Worker In WorkersList
            w.RemoveEmptyRotations()
        Next

    End Sub

    Private Function NextTask(ByVal Worker As Worker, ByVal
periodNo As Integer, ByVal isHard As Boolean) As Task

        Dim total As Single = 0

        Dim query As IEnumerable(Of Task) =
TasksList.Where(Function(Task) Task.IsHard = isHard And
Task.GetECVlaues(periodNo).PLC <> 0)

        If query.Count() = 0 Then
            Return Nothing
        End If

        '/////////////////////////////////////////////

        For Each task As Task In query

            task.temp = 0

            task.temp = hw1 * task.GetECVlaues(periodNo).PLC +
hw2 * task.GetECVlaues(periodNo).PSC + hw3 *
task.GetECVlaues(periodNo).PLS + _
                hw4 * task.cm1 + hw5 * task.cm2 + hw6 * task.CRT
+ hw7 * task.SR + _
                hw8 * Worker.SH + hw9 * Worker.am + hw10 *
Worker.ap + hw11 * Worker.O + hw12 * Worker.TR

            total += task.temp

        Next
```

```vb
'///////////////////////////////////////////////////////////////

        For Each task As Task In query
            task.Normalization = task.temp / total
        Next

        Dim BestTask As Task =
query.OrderByDescending(Function(Task)
Task.Normalization).First()

        Return BestTask

    End Function

    Private Sub UpdateWorkers()
        For Each w In WorkersList
            w.CurrentRotation = w.GetLastRotation()
            w.CurrentTask = w.CurrentRotation.Task
        Next
    End Sub

End Class
```

IV. Rotation.vb

```vb
Public Class Rotation

    Public StartDay As Integer
    Public Task As Task
    Public Duration As Integer

    Public Sub New(ByVal StartDay As Integer, ByVal Task As Task,
ByVal Duration As Integer)

        Me.StartDay = StartDay
        Me.Task = Task
        Me.Duration = Duration

    End Sub
```

V. Task.vb

```
Public Class Task

    Public ReadOnly taskID As Integer
    Public cm1 As Single
    Public cm2 As Single
    Public CRT As Single
    Public SR As Single

    Public IsHard As Boolean

    Public temp As Single
    Public Normalization As Single

    Private ECsList As New Collections.Generic.List(Of EC)

    Public Sub New(ByVal taskID As Integer, ByVal cm1 As Single,
ByVal cm2 As Single, _
                   ByVal CRT As Single, ByVal SR As Single, ByVal
IsHard As Boolean)

        Me.taskID = taskID
        Me.cm1 = cm1
        Me.cm2 = cm2
        Me.CRT = CRT
        Me.SR = SR
        Me.IsHard = IsHard

    End Sub

    Public ReadOnly Property Status() As String
        Get

            If IsHard Then
                Return "Hard    "
            Else
                Return "Comfort"
            End If

        End Get
    End Property


    Public Function GetECVlaues(ByVal PeriodNo) As EC
```

```
        Return ECsList.Item(PeriodNo - 1)

    End Function

    Public Sub AddEC(ByVal NewEC As EC)

        ECsList.Add(NewEC)

    End Sub

    Public Overrides Function ToString() As String
        Return taskID
    End Function

End Class
```

VI. Worker.vb

```
Public Class Worker

    Public ReadOnly workerID As Integer

    Public IsVolunteer As Boolean

    Public CurrentTask As Task

    Public Violation As Integer

    Public CurrentRotation As Rotation

    Public SH, TR, am, ap, O As Single

    Private RotationsList As New Collections.Generic.List(Of
Rotation)

    Private MCsList As New Collections.Generic.List(Of Single)

    Public Sub New(ByVal workerID As Integer, ByVal SH As Single,
ByVal TR As Single, _
                   ByVal am As Single, ByVal ap As Single, ByVal
O As Single, ByVal task As Task)

        Me.workerID = workerID
```

```
        Me.SH = SH
        Me.TR = TR
        Me.am = am
        Me.ap = ap
        Me.O = O

    End Sub

    Public Function GetRotation(ByVal RotationNo As Integer) As
Rotation

        Return RotationsList.Item(RotationNo - 1)

    End Function

    Public Function GetLastRotation() As Rotation

        Return RotationsList.Item(RotationsList.Count - 1)

    End Function

    Public Sub AddRotation(ByVal NewRotation As Rotation)

        RotationsList.Add(NewRotation)

    End Sub

    Public Sub AddMC(ByVal NewMC As Single)

        MCsList.Add(NewMC)

    End Sub

    Public Function GetRotationCount() As Integer

        Return RotationsList.Count

    End Function

    Public Function GetMC(ByVal index As Integer) As Single

        Return MCsList.Item(index - 1)

    End Function

    Sub Clear()
```

```
        Dim firstRotation As Rotation = RotationsList.Item(0)
        RotationsList.Clear()
        RotationsList.Add(firstRotation)


        MCsList.Clear()

    End Sub

    Public Overrides Function ToString() As String
        Return workerID
    End Function

    Sub RemoveEmptyRotations()
        RotationsList = RotationsList.Where(Function(r)
r.Duration <> 0).ToList()
    End Sub

End Class
```

VII. EC.vb

```
Public Class EC

    Public ReadOnly PLC As Single
    Public ReadOnly PSC As Single
    Public ReadOnly PLS As Single

    Public Sub New(ByVal PLC As Single, ByVal PSC As Single,
ByVal PLS As Single)

        Me.PLC = PLC
        Me.PSC = PSC
        Me.PLS = PLC

    End Sub

End Class
```

Appendix B   Lee's Job Boredom Scale

1	Do you often get bored with your work?
2	Is your work monotonous?
3	Would you like to change from your type of work to another from time to time (if the pay were the same)?
4	How well do you like the work you do?
5	Do you often get tired on the job?
6	Do you find the job dull?
7	Does the job go by slowly?
8	Do you become irritable on the job?
9	Do you get apathetic on the job?
10	Do you get mentally sluggish during the day?
11	Do you get drowsy on the job?
12	Does the time seem to go by slowly?
13	Are there long periods of boredom on the job?
14	Does the job seem repetitive?
15	During the day, do you think about doing another task?
16	Does monotony describe your job?
17	Is your work pretty much the same day after day?