KOLEJ UNIVERSITI KEJURUTERAAN DAN TEKNOLOGI MALAYSIA

BORANG PENGESAHAN STATUS TESIS

JUDUL **CHESS TOURNAMENT MANAGEMENT SYSTEM (SCHEDULING MODULE)**

SESI PENGAJIAN: **2004/2005**

Saya **LIM JIIN KANG**
(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah)* ini disimpan di Perpustakaan Kolej Universiti Kejuruteraan dan Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Kolej Universiti Kejuruteraan dan Teknologi Malaysia.
2. Perpustakaan Kolej Universiti Kejuruteraan dan Teknologi Malaysia dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (✓)

☐	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
✓	TIDAK TERHAD	

Disahkan oleh

(TANDATANGAN PENULIS) (TANDATANGAN PENYELIA)

Alamat Tetap: **35, JLN MERBOK 2, TMN BLOSSOM PARK, 70100 SEREMBAN, NS**

SUPERVISOR
Wan Muhammad Syahrir

Tarikh: 22nd March 2005

Tarikh: 22nd March 2005

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.
◆ Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertai bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjans Muda (PSM).

"I hereby acknowledge that I had read this technical writing and in my opinion this technical writing is sufficient in terms of scope and quality for the purpose of the granting of Bachelor of Computer Technology (Software Engineering)."

Signature : ……………………………………………

Name of Supervisor : Wan Muhammad Syahrir

Date : 22nd March 2005

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

DECLARATION

I declare that this thesis entitled "Chess Tournament Management System (Scheduling module)" is the result of my own research except as cited in references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree."

Signature : ....

Name of Candidate : Lim Jiin Kang

Date : March 22nd , 2002

DEDICATION

Special Dedication to my family members that always love me,
My friends, my fellow colleague
and all faculty members

For all your Care, Support and Believe in me.

Sincerely
Lim Jiin Kang

ACKNOWLEDGEMENTS

I would like to forward my appreciation to my thesis supervisor, Mr. Wan Muhammad Syahrir, for his guidance and support. I would also very thankful to my academic advisor, Assoc. Prof. Ruzaini Abdullah Arshah, for his support and believe in me during my studies. Special thanks to, Mr. Adzhar Kamaluddin, whom build my interest in computer technology field.

I'm very thankful to University College of Engineering & Technology Malaysia (UCET) for providing good facilities in the campus. To all the staff in Faculty of Computer System & Software Engineering, a very big thanks you to all.

My fellow colleague should be noted for their support. Thank you for the time sacrificed to accompany me when I'm down and the time we share in our University life.

ABSTRACT

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

Masalah penjadualan telah menjadi masalah utama di pelbagai bidang. Setiap penyelesaian memerlukan sistem penjadualan untuk menjalankan tugas yang berlainan mengikut keperluan bidang tersebut. Setiap penyelesaian merangkumi pelbagai jenis sumber dan kekangan, dan sesetengah kekangan ini adalah dinamik. Disebabkan oleh ketidakpastian penjadualan, adalah mustahil untuk menghasilkan satu penyelesaian yang sempurna. Penghasilan jadual untuk Pertandingan Catur telah dianggap sebagai tugas yang berat kerana kekangan yang wujud adalah dinamik. Penggunaan kesemua sumber yang sedia ada untuk pertandingan dengan optimum adalah mustahil. Algoritma Genetik telah digunakan untuk menyelesaikan masalah penggunaan sumber dengan optimum dalam satu pertandingan. Algoritma Genetik telah digunakan untuk optimumkan penggunaan sumber. Satu sistem penjadualan yang dibina menggunakan Algoritma Genetik dijangka akan menghasilkan jadual yang optimum dalam penggunaan sumber.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

GA	-	Genetic Algorithms
VB .NET	-	Microsoft Visual Basic .NET
MSSQL	-	Microsoft SQL Server 2003
OO	-	Object-Oriented
RCPSP	-	Resource-constrained project scheduling problem

LIST OF APPENDICES

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 $\mathrm{J} = \{1, \ldots, J\}$. 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 2 K the per-period-availability is constant and given by Rk. The processing time (duration) of an activity j is denoted as pj, 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:

Table 2-1 : Representation of a chromosome

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

Each chromosome is represented by a binary string. Each bit in the string can represent some characteristics of the solution. Another possibility is that the whole string can represent a number.

Of course, there are many other ways of encoding. The encoding depends mainly on the solved problem. For example, one can encode directly integer or real numbers; sometimes it is useful to encode some permutations (Hartmann, 1998) and so on.

2.4.3 Crossover

After we have decided what encoding we will use, we can proceed to crossover operation. Crossover operates on selected genes from parent chromosomes and creates new offspring. The simplest way how to do that is to choose randomly some crossover point and copy everything before this point from the first parent and then copy everything after the crossover point from the other parent.

Crossover can be illustrated as follows: (* is the crossover point):

Table 2-2 : Examples of chromosome crossover

Chromosome 1	11011*00100110110
Chromosome 2	11011*11000011110
Offspring 1	11011*11000011110
Offspring 2	11011*00100110110

There are other ways how to make crossover, for example we can choose more crossover points. Crossover can be quite complicated and depends mainly on the encoding of chromosomes (Hartmann, 2002). Specific crossover made for a specific problem can improve performance of the genetic algorithm.

2.4.4 Mutation

After a crossover is performed, mutation takes place. Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem (Hartmann, 1998). Mutation operation randomly changes the offspring resulted from crossover. In case of binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can be then illustrated as follows:

Table 2-3 : Examples of chromosome mutation

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110100

The technique of mutation (as well as crossover) depends mainly on the encoding of chromosomes. For example when we are encoding permutations, mutation could be performed as an exchange of two genes

2.5 The .NET Platform

The Microsoft .NET Platform consists of five main components, as shown in Figure 2-1. At the lowest layer lies the operating system (OS), which can be one of a variety of Windows platforms, including Windows XP, Windows 2000, Windows

Me, and Windows CE or other operating system like Linux or MAC OS. As part of the .NET strategy, Microsoft has promised to deliver more .NET device software to facilitate a new generation of smart devices.

On top of the operating system is a series of .NET Enterprise Server products that simplify and shorten the time required to develop and manage large-scale business systems. These server products include Application Center 2000, BizTalk Server 2000, Commerce Server 2000, Exchange Server 2000, Host Integration Server 2000, Internet Security and Acceleration Server 2000, and SQL Server 2000.

Since Web Services are highly reusable across the Web, Microsoft plans to provide a number of building-block services that application developers can use, for a fee. An example of building –block service is Microsoft Passport, which allows you to use a single username and password at all web sites that support Passport authentication. On March 19, 2001, Microsoft announced another set of Web Services w ith the codename HailStorm. This product encompasses a set of building –block services that support personalization, centered entirely on consistent user experiences. Microsoft plans to add newer services, such as calendar, directory, and search services. Third-party vendors are also creating new Web Services of their own.

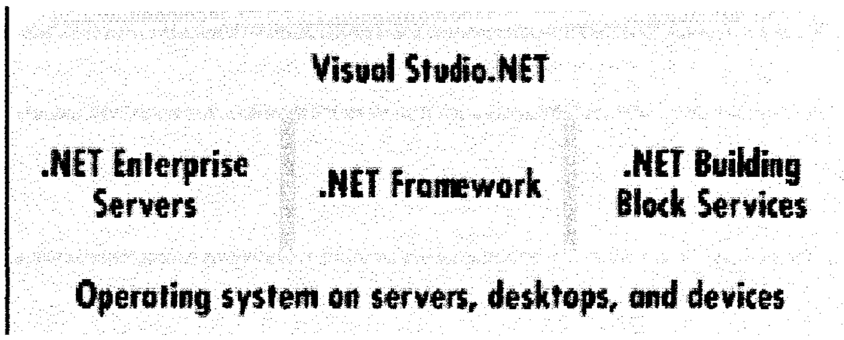


Figure 2-1: The Microsoft .NET Platform

At the top layer of the .NET architecture is a brand new development tool called Visual Studio.NET (VS.NET), which makes possible the rapid development of Web Services and other applications. A successor of Microsoft Visual Studio 6.0, VS.NET is an Integrated Development Environment (IDE) that supports four different languages and features such as cross-language debugging and the XML Schema Editor.

And at the center of .NET is the Microsoft .NET Framework. The .NET Framework is a new development and runtime infrastructure that will change the development of business applications on the Windows platform. It includes the Common Language Runtime (CLR) and a common framework of classes that can be used by all .NET languages.

CHAPTER 3

METHODOLOGY

This chapter will be devoted to discuss the project planning, analysis, design and developmental issues.

3.1 Project Analysis

3.1.1 Tournament Systems

Basically there are two different systems being used at most tournaments; the knock-out (elimination) system and the league system. There are also tournaments that used a hybrid system where it will start with a league system and end using a knock-out system.

A knock-out or elimination system is used to rapidly find the best player in a large pool of participants. This is archive by eliminating the loser of each round from the tournament. A standard knock-out system will used 7 rounds for a 128 participant's tournament.

A league system is used to find the best player in a large pool of participants by having each player playing for each round and calculate the cumulative points gain each round. The league system can further divided into round-robin system, multi-round round-robin system and the ever popular Swiss system.

In a round-robin system, every participant will play each other participant. The system requires N-1 rounds of play for N participants. The cumulative points at the end of the tournament will be used to decide the winner.

In a multi-round round-robin system, every participant will play each other participant more than once. For example, in a double round-robin system, the same two players will play each other twice with the reverse home venue.

The Swiss system is a pairing system where round by round players with equal or most similar number of points so far will meet each other at the board. Thus, with a round number far less than in round robin tournaments the most accurate ranking can be achieved. Most tournaments are played using the Swiss pairing system.

3.1.2 Tournament's Resource

In a tournament there are many resources needed to enable the tournament to run smoothly. In general, all tournaments involve the allocation of venue and judges. In particular, Chess tournament involve the allocation of equipments like the chess set which consist of a chess board and chess pieces, chess clock, tables and chairs to name a few.

All these resources each have their own constraints and limit. Examples of constraint would be the availability of a venue or arbiter. Other more critical constraints would be the limitation in the quantity of certain equipments.

3.2 Project Design

This part of the chapter will be devoted to discuss the design of the project in term of program architecture and chromosome encoding.

3.2.1 Encoding of GA

In GA, a population was form as a group of chromosome. A chromosome was form as a series of genes and a gene was form as a collection of information.

In this project, as OO programming was implemented, a gene was represented as an object which stores certain information. A chromosome was in turn form as an object by itself which contains a series of the gene objects. A population will be a group of chromosome objects. A representation on the relationship between chromosomes and genes can be seen in Figure 3-1.

As been seen in Figure 3-1, a solution was affected by the characteristics of a chromosome. The characteristics of a chromosome were form using the information carried by the genes that form the chromosome. As in this project, the information that affects the outcome of the solution would be the number of arbiter and venue available as resources.

A population of schedules will be initialized and assigned a fitness value. Then the population will be going through a series of evolution which consists of crossover, mutation and the selection process.

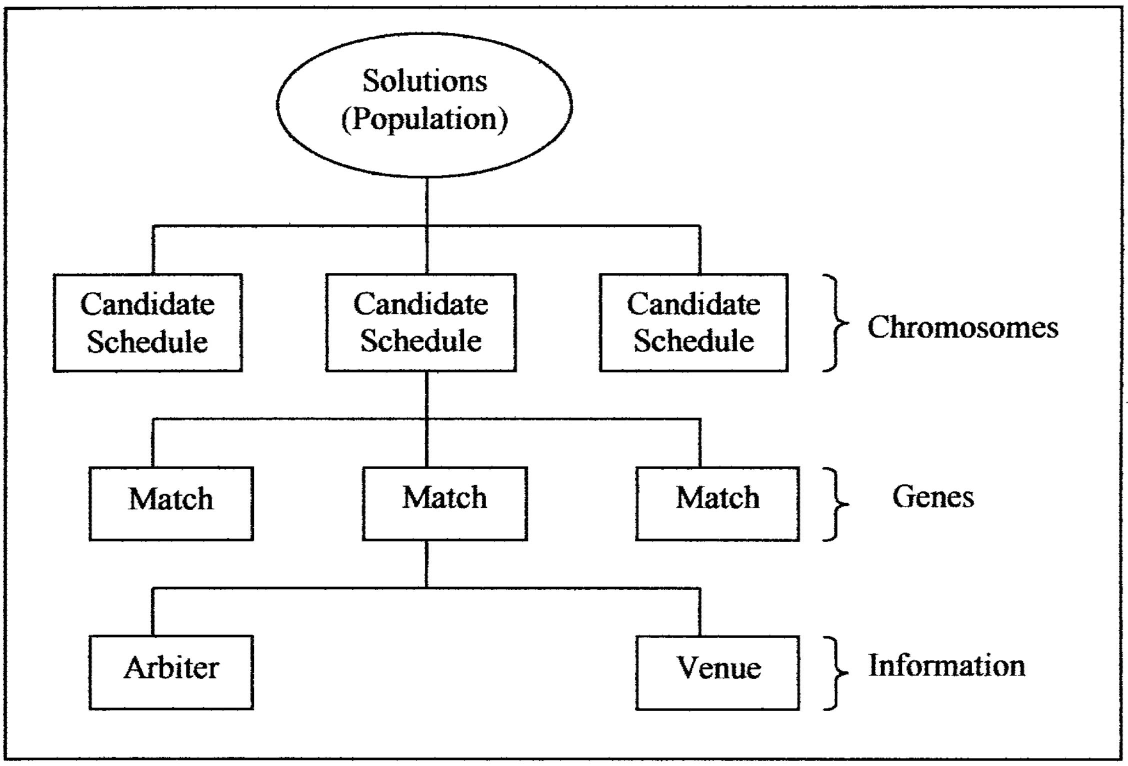


Figure 3-1 : Representation of GA in this project

3.2.2 Fitness Function

A fitness function was used to determine the candidate schedule that will be chose to form the new generation of solutions. A candidate schedule with higher fitness value will have more chances for it to be chosen to form the new generation.

The fitness value of a candidate schedule was calculated based on the following rules:

1) The time span for the schedule.
2) The number of resources consumes.

A shorter time span schedule will score a higher fitness value than a longer time span schedule. The fitness value will be penalized by the number of resources consumes. Each resource will have a multiplier to be included in the calculation of the fitness value. The more resources consumes, the more fitness value will be deducted. This fitness function was form to generate an optimize schedule than have the shortest time span and consuming fewest resources. The formula to calculate the fitness value was presented as below:

$$F = \frac{T - t}{\sum (R_i * m_i)}$$

where

F	: Fitness Value
T	: Expected time span
t	: Solution time span
Ri	: Resource i
mi	: Resource i multiplier

Figure 3-2: Fitness Function

3.2.3 Crossover

After all schedules have been assigning a fitness value, crossover of the characteristics that form the schedules will be executed to form a new generation of schedules. Crossover will be made on the list of resources used in the selected schedules. An example of crossover is illustrated in Figure 3-2.

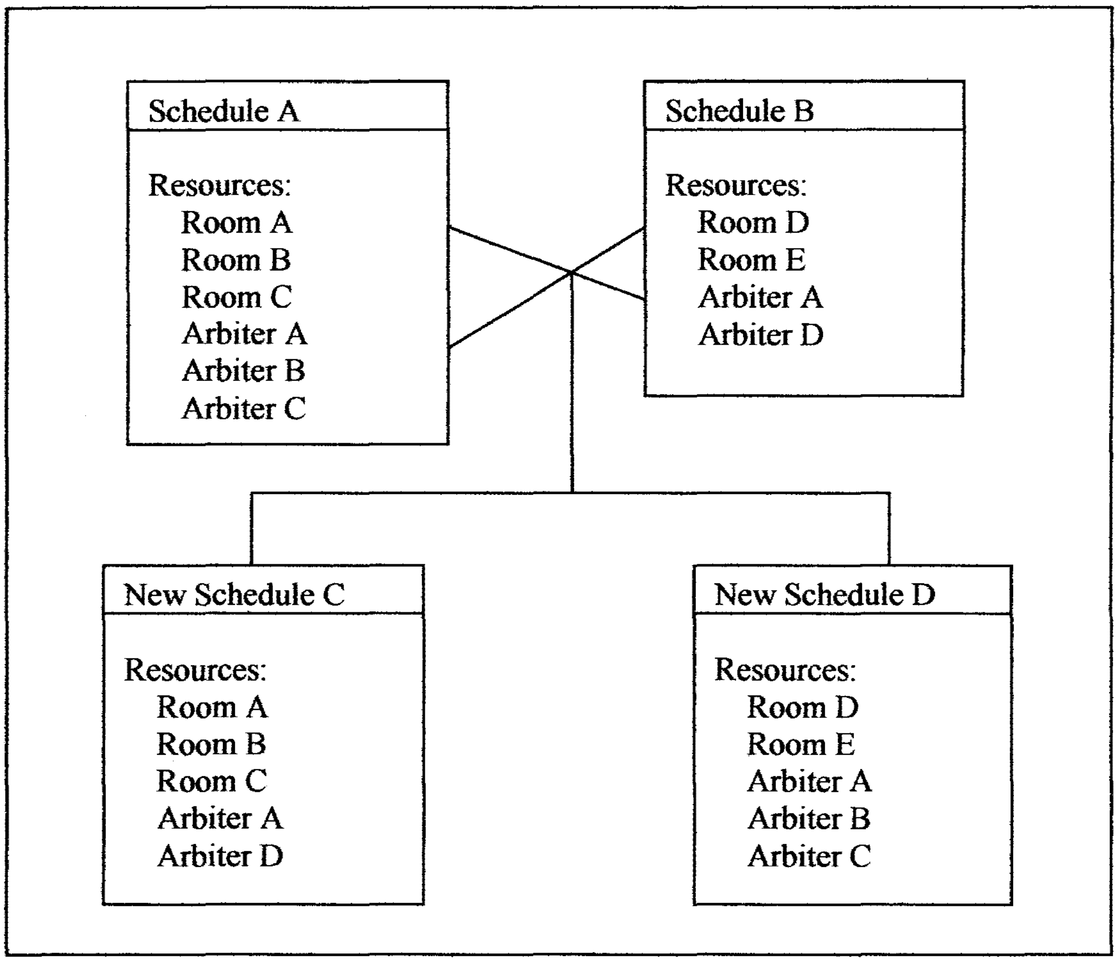


Figure 3-3 : Crossover of chromosome

3.2.4 Mutation

After the crossover process, part of the newly generated schedules will undergo a mutation process. The mutation process will randomly change the resources consume by the selected schedule to a new valid quantity. An example of mutation is illustrated in Figure 3-3.

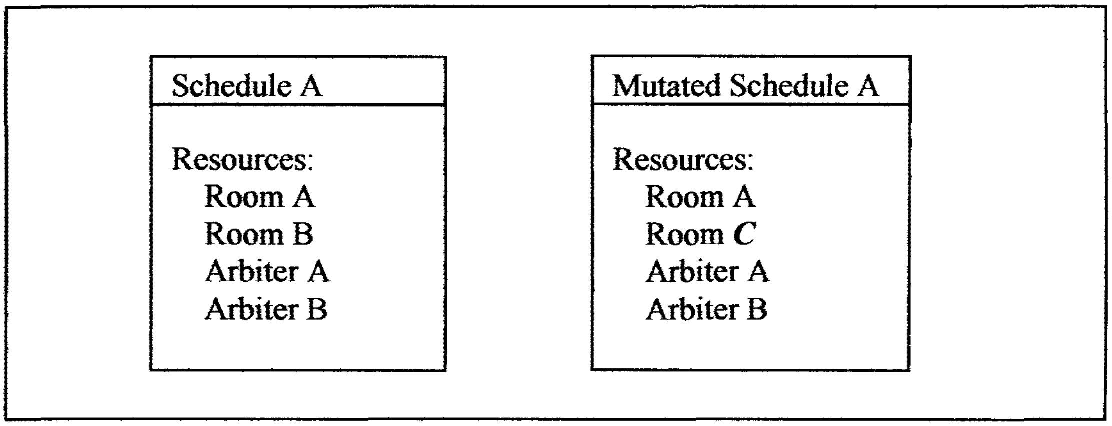


Figure 3-4 : Mutation on chromosome

3.3 Development and Deployment

Due to that this project only develop a prototype for the Chess Tournament Management System Scheduling sub-system, the front end of the system was build simple using VB .NET and the database used was MSSQL for easy connection and management.

The system will be deployed as a window application running in any operating system environment with the .NET Framework installed.

3.3.1 Hardware and Software Requirements

The hardware used to develop the system is as listed below:

Table 3-1 : Hardware requirements

Item	Required
Processor	Pentium 4 1.4Ghz
RAM	128Mb
CD-ROM	24x CD-ROM
Hard Disk	20 GB
Wireless Card	Hyundai e-Life wireless LAN PCMCIA Card

The software required to develop the system are as listed below:

Table 3-2 : Software requirements

Item	Name	Purposes
Operating System	Microsoft Windows XP Professional	The environment to run the .NET framework.
Software	Microsoft Visual Basic .NET	Develop the front end and back end of the system.
	Microsoft SQL Server 2003	Store data used by the system.
	Microsoft Office Word 2003	Document the thesis.
	Microsoft Office Project 2003	Document the Project Gantt Chart.
	Adobe Acrobat Reader 5.0	To read internet resources.

CHAPTER 4

RESULT AND DISCUSSION

This chapter will be devoted for the project output analysis and discussion on the constraints and future enhancements on the project.

4.1 Expected Result

The system was expected to run without errors and capable of generating a near optimized schedule every time proper input of data was provided.

4.2 Testing Result

After a series of testing with different value of input, a set of testing result was produced. The result set show that with the increasing of the initial population value or the generation value, the schedules that produced by the system are more optimized but significantly degrade the performance of the system. Below is shown the result for the testing on the system:

Table 4-1: Testing Result

ACT	VENUE	ARBITER	POP	GEN	MUTATION	FITNESS	TIME *SEC)
15	*2*	*2*	*50*	*5*	*20*	77.57	3.18
15	2	2	**100**	5	20	82.46	10.81
15	2	2	50	**10**	20	98.69	6.17
15	2	2	50	5	**40**	80.54	3.05
15	2	2	**150**	5	20	91.66	23.54
15	2	2	50	**15**	20	109.44	9.22
15	2	2	50	5	**60**	77.95	3.06
24	*2*	*2*	*50*	*5*	*20*	71.97	3.06
24	2	2	**100**	5	20	90.52	10.86
24	2	2	50	**10**	20	69.33	6.15
24	2	2	50	5	**40**	74.19	3.07
24	2	2	**150**	5	20	78.27	23.57
24	2	2	50	**15**	20	83.98	9.18
24	2	2	50	5	**60**	74.74	3.16
64	*2*	*2*	*50*	*5*	*20*	50.96	3.3
64	2	2	**100**	5	20	51.84	11.25
64	2	2	50	**10**	20	52.06	6.45
64	2	2	50	5	**40**	50.69	3.37
64	2	2	**150**	5	20	52.06	24.26
64	2	2	50	**15**	20	51.53	9.66
64	2	2	50	5	**60**	51.18	3.4

The increase in mutation rate doesn't guaranteed a more optimize schedule. This is due to the fact that the mutation process is a random process. The system will randomly mutate the solution, in hope, to generate a more optimize schedule. This doesn't always work the way it plan. Random mutation sometimes will mutate a high fitness value schedule and thus producing a lower fitness value schedule.

The increase of population limit guaranteed a more optimized schedule. But with a small increase in population limit degrade the performance of the system to about 30 percent. This is due the fact that the numbers of schedule need to be process were increase.

The increase of generation also guaranteed a more optimized schedule. Although the value increases cause the degradation of the performance, but the system still run at a reasonable performance level. This is due to the fact that the increases of generation value only cause the system to repeat the same process more.

4.3 Further Research

Further research and development can be done to enhance the system by developing a more user friendly front end.

Enhancement can be done at the scheduling engine. Rebuilding the engine using expert system can greatly improve the system performance in generating an optimized and flexible schedule.

To further enhance the performance of the system, an intelligent agent can be develop to determine the optimize value for population limit, generation and mutation rate to be used for scheduling.

The chromosome representation method proposes in this thesis can be used to implemented in scheduling solution for other field of research.

4.4 Constraint

Following are among the constraints of the system that were identified during the project development process. They are:

1) Maximum Population Limit

The population limit used to optimize the scheduling process is not advisable to go beyond 200 and always set to an even number. This is because the performance of the system degrades tremendously with the increase in the population limit. Furthermore, the increase in the population limit does not guaranteed to produce a more optimize schedule. An even number makes the algorithms easier to manage in the selection phase.

2) Resource Constraints Management

The system were capable to generate an optimize schedule with resources constraints. But the number of resources involves are limited to 10 due to the lack of a Resource Constraints Management module in the system.

3) Pairing system

The system is lacking with a input module. A pairing system is needed to generate the input needed for the scheduling to work properly. Currently, all input is directly store into the database without using a pairing module.

CHAPTER 5

CONCLUSION

This chapter is devoted to conclude and summarize the project overall performance.

5.1 Summary

As a conclusion, the overall performance of this project is acceptable. Although there are certain constraints arise while developing the system, the development of the system was able to meet the due date. The system has shown that scheduling can be optimized with the used of GA. GA is used in the scheduling process to fully optimized the usage of resources involved in the schedule. To develop a more powerful and flexible scheduling system, the usage of other scheduling techniques should be used in conjunction with GA to produce a flexible and optimize schedule. All identified constraints and lesson learnt during the project development were well documented for reference purposes.

5.2 Achieved Objectives

All objectives stated in this thesis are archived. A prototype for the Chess Tournament Management System Scheduling sub-system was developed. The prototype system was running well and is capable of generating an optimize schedule.

5.3 Lesson Learnt

1) Project Planning

Proper project planning is needed in order to successfully develop a high quality product. Without a good project planning, everything will become a mess. A good project planning and execution ensure the project finish on time and all due date is met.

2) Time Management

Time management really plays a huge role in this project. Besides having to research and develop this project, there are others subjects that require a lot of attentions. Assignments and tests has become one major barrier in this project. Another major barrier in this project is the amount of official holiday during the development of this project.

3) Rapid Application Development with Microsoft Visual Basic .NET

Developing application has never been any easier with VB .NET and SQL Server 2003. A user friendly front end can be develop in a matter of time.

REFERENCES

Amrine, H.T., Ritchey, J.A., Moodie, C.L., Kmec, J.F, (1993). *Manufacturing, organizing and management.* Prentice Hall International Inc., USA.

Black, W.J. (1986). *Intelligent knowledge-based system: An Introduction.* Van Nostrand Reinhold (UK) Co. Ltd.

Hartmann, S (1998). *A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling.* Naval Research Logistics, 45:733-750.

Hartmann, S. (2002). *A Self-Adapting Genetic Algorithm for Project Scheduling under Resource Constraints.* Naval Research Logistics, 49:433-448.

Kareem, S.A., and Moitra, A. (2003). *Advances in Schedule Optimization With Genetic Algorithms.* GE Global Research, 2003GRC111.

McMahon, C., Browne, J. (1993), *CADCAM, From principles to practice,* Addison-Wesley, England.

Smith, S.F. (1988). *A constraint-based framework for reactive management of factory schedules.* Intelligent Manufacturing Proceedings.

Sriprasert, E., and Dawood, N. (2003). *Genetic Algorithms for Multi-Constraint Scheduling: An Application for the Construction Industry.* Centre of Construction Innovation Research, University of Teesside.

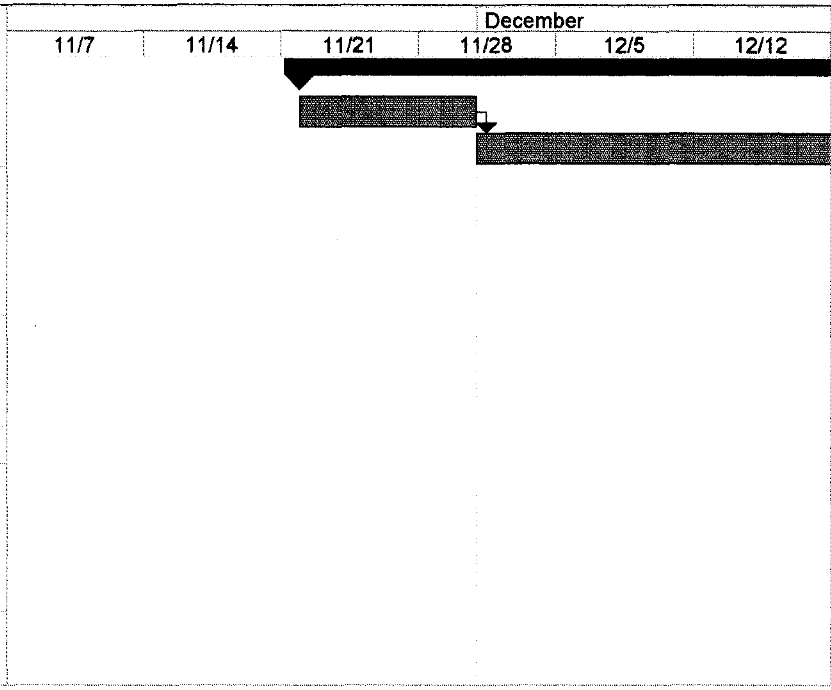
Zhang, X. (2003). *Interactive event-based intelligent scheduling.* Rand Afrikaans University.

Appendix A

Project Gantt Chart

Tournament Management System (TMS)

ID	Task Name	Duration	Start	Finish
1	**Tournament Management System (TMS)**	**85 days**	**Mon 11/22/04**	**Sun 3/20/05**
2	System Development Plan	7 days	Mon 11/22/04	Tue 11/30/04
3	System Requirement Analysis	20 days	Wed 12/1/04	Tue 12/28/04
4	**System Design**	**23 days**	**Wed 12/29/04**	**Fri 1/28/05**
5	System Flow Design	15 days	Wed 12/29/04	Tue 1/18/05
6	Front-end Interface Design	4 days	Wed 1/19/05	Mon 1/24/05
7	Back-end Interface Design	4 days	Tue 1/25/05	Fri 1/28/05
8	**System Development**	**20 days**	**Mon 1/31/05**	**Fri 2/25/05**
9	Resource Scheduling Engine Development	20 days	Mon 1/31/05	Fri 2/25/05
10	Presentation on Proposal Progress	0 days	Thu 1/27/05	Thu 1/27/05
11	**Integrated Testing**	**2 days**	**Mon 2/28/05**	**Tue 3/1/05**
12	Testing	2 days	Mon 2/28/05	Tue 3/1/05
13	Presentation on PSM	0 days	Wed 3/2/05	Wed 3/2/05
14	**System Debugging**	**13 days**	**Wed 3/2/05**	**Fri 3/18/05**
15	Debugging	11 days	Wed 3/2/05	Wed 3/16/05
16	Testing	2 days	Thu 3/17/05	Fri 3/18/05
17	Submission on PSM	0 days	Sun 3/20/05	Sun 3/20/05



Project: Tournament Management Sys
Date: Tue 3/22/05

Task		Milestone	◆	External Tasks	
Split		Summary		External Milestone	◆
Progress		Project Summary		Deadline	

Tournament Management System (TMS)

Project: Tournament Management Sys Date: Tue 3/22/05	Task		Milestone	◆	External Tasks	
	Split		Summary		External Milestone	◆
	Progress		Project Summary		Deadline	

Appendix B

Testing Result

ACT	VENT	ARBITER	POPULATION	GENERATION	MUTATION	FITNESS	TIME SEC
15	2	2	50	5	20	60	3.63
15	2	2	50	5	20	69.23	3.1
15	2	2	50	5	20	92.31	3.05
15	2	2	50	5	20	77.42	3.07
15	2	2	50	5	20	88.89	3.06
						77.57	3.18
15	2	2	100	5	20	72	10.89
15	2	2	100	5	20	82.76	10.82
15	2	2	100	5	20	69.23	10.77
15	2	2	100	5	20	92.31	10.8
15	2	2	100	5	20	96	10.77
						82.46	10.81
15	2	2	50	10	20	96	6.23
15	2	2	50	10	20	80	6.19
15	2	2	50	10	20	114.29	6.12
15	2	2	50	10	20	114.29	6.18
15	2	2	50	10	20	88.89	6.11
						98.69	6.17
15	2	2	50	5	40	85.71	3.13
15	2	2	50	5	40	63.16	2.99
15	2	2	50	5	40	61.54	3.08
15	2	2	50	5	40	100	2.98
15	2	2	50	5	40	92.31	3.06
						80.54	3.05
15	2	2	150	5	20	85.71	23.73
15	2	2	150	5	20	85.71	23.55
15	2	2	150	5	20	88.89	23.56
15	2	2	150	5	20	109.09	23.46
15	2	2	150	5	20	88.89	23.42
						91.66	23.54
15	2	2	50	15	20	104.35	9.16
15	2	2	50	15	20	114.29	9.41
15	2	2	50	15	20	114.29	9.16
15	2	2	50	15	20	100	9.18
15	2	2	50	15	20	114.29	9.21
						109.44	9.22
15	2	2	50	5	60	85.71	3.09
15	2	2	50	5	60	92.31	3.04
15	2	2	50	5	60	58.54	3.06
15	2	2	50	5	60	88.89	3.02
15	2	2	50	5	60	64.29	3.08
						77.95	3.06
24	2	2	50	5	20	63.16	3.09
24	2	2	50	5	20	61.22	3.06
24	2	2	50	5	20	63.16	3.04

24	2	2	50	5	20	80	3.08
24	2	2	50	5	20	92.31	3.05
						71.97	3.06
24	2	2	100	5	20	81.82	10.97
24	2	2	100	5	20	96	10.83
24	2	2	100	5	20	82.76	10.87
24	2	2	100	5	20	96	10.7
24	2	2	100	5	20	96	10.91
						90.52	10.86
24	2	2	50	10	20	72	6.2
24	2	2	50	10	20	64.86	6.11
24	2	2	50	10	20	68.57	6.13
24	2	2	50	10	20	69.23	6.18
24	2	2	50	10	20	72	6.11
						69.33	6.15
24	2	2	50	5	40	82.76	3.06
24	2	2	50	5	40	55.81	3.1
24	2	2	50	5	40	80	3.04
24	2	2	50	5	40	66.67	3.07
24	2	2	50	5	40	85.71	3.06
						74.19	3.07
24	2	2	150	5	20	85.71	23.57
24	2	2	150	5	20	68.57	23.46
24	2	2	150	5	20	75	23.67
24	2	2	150	5	20	92.31	23.51
24	2	2	150	5	20	69.77	23.64
						78.27	23.57
24	2	2	50	15	20	100	9.12
24	2	2	50	15	20	104.35	9.19
24	2	2	50	15	20	66.67	9.14
24	2	2	50	15	20	85.71	9.29
24	2	2	50	15	20	63.16	9.17
						83.98	9.18
24	2	2	50	5	60	96	3.1
24	2	2	50	5	60	68.57	3.11
24	2	2	50	5	60	68.57	3.39
24	2	2	50	5	60	77.42	3.13
24	2	2	50	5	60	63.16	3.09
						74.74	3.16
64	2	2	50	5	20	51.61	3.3
64	2	2	50	5	20	51.06	3.29
64	2	2	50	5	20	51.61	3.28
64	2	2	50	5	20	50	3.3
64	2	2	50	5	20	50.53	3.32
						50.96	3.3
64	2	2	100	5	20	52.17	11.27

64	2	2	100	5	20	52.17	11.23
64	2	2	100	5	20	52.17	11.21
64	2	2	100	5	20	51.61	11.25
64	2	2	100	5	20	51.06	11.25
						51.84	11.25
64	2	2	50	10	20	52.17	6.44
64	2	2	50	10	20	52.17	6.45
64	2	2	50	10	20	52.17	6.43
64	2	2	50	10	20	51.61	6.5
64	2	2	50	10	20	52.17	6.43
						52.06	6.45
64	2	2	50	5	40	47.52	3.38
64	2	2	50	5	40	51.61	3.33
64	2	2	50	5	40	50	3.38
64	2	2	50	5	40	52.17	3.36
64	2	2	50	5	40	52.17	3.38
						50.69	3.37
64	2	2	150	5	20	51.61	24.22
64	2	2	150	5	20	52.17	24.33
64	2	2	150	5	20	52.17	24.21
64	2	2	150	5	20	52.17	24.32
64	2	2	150	5	20	52.17	24.2
						52.06	24.26
64	2	2	50	15	20	52.17	9.63
64	2	2	50	15	20	52.17	9.57
64	2	2	50	15	20	52.17	9.7
64	2	2	50	15	20	52.17	9.63
64	2	2	50	15	20	48.98	9.76
						51.53	9.66
64	2	2	50	5	60	51.61	3.41
64	2	2	50	5	60	49.48	3.39
64	2	2	50	5	60	51.61	3.38
64	2	2	50	5	60	51.61	3.42
64	2	2	50	5	60	51.61	3.39
						51.18	3.4

Appendix C

User Manual

CHESS TOURNAMENT MANAGEMENT SYSTEM

SCHEDULING SUB-SYSTEM

USER MANUAL

Interface Description

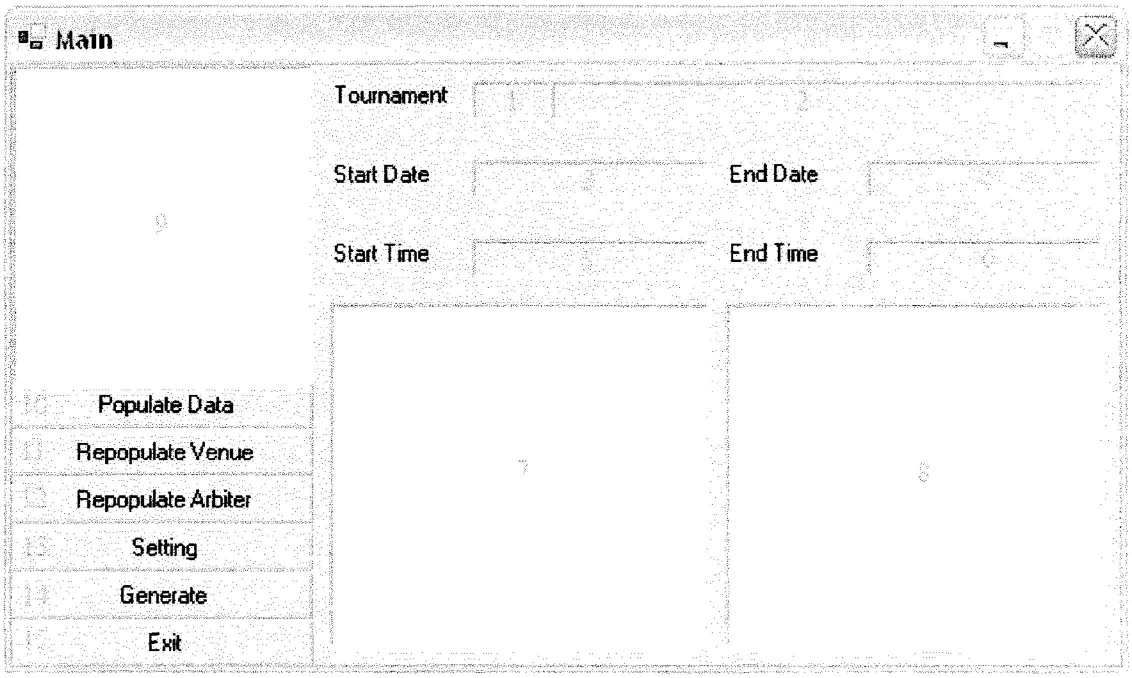


Figure 1: Program main interface

	Name	Usage
1	Tournament ID	Indicating the ID of currently selected tournament.
2	Tournament Name	Indicating the name of currently selected tournament.
3	Start Date	Indicating the Start Date of currently selected tournament.
4	End Date	Indicating the End Date of currently selected tournament.
5	Start Time	Indicating the Start Time of currently selected tournament.
6	End Time	Indicating the End Time of currently selected tournament.
7	Venue List	List all available Venues.
8	Arbiter List	List all available Arbiters.
9	Tournament List	List all Tournaments.

10	**Populate Data**	**Load the data from databases into the program.**
11	**Repopulate Venue**	**Reloading the Venue data.**
12	**Repopulate Arbiter**	**Reloading the Arbiter data.**
13	**Setting**	**Set the setting for the program.**
14	**Generate**	**Start generating the schedule.**
15	**Exit**	**Exit the program.**

Table 1: Listing of items

1. **Start the program by populating the data into the program. Figure 2 show the screen shot for the program after populated with data.**

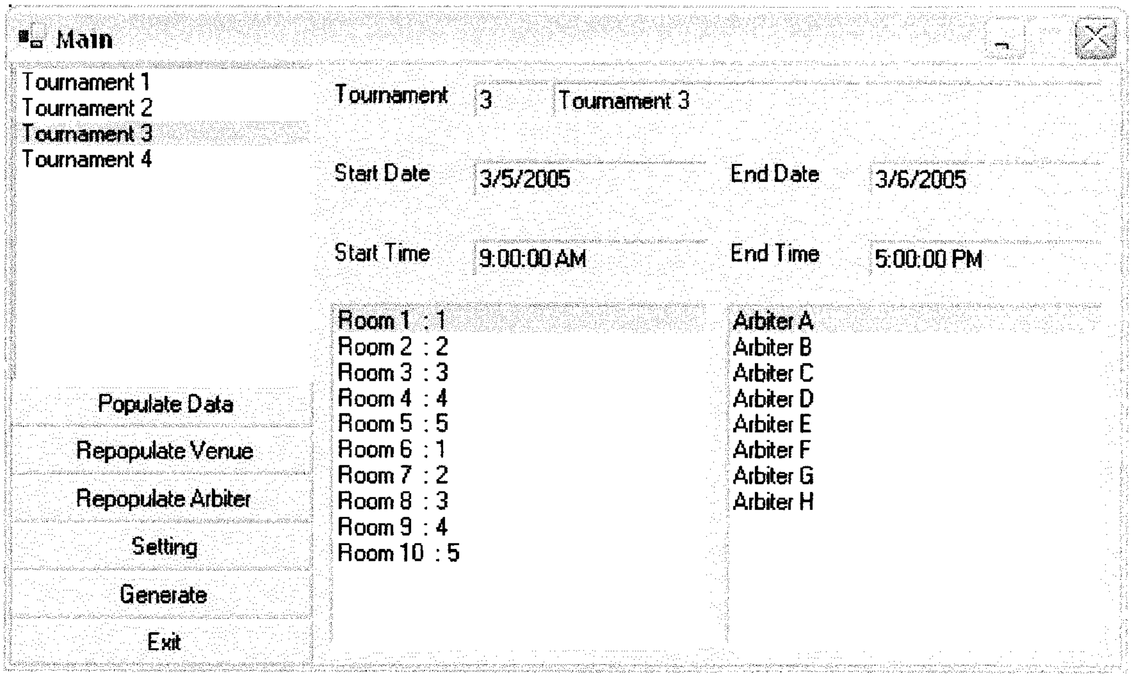


Figure 2: Data loaded into the program

2. **Select the desired Tournament to be schedule.**
3. **The information for the selected Tournament will be available in item 1 – 6.**
4. **Click on the unwanted resources to remove it from the list. Figure 3 show the screen shot after the unwanted resources are remove.**

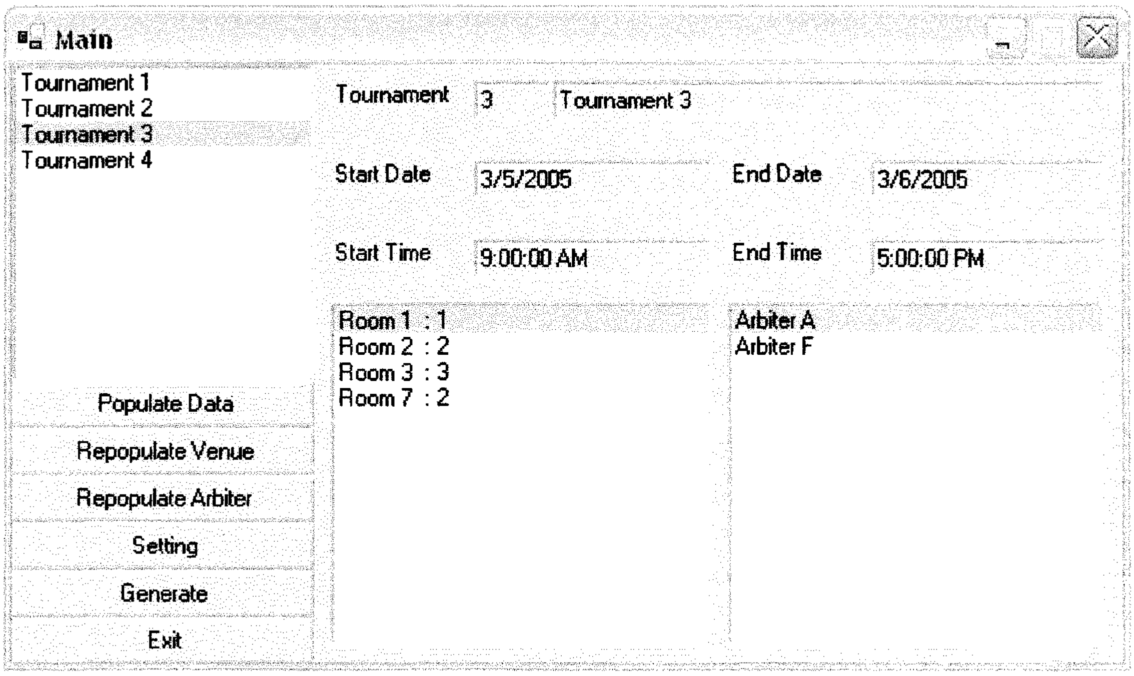


Figure 3: Unwanted resources removed

5. **Click on the Setting button to open the setting dialog box. Hit the Random button the randomly fill the setting dialog box. Click Save to exit Setting.**

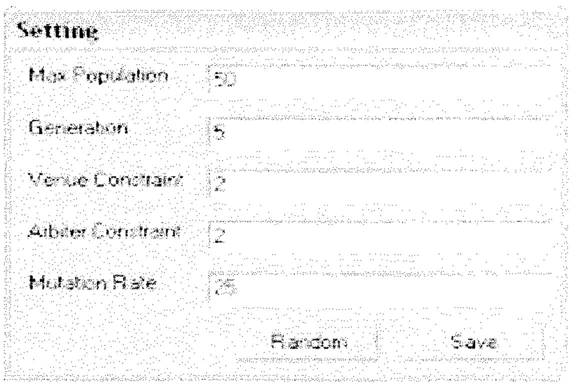


Figure 4: Setting dialog box

6. **Click on the Generate button to start generating the schedule.**

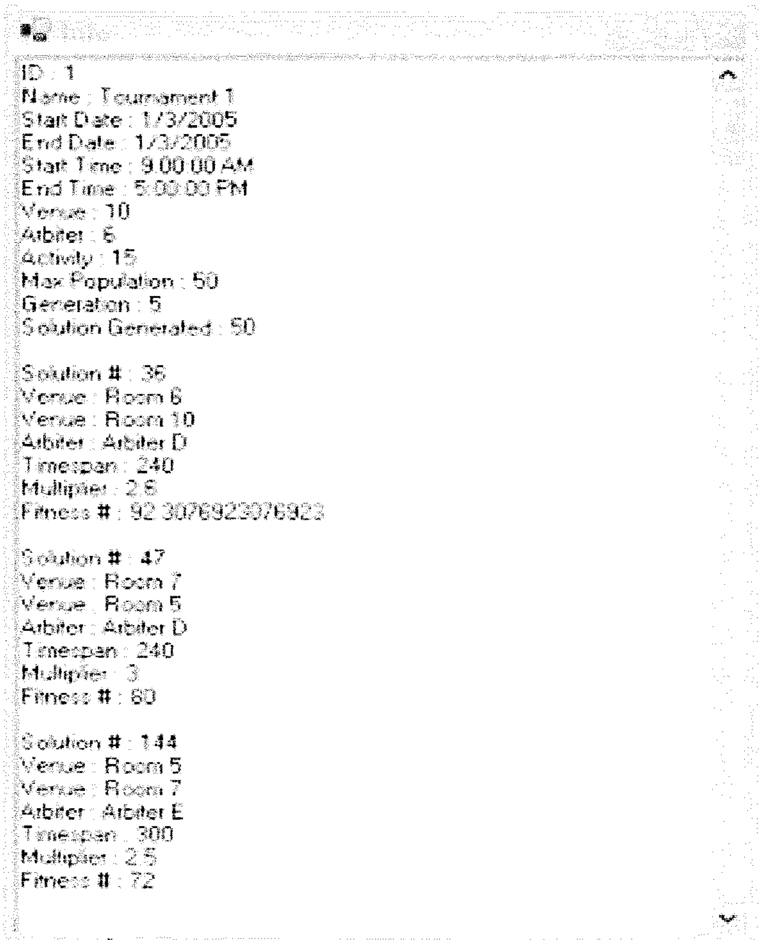


Figure 5: Solution window

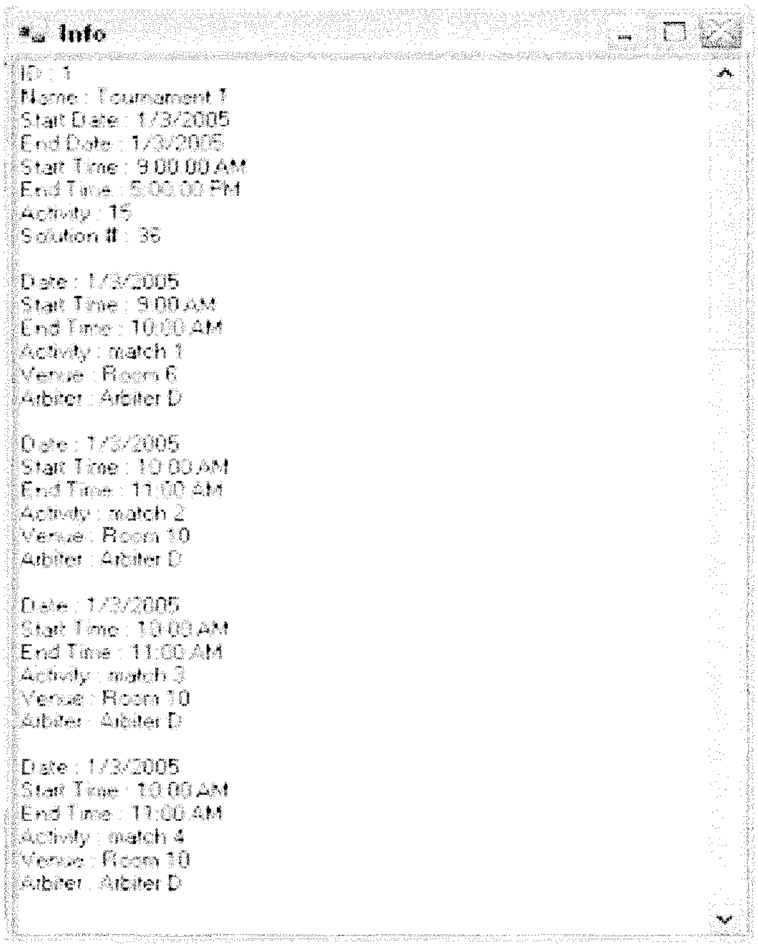


Figure 6: Schedule window

7. Click Exit to end the program.