CHESS TOURNAMENT MANAGEMENT SYSTEM
(SCHEDULING MODULE)

LIM JIIN KANG

A thesis submitted in fulfillment
of the requirements for the award of the degree of
Bachelor of Computer Technology ( Software Engineering )

Faculty of Computer System & Software Engineering
University College of Engineering & Technology Malaysia

MARCH, 2005
Scheduling problems has been a major problem in all fields. Each solution required a scheduling system to perform different task according to the need of that solution. Each solution involved different kind of resources and constraints, and some of these constraints are dynamic. Given the uncertainty of a schedule, it is impossible to produce a perfect solution to a problem. Producing a schedule for Chess Tournament has been considering a difficult task because of the constraints that need to be concern in the scheduling process are all dynamic. To fully utilize the resources available in a tournament are almost impossible. To solve the problem in utilizing the resources in a tournament, Genetic Algorithms approach has been used. Genetic Algorithms has been used as an approach in optimizing a solution. A scheduling system build using Genetic Algorithms is expected to produce a schedule that will utilize the usage of resources available.
ABSTRAK

TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DECLARATION</td>
<td>ii</td>
</tr>
<tr>
<td></td>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>ACKNOWLEDGMENTS</td>
<td>iv</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>ABSTRAK</td>
<td>vi</td>
</tr>
<tr>
<td></td>
<td>TABLE OF CONTENTS</td>
<td>vii</td>
</tr>
<tr>
<td></td>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>LIST OF FIGURES</td>
<td>xi</td>
</tr>
<tr>
<td></td>
<td>LIST OF SYMBOLS</td>
<td>xii</td>
</tr>
<tr>
<td></td>
<td>LIST OF APPENDICES</td>
<td>xiii</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1</td>
<td>Scheduling System</td>
<td>1</td>
</tr>
<tr>
<td>1.1.2</td>
<td>Scheduling System Usage</td>
<td>3</td>
</tr>
<tr>
<td>1.1.3</td>
<td>Chess Tournament Management System</td>
<td>4</td>
</tr>
<tr>
<td>1.1.4</td>
<td>Current Scheduling System</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>Problem Statement</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>Objectives</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Scopes</td>
<td>6</td>
</tr>
<tr>
<td>1.4.1</td>
<td>Method and Techniques</td>
<td>6</td>
</tr>
<tr>
<td>1.4.2</td>
<td>Data Range and Limitation</td>
<td>6</td>
</tr>
<tr>
<td>1.5</td>
<td>Thesis Organization</td>
<td>6</td>
</tr>
</tbody>
</table>
2 LITERATURE REVIEW
2.1 The Resource-Constrained Project Scheduling Problem 8
2.2 Scheduling Classification 9
   2.2.1 Static Scheduling vs. Dynamic Scheduling 9
   2.2.2 On-line Scheduling vs. Off-line Scheduling 10
   2.2.3 Real-time Scheduling vs. Dynamic Scheduling 10
2.3 Scheduling Techniques 10
2.4 Genetic Algorithms 13
   2.4.1 Outline of a Basic Genetic Algorithms 14
   2.4.2 Encoding of a Chromosome 14
   2.4.3 Crossover 15
   2.4.4 Mutation 16
2.5 The .NET Platform 16

3 METHODOLOGY 19
3.1 Project Analysis 19
   3.1.1 Tournament System 19
   3.1.2 Tournament Resources 20
3.2 Project Design 21
   3.2.1 Encoding of GA 21
   3.2.2 Fitness Function 22
   3.2.3 Crossover 23
   3.2.4 Mutation 24
3.3 Development and Deployment 25
   3.3.1 Hardware and Software Requirements 25

4 RESULT AND DISCUSSION 27
4.1 Expected Result 27
4.2 Testing Result 27
4.3 Further Research 29
4.4 Constraint

5 Conclusion
5.1 Summary 31
5.2 Archived Objectives 32
5.3 Lesson Learnt 32

REFERENCES 33
APPENDICES 34
LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Representation of a chromosome</td>
<td>15</td>
</tr>
<tr>
<td>2-2</td>
<td>Examples of chromosome crossover</td>
<td>15</td>
</tr>
<tr>
<td>2-3</td>
<td>Examples of chromosome mutation</td>
<td>16</td>
</tr>
<tr>
<td>3-1</td>
<td>Hardware requirements</td>
<td>26</td>
</tr>
<tr>
<td>3-2</td>
<td>Software requirements</td>
<td>26</td>
</tr>
<tr>
<td>4-1</td>
<td>Testing Result</td>
<td>28</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>The Microsoft .NET Platform</td>
<td>17</td>
</tr>
<tr>
<td>3-1</td>
<td>Representation of GA in this project</td>
<td>22</td>
</tr>
<tr>
<td>3-2</td>
<td>Fitness Function</td>
<td>23</td>
</tr>
<tr>
<td>3-3</td>
<td>Crossover of chromosome</td>
<td>24</td>
</tr>
<tr>
<td>3-4</td>
<td>Mutation of chromosome</td>
<td>25</td>
</tr>
</tbody>
</table>
**LIST OF SYMBOLS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>Genetic Algorithms</td>
</tr>
<tr>
<td>VB .NET</td>
<td>Microsoft Visual Basic .NET</td>
</tr>
<tr>
<td>MSSQL</td>
<td>Microsoft SQL Server 2003</td>
</tr>
<tr>
<td>OO</td>
<td>Object-Oriented</td>
</tr>
<tr>
<td>RCPSP</td>
<td>Resource-constrained project scheduling problem</td>
</tr>
</tbody>
</table>
## LIST OF APPENDICES

<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Project Gantt Chart</td>
<td>34</td>
</tr>
<tr>
<td>B</td>
<td>Testing Result</td>
<td>35</td>
</tr>
<tr>
<td>C</td>
<td>User Manual</td>
<td>36</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

In this chapter, an introduction to scheduling will be presented, followed by the problem statement, the objective and scope of the project.

1.1 Introduction

In this part, a brief introduction to scheduling system, its usage and its role in management will be presented.

1.1.1 Scheduling System

Scheduling problems are highly complex problems and, given the uncertain nature of these problems over time, they are that much more difficult to solve. Despite these constraints, however, many scheduling solutions have been proposed.

Although the term “scheduling” can be defined as “that plan for the optimum utilization of production capacity”, successful scheduling is heavily dependent upon
the identification of and "feeding in" of changes that, for a number of reasons, occur in real time. The term "scheduling" has, on various occasions, been defined in other terms too:

1) "Scheduling is the most dynamic activity in cost-effective production (Black, 1986)".

2) "Scheduling is a very important management activity. Effective scheduling in manufacturing and office environments can increase productivity and decrease cost by a significant factor (Amrine et al., 1993)".

3) "Scheduling is a constraint satisfaction activity. The three most important types of scheduling constraints can be seen as time constraints, resource constraints and casual constraints caused by internal or external factors (Smith, 1988)".

"From a functional point of view, scheduling can be defined as a multiple-problem process affected in order to achieve multiple scheduling objectives through performing multiple scheduling tasks in multiple scheduling positions by multiple schedulers (Zhang, 2003)." It becomes abundantly clear, then, that even though the definitions for scheduling may vary, they all have the main objective of scheduling in common, namely to design a reasonable schedule and to execute that schedule on time.

As a schedule is applied in a dynamic environment, it is difficult fully to meet requirements the first time round. Normally, many attempts have to be made at rescheduling before any measure of success is achieved.

The characteristics of scheduling problems can be summed up as follows:

1) Scheduling problems are uncertain, temporary and changeable over time.

2) The three main factors pertaining to these problems, namely the scheduling act, the scheduling environment and the people effecting the scheduling, are all dynamic.
In order to solve a scheduling problem, one must develop a comprehensive scheduling system (Kareem and Moitra 2003), which system should be able to perform the following functions:

1) Absorb and accommodate unforeseen changes.
2) Obtain visual information and knowledge of scheduling, and record all variables.
3) Complete intelligent scheduling strategies.
4) Produce real-time scheduling solutions.
5) Control the scheduling tasks in a dynamic environment.

Scheduling strategies are dynamic and must, therefore, be updated continuously.

1.1.2 Scheduling system usage

Scheduling system has been widely used across all field of expertise. The system were used to schedule for optimize usage of resources in manufacturing sites, flexible schedule for school timetabling, and many more.

Scheduling system mostly used by administrators, managers or authorize personnel. The usage of scheduling system is widely differing from one field to the other. The main reason is because the constraints of each field are different.

Although the system was functioning in a different manner in different field, the purpose of the system is basically the same. The system was used to reduce the burden of the administrator in managing the schedule, the cost of scheduling, and the time of scheduling.

Some scheduling system aim for producing an optimize schedule in the usage of resources, and some aim for producing a flexible schedule where sudden changes can be handles.
1.1.3 Chess Tournament Management System

A Chess Tournament Management System was used by the administrator of the tournament to organize and manage the tournament.

A Chess Tournament Management System should provide the functionality to record the profile of each player, their rating and games played. The system should also manage to perform automatic pairing for different type of tournament system available. The system should also be capable to record all games played throughout the tournament.

1.1.4 Current Scheduling System

Over the past few years, scheduling system has been developed tremendously. Scheduling system is used in manufacturing, management and production site. Scheduling system has been default included in many applications used by the manufacturing, management and production site.

Currently, scheduling system is not included in any Chess Tournament Management System. The new version of Chess Tournament Management System developed by Swiss Perfect Pty. Ltd. will include a scheduling system for chess tournament. The system is expected to be release during the first quarter of 2004.
1.2 Problem Statement

Over the years, Chess has been developed to become the world most popular board game in the western country. Hundreds of tournaments were held each year to acknowledge the superiority of the players.

Chess adopted a very sophisticated system for pairing and scoring in the tournament. These systems can be easily adopted and handle by human if the number of participant is small. If the number of participant get large, it would be troublesome for human to manage the tournament effectively and efficiently.

For a large scale tournament, managing resources and scheduling the date and venue for a game would create a fuss for the tournament administrator since there will be hundreds of players, few dozens of chess equipment, and few tens of venues to choose from. To create more fuss, administrator should consider the resource and time constraint on certain venue.

To overcome the scheduling problem faced by the tournament administrator, a Scheduling module should be included in the Chess Tournament Management System.

1.3 Objectives

The objectives of the project are to:

1) Build a prototype for Chess Tournament Management System Scheduling module.

2) Generate an optimal schedule for tournament with resource constraint.
1.4 Scopes

1.4.1 Method and Technique

The project will be developed with Microsoft Visual Basic .NET (VB .NET) and Microsoft SQL Server 2003 (MSSQL) using Genetic Algorithm (GA). As VB .NET is an Object-Oriented (OO) programming language, the project will use Object-Oriented Modeling as the primary design.

1.4.2 Data Range and Limitation

All data used by the project will be stored in the database. The project will be limited to schedule only for the league system with the constraints of time, matches, venues and arbiters.

1.5 Thesis Organization

This thesis is divided into 5 chapters and each chapter is devoted to discuss different issue in the project. Below is a summary of the content for each chapter:

1) Chapter 1

Introduction to the project is presented along with the project’s problem statement, objectives of the project and the scopes of the project.
2) Chapter 2

Research and literature review related to the project is presented.

3) Chapter 3

Project analysis, design and methodology is presented.

4) Chapter 4

Result from the testing of the system is presented along with statistic and discussion on the result.

5) Chapter 5

Summary of the project is presented.
CHAPTER 2

LITERATURE REVIEW

This chapter will be devoted to a survey of the scheduling concepts found in literature, which include scheduling terminology, types, methods, technology and strategies.

2.1 The Resource-Constrained Project Scheduling Problem

The classical resource-constrained project scheduling problem (RCPSP) discuss by Hartmann (2002) can be summarized as follows. We consider a project which consists of \( J \) activities (jobs) labeled \( j = 1, \ldots, J \). The set of activities is referred to as \( J = \{1, \ldots, J\} \). Due to technological requirements, there are precedence relations between some of the jobs. These precedence relations are given by sets of immediate predecessors \( P_j \) indicating that an activity \( j \) may not be started before all of its predecessors are completed. Analogously, \( S_j \) is the set of the immediate successors of activity \( j \). The precedence relations can be represented by an activity-on-node network which is assumed to be acyclic. We consider additional activities \( j = 0 \) representing the single source and \( j = J + 1 \) representing the single sink activity of the network.
With the exception of the (dummy) source and (dummy) sink activity, each activity requires certain amounts of (renewable) resources to be performed. The set of resources is referred to as $K$. For each resource $k \in K$ the per-period-availability is constant and given by $R_k$. The processing time (duration) of an activity $j$ is denoted as $p_j$, its request for resource $k$ is given by $r_{jk}$. Once started, an activity may not be interrupted. Without loss of generality, we assume that the dummy source and the dummy sink activity have duration of zero periods and no request for any resource. The parameters are assumed to be nonnegative and integer valued. The objective is to determine a schedule with minimal makespan such that both the precedence and resource constraints are fulfilled.

2.2 Scheduling Classification

Scheduling can be classified into three categories: Static Scheduling vs. Dynamic Scheduling; On-line Scheduling vs. Off-line Scheduling; and Real-time Scheduling vs. Dynamic Scheduling.

2.2.1 Static Scheduling vs. Dynamic Scheduling

Static scheduling means the schedule is produce in the preproduction period or the environment where the schedule applied is static. Dynamic scheduling means the schedule can be produce within a volatile environment with necessitates frequent changes to requirements.
2.2.2 On-line Scheduling vs. Off-line Scheduling

On-line scheduling means the scheduling process is done while the production is in progress without influencing the production process. Off-line scheduling, on the other hand, means the scheduling process is being done while production is in down time.

2.2.3 Real-time Scheduling vs. Dynamic Scheduling

Real-time scheduling means the schedule activities is being executed in a real-time environment. All these activities are varying with changes from time to constraint. Dynamic scheduling means the schedule activities has to be revised, owing to the dynamic changes to the environment.

2.3 Scheduling Technique

An overview of scheduling is given in McMahon and Browne (1993). Commonly used scheduling techniques include the following.

1) Gantt chart
2) Operations Research Approaches
3) Algorithms
4) Heuristic approaches
5) Expert Systems

Criteria or performance measures often used within scheduling for ranking jobs into a processing order include, earliest due-date, minimum processing time,
shortest waiting time, lateness, idle-time, number of waiting jobs, processing cost etc., and of course do the job the boss wants.

A Gantt chart is a commonly used manual scheduling tool. The chart consists of a row for each machine and a time axis along the bottom. Jobs are loaded on to machines using colored blocks in the operation sequences of the various jobs. By shuffling the sequence of jobs on individual machines, gaps (idle periods) in the schedule can be reduced. The technique is intuitive but limited to small numbers of both jobs and machines. Computer based Gantt chart type scheduling packages are available.

Operations research methods of scheduling, such as 'dynamic programming' or 'branch and bound' techniques, by searching through multi-dimensional spaces for the 'best' solutions, will give optimal results for the given decision criteria. As long as the decision criteria are weighted correctly the results will be valid. Unfortunately the assigning of weightings, such as cost or set-up time is often arbitrary and relies on value judgments. So the results are often less than optimal. Another problem with this approach to scheduling is that the computational efforts can be vast due to reasons of combinatorial explosion. The computational load can approach that of complete enumeration of the search space.

Scheduling algorithms have been developed but in general they are only applicable to a small number of scheduling problems. Algorithms, including Johnson's and Moore's algorithms, have been developed for single machine, two or three machine and job-shop scheduling problems, but not for more complicated systems which are typical in the modern manufacturing environment. These algorithms could however be used in the multi-machine set-up in the circumstance that one of the machines is a bottleneck to production for example, or where two machines are very closely associated based on product routings. The algorithms could then be applied to the appropriate machine and the rest of the schedule developed around this core by forward and backward scheduling.
For most manufacturing systems with large numbers of machines and many jobs with various routings competing for the various resources an algorithmic solution to the scheduling problem is not possible. In these instances heuristics or 'rules of thumb' are often used for scheduling. These rules of thumb evolve over time through trial and error and based on past experience of what worked. For large scheduling problems 'the best' solution can not be found within real-world time constraints. So the heuristic approach is to develop a schedule based on experience which will work and will also be better than a random or unplanned schedule, a situation which may prevail in some situations e.g. where operators are free to chose the next job which improves their bonus chances based on an piece-work incentive scheme. Heuristic approaches can take considerable computer power and processing time. Examples of individual heuristics would include, the first in first out (FIFO) heuristic, the shortest processing time (SPT) heuristic where jobs are queued in order of ascending processing times, or the modified SPT heuristic which is like the SPT rule but puts jobs that have been waiting over-long to the top of the queue, or the earliest due-date heuristic where jobs are processed in order of due-dates.

The use of expert systems within scheduling can be seen as an extension of the use of heuristics, where the selection of the rules to apply are suggested by the ES based on the encoding of an experts domain specific knowledge. This allows the non-expert to apply the heuristics as the expert scheduler would. By eliciting the expertise of certain key individuals, such as for example a section foreman and encoding this expertise within a set of rules the expertise can be called upon repeatedly and reliably even when the said foreman goes on holidays.

Modern scheduling approaches often use expert systems front end software linked to possibly more conventional scheduling heuristics or algorithms for developing detailed schedules. The expert systems may often act as expert scheduler assistant rather than a stand alone 'expert' scheduler. The systems may suggest search heuristics to the scheduler / operator in certain conditions, and then the system would carry out the heuristic search with possibly an algorithmic base to kick off.
2.4 Genetic Algorithms

Genetic Algorithms are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of Genetic Algorithms is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

First pioneered by John Holland in the 60s, Genetic Algorithms has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide an alternative method to solving problem, it consistently outperforms other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for Genetic Algorithms.

Genetic Algorithms was introduced as a computational analogy of adaptive systems. They are modeled loosely on the principles of the evolution via natural selection, employing a population of individuals that undergo selection in the presence of variation-inducing operators such as mutation and recombination (crossover). A fitness function is used to evaluate individuals, and reproductive success varies with fitness.

Algorithm begins with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are then selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce.

This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.
2.4.1 Outline of a Basic Genetic Algorithms

Below is the basic algorithm for GA:

1) [Start] Generate random population of n chromosomes (suitable solutions for the problem).
2) [Fitness] Evaluate the fitness \( f(x) \) of each chromosome \( x \) in the population.
3) [New population] Create a new population by repeating following steps until the new population is complete.
   a. [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).
   b. [Crossover] With a crossover probability cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
   c. [Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome).
   d. [Accepting] Place new offspring in the new population.
4) [Replace] Use new generated population for a further run of the algorithm.
5) [Test] If the end condition is satisfied, stop, and return the best solution in current population.
6) [Loop] Go to step 2.

2.4.2 Encoding of a Chromosome

A chromosome should in some way contain information about solution that it represents. The most used way of encoding is a binary string. A chromosome then could look like this: