Kontigensi: Scientific Journal of Management
Vol 6 , No. 2, June 2018, pp. 112 - 118
ISSN 2088-4877

Implementation of the Harmony Search Algorithm in Completing Lecture Scheduling

Abd. Rahman
UPT TIPD IAIN Palopo
E-mail : abdrahman@iainpalopo.ac.id

ABSTRACT

Scheduling lectures at a college is a routine activity every semester and is a process to implement events that contain components of courses and classes in time slots that contain components of time and space. The problem that often occurs in scheduling activities is the occurrence of conflicts between one schedule and another. One method to solve these problems is to use artificial intelligence (AI). One method in AI that is considered to provide a solution to scheduling problems is Harmony Search. Harmony Search is an area of computer science that bases its algorithms on music. The Harmony Search algorithm compares music with all its devices to optimization problems. For example, each musical instrument is associated with a decision variable; the musical pitch is associated with a variable value, harmony is associated with a solution vector. Like a musician who plays certain music, improvises playing a random tone, or based on experience to find a beautiful harmony, the variables in Harmony Search have random values or values obtained from iterations (memory) to find the optimal solution. By applying the Harmony Search algorithm to prepare the lecture schedule, it is hoped that an optimal arrangement of the lecture schedule can be created.

Keywords: Optimization, Scheduling, Artificial Intelligence, Harmony Search

INTRODUCTION

Setting the lecture schedule at a university is very important because the implementation of lecture activities involves many interrelated aspects, and each has different priorities. Aspects considered in scheduling include the number of students, lecturers, time, room, and other requirements.

The teaching and learning activities at a university cannot be separated from the lecture scheduling factor. The better the lecture scheduling system, the more organized and optimal the teaching and learning process will be at the university. However, compiling a lecture schedule is not an easy job. Many obstacles can make scheduling difficult.

Other factors that make it difficult to schedule lectures are the limited number of competent lecturers while the courses taught are quite a lot, limited classrooms, and the addition of new classes in line with the increase in the number of students each year. Because the process is still completely manual and made worse by the level of accuracy that is less consistent, causing the resulting schedule to be less than perfect, this is because there are still violations of existing requirements, such as conflicting schedules in the classroom or teaching hours, lecturers who teach unable to attend and others.

The lecture scheduling system in several higher education institutions is still done manually, namely by searching which blocks or columns are still empty then placing the schedule in that block or column. Schedules generated in this way take a long time and tend to ignore these aspects. So that class schedules and exams that have been made often need to be repaired again. Therefore, it is necessary to develop a system of scheduling lectures and exams to accommodate the various aspects considered above.

Harmony Search includes a metaheuristic approach that bases its algorithm on music. There is an analogy between music and all its tools and optimization problems (Geem et al., 2001). For example, each musical instrument is associated with a decision variable; a musical note is associated with a variable value, harmony is associated with a solution vector. Like a musician who plays certain music, improvises playing random notes, or, based on experience to find beautiful harmonies, the variables in Harmony Search have random values or values obtained from iterations (memory) to find the optimal solution. As an optimum solution to an optimization problem, the preparation of the lecture schedule (Santosa & Willy, 2011). By using the Harmony Search algorithm, it is hoped that the optimal schedule can be obtained, namely the best combination conditions in the scheduling.

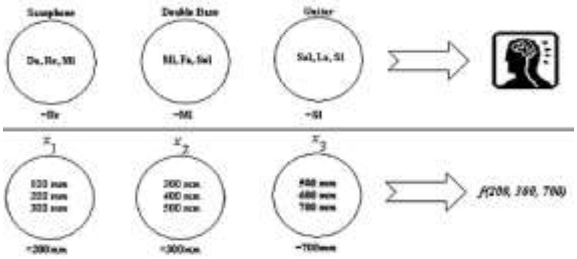

Figure 1.
Harmony Search Algorithm Structure

In Figure 1, each music player (saxophonist, double bassist, and guitarist) represents a decision variable (x1, x2, and x3). The set of sounds produced by each musical instrument (saxophone = {Do, Re, Mi}; double bass = {Mi, Fa, Sol}; guitar = {Sol, La, Si}) represents a range of variable values (x1 = { 100, 200, 300 }; x2 = { 300, 400, 500 }; x3 = { 500, 600, 700}). For example, the saxophonist sounds Re, the double bassist sounds Fa, and the guitarist sounds La, then the three of them build a new harmony (Re, Fa, La). If this harmony is more beautiful than the current harmony, then this new harmony is preserved. The harmony obtained in the optimization world is called a solution represented in the form of a solution vector dimension (Desiani & Arhami, 2006).

Based on the above concept, the Harmony Search algorithm consists of five stages, namely:

1. Initialize Harmony Search parameters

Minimize (or maximize) f(x)
With xi, $\in$ Xi , i = 1,2,...N
Where:
f(x) is an objective function
xi is the decision variable to i
Xi is the set of decision variables
N is the number of decision variables
The parameters needed are as follows

a. Harmony Memory Size (HMS) is the number of vectors formed solution
b. The Harmony Memory Consideration Rate (HMCR) is $0 \leq$ HMCR ≤ 1. Generally, ranges from 0,7 to 0,99
c. The Pitch Adjustment Rate (PAR) is $0 \leq$ PAR ≤ 1. Generally, range from 0.1 to 0.5
d. The stopping criterion is the number of iterations for improvisation.

2. Initialize Harmony Memory

Equation 1 represents the initialization of HM obtained by randomly generating the decision variable xi to form a solution vector Xi. Then calculate the value of the objective function f(x) for each solution vector.

$$HM = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & \cdots & x_{1N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_{N-1}^2 & x_p^2 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_1^{HMS-1} & x_2^{HMS-1} & x_3^{HMS-1} & \cdots & x_{p-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & x_3^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix}$$

Equation 1.

where each solution vector (each row) will be evaluated for its function value

$$f(x)=\begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HMS-1}) \\ f(x^{HMS}) \end{bmatrix}$$

Equation 2.

In this study, members of the set of decision variables represent the number of courses processed. For example, if there are six courses to be processed in 5-time slots, then the set of decision variables formed is X = {1,2,3,4,5,6}. So that the HM initialization that may be formed is as follows:

$$HM=\begin{bmatrix} 2 & 3 & 1 & 4 & 5 \\ 1 & 4 & 2 & 3 & 5 \\ 4 & 1 & 3 & 2 & 5 \\ 3 & 1 & 4 & 5 & 2 \end{bmatrix}$$

After obtaining the solution vector as many as HMS, namely the number of rows from HM, calculate the value of the objective function of each vector.

3. New Harmony Improvisation

Harmonic improvisation generates a new harmony xi to form a new solution vector x = (x1, x2, … xn). The generation of that new harmony

done with two rules, namely:

a) Harmony Memory Consideration

At this stage, the value of the decision variable X is chosen randomly from any stored variables in HM (xi1, xi2, xi3, xiHMS) with probability x's.

Generation of decision variables not in HM will be chosen randomly from the set variable X with probability 1–HMCR. Generation of new decision variables at this stage as in Equation 3.

$$x_1^{'}=\begin{cases} x_1^{'}\in x_i^1, x_i^2, \ldots, x_i^{HMS}\}) & HMCR \\ x_1^{'}\in X_i & HMCR-1 \end{cases}$$

Equation 3

b) Pitch Adjustment

This stage is the adjustment stage for the new decision variable xi generated at the harmony stage memory considerations. The decision variable will be adjusted to its neighboring variables with HMCRxPAR probability. Decision variables xi generated by the harmony memory consideration is maintained with probability HMCRx(1–PAR) (Mahdavi et al., 2007)

4. HM Updates

If the objective function value of the new solution vector is worse than the vectors in HM, then the solution vectors in HM are retained.

5. Stopping Criterion

The stop criteria determine the number of processes for generating new solutions and updating the HM. If the stopping criteria are met, the iteration process will stop.

Several researchers have researched lecture schedules using different algorithms. For example, the application of the Genetic Algorithm in the lecture schedule has been successfully carried out by Teddy (Teddy, 2009). Meanwhile, Komang Siteman succeeded in combining the Genetic Algorithm with the Tabu Search algorithm in compiling the lecture schedule (Setemen, 2008) and Heni Rahmawati, who researched the use of graph coloring techniques and bee colony algorithms in solving scheduling problems (Rachmawati, 2012). Some researchers apply the particle swarm optimization algorithm in compiling the lecture schedule. (Ana Ratna Wati & Agusti Rochman, 2008)

The use of the Harmony Search algorithm has also been investigated for application in various fields, as done by Indra Aulia, who uses the Harmony Search algorithm in scheduling the flow shop (Aulia et al., 2012). Likewise, Niko Setiono applies the Harmony Search algorithm to automate timetabling (Sutiono et al., 2012).

Likewise, research on the application of the harmony search algorithm has been carried out by Indra Aulia. The research was conducted by applying the harmony search algorithm in solving flow shop scheduling problems (Priyo et al., 2016). In this article, research is carried out whose final results can be used to overcome difficulties in making lecture schedules. The Harmony Search algorithm is used as a scheduling optimization method in this study.

Harmony Search includes a metaheuristic approach that bases its algorithm on music. There is an analogy between music and all its tools and optimization problems. For example, each musical instrument is associated with a decision variable; a musical note is associated with a variable value, harmony is associated with a solution vector. Like a musician who plays certain music, improvises playing random notes, or, based on experience to find beautiful harmonies, the variables in Harmony Search have random values or values obtained from iterations (memory) to find the optimal solution. As an optimum solution to an optimization problem, the preparation of the lecture schedule (Suyanto, 2007). By using the Harmony Search algorithm, it is hoped that the optimal schedule can be obtained, namely the best combination conditions in the scheduling.

This research aims to produce an optimal lecture scheduling system using the Harmony Search algorithm as an optimization technique.

METHOD

The stages that will be carried out in this research are as follows:

1. Research stages
   a. Problem identification

At this stage, observations were made on the lecture schedule prepared using the manual method. The observations found several problems in scheduling, including schedule collisions between lecturers, rooms, and others

b. Problem formulation

From the results of the problem inventory that has been carried out, the search for solutions to these problems is carried out. The main solution used is to use a computerized system for problem-solving.

c. Research design

At this stage, several simulation methods are carried out to be used in a computerized system to solve the problem. Then from several simulations, it was decided to use the harmony search method.

d. Data collection

Some of the data collection techniques used are:

1) Observation; by observing the ongoing process of making the lecture schedule.
2) A literature study was carried out to extract references to decision support systems, methods, and other matters.
3) Academic data; This is done by digging up information from the manager regarding the data used to make the lecture schedule.

2. Data analysis

Data analysis was carried out after the research stage was carried out. In this data analysis, what is done in the study is to analyze data from the stages of research that have been carried out, namely analyzing the system testing questionnaire used in the study.

3. System development

At this stage, the system is developed using the prototyping method. The prototyping technique is rapid system development and testing of working models (prototypes) of new applications through repeated and interactive processes commonly used by information systems experts and business experts. Prototyping is also called rapid application design (RAD) because it simplifies and speeds up system design.

There are several stages in the prototyping technique, which are as follows:

a. Data collection

   Users and developers together define the format of the full software. Define all requirements, and outline the system to be made.

b. Build Prototype

   Build prototypes by making temporary designs that focus on presenting to users (e.g.by creating input and output formats). In this stage, the Simple Additive Weighting method is integrated into the system that is being built.

c. Prototype evaluation

   The user carries out this evaluation of whether the prototype has been built in by the user's wishes.

d. Coding system

   The agreed prototype is translated into the appropriate programming language in this stage.

e. System testing

Kontigensi: Scientific Journal of Management
Vol 6 , No. 2, June 2018, pp. 112 - 118
ISSN 2088-4877

After the system has become ready-to-use software, it must be tested before use. This test is carried out using several methods, including White Box, Black Box, Base Path.

f. System evaluation

Users evaluate the finished system as expected; if approved, then proceed to the next step.

g. Using the system

Software that has been tested and accepted by users is ready to use.

4. Testing and implementation

What is done at this stage is to implement the results of the research conducted. In the implementation of this software, the things that will be done include inputting data for all lectures, classes, rooms, and so on

RESULT and DISCUSSION

The results present the results of the final data analysis instead of unprocessed raw data.

To test the performance of the system, trials were conducted by looking at the results of several criteria as follows:

Percentage of successfully scheduled lectures.

1. Percentage of violations against hard constraints.
2. Percentage of violations against soft constraints.
3. Computational time is used to generate solutions.

The data used is lecture data for the even semester of the 2013/2014 academic year at the Tarbiyah Department, STAIN Palopo. The data was obtained from the Academic Sub Division of STAIN Palopo. It consists of 332 lectures, 129 courses, and 109 lecturers. The number of rooms is as many as 17 rooms. The available time slots are eight slots per day divided into two sessions, namely the morning session and the afternoon session. One credit uses a one-time slot, which is 45 minutes long, with lecture days starting from Monday to Saturday.

The determining factor in scheduling is the constraint, divided into constraints, namely constraints that should not be violated, and soft constraints, namely limits that can be violated but as far as possible avoided (minimized). In this study, the hard constraints are as follows:

1. No lecturer teaches more than one subject study at the same time.
2. No lecturer is in more than one room at the same time.
3. A room can only be used by one class at one particular time.
4. Room capacity must be equal to or greater than the capacity of the class to be held. While those that are set as soft constraints are as follows:
   a. Strive to accommodate the lecturer's request not to teach at certain times.
   b. It endeavors that a lecturer does not teach more than twice a day.

Table 1. Test parameters

No.	Parameters	Parameter Value
1.	Number of lectures	332
2.	Number of rooms	17
3.	Number of lecturers	109
4.	Number of time slots	48
5.	HMCR	0,98
6.	PAR	0,02
7.	HMS	48

8.	NI	100

The test was carried out on a computer with an Intel Core i5 2.30 GHz processor and 4 GB RAM. Using the help of application programs built using the Java programming language with Netbean 7.5 environment. As for the database management using MySql ver. 5.0.65.

The application program carries out the scheduling process by calculating input data such as lecturer data, course data, and class data. First, the data is stored in each database table. Then the application program is taken and entered into a matrix with column dimensions representing space and rows representing time slots. The matrix is then calculated using the principles of the harmony search algorithm as described in the previous section. The purpose of the calculations carried out is to produce an array of matrix vectors with the minimum possible objective function value or, in other words, to produce the most feasible solution vector.

After several experiments with various combinations of parameter values (HMCR, PAR, HMS, NI), the best solution was finally obtained, as described in Table 2.

Table 2. Test Results

No.	Parameters	Description
1.	Percentage of successful lectures scheduled	100%
2.	Percentage of violation of hard constraints	0%
3.	Percentage of violation of soft constraints	0%
4.	Computing time	53 second

From the test results as described in Table 2, from the total number of lectures, as many as 332 were successfully scheduled, or in other words, no lectures were not successfully scheduled. Of the four hard constraints set, none was violated, as well as the two soft constraints; none was violated. The amount of time it takes to build a solution is 53 seconds.

CONCLUSION

Based on the research that has been done, the Harmony Search algorithm can be used to solve lecture scheduling problems. Furthermore, the Harmony Search algorithm also successfully provided the optimum solution without any violations, either hard or soft constraints.

For development purposes, follow-up of this research is needed, including:

1. The addition of other constraints to produce a more optimal system.
2. Trying other optimization methods, which can later be compared with this research.

REFERENCES

Ana Ratna Wati, D., & Agusti Rochman, Y. (2008). Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma Particle Swarm Optimization (PSO). UII Yogyakarta.

Aulia, I., Nababan, E. B., & Muchtar, M. A. (2012). Penerapan Harmony Search Algorithm dalam Permasalahan Penjadwalan Flow Shop. Jurnal Dunia Teknologi Informasi, 1, 1–7.

Desiani, A., & Arhami, M. (2006). Konsep Kecerdasan Buatan. Andi Offset.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. Simulation, 76(2), 60–68. https://doi.org/10.1177/003754970107600201

Mahdavi, M., Fesanghary, A., & Damangir, E. (2007). An. Improved Harmony Search Algorithm for Solving Optimization Problems. Applied Mathematics and Computation, 188, 1567–1579.

Priyo, S. S., Helmie, A. W., Indra, W., Nurdin, B., & K, T. W. (2016). Chapter 7 Backpropagation dan Aplikasinya (pp. 135–146).

Rachmawati, H. (2012). Analisis Penyelesaian Masalah Penjadwalan Kuliah Menggunakan Pewarnaan Graph oleh Algoritma Koloni Lebah dengan Parameter Soft Contsraints Prioritas Dosen. ITS Surabaya.

Santosa, B., & Willy, P. (2011). Metoda Heuristik, Konsep dan Implementasi. Guna Widya.

Setemen, K. (2008). Kombinasi Algoritma Genetika dan Tabu Search dalalm Penyusunan Jadwal Kuliah. ITS Surabaya.

Sutiono, N., Imanuel Malik, N., & Rojali. (2012). Automatisasi Timetabling Asisten Pengajar pada Software Laboratory Center menggunakan Harmony Search. Binus University.

Suyanto. (2007). Artificial Intelligence; Searching, Reasoning, Planning and Learning. Informatika.

Teddy. (2009). Penyelesaian Penjadwalan Kuliah Sebagai Constraint Satisfaction Problem dengan Genetic Algorithm. Universitas Indonesia.