Using Nested Tables and Multi-pitch Adjusting in Harmony Search (NTMHS) to Solve Timetabling Problem in Object-Relational Model

Wadee Al-Qubati¹, Ammar Zahary², Abdualkadir Alabadi³

¹,²,³ Department of Computer Science, Faculty of Computing and Information Technology, University of Science and Technology, Sana’a, Yemen

E-mails: ¹wadee_alkopaty@yahoo.com, ²aalzahary@gmail.com, ³amayemen@yahoo.com

ABSTRACT
In this paper, we conduct our novel approach called NTMHS that employs nested tables and multi-pitch adjusting in harmony search algorithm to solve timetabling problem in database object-relational model. Our approach examines the ability of object-relational model to provide structural storage, indexed and organized processing for query. In NTMHS, timetables are produced depending on structure of nested tables that effectively deal with multi-pitch adjusting within the algorithm of harmony search. NTMHS approach has been implemented and applied to a proposed model for semester-based course timetables of Computing and Information Technology Faculty at the University of Science and Technology. Results show that NTMHS idea led to overcome hard and soft constraints of timetabling process in an effective way, short time, and the absence of conflicts.

Keywords: Object relational modal, nested table , Harmony Search (HS), Timetabling, Multi-pitch Adjusting, New Harmony (NH), pitch Adjusting Rate (PAR), harmony memory (HM)

1. INTRODUCTION
Timetabling process is a hard and difficult challenge for academic institutions, schools, hospitals, airports, and so many projects that need timetabling to manage and organize tasks and resources. The difficulty of timetabling process comes from the long time it takes and the multiple soft constraints that must be taken into account throughout the whole process. Examples of such constraints are non-contradiction in time, place and other determinants that might happen during the timetabling process. Therefore, there was so many studies and research introduced to solve this problem by proposed many approaches that can help doing timetabling process. Most of those approaches aim to produce schedules with free of contradiction. In the next Section, we will present our related work including many researches related to our study. In this paper, we have developed our novel approach called NTMHS that employs what is known as nested tables [1] and multi-pitch adjusting in Harmony Search (HS) algorithm [2] to solve timetabling problem in database object-relational model. NTMHS provides effective structural storage, indexing, and good processing for query and it explores the supportive role when combining our previous approach NTMGA [1] with the Harmony Search Algorithm to solve timetabling problem in database object-relational model. Our previous approach NTMGA was based on the idea of applying nested tables to solve such problem using Genetic Algorithms (GA).

The contributive idea behind our new approach NTMHS is that the object-relation model can provide structured solutions through the use of nested tables, which can be structured to help improving HS algorithm performance to solve the problem of timetabling. By improving the operations of the Pitch Adjusting by making it Multi-pitch Adjusting, we supposed this will accelerate the process of reaching the optimal solution which has been proven in implementation and experiments of this paper. NTMHS approach has been implemented and applied to a semester-based course timetables of the Faculty of Computing and Information Technology at the University of Science and Technology.

2. RELATED WORK
This section reviews the most important studies starting with the our previous approach called NTMGA [1]. In [1], we use object-relational database model with genetic algorithm to solve the problem of timetabling. We utilized an advantage from the idea of nested tables to eliminate so
many contradictions and improve the process of Mutation to reach multiple optimal solutions with fewest number of generations in a short time. NTMGA has developed a proposed model and implemented for timetabling of academic institutions. NTMGA performed well and achieved a satisfactory results in terms of the absence of contradiction and short time needed to produce timetables. The authors in [1] have concentrated on applying hard constraints only.

Authors in [2] have applied the harmony search algorithm in course timetabling by dividing the pitch adjustment operator to eight procedures, each procedure is controlled by its pitch adjusting Rate value range to produce new harmony as a result of doing each procedure and to achieve the optimal solutions. Study results show the ability to provide effective course timetables, the authors have concentrated on applying three hard constraints and three soft constraints only.

Authors in [3] proposed a method to solve the problem of course timetabling using genetic algorithms. The authors developed the algorithm using matrix-dimensional by which each row in the table represents the hall and the entire rows represent chromosomes. Based on this matrix, Mutation and Selection process is applied to improve the chromosomes from one generation to another until reaching the optimal solution. Authors in [3] depend on a set of standards or constraint to produce schedules which are close to the optimal solution. Authors have concentrated on applying some hard constraints.

3. THE NTMHS APPROACH

This Section introduces our novel approach called NTMHS that employs nested tables and multi-pitch adjusting in harmony search algorithm to solve timetabling problem in database object-relational model. Developing NTMHS approach has been relying on four hypotheses:

- A class lecture time may be two or three hours.
- A class lecture time ranging from 8:00am to 5:00pm.
- Halls and laboratories are available for all departments of the faculty.
- For any course, student intakes might be divided into more than one group according to the hall capacity.

All hard constraints and some soft constraints should be met in the course timetable and represented as following:

- No more than two classes in a day for any lecturer.
- If a lecturer has two classes in a day, the break between the classes must not be more than one hour.
- The break between any two classes for any student group must not be more than one hour.
- A lecturer can determined the suitable times of his/her classes.

3.1. The Mechanism of NTMHS Approach

The idea behind our approach NTMHS is based on the university database that provides all the information about the courses, teachers and students, the halls and laboratories, through this information the solution was represent according to three basic dimensions depend on the timetabling shown in figure (1).

Figure 1: the three basic dimensions

Figure 1 represents the three-dimensional through creation nested table consists of six columns represent the days of study during the week (first dimension), an additional column to determine the halls, and number of rows determine the number of halls and laboratories in the faculty (second dimension). The intersection of any row with any column is identified as object which will represent teaching information (third dimension).
Figure 2 illustrates the proposed model to represent the solution, which will achieve the improvisation process by dividing the pitch adjustment to three procedures each of them is controlled by PAR value range that repeatedly provides a new harmony until reaching the optimal solution that achieves all hard and soft constraint which have been identified.

3.2. Stages of NTMGA approach

Figure 3 illustrates the general steps that have been followed for the implementation of HS algorithm.

3.2.1 Initialize the HSA and timetable parameters stage

At this stage, all transactions have been initialized relating to the timetabling in order to deal with them in HS algorithm by encoding all information about teaching and represent it as object within the nested tables shown by Figure 4. Table 1 represents the timetable coefficients which are needed to create a solution.

Table 1: Encoding all timetable parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>represents the lecture-time code (beginning and end) and consists of three characters</td>
<td>P</td>
</tr>
<tr>
<td>represents the course code consists of three characters</td>
<td>T</td>
</tr>
<tr>
<td>represents the course instructor code consists of two characters</td>
<td>M</td>
</tr>
<tr>
<td>represents the specialty code that belongs to the course and consists of a single character (one character)</td>
<td>L</td>
</tr>
<tr>
<td>represents the student's level code and consists of a single character</td>
<td>G</td>
</tr>
</tbody>
</table>

3.2.2 Initialize the harmony memory

At this stage, we provide a range of primary solutions. Each solution is distributed to a set of objects loaded with a randomly teaching information within nested tables taking into account that the total number of hours/lecture for all courses is equal to or less than the total hours of working halls, and in the case they are less, it will charge the object with values (00000000) which represents the times available to the halls.

3.2.3 Performing the improvise

This stage depends on pitch adjustment operators to create NH, in which the pitch adjustment operator is divided into three procedures as shown by Table 2.

Table 2: pitch adjustment operator

<table>
<thead>
<tr>
<th>Description</th>
<th>procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>The lecture is moved to any empty site on a randomly way (this procedure is implemented in the case that the total number of lecture hour for all courses is less than the total hours of working of the halls)</td>
<td>Move-infoteach</td>
</tr>
<tr>
<td>In this procedure, it swaps between the two lectures randomly</td>
<td>exchange-infoteach</td>
</tr>
<tr>
<td>it is extend or shorten the lectures time code for a certain hall</td>
<td>Shift-timeslot</td>
</tr>
</tbody>
</table>

Shift-timeslot procedure:
Intersection of any row with any column within the nested tables represents lectures of a particular day in a certain hall. The process *Shift-timeslot* represents rearrangement of each lecture to fit the switching (swapping) or transmission action of lectures. Figure 5 shows a *Shift-timeslot* for the first and second lectures on Sunday in Hall number 3.

**Move-infoteatch procedure:**
In this procedure, the lecture randomly moved to any empty site. Figure 6 shows the transfer of the first lecture on Sunday (8:00am-10:00am) to Hall 7, which will be available in time 09 (1:00pm-3:00pm).

**Exchange-infoteatch procedure:**
In this procedure, we can switch between two lectures at random. Figure 7 shows switching between the first lecture on Sunday, at 03 (10:00am-1:00pm) with the lecture in hall 7 on Monday at 02 (8:00am-11:00am).

**3.2.4 Update the harmony memory**
At this stage, quality of a solution is measured by checking that all hard and soft constraints that have been identified have been achieved. Quality of a solution is calculated by the following Equation (1).

\[ F(x) = \frac{\sum \text{conf}}{\sum \text{AllCour}} \quad \text{......(1)} \]

Where *AllCour* represents all objects within the nested table which is loaded with teaching information, *conf* all objects within the nested table which did not realize hard or soft constraints.

The result of the previous equation gives the solution quality which ranges between 0 and the 1, higher rate in each reiteration reflects the improvement in the solution.

### 3.2.4 Stopping criterion
When the value of calculating quality of the solution process is equal one, it must stop the process of calculation of Equation 1. Which means that all hard and soft constraints that have been identified to the timetabling have been realized.

### 4. EXPERIMENTAL STUDY
This Section presents the experimental results and evaluation of our approach NTMHS that has been implemented oracle 11.2.0.3 by using tools Toad9 Produced by Quest Company, and applied to the timetable of Faculty of Computing and IT at University of Science and Technology which consists of three departments and nine programs with dividing all intakes into two groups (divisions). Preparing for HS Memory (HSM), it consists of ten solutions conducting the Multi-pitch adjustment on each solution separately and the results were issued ten different versions of the timetables in quick time and full resolution(100%) realized all hard and soft constraints that have been identified.

The following example shows the versions of a timetable with the analysis of the results by more than one solution.

1. At the beginning, initial parameters have been initialized for the problem by preparing an encoding timetabling. Figure 8 shows this encoding for all lectures.
2. Initialize for the preparation of HSM as illustrated by Figure 9.

![Figure 9: Initialize the harmony memory](image)

It is clear from Figure 9 that the lectures in hall 2 on Saturday was represented as three lectures and the fourth time ($T = 13$), which represents the encoding for the time period which extending from 3 pm to 5 pm represents free time (00000000) and we can make Move-infoteach process for it when a pitch adjustment process is making.

3. The process of multi-pitch adjustment has been applied for each solution within the harmony memory and we got a timetable free from contradiction. Figure 10 shows timetable in the nested table.

![Figure 10: Nested table represents the timetable of the optimal solution](image)

4. We can get different important reports in timetables by making a simple query in the nested table because of all data has stored only in object, Figure 11 shows the results of a query to view the schedule of classes of lecturers.

![Figure 11: Schedule of classes of teachers](image)

5. At this point, we will analyze the results which we got by implementing the proposed approach for several solutions. Table 3 shows the values of parameters used in the implementation.

It is clear from Figure 12 that the solution is exponentially improved with each iteration of the multi-pitch adjustment. First, improving rate moves from 70% to 80% and then improvement increases with fluctuate rate until reaching the optimum solution.

Figure 13 illustrates the convergence process of solutions improvement when implementing the approach every time.
Table 3: parameter values

<table>
<thead>
<tr>
<th>No.</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lectures</td>
<td>211</td>
</tr>
<tr>
<td>2</td>
<td>Teachers</td>
<td>43</td>
</tr>
<tr>
<td>3</td>
<td>Halls</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Labs</td>
<td>6</td>
</tr>
</tbody>
</table>

5. CONCLUSION

This paper presents our novel approach called NTMHS that utilizes the idea of nested tables and multi-pitch adjusting in Harmony Search algorithm to solve timetabling problem in database object-relational model. It takes the advantage of the technology object-relational databases in the application of HS algorithm to create timetables and study the achievement of hard and soft constraints to get multiple optimal solutions in easy and fast way. NTMHS approach has been implemented and applied to a semester-based course timetables of the Faculty of Computing and Information Technology at the University of Science and Technology. Results show that NTMHS idea led to overcome hard and soft constraints of timetabling process in an effective way, short time, and the absence of conflicts. As a future work, a researcher can improve and develop NTMHS to deal with different types of scheduling and achieve flexibility in dealing with other constraints on scheduling such as determining teachers time, study time, number of lecture hours, and dedicate certain halls for certain lectures and other constraints. This can be achieved through appropriate adjustments to the nested table structure to adapt with applying various constraints to tables.

REFERENCES


[5] M. Rui, G. Jinyu and Li Bo, "University Course Timetable System Design and Implementation Based


