A Two Stage Approach for High School Timetabling

Moh'd Khaled Yousef Shambour[1], Ahamad Tajudin Khader[1], Ahmed Kheiri[2], and Ender Özcan[2]

[1] School of Computer Sciences, Universiti Sains Malaysia (USM), 11800, Pulau Pinang, Malaysia
[2] School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK

Abstract. There are different types of educational timetabling problems which are computationally difficult to solve. In this study, we deal with the High School Timetabling Problem which requires assignment of events, such as courses, and resources, such as classrooms, to time-slots under a set of different types of constraints. We describe an approach that hybridises an Evolutionary Algorithm variant and Simulated Annealing methods to solve this problem. This approach is tested over a set of real world instances obtained across different countries. The empirical results demonstrate the viability of the hybrid approach when compared to the previously proposed techniques.

1 Introduction

High school timetabling problem (HSTP) is a real-world hard combinatorial optimisation problem [7]. It seeks a search for the best event schedule and the best allocation of resources including the scheduling of classes, teachers, courses and students in a time slots in a high school institution subject to a set of constraints. These constraints are classified as *hard* and *soft* constraints. The hard constraints must be satisfied to ensure the feasibility of the solution whereas the soft constraints characterise preferences. The violation of the soft constraint does not destroy the feasibility but rather affecting the quality of the solution.

Due to the NP nature of high school timetabling problems, meta-heuristics are preferred in most of the previous studies. Approaches used in HSTP include simulated annealing [8], evolutionary algorithms [6], tabu search [15], adaptive large neighborhood search [20], hyper-heuristic [11], particle swarm optimisation [21], bee algorithms [14], integer programming [5], tiling algorithms [13], walk down jump up algorithm [24] and constraint programming approach [22]. [16] provided a recent survey on HSTP. In this study, we introduce a high school timetabling problem and combining Harmony Search Algorithm as an Evolution Strategy with Simulated Annealing approach (HSA-SA) as a solver to the problem. To the best knowledge of authors, this is the first work that employs HSA-SA for HSTP.

Harmony search algorithm (HSA) is an Evolutionary Algorithm variant. It is a population-based metaheuristic named by Geem et al. (2001) [9]. Although it was introduced as new, soon after, HSA was classified as a special case of Evolution Strategies in [23]. In this study, we will adapt the terminology as used in [9] while discussing the algorithmic components. The terminology used in HSA is inspired from the process of musical improvisation where a group of musicians play the pitches of their musical instruments together seeking for a pleasing harmony as determined by aesthetic standards. Based on harmony memory size (HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR) and the number of improvisations (NI), a new harmony vector X=x_1,x_2,.,x_n is generated. In optimisation process, musician (decision variable) plays (generates) a possible note (value of decision variable) for finding the best harmony (optimal solution). At each iteration, a candidate solution (harmony) is evaluated by objective function. The move acceptance criteria considers whether to keep or replace the new solution with the respect to the worst solution in the harmony memory. This cycle continues until the termination criteria (maximum NI or condition) is satisfied. HSA has been successfully applied to a range of real world optimisation problems, including job shop scheduling [25], task assignment problem [26], optimising energy consumption [10], nurse rostering problem [4] and university course timetabling [2]. More on HSA can be found in [3].

Simulated Annealing [1] is a probabilistic metaheuristic method in which at each iteration a new solution is generated. The new solution is accepted if it improved the previous solution. The non-improving solutions are accepted with a probability of $p_t = e^{-\frac{\Delta}{T}}$, where Δ is the quality change, and T is the method parameter, called temperature which regulates the probability to accept solutions with higher objective value (cost) [1].

Section 2 provides an overview of the high school timetabling problem dealt with in this work. Section 3 describes the proposed solver components that are tested for solving the high school timetabling problem. Section 4 provides the empirical results. Finally, Section 5 presents the conclusion.

2 International Timetabling Competition 2011 Datasets

Recently, the challenge has become increasingly highlighted when a group of researchers run the Third International Timetabling Competition (ITC2011) in 2011-2012 [17], with the goal of raising the profile of automated high school timetabling. The ITC2011 was run by the Centre for Telematics and Information Technology at the University of Twente in the Netherlands, aiming to drive a new era of research of automated high school timetabling. The participants of ITC2011 tackled 35 instances of the high school timetabling problem, taken from schools in 10 countries. The instances are defined by a standard data format based on XML schema called XHSTT (XML High School Timetabling) [18, 19]. Out of 17 registered participants to the competition, only 5 teams submitted solutions. The reason is unknown, but could be due to the large number of imposed constraints which makes the problem hard to handle in practice [19].

Briefly, the ITC2011 problem instances [19] contain four components including *times*, *resources*, *events/meetings* and *constrains*. Time represents indivisible interval of time during which event run. Resource represents the entity that attend event. For example: "T1" resource in "BrazilInstance3_XHSTT2013" instance refers to a teacher while resource "S1" in the same instance refers to a class. Event is a meeting between resources. Some events might be pre-assigned with time or resource; For example, the event "RE-3A" predefined the time to "Mon_1" in "ItalyInstance1_XHSTT2012A" instance. Also an event can be split into sub-events. Constraint represents the condition that a solution should satisfy, if possible. The ITC2011 problem instances contain 15 types of constraints. All constraints could be hard and soft according to a given instance. For more description, see [17–19].

2.1 Characteristics of the problem instances

In this work, twelve ITC2011 high school timetabling problem instances are used to study the effectiveness of the proposed HSA. The instances are small and medium in size taken from three countries: Brazil, Finland and Italy. Table 1 summarises the main characteristics of these instances. The datasets can be downloaded from the ITC2011 website [17].

Table 1. Datasets characteristics (*Times* total number of times. *Teachers*, *Rooms* and *Classes* are the total number of resources of resource type "Teacher", "Room" and "Class", respectively. *Events* total number of events.)

Country	Instance Name	Times	Teachers	Rooms	Classes	Events
Brazil	Instance1	25	8		3	21
Brazil	Instance2	25	14		6	63
Brazil	Instance3	25	16		8	69
Brazil	Instance4	25	23		12	127
Brazil	Instance5	25	31		13	119
Brazil	Instance6	25	30		14	140
Brazil	Instance7	25	33		20	205
Finland	ElementarySchool	35	22	21	60	291
Finland	SecondarySchool	35	25	25	14	280
Finland	SecondarySchool2	40	22	21	36	469
Italy	Instance1	36	13		3	42
Italy	Instance4	36	61		38	748

3 Harmony Search Algorithm and Simulated Annealing

HSA is a population-based meta-heuristic approach inspired by the musical improvisation process [9]. It iteratively improves the solutions stored in the harmony memory. It randomly perturbs a solution (vector of decision variables) via

a number of improvisations and the harmony memory is updated during the improvisations. At each improvisation, stored values of decision variables in the harmony memory are adapted according to Harmony Memory Consideration Rate (HMCR); and the variable values in the solution are adjusted according to a Pitch Adjustment Rate (PAR). This cycle continues until the stop criteria is met. The steps of the HSA are as follows [9].

Step 1. Initialise the parameters: The HSTP variables are extracted from the HSTP instances. The variables vary from instance to instance such as existence of certain resource type(s), number of events (meetings) and number of hard and soft constraints. The parameters of HSA including: (i) Harmony Memory Consideration Rate (HMCR), the rate of selecting a value from Harmony Memory HM (memory consideration) or taking into account a random consideration; (ii) Harmony Memory Size (HMS), the number of solution vectors in the HM; (iii) Pitch Adjustment Rate (PAR), the rate of adjusting the values; and (iv) Number of Improvisations (NI), the number of iterations.

Step 2. Initialise the harmony memory: The HMS initial solutions are generated using the general solver of KHE implemented by Kingston [12]. The hard constraints of the initial solutions are, likely, violated and the following steps need to fix the violation of hard constraints. Each row in HM represents one solution of HSTP and $f(x)$ is the objective function value (Equation 1).

$$\mathsf{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_N^{HMS} \end{bmatrix} = \begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HMS}) \end{bmatrix} \tag{1}$$

Step 3. Improvise a new harmony: The essential step in HSA is to improvise a new harmony vector $x'=(x'_1,x'_2,\dots,x'_N)$ based on three operators: memory consideration, pitch adjustment and random consideration. In *memory consideration*, the values of the new harmony vector are randomly selected from the current values stored in HM $x_i^{'}$ with a probability of HMCR where HMCR $\in$ [0, 1]. In *random consideration*, decision variables are randomly assigned according to their possible range with a probability of (1-HMCR). Furthermore, if memory consideration failed to maintain the feasibility of the new solution, then the random consideration will be called. In *pitch adjustment*, the decision variable x_i of a new harmony vector is pitch adjusted with a probability of PAR, where PAR $\in$ [0,1]. In HSTP, pitch adjustment is a neighbourhood move. In this work, we have implemented three neighbourhood moves:

- Move Meeting (MM) moves the time slot t_1 of meeting m to time slot t_2.
- Swap Meeting (SM) swaps two meetings m_1 and m_2.
- Do Nothing (DN); each of which is selected with equal probability as shown in Equation 2.

$$\textbf{Adjust } x' = \begin{cases} MM & \text{if } 0 \leq \text{rand} < (PAR/3), \\ SM & \text{if } (PAR/3) \leq \text{rand} < (2.PAR/3), \quad s.t\ rand \in [0,1] \\ DN & \text{if } (2.PAR/3) \leq \text{rand} < (PAR). \end{cases} \tag{2}$$

Step 4. Update the harmony memory: If the new generated harmony solution vector x'=(x'_1,x'_2,..,x'_N) has better quality than the worst harmony vector stored in HM in terms of the objective function (cost) value, then the worst harmony in HM will be replaced by the new harmony.
Step 5. Check the stopping criteria: Steps 3 and 4 are repeated until the maximum number of improvisations (NI) is reached.

After applying an iteration of the HSA, the Simulated Annealing method will be employed as a polishing procedure with the goal of improving the current best solution obtained. At each iteration one of 5 neighbourhoods is randomly selected and applied to the candidate solution. The neighbourhood moves are: Move Meeting; Swap Meeting; Swap Three Meetings (swaps meetings m_1 and m_2 then m_2 and m_3); Swap Block of Meetings (swaps time slot of meetings m_1 and m_2, but if the meetings have different duration, m_1 is moved to the following the last time slot occupied by m_2); and Task Split (split task into two). The first two moves are explained in step 3 of HSA. The temperature of the SA method is a function of the number of meetings.

4 Results

The parameter values of the HSA are chosen as HMS=5, HMCR=0.99, PAR=0.66 and NI=50. These values are decided after a set of exhaustive experiments using different combinations of values which is not reported in this paper due to space requirements. The proposed hybrid approach is implemented in C, under CenTos 6.4 operating system. The experiments are performed on an Intel(R) Xeon(R) CPU X7560 @2.27GHz with a memory of 4.00G.

The HSA-SA is tested on ITC2011 instances described in table 1. Solutions are evaluated in terms of *feasibility* (sum of weighted hard constraints violations) and *preferences* (sum of weighted soft constraints violations); and the goal is to minimise it. A solution with a cost value of 42.0013 indicates an infeasibility value of 42 and objective value of 13. Five competitors were submitted solutions to the ITC2011. HySST [11] applies a multi-stage hyper-heuristic on a set of mutational heuristics and hill climbers, GOAL [8] combines the iterated local search with simulated annealing, HFT [6] uses the evolutionary algorithms as a meta-heuristic to solve the problem, Lectio [20] applies an algorithm based on adaptive large neighbourhood search, and VAGOS [17].

Table 2 presents the performance comparison of the HSA-SA to the five competing approaches and the best publicly known (BPK) solutions before the five approaches found better solutions. The table shows the cost of the solutions obtained by the different approaches. The column KHE in the table shows the average of five solutions produced by the general solver of KHE implemented

by Kingston [12]. From the table, HSA-SA was able to produce feasible solutions to all the tested instances and enhance the initial solutions generated by the general solver of KHE. The proposed HSA-SA improved upon BPK in six out of seven instances. The HSA-SA performs the best on ElementarySchool and SecondarySchool2 instances. The HSA-SA produces a solution satisfying all the hard and soft constraints on SecondarySchool2 instance. The HSA-SA outperforms the HFT approach in all the instances. In BrazilInstance4, HSA-SA generates better solution than HySST. Note that in the table, "-" are shown to indicate that the solver did not submit solution for the instance. Most of the tested instances are taken from instances used during Round-1 of the ITC2011 competition, and the competitors were expected to submit solutions if it improved on the best solution previously known to the organisers of the competition. No restrictions were placed on the time limit or how the solutions could be obtained. For more details about the competition, see [19].

Table 2. The cost values obtained by KHE, HSA-SA, the best publicly known solution before the start of the ITC2011 (BPK), and the competitors solvers

Instance Name	KHE	HSA-SA	BPK	GOAL	HySST	Lectio	HFT	VAGOS
BrazilInstance1	0.0081	0.0020	0.0024	-	-	-	-	0.0011
Instance2	3.9999	0.0048	-	-	0.0044	0.0005	0.0082	0.0026
Instance3	3.9999	0.0154	-	-	0.0084	0.0048	0.0212	0.0047
BrazilInstance4	39.9999	0.0162	0.0112	-	0.0176	0.0090	0.0205	0.0078
Instance5	12.9999	0.0148	0.0225	-	-	-	-	0.0043
Instance6	11.9999	0.0186	0.0209	-	0.0150	0.0060	0.0347	0.0074
Instance7	22.9999	0.0234	0.0330	-	-	-	-	0.0122
ElementarySchool	10.0306	0.0003	-	0.0003	0.0003	0.0003	0.0003	-
SecondarySchool	43.9999	0.0090	0.0106	-	-	0.0088	-	-
SecondarySchool2	2.9999	0.0000	-	0.0000	0.0000	0.0000	0.0576	-
ItalyInstance1	0.3022	0.0020	0.0028	-	-	-	-	0.0012
ItalyInstance4	38.9999	0.0082	-	0.0061	0.0052	0.0078	0.8623	-

5 Conclusion and Future Directions

A unified high school timetabling problem which was a topic of a recent competition, referred to as ITC2011, is described in this study. ITC2011 provided a collection of high school timetabling problem instances collected from different countries across the world. The goal of the competition was to promote researchers and practitioners to deal with the real world complexities of the problem. We have combined an Evolution Algorithm variant and Simulated Annealing methods (HSA-SA) and tested on twelve ITC2011 benchmark instances as an initial study. The results show that the approach is sufficiently powerful producing high quality solutions. The proposed approach was able to generate solutions to the problem matching the quality of the best known solutions in two

occasions and near to the best known solutions for the other instances. As future work, we would like to analyse the influence of choice of parameter values of the approach and extend our experiments to all ITC2011 benchmark instances. Although the performance of the HSA-SA is evaluated on high school timetabling, it is also our intention to investigate its performance on the other educational timetabling problems, such as university course timetabling. Finally, we aim to improve the performance of the approach further by incorporating new move operators including particularly local search.

References

1. Abramson, D.A., Dang, H., Krisnamoorthy, M.: Simulated annealing cooling schedules for the school timetabling problem. Asia-Pacific Journal of Operational Research 16(1), 1–22 (1999)
2. Al-Betar, M.A., Khader, A.T.: A harmony search algorithm for university course timetabling. Annals of Operations Research 194(1), 3–31 (2012)
3. Alia, O.M., Mandava, R.: The variants of the harmony search algorithm: an overview. Artificial Intelligence Review 36(1), 49–68 (2011)
4. Awadallah, M.A., Khader, A.T., Al-Betar, M.A., Bolaji, A.: Nurse rostering using modified harmony search algorithm. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) Swarm, Evolutionary, and Memetic Computing, Lecture Notes in Computer Science, vol. 7077, pp. 27–37. Springer Berlin Heidelberg (2011)
5. Birbas, T., Daskalaki, S., Housos, E.: School timetabling for quality student and teacher schedules. Journal of Scheduling 12(2), 177–197 (2009)
6. Domrös, J., Homberger, J.: An evolutionary algorithm for high school timetabling. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012). pp. 485–488 (2012)
7. Even, S., Itai, A., Shamir, A.: On the Complexity of Timetable and Multicommodity Flow Problems. SIAM Journal on Computing 5(4), 691–703 (1976)
8. Fonseca, G.H., Brito, S.S., Santos, H.G.: A simulated annealing based approach to the high school timetabling problem. In: Yin, H., Costa, J.A., Barreto, G. (eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2012, Lecture Notes in Computer Science, vol. 7435, pp. 540–549. Springer Berlin Heidelberg (2012)
9. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: Harmony search. Simulation 76(2), 60–68 (2001)
10. Hoang, D.C., Yadav, P., Kumar, R., Panda, S.: A robust harmony search algorithm based clustering protocol for wireless sensor networks. In: Communications Workshops (ICC), 2010 IEEE International Conference on. pp. 1–5 (2010)
11. Kheiri, A., Özcan, E., Parkes, A.J.: Hysst: Hyper-heuristic search strategies and timetabling. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012). pp. 497–499 (2012)
12. Kingston, J.H.: A software library for high school timetabling. http://sydney.edu.au/engineering/it/~jeff/khe/ (2009)
13. Kingston, J.H.: A tiling algorithm for high school timetabling. In: Burke, E., Trick, M. (eds.) Practice and Theory of Automated Timetabling V, Lecture Notes in Computer Science, vol. 3616, pp. 208–225. Springer Berlin Heidelberg (2005)

14. Lara, C., Flores, J.J., Calderon, F.: Solving a school timetabling problem using a bee algorithm. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 5317, pp. 664–674. Springer Berlin Heidelberg (2008)
15. Minh, K.N.T.T., Thanh, N.D.T., Trang, K.T., Hue, N.T.T.: Using tabu search for solving a high school timetabling problem. In: Nguyen, N.T., Katarzyniak, R., Chen, S.M. (eds.) Advances in Intelligent Information and Database Systems, Studies in Computational Intelligence, vol. 283, pp. 305–313. Springer Berlin Heidelberg (2010)
16. Pillay, N.: A survey of school timetabling research. Annals of Operations Research pp. 1–33 (2013)
17. Post, G.: Benchmarking project for high school timetabling. http://www.utwente.nl/ctit/hstt/ (2011)
18. Post, G., Ahmadi, S., Daskalaki, S., Kingston, J.H., Kyngas, J., Nurmi, C., Ranson, D.: An xml format for benchmarks in high school timetabling. Annals of Operations Research 194(1), 385–397 (2012)
19. Post, G., Gaspero, L., Kingston, J.H., McCollum, B., Schaerf, A.: The third international timetabling competition. Annals of Operations Research pp. 1–7 (2013)
20. Sørensen, M., Kristiansen, S., Stidsen, T.R.: International timetabling competition 2011: an adaptive large neighborhood search algorithm. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012). pp. 489–492 (2012)
21. Tassopoulos, I.X., Beligiannis, G.N.: A hybrid particle swarm optimization based algorithm for high school timetabling problems. Applied Soft Computing 12(11), 3472 – 3489 (2012)
22. Valouxis, C., Housos, E.: Constraint programming approach for school timetabling. Computers and Operations Research 30(10), 1555–1572 (2003)
23. Weyland, D.: A rigorous analysis of the harmony search algorithm: How the research community can be misled by a "novel" methodology. International Journal of Applied Metaheuristic Computing 1(2), 50–60 (2010)
24. Wilke, P., Killer, H.: Walk down jump up algorithm a new hybrid algorithm for timetabling problems. In: Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2010). pp. 440–446 (2010)
25. Yuan, Y., Xu, H., Yang, J.: A hybrid harmony search algorithm for the flexible job shop scheduling problem. Applied Soft Computing 13(7), 3259 – 3272 (2013)
26. Zou, D., Gao, L., Li, S., Wu, J., Wang, X.: A novel global harmony search algorithm for task assignment problem. Journal of Systems and Software 83(10), 1678 – 1688 (2010)