Chapter 9

Conclusion and Further Work

9.1 Introduction

Utilization of Timetable Management System to Faculty of IT is successfully achieved throughout this research and accomplished tasks by the TMSFIT as follows.

- Generation of the timetable management system
- Resource optimization
- View the resources available

9.2 Conclusion

Utilization of Timetable Management System to Faculty of IT at University of Moratuwa is successfully achieved through this research process. From a thorough study of literature survey a proper algorithm could find as Genetic Algorithm. According to Prof Kaunananda, Genetic Algorithm can be applied to problems which are unsolvable [19]. The proposed timetabling system for this project seeks to generate near optimal timetables using the principles of genetic algorithm (selection, mutation and crossover). It is easily understandable, efficient and automated system, which is helpful for authorities of the IT faculty, lecturers and students. In addition, it is less paper work.

I have pleased to state; that I could get experience in Yii framework and the MVC design patterns. As a result, now I can develop any Yii component by my own. By following, this research my patience quality increased. For example, I spent number of days to find a searching algorithm. Writing about literature review and find the theories of that. Find information about research methodology and writing of interim document also interesting. There, I could learn about number of algorithms, about knowledgebase systems, searching methods, technologies used by others and Genetic algorithm and Programming languages such as PHP and DBMS such as MYSQL.

As conclusion, the generating timetables and viewing available resources tasks are successfully accomplished by the system within five minutes.
9.3 Further Work

Even this solution is ideal for a middle-sized campus as Moratuwa, large universities, cannot get the best results due to complexity of their problem in case of fitness value is getting decrease. Therefore, I propose those universities can use a combination of ruled base and Genetic Algorithm solution to develop such kind of system.

9.4 Summery

This chapter provided a conclusion of overall achievement met through the research project called utilization of timetable management system to faculty of IT at University of Moratuwa and further work in future as an enhancement of the current solution.
References


66


Appendix A

Interfaces of the TMSFIT

Before saving the Timetable which the fitness value = 1

![Generated Timetable Before Saved]

Error Messages given by the TMSFIT

![Error Message]

Figure A-1 Generated Timetable Before Saved

Figure A-2 Error Message
Manage Students Interface

Figure A-3 Manage Student Interface

Preview of tables

Figure A-4 Table Preview

SQL result

Table: students

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Contact</th>
<th>Address</th>
<th>Gender</th>
<th>Batch</th>
<th>Roll</th>
<th>Year</th>
<th>Fees</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>abc@email</td>
<td>123-456-7890</td>
<td>123 Main St</td>
<td>Male</td>
<td>1st</td>
<td>123</td>
<td>201</td>
<td>1000</td>
</tr>
<tr>
<td>DEF</td>
<td>def@email</td>
<td>456-789-0123</td>
<td>456 Main St</td>
<td>Female</td>
<td>2nd</td>
<td>456</td>
<td>2011</td>
<td>2000</td>
</tr>
<tr>
<td>GHI</td>
<td>ghi@email</td>
<td>789-012-3456</td>
<td>789 Main St</td>
<td>Male</td>
<td>3rd</td>
<td>789</td>
<td>2012</td>
<td>3000</td>
</tr>
</tbody>
</table>

Figure A-4 Table Preview
Appendix B

Code segments of the TMSFIT

Sample code for batch form

```php
<?php
/* @var $this BatchController */
/* @var $model Batch */
/* @var $form CActiveForm */
?>

<div class="form">

<?php $form=$this->beginWidget ('CActiveForm', array(
    'id'=>'batch-form',
    // Please note: When you enable ajax validation, make sure the
    // controller action is handling ajax validation correctly.
    // There is a call to performAjaxValidation() commented in
    // generated controller code.
    // See class documentation of CActiveForm for details on this.
    'enableAjaxValidation'=>false,
)); ?>

<p class="note">Fields with <span class="required">*</span> are required.</p>

<?php echo $form->errorSummary($model); ?>

<div class="row">
    <?php echo $form->labelEx($model,'name'); ?>
    <?php echo $form->textField($model,'name',array('size'=>50,'maxlength'=>50)); ?>

    <?php echo $form->error($model,'name'); ?>
</div>

<div class="row">
    <?php echo $form->labelEx($model,'description'); ?>
</div>
```

<?php echo $form->textField($model, 'description', array('size'=>60,'maxlength'=>200)); ?>

<?php echo $form->error($model, 'description'); ?>

</div>

<div class="row">
<?php echo $form->textField($model, 'startyear', array('size'=>4,'maxlength'=>4)); ?>

<?php echo $form->error($model, 'startyear'); ?>

</div>

<div class="row buttons">
<?php echo CHtml::submitButton($model->isNewRecord ? 'Create' : 'Save', array('class'=>'btn btn-primary')); ?>

</div>

<?php $this->endWidget(); ?>

</div><!-- form -->

Calculate the fitness of the chromosome

// Calculate fitness value for chromosome public function calcFitness(){
    // chromosome's score
    $score = 0;

    $numberOfRooms = $this->_noOfClassRoom;
    $roomDaySize = Schedule::DAY_HOURS * $numberOfRooms;

    $ci = 0;
    // check criteria and calculate scores for each class in schedule
    foreach($this->_classes as $value)
    {
        // coordinate of time-space slot
        $p = $value[0]; // slot number
        $day = (int)($p / $roomDaySize);
$time = $p % $roomDaySize;
$room = (int)($time / Schedule::DAY_HOURS);
$time = $time % Schedule::DAY_HOURS;

//
echo "<hr>rooms ******* $room." /// slot = "$p :
$dur = round($value[1]->duration, 0, PHP_ROUND_HALF_UP);
// check for room overlapping of classes
$ro = false;
for( $i = $dur - 1; $i >= 0; $i-- )
{
    if( array_key_exists ((int)($p + $i),$this->_slots)
&& sizeof($this->_slots[ (int)($p + $i) ]) > 1 )
    {
        $ro = true;
        break;
    }
}
// on room overlapping
if( !$ro ){
    $score++;
    //
    echo "<br>room overlapping";
}
$this->_criteria[ (int)($ci + 0 ) ] = !$ro;
$cc = $value[1];
$r = $this->_allClassRoom[$room];
// does current room have enough seats
$roomEnough = false;
if($r->capacity >= count($cc->students)){
    $roomEnough = true;
}
$this->_criteria[ (int)($ci + 1 ) ] = $roomEnough;
if($roomEnough ){
    $score++;
    //
    echo "<br>Room enough";
}

// does current room have computers if they are required
$labRequired = false;
if($cc->reqLab){
    $labRequired = true;
}
$resourseType = false;
if ($r->type == 'lab') {
    $score++;
    $resourceType = true;
} else {
    //
    //
}

if ($r->type != 'lab') {
    $score++;
    $resourceType = true;
} else {
    //
    //
}

$this->_criteria[(int) ($ci + 2)] = $resourceType;
set_time_limit(20);
$po = false;
$go = false;

// check overlapping of classes for professors and student groups
for ($ii = $numberOfRooms, $t = $day * $roomDaySize + $time; $ii > 0; $ii--, $t += Schedule::DAY_HOURS )
{
    // for each hour of class
    for ($i = $dur - 1; $i >= 0; $i--)
    {
        // check for overlapping with other classes at same time
        if (array_key_exists((int)($t + $i), $this->_slots) && isset($this->_slots[($t + $i)])) {
            //echo "<p>" . ($t + $i) . "</p>
            $classes = $this->_slots[($t + $i)];
        }
if( $cc != $cls )
{
    // professor overlaps?
    if( !$po && $cc->lecturerOverlaps( $cls ) )
        $po = true;

    // student group overlaps?
    if( !$go && $cc->studentOverlap( $cls ) )
        $go = true;

    // both type of overlapping? no need to check more
    if( $po && $go )
        goto total_overlap;
};

if( !$po )
{
    $lectpref = $cc->checkLectPrefTime($cc->lecturer->id,$day,$time,$dur,$cc->semester,$cc->year);
    if($lectpref)
        $score++;
    else {
        $po = true;
    }
}

$this->_overlaping_classes = !$po;

// student groups has no overlapping classes:
if( !$go )
{
    $score++;
    echo "<br>Student overlap";
}
$this->_criteria[$ci + 4] = !$go;

$ci += 5;

// calculate fitness value based on score
$this->_fitness = $score / (count($this->_subjectClass) * Schedule::DAYS_NUM);

}{

Cross over operation
// Performs crossover operation using two chromosomes and returns two offspring
public function crossover($parent2)
{
    // Check probability of crossover operation
    if (rand() % 100 > $this->_crossoverProbability)
    {
        // No crossover, just copy first parent
        return $this->makeCopy(false);
    }

    // Echo
    "<br><hr>crossover=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
    ===<br>";
    // New chromosome object, copy chromosome setup
    $n = $this->makeCopy(true);

    // Number of classes
    $size = (int)count($this->_classes);

    $cp = array();

    // Determine crossover point (randomly)
    for ($i = $this->_numberOfCrossoverPoints; $i > 0; $i--)
    {
        while (true)
        {
            $p = rand() % $size;
if(!array_key_exists($p, $cp))
{
    $cp[$p] = true;
    break;
}
}

$it1 = $this->_classes;
$it2 = $parent2->_classes;
$pindex1 = 0;
$pindex2 = 0;

// make new code by combining parent codes
$first = rand() % 2 == 0;
for( $i = 0; $i < $size; $i++)
{
    if( $first )
    {
        // insert class from first parent into new chromosome's class table
        $n->_classes[$pindex1] =
            array($it1[$pindex1][0], $it1[$pindex1][1]);

        // all time-space slots of class are copied
        for( $ii = round($it1[$pindex1][1]->duration, 0, PHP_ROUND_HALF_UP) - 1; $ii >= 0; $ii-- )
        {
            if($n->_slots[(int)($it1[$pindex1][0] + $ii)] == null) {
                // if array is null then initialize new array and add class
                $n->_slots[(int)($it1[$pindex1][0] + $ii)] =
                    array($it1[$pindex1][1]);
            } else {
                $array = $n->_slots[(int)($it1[$pindex1][0] + $ii)];
                $array[] = $it1[$pindex1][1];
                $n->_slots[(int)($it1[$pindex1][0] + $ii)] = $array;
            }
        }
    }
    else {
        // insert class from second parent into new chromosome's class table
    }
}
$n->classes[$pkindex1] = array($it2[$pkindex2][0],$it2[$pkindex2][1]);

// all time-space slots of class are copied
for( $ii = round($it2[$pkindex2][1]->duration, 0, PHP_ROUND_HALF_UP) - 1; $ii >= 0; $ii-- ){
if($n->_slots[($it2[$pkindex2][0] + $ii )] == null){
    // if array is null then initialize new array and add class
    $n->_slots[($it2[$pkindex2][0] + $ii )] = array($it2[$pkindex2][1]);
} else{
    $array = $n->_slots[($it2[$pkindex2][0] + $ii )];
    $array[] = $it2[$pkindex2][1];
    $n->_slots[($it2[$pkindex2][0] + $ii )] = $array;
}
}

// crossover point
if( array_key_exists($i, $cp)){
    // change source chromosome
    $first = !$first;
}

$pkindex1++;
$pkindex2++;}

$n->calFitness();

// crossover point
if( array_key_exists($i, $cp)){
    // change source chromosome
    $first = !$first;
}

$pkindex1++;
$pkindex2++;

$n->calFitness();

// Performs mutation on chromosome
public function mutation()
{
// echo "<br>---------------------------------&gt;mutation--------------------------------";
// check probability of mutation operation
if( rand() % 100 > $this->_mutationProbability )
    return;

Mutation code segment

// Performs mutation on chromosome
public function mutation()
{
// echo "<br>---------------------------------&gt;mutation--------------------------------";
// check probability of mutation operation
if( rand() % 100 > $this->_mutationProbability )
    return;
// number of classes
$numberOfClasses = (int)count($this->_classes);
// number of time-space slots
$size = (int)count($this->_slots);
// move selected number of classes at random position
for($ii = $this->_mutationSize; $ii > 0; $ii--)
{
    // select random chromosome for movement
    $mpos = rand() % $numberOfClasses;
    $pos1 = 0;
    $classIndex = array_keys($this->_classes);
    // current time-space slot used by class
    $pos1 = $this->_classes[$mpos][0];
    $scl = $this->_classes[$mpos][1];
    // determine position of class randomly
    $nr = $this->_noOfClassRoom;
    $dur = round($scl->duration, 0, PHP_ROUND_HALF_UP);
    $day = rand() % Schedule::DAYS_NUM;
    $room = rand() % $nr;
    $time = rand() % (Schedule::DAY_HOURS + 1 - $dur);
    $pos2 = $day * $nr * Schedule::DAY_HOURS + $room * Schedule::DAY_HOURS + $time;
    // move all time-space slots
    for($i = $dur - 1; $i >= 0; $i--)
    {
        // remove class hour from current time-space slot
        $cl = $this->_slots[$pos1 + $i];
        foreach($cl as $key=>$cls)
        {
            if($cls == $scl)
            {
                unset($cl[$key]);
                break;
            }
        }
    }
    // move class hour to new time-space slot
    if($this->_slots[(int)($pos2 + $i)] == null)
    {
        // if array is null then initialize new array and add class
    }
}
$this->_slots[(int)($pos2 + $i)] = array($scl);

} else {
    $array = $this->_slots[(int)($pos2 + $i)];
    $array[] = $scl;
    $this->_slots[(int)($pos2 + $i)] = $array;
}

// change entry of class table to point to new time-space slots
unset($this->_classes[$mpos]);
$this->_classes[$mpos] = array($pos2, $scl);

$this->calFitness();

Dashboard code segment

<?php
// @var $this DashboardController

// $this->breadcrumbs=array(  // Dashboard',  // );?
?>
<div class="dashboard">
    <div class="dashboard-wraps">
        <!-- Dashboard for Administrators -->
        <?php if (Yii::app()->user->getState('roles') == "admin"): ?>
        <!-- <div class="dash-icon">
            <a class="dash-link" href="<?php echo Yii::app()->
>createUrl('candidate/admin'); ?>"/>
        </a> -->
    <img src="<?php echo Yii::app()->
>baseUrl.'/images/dashboard/add-new-applicant.png'; ?>" />
    </div>
</div>

79
<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()->createUrl('degree/admin') ?>">
        <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/degrees.png'; ?>" />
    </a>
</div>

<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()->createUrl('student/admin') ?>">
        <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/manage-students.png'; ?>" />
    </a>
</div>

<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()->createUrl('employee/admin') ?>">
        <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/manage-employees.png'; ?>" />
    </a>
</div>

<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()->createUrl('subject/admin') ?>">
        <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/subject.png'; ?>" />
    </a>
</div>

<div class="link-text">New Applicants</div>
<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()->createUrl('resource/admin'); ?>">
        <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/class-room.png'; ?>" />
    </a>
</div>

<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()->createUrl('batch/admin'); ?>">
        <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/batch.png'; ?>" />
    </a>
</div>

<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()->createUrl('algorithm/load'); ?>">
        <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/time-table.png'; ?>" />
    </a>
</div>

<?php endif; ?>

<!-- Lecturer Dashboard -->

<?php if ((Yii::app()->user->getState('roles') == 'lecturer')): ?>
    <div class="dash-icon">
        <a class="dash-link" href="<?php echo Yii::app()->createUrl('employee/time', array('id'=>Yii::app()->user->getState('logeduserid'))); ?>">
            <img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/time.png'; ?>" />
        </a>
    </div>
<?php endif; ?>
<?php endif; ?>

<!-- Dashboard for Students -->

<?php if (Yii::app()->user->getState('roles') == "student") { ?>

<div class="dash-icon"> <a class="dash-link" href="<?php echo Yii::app()->createUrl('student/view', array('id' => Yii::app()->user->getState('loggeduserid'))); ?>" >
<img src="<?php echo Yii::app()->baseUrl.'/images/dashboard/profile.png'; ?>" />
</a>
</div>

<?php } ?>

</div>
<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::$app->createUrl('student/timetable'); ?>">
        <img src="<?php echo Yii::$app->baseUrl.'/images/dashboard/time-table.png' ?>" />
    </a>
</div>

<?php endif; ?>
</div>
# Appendix C

## Testing and Evaluation with Test data

### Sample Test cases for Black box Testing

<table>
<thead>
<tr>
<th>Test No</th>
<th>Test Data</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Logging with incorrect user name</td>
<td>Operation must be rejected</td>
<td>Message - Incorrect Username or password</td>
<td>Achieved</td>
</tr>
<tr>
<td>2</td>
<td>Logging with incorrect password</td>
<td>Operation must be rejected</td>
<td>Message - Incorrect Username or password</td>
<td>Achieved</td>
</tr>
<tr>
<td>3</td>
<td>Logging with incorrect user name and password</td>
<td>Operation must be rejected</td>
<td>Message - Incorrect Username or password</td>
<td>Achieved</td>
</tr>
<tr>
<td>4</td>
<td>Logging with empty user name</td>
<td>Operation must be rejected</td>
<td>Message - Incorrect Username or password</td>
<td>Achieved</td>
</tr>
<tr>
<td>5</td>
<td>Logging with empty password</td>
<td>Operation must be rejected</td>
<td>Message - Incorrect Username or password</td>
<td>Achieved</td>
</tr>
<tr>
<td>6</td>
<td>Logging with empty Username and password</td>
<td>Operation must be rejected</td>
<td>Message - Incorrect Username or password</td>
<td>Achieved</td>
</tr>
<tr>
<td>7</td>
<td>Search with incorrect name</td>
<td>Empty results</td>
<td>Message - No results found</td>
<td>Achieved</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Operation</td>
<td>Message</td>
<td>Result</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>8</td>
<td>Input Empty Field Name to a field with astric</td>
<td>Operation must be rejected</td>
<td>Message - Name cannot be blank.</td>
<td>Achieved</td>
</tr>
<tr>
<td>9</td>
<td>Input value to a field with astric</td>
<td>Operation must be success</td>
<td>No error Message</td>
<td>Achieved</td>
</tr>
<tr>
<td>10</td>
<td>Input submit number of Empty Fields with astric</td>
<td>Operation must be rejected</td>
<td>Message - Name cannot be blank.</td>
<td>Achieved</td>
</tr>
<tr>
<td>11</td>
<td>Search with correct No</td>
<td>Operation success</td>
<td>Show the searched data</td>
<td>Achieved</td>
</tr>
<tr>
<td>12</td>
<td>Search with correct batch No</td>
<td>Operation success</td>
<td>Show the searched data</td>
<td>Achieved</td>
</tr>
<tr>
<td>13</td>
<td>Advanced Search with one the data Name Perera</td>
<td>Operation success</td>
<td>Show all the searched name with Perea</td>
<td>Achieved</td>
</tr>
<tr>
<td>14</td>
<td>Update student without registration No</td>
<td>Operation success</td>
<td>Message-Registration No cannot be blank</td>
<td>Achieved</td>
</tr>
</tbody>
</table>

Table C-1 Test Cases
White Box Testing with Understand Software

Analysis Log

Figure C-1 White box Testing

Results log
Metric Summery
## Evaluation Questionnaire

**Evaluation of the Timetable Management System of Faculty of IT at University of Moratuwa**

1. Strongly agree  
2. Fairly agree  
3. Agree  
4. Disagree  
5. Strongly disagree

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>N\A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I'm satisfying with the feature of effective use of time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>This system is simple and easy to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Interfaces of the system are good and attractive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>It is easy to search information from this system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>It is easy to recover and correct from the system mistakes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>System gives error messages which clearly tells me how to fix problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>I am satisfying with the functionalities of this system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>System functionalities are user friendly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>System's manually changing part also, can be tolerate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Overall, I am satisfied with TMSFIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table C-2 Questionnaire Evaluation Table