INTELLIGENT EXAMINATION TIMETABLING SYSTEM USING HYBRID INTELLIGENT WATER DROPS ALGORITHM

Bashar A. AlDeeb[1], Norita Md Norwawi[2], Mohammed A. Al-Betar[3], and Mohd Z. Jali[4]

[1]*Universiti Sains Islam Malaysia (USIM), Malaysia, Bashar_Deeb@hotmail.com*
[2]*Universiti Sains Islam Malaysia (USIM), Malaysia, Norita@usim.edu.my*
[3]*Al-Balqa Applied University, Jordan, betar79@gmail.com*
[4]*Universiti Sains Islam Malaysia (USIM), Malaysia, Zalisham@usim.edu.my*

ABSTRACT. This paper proposes Hybrid Intelligent Water Drops (H-IWD) algorithm to solve Tamhidi programs uncapacitated examination timetabling problem in Universiti Sains Islamic Malaysia (USIM). Intelligent Water Drops algorithm (IWD) is a population-based algorithm where each drop represents a solution and the sharing between the drops during the search lead to better drops. The results of this study prove that the proposed algorithm can produce a high quality examination timetable in shorter time in comparison with the manual timetable.

Keywords: optimization, intelligent water drops algorithm, examination timetable, Universiti Sains Islamic Malaysia

INTRODUCTION

When exams or courses are scheduled, education timetabling is supposed to be an ongoing challenge faced by most institutions either they are academic or professional. The reason for this is a large number of constraints which are to be accommodated in the schedule. Departments and individual faculties often schedule the courses. On the other hand, timetable of examination is generally generated on central basis in order to facilitate whole university. The problems in this manner are supposed to be complex that involves various constraints. Thus, it is found a challenging task for both the practitioners and researchers to create timetable or to conduct a research study respectively (Al-Betar & Khader, 2012).

Many approaches have been developed for the purpose of solving problems regarding timetabling for the previous decades. The approaches, in this regard include graph based sequential techniques (Badoni & Gupta, 2014; Saharan, 2014; Sabar et al., 2012) and constraint based techniques (Müller, 2009). Other methods are local search methods such as simulated annealing (Battistutta et al., 2014; Burke et al., 2004; Gogos et al., 2012), tabu search. Several population based algorithms include ant colony optimization (Thepphakorn et al., 2014; Eley, 2007; Dowsland & Thompson, 2005), genetic algorithms (Innet, 2013), scatter search (Sabar & Ayob, 2009) and hybrid approaches (Alzaqebah & Abdullah, 2015; Abdullah & Alzaqebah, 2013) and etc.

In recent two decades, the researchers have found different algorithms by simulating the behaviors of various swarms of insects and animals such as fishes, bees, and ants, this type of algorithms called swarm intelligence algorithms. Many of swarm intelligence algorithms have been designed and applied to timetabling problem in the educational organizations (Fong et al., 2014; Alzaqebah & Abdullah, 2015; Turabieh & Abdullah, 2011; Dowsland & Thompson, 2005). Since, our interest lies in Intelligent Water Drops (IWD) algorithm which is one of the recently developed algorithms in the swarm intelligence field, this algorithm mimicked the dynamic of river systems (Shah-Hosseini, 2007).

TAMHIDI PROGRAM OF USIM

Tamhidi Program of USIM is a program for the students who have the Certificate of Malaysian Education (SPM which is referring to Sijil Pelajaran Malaysia). Tamhidi Program aims to prepare its students to the first degree level. The term "Tamhidi" is an Arabic word, which means "Matriculation" that has the same meaning with "Preparatory Year", "Preparation" or "Foundation Program". The duration of Tamhidi is one year. Through this year, students should be qualified for admission into the undergraduate programs which offered by the USIM. There are five undergraduate programs in Tamhidi namely:

1. Science and Technology (TST).
2. Medicine (TM).
3. Accounting and Muamalat (TAC).
4. Syariah and Law (TSU).
5. Dentistry (TD).

In the first semester of 2014/2015, Tamhidi program included approximately 900 students spread over all programs. In each semester, each student has to take 25 credit hours with minimum of 6 subjects. At the moment, final exams timetable is construct manually by the administration of the tamhidi program, it requires more persons and days to do that, specially, if the university administration increases the number of programs and students in this faculty. To construct exam timetable, the tamhidi programs administration need to schedule a number of subjects or courses to a number of timeslots to achieve soft constraint as much as possible. At the same time, hard constraints must be achieved. Table 1 shows the Uncapacitated Tamhidi Program dataset that will use in this study.

Table 1. Characteristics of the Uncapacitated Tamhidi Program dataset

Institution	Timeslots	Exams	Students	Enrolment	Density
Tamhidi Programs, USIM.	20	18	854	6104	0.51

HYBRID INTELLIGENT WATER DROPS ALGORITHM

IWD algorithm is a population based approach inspired by nature optimization algorithm that replicates certain natural phenomena of a swarm of water drops with the soil onto the river bed. The solutions (path) are recursively constructed by the IWD algorithm. It was investigated thet the utilization of IWD theory has shown success in various optimization problems (Alijla et al., 2014; Shah-Hosseini, 2013; Wood & Wollenberg, 2012; Lenin & Kalavathi, 2012; Shah-Hosseini, 2009, 2007), this is motivate us to investigate IWD algorithm for solving other optimization problem. Therefore, this paper introduces Hybrid IWD algorithm, the proposed algorithm employed to overcome the UETP.

An optimal solution can be constructed with the help of IWD algorithm. The cooperation that among the different groups in which agents lies is known as Water Drops. The phenomenon of this algorithm takes into account the imitation of a water drops swarm and that flows along with the river bed with soil (Shah-Hosseini, 2013). In the process of swarm of water drops, the solution is incrementally constructed by each water drop with the help of series of water iterative solutions initializes from one node to the another single node until the solution with the complete outcome is attained. The communication of water drops with one another is done with the attribute generally known as soil. This soil is collaborated with the ways or paths between dual points. The determination of the movement direction can be done with the value of soil from present node to the next node. Meanwhile, the path with the lesser quantity of soil is certain to be followed (Alijla et al., 2014).

Population-based techniques are able to explore multiple search space regions at a time. However, some limitations occur in the phases of these techniques such as exploitation is less concerned with these techniques as compared to exploration including low convergence speed and premature convergence (Al-Betar & Khader, 2014). In contrast, local search based techniques are able to fine tune the search space region to which they converge and thus find a precise local optimal solution. Therefore, this paper proposes the hybridization of Local Search Optimizer (LSO) with IWD algorithm for UETP to further improve the IWD algorithm exploitation capability.

In the Hybrid-IWD algorithm, the major modification that applies to the IWD algorithm was hybridization of LSO with IWD algorithm. The position where the LSO is placed within the pseudo-code of Hybrid-IWD algorithm is after the solutions construction phase as shown in Algorithm 6.1 lines (21-23). This combined aimed to balance between exploration and exploitation of the search space, thus enhance the solution quality.

Algorithm 1 shows how local search optimizer is hybridized with IWD algorithm as a new operator. In Hybrid-IWD algorithm, LSO is invoked to improve the constructed solutions locally with a probability rate of Local Search Rate (LSR), where LSR belong to (0-1).

Algorithm 1: The main steps for Hybrid IWD algorithm

1 Initialize static parameters.
2 For (NI = 1 to It_{max})
3 Initialize dynamic parameters.
4 Spread one random timeslot to a random exam for all IWD_s.
5 Update the list of visited node $V_s(IWD)$.
6 For (e = 1 to M-1)
7 For (s = 1 to $Drops$)
8 Determine next exam n_e to be scheduled in the solution IWD_s using saturation degree
9 If ($IWD_{exam\text{-}timeslot}(s,n_e,T^{Tbest}(n_e)$=1))
10 $IWD_s(n_e)=T^{Tbest}(n_e)$ Else
11 For(t=0 to P)
12 If ($IWD_{exam\text{-}timeslot}(s,n_e,p)$=1) Push timeslot p in to vector φ.
13 EndFor
14 τ = select random timeslot from φ.
15 $IWD_s(n_e) = \tau$.
16 EndIf
17 Update Velocity of the solution IWD_s.
18 Update Soil of the solution IWD_s.
19 EndFor
20 EndFor
21 If ($RAND(0,1)<LSR$) // Where LSR is to determine the utilization of LSO.
22 $IWD_s=LSO(\,)$
23 EndIf
24 Update soil value for all edges included in the T^{IBest}.
25 Update global best solution T^{TBest}.
26 If (value of $f(T^{TBest})>f(T^{IBest})$)
27 $f(T^{TBest})=f(T^{IBest})$.
28 EndIf EndFor
29 Stops with the global best solution.

This paper employed Hybrid-IWD algorithm to construct final exams timetabling for tamhidi programs in USIM. Administration staff in tamhidi programs is constructing the final exams timetabling at the end of each semester. The examination period is two weeks (five days in each week). Each day includes 2 timeslots (AM and PM). There are common exams between all programs.

In order to construct the final examination timetable, we have to consider several inputs. These inputs grouped into three files:

1. List of courses and their enrolments: This list was included in a text file which consists of two columns: (i) The first column contains the course numbers for all programs in the faculty, (ii) The second column contains the number of students enrolled in each course (Fig. 1).

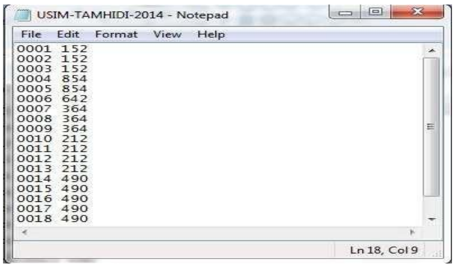

Figure 1. List of Courses and their Enrolment

2. List of students and their course selections: This list was also included in a text file which consists of a number of rows equal to number of students in all programs; each row represents the subjects enrolled by each student.

3. Other parameters (i.e. Exam start date, Number of timeslots, programs names).

Therefore, in this paper, we need to exploit a better solution in searching the best suitable timeslot for the best suitable subject or exam, the quality of the solutions is measured by the value of the penalty cost function. Before developing the examination timetable, we need to take some different hard and soft constraints into account. These constraints may be conflicting; i.e. when you try to satisfying one of the constraints, another constraint will be violated. Therefore, in order to achieve best results, we need to utilize from an algorithm that can be used as an intelligent way for the development of an optimized timetable at every academic year. The hard constraint that we take into account in this paper is: "*No student can sit for two exams simultaneously*" Whilst, "*The exams taken by the same students should be spread out evenly across a timetable*" Is the soft constraint that we take into account in this study and which is desirable to satisfy as much as possible. In real world examination timetabling problem is usually impossible to satisfy all of the soft constraints.

PROBLEM FORMULATION

The notations for tamhidi examination timetabling formulation are stated in Table 2. The hard constraint of tamhidi dataset is:

"***H1**: No student can sit for two exams simultaneously*"

The satisfaction of hard constraint in the solution of tamhidi dataset is given by Eq. 1.

$$x_i \neq x_j \qquad \forall x_i, x_j \in x \wedge c_{i,j} \geq 1 \tag{1}$$

In addition, as in standard benchmark dataset, wherever possible, examinations should be spread out over timeslots so that students have large gaps in between exams, this represent the soft constraint of tamhidi dataset:

"***S1:*** *The exams taken by the same students should be spread out evenly across a timetable*"

Table 2. Notations used to formalize the UETP

Symbol	Description
N	Total number of exams.
P	Total number of timeslots.
M	Total number of students.
$\mathcal{E}$	Set of exams, $\mathcal{E}$= {1, 2, 3... *N*}.
S	Set of students, *S* = {1, 2, 3... *M*}.
T	Set of timeslots, *T* = {1, 2, 3... *T*}.
x	A solution $\mathbf{x} = (x_1, x_2..., x_N)$.
x_i	The timeslot of exam *i*.
$c_{i,j}$	Conflict matrix element: total number of students sharing exam *i* and exam *j*. $$c_{i,j} = \sum_{k=1}^{M} u_{k,i} \times u_{k,j} \quad \forall i,j \in \mathcal{E}$$
$a_{i,j}$	Proximity coefficient matrix element: whether the timetable *x* is penalized based on the distance between timeslot of exam *i* in timeslot of exam *j*. $$a_{i,j} = \begin{cases} 2^{5-\lvert x_j - x_j \rvert} & \text{if } 1 \le \lvert x_j - x_j \rvert \le 5. \\ 0 & \text{Otherwise.} \end{cases}$$
$u_{i,j}$	Student, exam matrix elements: whether student s_i is sitting for exam *j* $$u_{i,j} = \begin{cases} 1 & \text{if student } i \text{ sitting in exam } j. \\ 0 & \text{Otherwise.} \end{cases}$$

OBJECTIVE FUNCTION

In order to adhere with practical issues, we used Burke & Newall (2004) objective function (named as Penalty Cost) which is formulated as the minimization of the sum of proximity costs (See Equation 2):

$$\min \quad f(x) = \frac{1}{M} \times \sum_{i=0}^{N-1} \sum_{j=j+1}^{N} c_{i,j} \times a_{i,j} \tag{2}$$

THE PROPOSED TIMETABLE

Figure 2 shows the main page of tamhidi examination timetabling system. The proposed system involves two phases. The first phase used Hybrid-IWD algorithm to generate feasible solutions that meet the hard constraint, also meet soft constrains as much as possible. Whilst, in second phase, the system read the results file to visualize the solution as shown in Figure 4.

Figure 2. GUI of the proposed system

As shown in Figure 2, the system has a menu bar which contains three options. The first option "Main Menu" consist of four options detailed in the following:

System Configuration "App Config": The user can use this option to: (i) Determine the starting examination timetabling date, (ii) Manage the Tamhidi programs groups, (iii) Change password of the current user, (iv) Add/Delete users.

Running the System: by using this option, the system will read proceed all inputs which include: (i) Courses' data (course code, name, number of students enrolled in each course), (ii) Students' Data (Student ID, name, specialization, group name and the courses enrolled by the students in the current semester) and Number of timeslots needed to the examination timetabling. Note that, there is to methods to input the data (Load txt or excel files and Entire the data using related the window).

Print: This option enables the user to print the examination timetable (See Figure 4).

System Log out "Exit": Use this option is preferable to logout of the system.

The second option in the main menu is "Data" which enables the user to amendment to the courses and students (Add/Delete course, Add/Delete Student, Add/Delete course for any student, change student group).

The last option in the main menu is "Timeslots", through this option the system display the examination timetabling in several forms:

- Display Examination Timetabling for all Tamhidi programs.
- Display Examination Timetabling for each program.
- Display Examination Timetabling for each group.
- Display Examination Timetabling for each student.

EXPERIMENT RESULTS AND DISCUSSION

Applying of Hybrid-IWD algorithm on Tamhidi programs in USIM obtained separated exams for each program. For example, Figure 3 represents the examination timetabling for Science and Technology (TST) program, this timetable shows separated exams which means that the system achieved minimum possible of violation for soft constraints, thus good quality solutions.

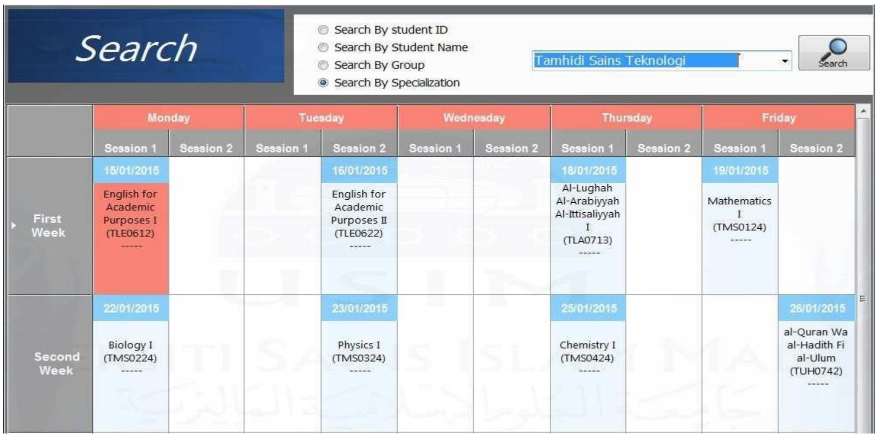

Figure 3. Examination Timetabling for Science and Technology program

The system calculates soft constraints violation (Penalty value) using Equation 2. This equation takes into account the number of common students for each adjacent two exams in the solution. For example, the number of common students between the exams (TLA0713, TMS0124) is 490. While there are 854 common students between the exams (TLE0612, TLE0622). Figure 4 shows that there is only one timeslot between the exams (TLA0713) and (TMS0124), while there are two timeslots between the exams (TLE0612) and (TLE0622).

In other hand, the comparison between the obtained timetable solution and manual solution that made by the staff of Tamhidi programs in USIM shows that the penalty value for the manual solution is (35.38), while the penalty value for the solution obtained by applying Hybrid-IWD algorithm was (23.27), which means that soft constraints violations were less than using the proposed method. This indicates that the results obtained by using Hybrid-IWD algorithm are better than those obtained manually.

CONCLUSION

In this study, the Hybrid-IWD algorithm has been employed for examination timetabling. Based on the comparisons between the results obtained by Hybrid-IWD algorithm and the manual examination timetable, prove that the proposed algorithm can produce a high quality examination timetable in shorter time.

REFERENCES

Abdullah, S. & Alzaqebah, M. (2013). A hybrid self-adaptive bees algorithm for examination timetabling problems. *Applied Soft Computing*, 13(8), 3608–3620.

Al-Betar, M. A. & Khader, A. T. (2012). A harmony search algorithm for university course timetabling. *Annals of Operations Research*, 194(1), 3–31.

Al-Betar, M. A., Khader, A. T. & Doush, I. A. (2014). Memetic techniques for examination timetabling. *Annals of Operations Research*, 218(1), 23–50.

Alijla, B. O., Wong, L. P., Lim, C. P., Khader, A. T., & Al-Betar, M. A. (2014). A modified Intelligent Water Drops algorithm and its application to optimization problems. *Expert Systems with Applications*, 41(15), 6555-6569.

Alzaqebah, M. & Abdullah, S. (2015). Hybrid bee colony optimization for examination timetabling problems. *Computers & Operations Research*.

Badoni, R. P. & Gupta, D. (2014). A graph edge colouring approach for school timetabling problems. *International Journal of Mathematics in Operational Research*, 6(1), 123–138.

Battistutta, M., Schaerf, A. & Urli, T. (2014). Feature-based tuning of single-stage simulated annealing for examination timetabling.

Burke, E., Bykov, Y., Newall, J., & Petrovic, S. (2004). A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36 (6), 509–528.

Dowsland, K. & Thompson, J. (2005). Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56(4), 426–438.

Eley, M., (2007). Ant algorithms for the exam timetabling problem. In *Practice and Theory of Automated Timetabling VI*, 364–382. Springer.

Fong, C. W., Asmuni, H., McCollum, B., McMullan, P, & Omatu, S. (2014). A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems. *Information Sciences*.

Gogos, C., Alefragis, P., & Housos, E., (2012). An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research*, 194(1), 203–221.

Innet, S., (2013). A noval approach of genetic algorithm for solving examination timetabling problems: A case study of Thai Universities. *13th International Symposium on Communications and Information Technologies (ISCIT)*, 233–237.

Lenin, K. & Kalavathi, M. S. (2012). An intelligent water drop algorithm for solving optimal reactive power dispatch problem. *Internat J on Elect Engin and Inform*, 4 (3), 450–463.

Müller, T., (2009). ITC2007 solver description: A hybrid approach. *Annals of Operations Research*, 172 (1), 429–446.

Sabar, N. R. & Ayob, M., (2009). Examination timetabling using scatter search hyper-heuristic. *2nd Conference on Data Mining and Optimization (DMO'09)*, 127–131.

Sabar, N. R., Ayob, M., Qu, R., & Kendall, G. (2012). A graph coloring constructive hyper-heuristic for examination timetabling problems. *Applied Intelligence*, 37(1), 1–11.

Saharan, S., (2014). Resource Optimization in Examination Timetabling using Graph Coloring. *Ph.D. Thesis*, Thapar University.

Shah-Hosseini, H., (2007). Problem solving by intelligent water drops. *IEEE Congress on Evolutionary Computation*, 3226–3231.

Shah-Hosseini, H., (2009). The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation*, 1(1), 71–79.

Shah-Hosseini, H., (2013). Improving K–means clustering algorithm with the intelligent water drops (IWD) algorithm. *International Journal of Data Mining, Modelling and Management*, 5(4), 301–317.

Thepphakorn, T., Pongcharoen, P., & Hicks, C. (2014). An ant colony based timetabling tool. *International Journal of Production Economics*, 149, 131–144.

Turabieh, H. & Abdullah, S. (2011). A hybrid fish swarm optimisation algorithm for solving examination timetabling problems. *Learning and Intelligent Optimization*, 539–551. Springer

Wood, A. J. & Wollenberg, B. F., (2012). *Power generation, operation, and control*. John Wiley & Sons.