Information Systems International Conference (ISICO), 2 – 4 December 2013

Solving University Timetabling Problem Using Harmony Search: a Case Study in STMIK Mikroskil

Harrison, Hardy, Ng Poi Wong
Department of Computer Science, STMIK Mikroskil

Keywords:

Harmony Search
University Timetabling
Machine Learning
Optimization

ABSTRACT

A University Timetabling Problem (UTP) belongs to NP-complete (Nondeterministic Polynomial-Complete) problem category since it has huge search space that consists of many students, classrooms, lectures, and courses permutation. Due to that, finding an optimal schedule using exact algorithm is impossible to be done under a reasonable time.

Harmony Search is a heuristic method that can solve NP-Complete problem such as UTP in reasonable time. However because of its heuristic nature, it can only provide solution which is close to the perfect solution. Therefore, to prove that a schedule is a feasible solution, several constraints have to be satisfied. This research used Harmony Search to solve the UTP whereby the dataset, constraints, and experiments are acquired and performed in STMIK Mikroskil.

This research also evaluates the speed performance and fitness value of Harmony Search using different parameter configuration. From its finding, the HMCR parameter value that is best suited for scheduling is in range of 0.7 to 0.95. The PAR value however, did not sufficiently contribute to the speed and fitness value.

Corresponding Author:

Harrison,
Department of Computer Science,
STMIK Mikroskil,
Jalan Thamrin No. 140, Medan, Indonesia.
Email: conan_teleport@hotmail.com

1. INTRODUCTION

A University Timetabling Problem (UTP) is a placement of a set of courses in predefined time period [1]. UTP consists of lecture and exam scheduling where both of them face common problem of finding a schedule over a limited time period while avoiding conflicts and satisfying a number of side-constraint [2]. UTP has long been known as NP-complete problem class [3]. NP-complete problem class can't be solved using exact algorithm since it has a huge search space whereby iterating each possible solutions can't be done under a reasonable time period. However, heuristic algorithms such as Genetic Algorithm [4], Particle Swarm Optimization [5], Harmony Search (HS) [6] and many others can be used to solve NP-complete problem. These heuristic algorithms can provide an optimal solution to UTP. HS is chosen as the preferred method of this research due to its performance that is better than other heuristic algorithms [6].

This research is perfomed in STMIK Mikroskil environment. All the data that are used as input to HS are collected from STMIK Mikroskil. The resulting schedule also have to conform to the constraints that are stated in the STMIK Mikroskil academic rules. The contribution from this research is majorly directed toward the academic improvement of STMIK Mikroskil and the evaluation of speed performance and fitness value of HS using different parameter configurations that are best suited for scheduling purpose.

2. LITERATURE REVIEW

University Timetabling Problem has been investigated in several researches such as Al-Betar et al. [7], Nguyen et al. [8], and Chen and Shih [9]. In Al-Betar et al. [7]'s work, HS is used with multi-pitch adjusting rate. Nguyen et al [8] used a hybrid algorithm of HS and Bees [10] algorithm. Chen and Shih [9] used constriction Particle Swarm Optimization with local search. HS is chosen as the prefered method in solving UTP in STMIK Mikroskil due to its good performance in Socha Benchmark[2] [7].

HS is inspired by musician's music improvisation [11]. It follows the pattern of a musician during improvisation processes which are: (1) famous tune retrieval from his or her memory; (2) adjusting the pitch of the aforementioned tune and replays it again; (3) composing new or random notes. Geem [11] formalize those three components into: usage of harmony memory, pitch adjusting and randomization. The algorithm of HS is presented in Algorithm 1.

Algoritma 1 Harmony Search algorithm [11].

```
begin
    Define objectife function f(x), x=(x1,x2, …, xd)^T
    Define harmony memory accepting rate (r_accept)
    Define pitch adjusting rate (r_pa) and other parameters
    Generate Harmony Memory with random harmonies
    while ( t< max number of iterations )
       while ( i<=number of variables)
          if (rand<r_accept), Choose a value from HM for the variable i
             if (rand<r_pa), Adjust the value by adding certain amount
             end if
          else Choose a random value
          end if
    end while
    Accept the new harmony (solution) if better
end while
Find the current best solution
```

Geem et al. [6] have done benchmarking in HS with other heuristic algorithms by using Braken and McCormick [12]'s problem. The problem statement is as follows.

Minimize $f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$

Subject to $g_1(x) = 0$

$g_2(x) \geq 0$

with

$g_1(x) = x_1 - 2x_2 + 1$

$g_2(x) = -x_1^2/4 - x_2^2 + 1$

The components x_1 and x_2 are both initialized randomly with uniform distribution over the range [-10.0, 10.0]. The results are then compared with results from other researches, such as: Homaifer et al. [13]'s comparison of using exact solution, Genetic Algorithm (GA) [4], and Generalized Reduced Gradient (GRG) [14]; and Fogel [15] using Evolutionary Programming [16].

Table 1 shows the comparison of results from various methods. HS shows good result with relative error between the algorithm results and the exact function value only +0.2153. Although GRG shows better result from the HS, however, it violates the inequality constraint.

Table 1. Comparison of Results from various methods for Braken and McCormick [12]'s problem [6]

	EXACT	GRG	GA	EP	HS (1)	HS (2)
f(x)	1.3935	1.3934	1.4339	1.3772	1.3771	1.3965
%	0.0000	-0.0072	+2.8992	-1.1697	-1.1769	+0.2153
x_1	0.82288	0.8229	0.8080	0.8350	0.8348	0.8290
x_2	0.91144	0.9115	0.8854	0.9125	0.9124	0.9080
g_1	7.05×10^{-9}	1.0×10^{-4}	3.7×10^{-2}	1.0×10^{-2}	1.0×10^{-2}	1.3×10^{-2}
g_2	1.73×10^{-8}	-5.2×10^{-5}	5.2×10^{-2}	-7.0×10^{-3}	-6.7×10^{-3}	3.7×10^{-3}

3. RESEARCH METHOD

This section presents a generalization of quantitative research method that are divided into three distinct phases which are: (1) data and constraint acquisition from STMIK Mikroskil, (2) harmony memory and HS parameters initialization, and (3) HS processing. The result evaluation is presented in the next section.

[2] http://iridia.ulb.ac.be/msampels/tt.data/

3.1. Data and Constraint Acquisition Phase

STMIK Mikroskil have four departments comprises of two undergraduate degree (S-1), and two vocational degree (D-3). The undergraduate degree further comprises of Computer Science department and Information System department, while the vocational degree comprises of Informatics Management department and Computerized Accounting department. Each department have their own curriculum, lecturer, and student data. However, several lecturers can teach different departments as long as they have the course prerequisite and there is no conflict between two or more taken courses in lecture schedule.

Since STMIK Mikroskil have two lecture sessions which are morning and afternoon sessions, therefore there must be two schedules generated for each period. There are also hard constraints and soft constraints that have to be satisfied. The hard constraints and soft constraints can be seen in Table 2.

Table 2. Hard constraint and soft constraint specification for scheduling criterias

Name of Constraint	Morning	Afternoon
Hard Constraint		
Each lecturer can only teach for maximum 18 units in one semester.	✓	✓
Each lecturer can not teach two courses in the same class.	✓	✓
Each lecturer can only teach for maximum two different courses in one semester	✓	✓
Afternoon classes must have at least one day schedule free.	×	✓
Soft Constraint		
In assigning lecturer, homebased lecturers have priority over other lecturers.	✓	✓
Monday until Friday are given higher priorities over other Saturday.	✓	×
It's preferable that lecturer can teach five units in one day.	✓	✓
It's preferable that student can take classes with total of five units in one day.	✓	✓
For religious purpose, it's preferable that on Friday student can take classes with total of three units where units allocation is on the first until the third hours of day.	✓	×

3.2. Harmony Memory and HS Parameters Initializing Phase

In this phase, Harmony Memory (HM) is initialized. The first step in initialization is to determine the HM size. The second step is to convert lectures, courses, available days, and status of lecturers (homebased or non-homebased) dataset into several matrixes. The following is the matrixes of Harmony Memory.

$$lecturer\_matrix = \begin{bmatrix} lid_1 & lname_1 & c_1 & s_1 & dtm_1 & dta_1 \\ lid_2 & lname_2 & c_2 & s_2 & dtm_2 & dta_2 \\ lid_3 & lname_3 & c_3 & s_3 & dtm_3 & dta_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ lid_n & lname_n & c_n & s_n & dtm_n & dta_n \end{bmatrix}$$

where:
lid_i = i^{th} lecture id
$lname_i$ = it^h lecture name
c_i = course that can be taught by i^{th} lecture
s_i = status of i^{th} lecture
dtm_i = date and time that is available to i^{th} lecture in morning session
dta_i = date and time that is available to i^{th} lecture in afternoon session

$$dcm = \begin{bmatrix} [cid_{11} & c_{11} & cu_{11}] & \dots & [cid_{1m} & c_{1m} & cu_{1m}] & s_1 & sm_1 & sa_1 \\ [cid_{21} & c_{21} & cu_{21}] & \dots & [cid_{2m} & c_{2m} & cu_{2m}] & s_2 & sm_2 & sa_2 \\ & \dots & & \dots & & \dots & & \dots & \dots & \dots \\ [cid_{n1} & c_{n1} & cu_{n1}] & \dots & [cid_{nm} & c_{nm} & cu_{nm}] & s_n & sm_n & sa_n \end{bmatrix}$$

where:
cid_{ij} = j^{th} course id of i^{th} semester
c_{ij} = j^{th} course name of i^{th} semester
cu_{ij} = sum of j^{th} course unit of i^{th} semester
s_i = i^{th} semester
sm_i = sum of morning classes of i^{th} semester
sa_i = sum of afternoon classes of i^{th} semester

$$course\_matrix = [dcm_1 \quad dcm_2 \quad \dots \quad dcm_n]$$

where:
dcm_i = course of i^{th} department

$$schedule\_matrix = \begin{bmatrix} [cc_{11} & lc_{11}] & \dots & [cc_{1m} & lc_{1m}] & tc_1 & dc_1 & sc_1 & c_1 \\ [cc_{21} & lc_{21}] & \dots & [cc_{2m} & lc_{2m}] & tc_2 & dc_2 & sc_2 & c_2 \\ \dots & & \dots & \dots & & \dots & \dots & \dots & \dots \\ [cc_{n1} & lc_{n1}] & \dots & [cc_{nm} & lc_{nm}] & tc_n & dc_n & sc_n & c_n \end{bmatrix}$$

where:
cc_{ij} = j^{th} course of i^{th} class
lc_{ij} = j^{th} lecturer of i^{th} class
tc_i = lecturing time of i^{th} class (morning and afternoon)

dc_i = department of i^{th} class
sc_i = semester of i^{th} class
c_i = i^{th} class

For the two parameter of HS, namely: HMCR and PAR, this research use the recommended value as seen in Geem [11]'s work. The recommended value for HMCR is between 0.1 to 0.4 while for PAR is between 0.7 to 0.95.

3.3. Harmony Search Processing Phase

In this phase, each class of morning and afternoon session schedule are generated through the improvisation process of HS. Steps of the improvisation process is as follows.

1. Generate random value between 0.0 to 1.0. The value will be compared to HMCR value. If it is higher or equal with HMCR, then a new random schedule is generated. However, if it is smaller than HMCR then randomly take the previous schedule in HM.
2. Generate another random value between 0.0 to 1.0. The value will be compared to PAR value. If it is higher or equal with PAR, then check whether there are collision in the schedule. When there are no conflict detected in the schedule, the schedule will be used as the new solution. If it is lower than PAR then the schedule will be adjusted in accordance to the bandwith value. Bandwidth value is generated by random with value from 0 to the number of available days.
3. If all the schedule in Harmony Memory have been used and all of them contain coflict then, a new schedule will be regenerated and return to step 1.
4. However, if the schedule have reached termination criterias, the improvisation process will be stopped. There are two termination criterias available: (1) if the generated schedule have satistied all the hard constraint, or (2) have run 500 iteration.

4. RESULTS AND ANALYSIS

This section presents the results that were obtained and the analysis of those results. Since HS has two parameters (HMCR and PAR), this research also evaluate which configuration of parameter is the best for scheduling purpose. For evaluation, the program is run 25 times where each run has 500 iteration termination criteria. The academic data that are used in each run is as follows.

1. Three department data are used, namely Computer Science, Information System, and Computerized Accounting department.
2. Computer Science department has 13 classes in morning session and 9 classes in afternoon session.
3. Information System department has 8 classes in morning session and 7 classes in afternoon session.
4. Computerized Accounting department has 12 classes in morning session and 12 classes in afternoon session.
5. Total lecturers that are available for three departments is 84 people.

Table 3 shows the speed performance and fitness value for the best schedule obtained using different parameter configuration. Every generated schedule that is stated in Table 3 have satisfied all the hard constraint.

Table 3. Speed performance and fitness value for the best schedule obtained using different parameter configuration

Parameter Configuration	**HMCR = 0.8 PAR = 0.4**	**HMCR = 0.5 PAR = 0.4**	**HMCR = 0.8 PAR = 0.7**	**HMCR = 0.5 PAR = 0.7**
Time (s)	61.656	156.125	49.453	149.797
Fitness value (0~1)	0.98904	0.98780	0.98699	0.98699

From the result, the best parameter for HMCR is in the range of 0.7 to 0.95. The PAR value doesn't seem to significantly affect the result.

5. CONCLUSION

In this research, our implementation of Harmony Search has fulfilled every hard constraint of UTP problem in STMIK Mikroskil. This means that the generated schedule is feasible for usage in real academic situation. It is also found that HMCR value with range of 0.7 to 0.95 is best suited for scheduling purpose. For future work, it is hoped that the scheduler application scope can be widen into classroom allocation problem.

ACKNOWLEDGEMENTS

This research was supported by STMIK Mikroskil grant 1757/ST.45/01/PN/2012. Many thanks to STMIK Mikroskil for opening their dataset.

REFERENCES

[1] B. Edman, et al. "Automated University Timetabling: The State of the Art". University of Nottingham, UK, 2004.
[2] M.W. Carter, G. Laporte, and J.W. Chinneck, "A General Examination Scheduling System," *Interfaces*, vol. 24, no. 3, pp. 109-120, The Institute of Management Sciences, May-June 1994. Department of Industrial Engineering, University of Toronto, Ontario, Canada; Ecole des Hautes Etudes Commerciales de Montr´eal, Quebec, Canada; and Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada.
[3] T.B. Cooper and J.H. Kingston, "The Complexity of Timetable Construction Problems," in the *Practice and Theory of Automated Timetabling, ed. EK Burke and P Ross, pp. 283-295, Springer-Verlag (Lecture Notes in Computer Science)*, 1996. Basser Department of Computer Science, University of Sydney, Australia.
[4] D.E. Goldberg and J.H. Holland, "Genetic algorithms and machine learning". *Machine learning, 3(2), 95-99*, 1988
[5] R. Eberhart and J. Kennedy, October. "A new optimizer using particle swarm theory". In *Micro Machine and Human Science*, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE, 1995
[6] Z. W. Geem, J. H. Kim and G.V. Loganathan, "*A new heuristic optimization algorithm: harmony search. Simulation*", 76(2), 60-68, 2001
[7] Al-Betar, M. Azmi, A. T. Khader, and I. Y. Liao. "*A harmony search with multi-pitch adjusting rate for the university course timetabling*." Recent Advances In Harmony Search Algorithm. Springer Berlin Heidelberg, 2010.
[8] K. Nguyen, P. Nguyen, and N. Tran. "*A hybrid algorithm of Harmony Search and Bees Algorithm for a University Course Timetabling Problem*." International Journal of Computer Science Issues(IJCSI) 9.1, 2012.
[9] R. M. Chen and H. F. Shih. "*Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search*." Algorithms 6.2., 2013.
[10] D. T. Pham, Ghanbarzadeh A., Koc E., Otri S., Rahim S., and M.Zaidi. "*The Bees Algorithm - A Novel Tool for Complex Optimisation Problems*", Proceedings of IPROMS 2006 Conference, pp.454-461, 2006.
[11] Z. W. Geem, "Music-Inspired Harmony Search Algorithm: Theory and Applications", Springer-Verlag Berlin Heidelberg, 2009.
[12] J. Bracken and G. P. McCormick, "Selected Applications of Nonlinear Programming", John Wiley & Sons, 1968.
[13] A. Homaifar, C. X. Qi, and S. H. Lai, "Constrained Optimization via Genetic Algorithms." *SIMULATION*, Vol. 62, No. 4, pp 242-254, 1994.
[14] L.S. Lasdon, et al. "Design and testing of a generalized reduced gradient code for nonlinear programming.", *ACM Transactions on Mathematical Software (TOMS)* 4, no. 1: 34-50, 1976.
[15] D. B. Fogel, "A Comparison of Evolutionary Programming and Genetic Algorithms on Selected Constrained Optimization Problems." *SIMULATION*, Vol. 64, No. 6, 399-406, 1995.
[16] L. J. Fogel, A. J. Owens and M. J. Walsh, "Artificial Intelligence Through Simulated Evolution", John Wiley & Sons, New York, NY, 1966.

BIBLIOGRAPHY OF AUTHORS

	**Harrison** received the B.Sc. degree in Computer Science from STMIK Mikroskil in 2012. He is currently teaching in STMIK Mikroskil. His research interests include artificial intelligence and database.
	**Hardy** received the B.Sc. degree in Computer Science from STMIK Mikroskil in 2008 and M.Sc. degree in Computer Science from Universiti Sains Malaysia in 2011. He is currently serving as Head of Computer Science Department in STMIK Mikroskil. His research interests include machine learning, and optimization algorithms.

Ng Poi Wong received the B.Sc. degree in Computer Science from STMIK Mikroskil in 2003 and M.T.I. degree in Information Technology from Universitas Indonesia in 2008. He is currently serving as Head of Academic Administrative Department in STMIK Mikroskil. His research interests include database and image processing.