A hybrid algorithm of Harmony Search and Bees Algorithm for a University Course Timetabling Problem

Khang Nguyen[1], Phuc Nguyen[2] and Nuong Tran[3]

[1] HCMC University of Science, Vietnam

[2] Cao Thang Technical College, Vietnam

[3] HCMC University of Science, Vietnam

Abstract

This paper is concerned with the development of a new hybrid metaheuristic approach for solving a practical university course timetabling problem in Vietnam. Our hybrid method is a combination of Harmony Search (HS) algorithm and the Bees algorithm. The proposed method has been tested on 14 real-world data instances and compared with some other metaheurisitic approaches, which are Variable Neighborhood Search, Tabu Search and Bee Algorithms. Numerical results indicate the effectiveness of the hybrid HS – Bees algorithm over the others for the specific problem.

Keywords: *Bees algorithm, harmony search, university course timetabling.*

1. Introduction

The univeristy timetabling problems in general are known to be NP-complete problems [4]. It often involves in assignment of a set of courses to a given number of periods and rooms subject to some hard and soft constraints. Hard constraints must be satisfied to produce a feasible solution. Soft constraints should be satisfied as much as possible. In the timetabling literature, there are several meta-heuristic approaches have been proposed and efficiently applied for solving these problems, such as Tabu Search [3,9], Simulated Annealing [12], Variable Neighborhood Search (VNS) [10], Memetic Algorithm [11], Genetic Agorithm [13], Harmony Search [14,18], Great Deluge[15], etc. Many approaches have been introducted for solving specific curiculum-based course timetabling problem in Vietnam [2,3,4,5,15,16]. In this paper, we proposed the hybrid of Harmony Search (HS) [17] and Bees algorithm [1] for this problem. Our approach is tested over 14 real-world data which are taken from the Faculty of Information Technology, HCMC University of Science in Vietnam and compared with previous works [2,3,4]. The results show that our approach is capable of finding high quality solutions for this problem.

This paper is organized as follow: section 2 describes the details of the considered course university timetabling problem, section 3 explains details of the proposed algorithm, section 4 presents the experimental results and the comparison between proposed approach and the previous works, and section 5 shows some conclusions.

2. Problem Description

The general aim of the educational timetabling problem is to find an appropriate assignment of a set of courses (events) into a limited number of periods and rooms in such a way that satisfying a number of pre-defined constraints. The problem considered in this paper is a real-world curriculum – based course timetabling problem. This problem is highly constrained with 19 constraints, including 8 hard constraints, which must be satisfied, and 12 soft constraints, which should be satisfied as much as possible. Each soft constraint is associated with a weight factor to represent its level of importance.

- Time concepts: including the concepts of period, lecture and session.
  - + Period: each period often last about 45 minutes.
  - + Session: is a group of consecutive periods that forms a distinct part of the day, e.g., morning session includes all the periods in the morning.
  - + Lecture: is a part that lasts one period of a course
- Resource concepts: including the concepts of lecturer, class, room and building
  - + Class: is a group of students that will attend the same courses.

+ Building of the faculty: is a set of near rooms. Each building is often distanced far from the others, so there should be some constraints to eliminate the movement of lecturers and students between these buildings in the same day.

- Major concepts: including the concepts of course and curriculum
  + Course: is a group of lectures that have the same lecturer, the same attending classes and subject. The main information of a course includes the lecturer teaching this course, one or some attending classes and the number of lectures in a week that belonging to this course, i.e., the number of periods that this course must hold in a week. Note that some courses may be pre-assigned, i.e., their periods and rooms are pre-determined by the staff, these courses will not be re-scheduled again but their pre-assigned information (e.g., the assignment of their lecturer, classes) will be considered during the timetabling process of the other courses.
  + Curriculum: is a group of courses that should not be overlapped due to the special requirements of the university. In our considered problem, a course is allowed to belong to more than one curriculum.

Next, we present the constraints of the problems. As mentioned above, each soft constraint has a relevant weight and this value is shown just right before the description of the constraint.

- Eight hard constraints (denoted by letter H and their index):

[H1] Lecturers, classes and rooms could not be assigned to periods at which they're not available (the availability of those resources are pre-defined)
[H2] All the lectures of all courses must be assigned.
[H3] The pre-assignment of any course must be respected.
[H4] The capacity of a room that is assigned to a course must be equal or greater than the planned number of students attending that course.
[H5] Courses that have the same lecturer or class must not overlap.
[H6] No room could be assigned to two simultaneous courses.
[H7] The lectures of each course in a day must belong to the same session
[H8] All lectures that belong to the same course must be assigned to the same room

- Twelve soft constraints (including one high weighted soft constraint, denoted by HS and 11 low weighted soft constraints, denoted by letter S, followed by their weight values):

[HS1] (60) Courses that belong to the same curriculum should not be overlapped.
[LS1] (10) The movement of a lecturer between two buildings in the same day should be avoided.
[LS2] (5) The movement of a class between two buildings in the same day should be avoided.
[LS3] (10) Courses should be assigned to periods that they're preferred.
[LS4] (5) The idle time between consecutive courses in the same session for a lecturer or a class should be avoided.
[LS5] (5) The number of periods that a class is assigned in a day should be equal or less than maxStudyingPeriods.
[LS6] (5)The idle time between consecutive courses in the same day for a lecturer or a class should be avoided.
[LS7] (10) The number of sessions that a lecturer was assigned to should be as small as possible.
[LS8] (10) The number of periods that a lecturer is assigned in a day should be equal or less than maxTeachingPeriods.
[LS9] (5) The number of days that a class was assigned to should be as small as possible.
[LS10] (10) In the timetable of a class in session, if there are three periods in which this class is idle, these should be consecutive periods (this is a special constraint requested by the timetabling staff because they need use these idle periods for another purpose).
[LS11] (10) All courses that belong to the same curriculum which are scheduled to the same session should be assigned to the same building at that session.

3. The Algorithm

3.1. Hybrid HS-Bees Algorithm

In this subsection, we briefly review the general ideas of Harmony Search algorithm and Bees algorithm, besides that, the hybrid approach combining those two algorithms, namely the Hybrid HS-Bees algorithm is presented.

3.1.1. Harmony Search Algorithm

The harmony search (HS) algorithm is a new population-based metaheuristic algorithm proposed by Geem et al. [17] in 2001. The algorithm mimics the characteristics of a musician. The mimicked characteristics of a musician include the random selection, memory consideration and pitch adjustment. The goal of the process is to reach a perfect state of harmony. HS's parameters include the Harmony Memory Size (HMS), which is the number of harmonies (solutions) in the harmony memory; Harmony Memory Considering Rate (HMCR); Pitch Adjusting Rate (PAR) and the number of improvisations (NI). The harmony memory (HM) is a memory location (history) where all the harmonies (solutions) are stored. The four basic steps of HS algorithm are shown in Figure 1, and details of the improvising step are described in Figure 2.

1. Initialize a Harmony Memory (HM).
2. Improvise a new harmony from HM.
3. If the new harmony is better than the worst harmony in HM, put the new harmony in HM, and exclude the worst harmony from HM.
4. If stopping criteria are not satisfied, go to 2.

Fig. 1. The basic steps of the Harmony Search

Define a new harmony (solution) $x_{new}=(x_1, x_2,\ldots, x_n)$
1. While (i <= the number of components (n))
2. If (rand < HMCR)
3. Choose a random value from HM for the component x_i.
4. If (rand < PAR), Adjust the value of component x_i
5. End If
6. Else
7. Choose a random value for component x_i.
8. End If.
9. End while
10. Return x_{new}

Fig. 2. Improvise a new harmony from HM

3.1.2. The Bees Algorithm

The bees algorithm is a population-based search algorithm proposed by Pham et. al. [1]. It's inspired by the foraging behaviour of honey bees and performs a kind of neighbourhood search combined with random search to find the global optimum. The algorithm requires a number of parameters to be set, namely the number of scout bees (n), the number of sites selected for neighborhood search (out of n visited sites) (m), the number of elite bees among m selected sites (e), the number of bees recruited for patches visited by e elite bees (nep), the number of bees recruited for the other (m-e) selected patches (nsp) and size of patches (ngh).

1. Initialise the population with random solutions
2. Evaluate the fitness of the population
3. while (stopping criterion is not met)
//Forming new population
4. Select sites for neighbourhood search
5. Recruit bees for selected sites (the number of bees for a sites is proportional to the quality of that site) and evaluate the fitness.
6. Select the fittest bee from each patch
7. Assign remaining bees to search randomly and evaluate their fitness.
8. end while.

Fig. 3. The basic steps of the Bees Algorithm

The algorithm starts with the n scout bees being placed randomly in the search space. The fitness values of the sites that scout bees visited are evaluated in step 2. The bees having the highest fitness values are selected as "elite bees" in step 4 and sites visited by them are chosen for neighbourhood search in steps 5-6. The algorithm recruits more bees to follow the elite bees (e) than the other selected bees (m-e) in order to search around the neighbourhood of sites visited by the elite bees. In step 6, only one bee with the highest fitness value will be selected for each site to generate the next bee population. Besides, the remaining bees (n-m) in the bee population are randomly assigned around the search space in order to explore new potential solutions in step 7. These steps are repeated until the stopping criterion is met.

3.1.3. The Hybrid HS-Bees Algorithm

The Hybrid HS – Bees algorithm combines the optimization capabilities of harmony search algorithm and bees algorithm. The disadvantages of the basic bees algorithm are premature convergence and not obtaining experiences between solutions in a population. The Harmony search algorithm handles intensification and diversification. The diversification is controlled by pitch adjustment and random selection, which would help good local solutions to be retained. Random selection explores global search space more widely and efficiently while the pitch adjustment makes the new solution good enough and near the existing good solutions. The intensification in Harmony search algorithm is controlled by memory consideration, leading the searching process toward the searching space of good solutions. The hybrid HS-Bees algorithm involves two phases of optimization: (a) The bees algorithm using neighbourhood search and random search, (b) The harmony search using memory consideration, random selection and pitch adjustment. The general process of the algorithm is shown in Figure 4.

1. Initialise population with random solutions
2. Evaluate fitness of the population
3. While (stopping criterion not met)
// Bees algorithm
4. Select sites for neighbourhood search
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch
7. Assign remaining bees to search randomly and evaluate their fitnesses.
// Harmony Search Algorithm
8. Initialize Harmony Memory (HM).
9. Improvise a new harmony from HM.
10. Improve the new harmony using Hill Climbing

11. If the new harmony is better than the worst harmony in HM, include the new harmony in HM, and exclude the worst harmony from HM.
12. End While.

Fig. 4. The proposed hybrid HS - Bees Algorithm

3.2. The Hybrid HS-Bees Algorithm for the considered university course timetabling problem

3.2.1. Objective function

As described in section 2, the considered problem has 8 hard constraints and 12 soft constraints. Each soft constraint has a weighted value to represent its importance and the quality of a solution will be evaluated by the objective function $f(x)$, which is a linear weighted combination of the violation of soft constraints. In this approach, hard constraints H5 and H6 are relaxed. There're two reasons for this relaxation: first, those constraints is not easily to satisfied during the search because its relevant feasible assignment are dynamic, second, relaxing them will help to avoid the case of stucking in some local region by the lack of connectivity in the searching space. Those relaxed constraints will be treated as soft constraints during the searching process, with dominantly high weighted value compared to the original soft constraints (here we use the weighted value of 1000), helping the search not go too far from the real feasible region.

3.2.2. Creating random initial solutions

In those random initial solutions, every course is randomly assigned to periods and rooms in such a way that the non-relaxed hard constraints are not violated.

3.2.3. Moves used in the neighbourhood search

In Bees algorithm's phase, neighbour solution is created from current solution by randomly choosing a kind of move in the set of two kinds of moves, including: Single moves and Swap moves. These moves are used for searching in the neighbourhood of the selected sites with *ngh* parameter is 10 percent of a random available set of courses of current solution.

- Single moves: moves that changing either the period assignment or the room assignment of a course. Single move includes 3 components: a Course k, the new first period p and the new room r. We consider a set of courses (neighbourhood) (called N_{cur}) in current solution. Firstly, for each course k in N_{cur}, examine all available first period p for this course k. Secondly, find the best appropriate room r which is available at period p and its capacity must be equal or greater than the planned number of attending students. Finally, we choose the best single move in the set of single moves (k, p, r) which are created above.

- Swap moves: moves that exchanging either the period assignment or the room assignment of two courses. Each swap move includes 2 components: Course k_i and Course k_j. We consider a set of courses (neighbourhood) (called N_{cur}) in current solution. Firstly, for each course k_i in N_{cur}, examine all courses k_j in N_{cur} which is able to swap first period and room with course k_i. Secondly, we choose the best swap move in a set of swap moves (k_i, k_j) which are created above.

Create_A_Neighbour_Solution(X)
1. Randomly Select $\alpha \in \{1 \ldots ngh\}$
2. Set X' = X
3. Randomly choose a kind of move in two kinds of moves (Single move, Swap move)
4. Take a best solution X' in the neighbourhood of X created by the chosen kind of move with α percent of available of set of courses of X.
5. return X'

Fig. 5. The method to create a neighbour solution of X

3.2.4. Details of the application of Harmony Search Algorithm in proposed approach

Initialize Harmony Memory: In this step, all solutions in bees' population is sorted in ascending order by objective function values. Then, we initialize HM based on the the HMS best solutions in the population.

Improvise a new harmony: In Harmony search algorithm, three Harmony search operations are applied during each improvement of a new harmony, namely Memory consideration, random selection and pitch adjustment. Each course in a solution is a variable in the solution's relevant harmony.

a. *Memory consideration*: Memory consideration operators would decide the first period and room of every course of the new harmony based on solutions stored in HM with HMCR rate.

b. *Random selection*: In this operator, we choose randomly the new first period and new room for courses of the new harmony with (1-HMCR) rate.

c. *Pitch adjustment*: Pitch adjustment operator is applied with PARxHMCR rate, the first period and room of the chosen course of the new harmony are taken from the best harmony in HM.

d. *Additional Hill Climbing algorithm*: Hill Climbing is an iterated algorithm for optimization problems. In this problem, it attempts to improve a newly generated harmony as a new operator of Harmony search algorithm. A neighbour solution is created by using single move and swap move in the neighbourhood search above.

3.2.5. Stop criteria

The hybrid algorithm will stop when the stop criteria (the maximum number of allowance iterations) is met.

4. Experimental results

The hybrid HS-Bees algorithm is implemented in C++. The computer used to test has the configuration of Core2 Duo CPU, 4GB RAM and Windows Vista OS. Fourteen real- world data instances taken from the Faculty of Information Technology, HCMC University of Science in Vietnam are used to evaluate the hybrid algorithm.
The hybrid algorithm was run on each instance for ten times and the best and average results are reported in table I. In this implementation, the weights of soft constraints are: w(HS1) = 60, w(S1) = 15, w(S2) = 10, w(S3) = 10, w(S4) = 5, w(S5) = 2, w(S6) = 5, w(S7) = 10, w(S8) = 10, w(S9) = 5, w(S10) = 10, w(S11) = 10. In Bees algorithm's phase, we choose n (scout bees) in the set of {20, 30}, m (selected sites) in set of {4, 5}, e (number of elite bees) in set of {3, 4}, nep (number of bees recruited for patches visited by e elite bees) in set of {4, 5}, nsp (number of bees recruited for the other (m-e) selected patches) in set of {2, 3}, ngh (patch of size) in set of {5, 10}. Otherwise, In HS phase, we choose HMCR rate in the set of {0.7,0.8,0.95}, PAR rate in the set of {0.3, 0.4, 0.5}. We test all possible pairs of these parameters' values and find out that the pair {n=30, m=5, e=4, nep=4, nsp=2, ngh=10, HMCR=0.95, PAR=0.4} gives the best result.

Table 1. Details of 14 data instances

# instance	# courses	# teachers	# classes	# rooms	# curricula
Data 1	82	44	13	40	11
Data 2	80	44	10	46	10
Data 3	95	44	14	45	11
Data 4	84	45	11	40	10
Data 5	89	48	12	51	11
Data 6	51	31	19	45	0
Data 7	96	50	14	47	9
Data 8	88	44	11	47	5
Data 9	107	43	15	46	5
Data 10	78	47	12	48	3
Data 11	83	45	13	53	7
Data 12	87	52	21	53	5
Data 13	84	51	20	53	6
Data 14	102	55	26	53	9

Table 2. Best results on 14 data instances

#instance	Hybrid HS-Bees Algorithm	VNS	The Bees Algorithm	Tabu Search
Data 1	**172**	2647	435	272
Data 2	**194**	1209	382	1141
Data 3	**305**	767	569	577
Data 4	**188**	867	405	1657
Data 5	**142**	693	444	572
Data 6	**155**	1595	1876	1450
Data 7	213	1343	498	142
Data 8	1378	358	1420	154
Data 9	**269**	564	480	282
Data 10	1100	312	343	1167
Data 11	592	354	343	147
Data 12	1447	319	315	1125
Data 13	497	405	466	231
Data 14	**185**	490	297	210

5. Conclusion

In this paper, the hyrid HS - Bees algorithm is presented to solve a real-world university timetabling problem. The results show that the proposed algorithm is capable of efficiently solving this problem and has been confirmed by comparing with known results: VNS, The Bees Algorithm, Tabu Search. For future research, the hybrid HS - Bees algorithm can be applied other timetabling problems like high school timetabling, examination timetabling.

References

[1] 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.

[2] NTTM Khang, NB Phuc, TTH Nuong, "The Bees Algorithm for a Practical University Timetabling Problem in Vietnam", Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference, pp. 42-47, Shanghai, 2011.

[3] NTTM Khang, DTT Nguyen, TT Khon, TTH Nuong, "Automating a Real-World University Timetabling Problem with Tabu Search Algorithm", Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), pp. 1-6, Hanoi, 2010.

[4] NTTM Khang, NT Quang, TTT Hien, NB Phuc, TTH Nuong, "Variable Neighborhood Search for a Real-world Curriculum-based University Timetabling Problem", The Third International Conference on Knowledge and Systems Engineering (KSE 2011), Hanoi, 2011.

[5] NTTM Khang, NQ Nam, NQ Vi, NB Phuc, TTH Nuong, "Applications of particle swarm optimization algorithm to a practical university timetabling problem", 2011 International

Conference on Modeling and Optimization (ICMO 2011), Cairo, Egypt, 2011 (to appear).
[6] C Lara, J Flores, F Calderón, "Solving a School Timetabling Problem Using a Bee Algorithm", Proceedings of the 7th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence, pp. 664 – 674, Springer-Verlag Berlin, 2008.
[7] B. McCollum, P. McMullan, B. Paechter, R. Lewis, A. Schaerf, L. Di Gaspero, A. J. Parkes, R. Qu, E. Burke: Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition. Technical Report, Internaltional Timetabling Competition 07-08, 2007.
[8] TB Cooper and JH 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, 1996.
[9] LÄu Z, Hao JK, "Adaptive tabu search for course timetabling". European Journal of Operational Research doi: 10.1016/j.ejor.2008.12.007, 2009.
[10] Abdullah S.,Burke E.K. and McCollum B.:"An Investigation of a Variable Neighborhood Search Approach for Course Timetabling". Proceedings of the 2nd Multidisciplinary Conferenceon Scheduling: Theory and Applications, 18-21July, NY, USA, 413-427, July, 2005.
[11] S. N. Jat and S. Yang, "A memetic algorithm for the university course timetabling problem," Proc. 20th IEEE Int. Conf. Tools Artif. Intell. 2008, vol. 1, pp. 427–433, 2008.
[12] D. Abramson. "Constructing school timetables using simulated annealing: sequential and parallel algorithms". In Management Science, vol.37, pp. 98-113, 1991.
[13] Burke E.K., Elliman D.G. and Weare R.F. "A Genetic Algorithm Based University Timetabling System" East-West Conference on Computer Technologies in Education, Crimea, Ukraine pp. 35-40, 1994.
[14] M. A. Al-Betar, A.T. Khader, T.A. Gani, "A harmony search algorithm for university course timetabling", PATAT 2008.
[15] McMullan, "An Extended Implementation of the Great Deluge Algorithm for Course Timetabling", Lecture Notes in Computer Science, Springer, Vol 4487, 538-545, 2007.
[16] DT Anh, VH Tam, NQV Hung, "Combined Interactive and Automatic Course Timetabling", Proceedings of International Workshop on Advanced Computing and Applications, ACOMP'2007, HoChiMinh City, March 14-16, 2007, pp.2-7, 2007.
[17] Geem, Z.W., Kim, J.H., Loganathan, G.V.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation. 76(2), pp.60–68, 2001.
[18] M. A. Al-Betar, A.T. Khader, I.Y. Liao. "A Harmony Search with Multi-pitch Adjusting Rate for the University Course Timetabling", Recent Advances in Harmony Search Algorithm, SCI 270, pp. 147-161, Springer-Verlag Berlin Heidelberg, 2010.

Khang Nguyen received B.S degree and M.S degree in Computer Science from HCMC University of Science, Vietnam in 1996 and 2001. Currently, he is a Ph.D candidate in Computer Science in HCMC University of Science, Vietnam. Since 1996, he has been a lecturer at Faculty of Information Technology, HCMC University of Science, Vietnam. His research interests include timetabling problem, optimization problem. Khang Nguyen is co-author over ten papers in international conferences and national conferences.

Phuc Nguyen reveived B.S degree and M.S degree in Computer Science from HCMC University of Science, Vietnam in 2007 and 2011. Since 2008, he has been a lecturer at Faculty of Electronic and Informatics, Cao Thang Technical College, Vietnam. His research interests in timetabling problem.

Prof. Dr. Nuong Tran received Ph.D degree in Mathematics from University of Warsaw, Poland in 1989. Currently, she is a lecturer at Faculty of Mathematics, HCMC University of Science, Vietnam. Her research interests include timetabling problem, optimization problems, optimization global. Nuong Tran is co-author over thirty papers in international journals and international conferences.