

Heuristic Algorithm for Multi-Location Lecture Timetabling

San Nah Sze^{a1}, Huiggy Kuan^a, Kang Leng Chiew^{a2}, Wei King Tiong^{a3}, Chin Siong Heng^b

^a Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300, Sarawak, Malaysia
E-mail: ¹snsze@unimas.my, ²klchiew@unimas.my, ³wktiong@unimas.my

^b Faculty of Cognitive Sciences and Human Development, Universiti Malaysia Sarawak, 94300, Sarawak, Malaysia
E-mail: csheng@unimas.my

Abstract—This paper studies a real faculty timetabling problem with multi-location consideration. Faculty of Cognitive Sciences and Human Development, UNIMAS offers a master's program by coursework to postgraduate students. The construction of timetables for all the courses offered is a tedious process due to constraints such as team-teaching allocation, unavailability dates of lecturers, and multi-location considerations. Therefore, a manually designed timetable is not as practical as it is time consuming when operational constraints must be fulfilled. In this paper, a two-stage heuristic algorithm is proposed to solve this postgraduate coursework timetable problem. This is because the heuristic algorithm is easy to apply and able to generate a feasible solution in a short time. The proposed two-stage heuristic algorithm consists of Lecturer Grouping Stage and Group Allocation Stage. In Stage I, the lecturers are assigned into four lecturer groups with the condition of no identical lecturers in each of the groups. Then, in Stage II, these groups are allocated into a set of academic weeks throughout the semester. The timeslot for each course can be allocated, and the team-teaching slot for the lecturers can be assigned in this stage. The result from the two-stage heuristic algorithm shows remarkable improvement over the real timetables solution by analyzing the distribution of lecture sessions of the courses.

Keywords— university timetabling problem; heuristic algorithm; multi-location; team teaching; coursework timetabling.

I. INTRODUCTION

Timetable scheduling is a vital planning task for both lecturers and students as a well-designed timetable can maximize resource utilization. Although a manually designed timetable may still achieve results using a trial and error method, the feasibility and computational speed are impractical when more considerable operational constraints are introduced to this problem. Hence, creating a timetable is a tedious and time-consuming process [1]. Besides, this problem is NP-hard, imposing an excellent challenge for the manual timetable in getting a feasible solution as well as satisfying all operational constraints [2]–[4].

The timetabling in school or university area are different due to data requirements and the type of constraints [5]. It can be categorized into school timetabling, university course timetabling, and examination timetabling [6]. The university course timetable deals with making a schedule of a lecturers' set for different cohorts of students. This paper studies a university course timetabling where the lecturers are assigned to multi-location.

Generally, the university timetabling problem can be categorized into university timetabling problem (UTTP), course timetabling problem (CTTP), lecture timetabling

problem (LTTP), and examination timetabling problem (ETTP) [7], [8]. Each timetable category consists of different requirements to fulfill that can help in setting the constraints to get a feasible solution. An overview of the timetabling problem-solving approach, such as linear programming [9], constraint-satisfaction strategy, graph-coloring [10], meta-heuristic, and local search algorithms [11], and genetic algorithms [12], [13] are investigated [7], [14].

Consequently, it is essential to have a heuristic algorithm to solve the timetable scheduling problem. The heuristic algorithm is used because of its simplicity and fast response in getting a feasible solution. The literature review shows that various heuristic algorithms have been proposed to solve the timetabling problem in real life such as the Timetabling Heuristic Approach (THA) in Tunisian University [2], Linear Solution in Islamic Azad University [15], Modified Hybrid Particle Swarm Optimization (MHP SO) in Khulna University [16], and Genetic Algorithm in Jakarta University and a Brazilian university [4], [17].

The central aspect of the heuristic is easy to apply in the university timetabling problem to generate a feasible solution quickly by fulfilling the operational constraints in the problem [18]. The heuristic algorithm can generate a feasible solution in a short computational time [19]. A

heuristic approach is used on the specific problem, which does not guarantee an optimal solution but usually makes a better timetable for practical use [20]. Therefore, it can be observed that the heuristic algorithm approach is widely applied in the real-life university timetabling problem.

II. MATERIALS AND METHOD

Faculty of Cognitive Sciences and Human Development (FCSHD) offers a Master of Science in Human Resource Development (MSc HRD) program by coursework to cater to the need of working adults to upgrade themselves. This master program is conducted at multi-location, other than UNIMAS main campus. The program takes a total of two years or four semesters of study on the coursework program plan. Currently, the timetable is planned manually before the start of the new semester. The timetable goes through multiple revisions and adjustments to satisfy all the operational constraints and avoids the timetable clash. All lectures are conducted during the weekend, namely Saturday and Sunday. Each day will have two teaching slots: morning session, which is from 9.00 a.m. to 12.00 p.m. and afternoon session, which is from 2.00 p.m. to 5.00 p.m. as illustrated in Table 1.

TABLE I
TEACHING SLOT TYPE BASED ON LOCATION

Location	UNIMAS		Off-UNIMAS campuses
	Saturday	Sunday	Saturday and Sunday
Teaching Slot			
Morning	9.00 a.m. to 12.00 p.m.	9.00 a.m. to 12.00 p.m.	9.00 a.m. to 5.00 p.m.
Afternoon	2.00 p.m. to 5.00 p.m.	2.00 p.m. to 5.00 p.m.	
Total hours for a lecture session	6	6	14
Weeks to complete	7	7	3
Break	12.00 p.m. to 2.00 p.m.		One hour break

UNIMAS and off-UNIMAS campuses have 14 academic weeks of lectures in a semester and one week of semester break, where no classes are conducted. The academic calendar starts on the same date for all teaching locations. The exam week will be allocated in week 15. Besides that, no classes are allowed during the public holiday too.

The same group of manpower conducts all of the courses of MSc HRD in this multi-location. Limited manpower resources and more courses to conduct in multi-location cause difficulty in generating a feasible timetable solution. The current timetabling practice allows an individual or team to teach when conducting a course. This is because the lecturers may conduct the lecture sessions in other locations, and a secondary lecturer is needed to assist the principal lecturer in continuing that lecture session. Besides, a secondary lecturer can help in conducting the lecture sessions when the main lecturer is not available in some weeks.

Besides that, lecturers might not be available for some dates due to official duty or personal reasons. Hence, the current timetabling practices require a longer time and more

effort to generate a feasible timetable, and at the same time, avoid the unavailability dates of lecturers.

From the case study described above, the operational constraints can be categorized into a set of hard constraints (HC) and soft constraints (SC), which are listed as follows:

- HC₁ A lecturer can only conduct a course at one time.
- HC₂ A lecturer can only conduct in a location at one time.
- HC₃ A student can only attend one course at one time.
- HC₄ A student can only be involved in one cohort at one time.
- HC₅ Total teaching hours for a course must be fulfilled, i.e., 42 hours.
- HC₆ Fulfil the unavailability dates of lecturers.
- HC₇ Fulfil the teaching slot type based on the location, i.e., one course per week (off-UNIMAS campus) or two courses per week (UNIMAS campus).
- SC₁ Even distribution of timetable.
- SC₂ Even distribution of lecture duty.

In this multi-location timetabling problem, the following notation and parameters are used in mathematical modeling. For the UNIMAS campus, a lecturer is required to conduct only a one-day class (two lecture sessions or six hours of lectures) for a course on either Saturday or Sunday. For off-UNIMAS campuses, a lecturer is required to conduct two-day classes on both Saturday and Sunday. The main reason for the two consecutive day classes is due to the traveling budget for off-campus lectures. Consequently, a lecturer can only teach one course at one time, and one location at a time. Table 1 shows the teaching slot type and lecture session based on multi-location for the coursework program plan. The current coursework plan for multi-location is shown in Figure 1.

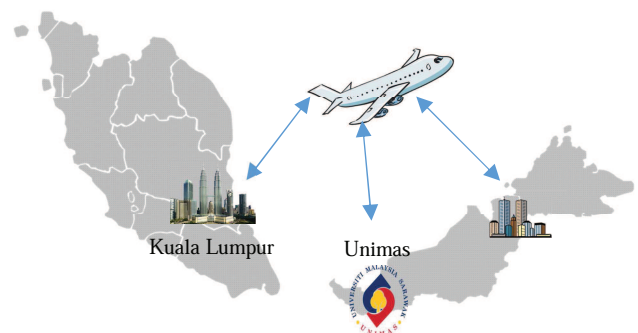


Fig. 1 Coursework program plan in multi-location.

- n = number of courses
- m = number of lecturers
- r = number of lecture sessions for a course
- q = number of locations
- p = cohort of students

C : a set of courses = $\{c_1, c_2, \dots, c_n\}$

L : a set of lecturers = $\{l_1, l_2, \dots, l_m\}$

S : a set of cohorts of students corresponding to locations

$$= \{s_{11}, s_{12}, \dots, s_{qp}\}$$

T : a set of teaching slot types = $\{t_1, t_2, \dots, t_r\}$

V : a set of locations = $\{v_1, v_2, \dots, v_q\}$

E_{c_i} : set of lecturers of courses $c_i l_m l_b$

Set C is a set of courses to be conducted by a set of lecturers L in a set of locations V . Set S is a set of cohorts of students who take a similar curriculum of courses in each area. Each course $c_i l_m l_b$ has a set of individual or team-teaching lecturers to conduct the lecture sessions which is defined as E_{c_i} . There is an assisting lecturer l_b to team-teach for course c_i , else $c_i l_m$ if no team-teaching for the course.

In this section, a two-stage heuristic algorithm is proposed to generate a feasible timetable in FCSHD. The advantage of a heuristic algorithm is its simplicity to cope with abundant constraints and fast response in getting a feasible solution. This section provides problems solving of the multi-location timetabling problem as described in the case study section, a two-stage heuristic algorithm is proposed. The stages in the heuristic algorithm are listed as follows, and a diagram for the process is illustrated in Figure 2:

- Stage I: The lecturer grouping stage clusters all the lecturers into the different groups with a simple rule that each group contains a unique set of teaching lecturers. This is to prevent the timetable from having a conflict during the group allocation stage.
- Stage II: The group allocation stage allocates each lecturer group into a set of academic weeks that is generated at a previous step. At this point, each lecturer in a group is determined by their unavailability dates and then assigned to the particular weeks. This stage ensures that all allocated weeks for each group would not have clashing problems when conducting the lecture sessions in multi-location.

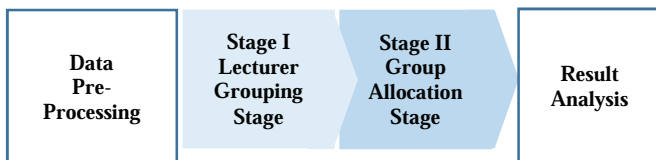


Fig. 2 Stages in the two-stage heuristic algorithm.

Before the two-stage heuristic algorithm is applied, the dataset must be pre-processed first to avoid a clash of lecturer resources. A lecturer is only able to conduct a limited number of courses per semester, as shown in Table 2. The pre-processing data stage is used to set a restriction on the data set and later processed by the heuristic algorithm.

In this paper, a cohort is defined as a group of students who share a similar curriculum or courses together. In this case study, the courses are conducted for three semesters for a group of students. Hence, the lecturers will have to conduct the lectures for three different cohorts in three teaching locations.

TABLE II
NUMBER OF COURSES BY LECTURER IN THREE TEACHING LOCATIONS

Location		
UNIMAS+Kuala Lumpur	UNIMAS+Sibu	Kuala Lumpur+Sibu
3 courses	3 courses	4 courses

A. Pre-Processing: Group Formation

Fourteen academic weeks of a semester are used as a basis to allocate and assign the lecturers to different teaching locations. At the same time, they must fulfill the 42 total teaching hours for a course. For the Saturday and Sunday courses, 12 hours is needed for a lecture. The maximum number of courses that can be conducted in one location is four courses, which is derived from equation (1). Based on equation (2), a course that leads a 12-hour lecture per week needs approximately four weeks to fulfill the total teaching hours. As in our case, the total teaching hours that are required by a course is 42. Hence, a total of four groups is used in this two-stage heuristic algorithm to split the lecturers, where a group will be allocated in any four weeks of a semester to conduct the lectures.

$$\max_{group} = \frac{\text{total hours per week} * 14 \text{ weeks}}{\text{total teaching hours}} \quad (1)$$

$$\text{week} = \frac{\text{total teaching hours}}{\text{total hours per week}} \quad (2)$$

B. Stage I. Lecturer Grouping Stage

In this stage, a set of lecturer groups is defined as $G = \{G1, G2, \dots, G_{|G|}\}$ and $|G| = g$ represents the number of lecturer groups. Each lecturer, l is then weighted and ranked before clustered into the lecturer groups, g . The lecturer grouping algorithm assigns the higher-ranked lecturer into the lecturer groups. Lecturers are allocated based on their teaching slot types of courses that they are required to conduct throughout the semester.

From equation (1), a total of four groups is used to organize the lecturers, and there should be no identical lecturers in each of the groups. Each group represents the courses to be conducted at multi-location in a week. This result will then be processed by the second stage of the heuristic algorithm. These lecturers are clustered to the same group if the conditions defined in the following are met:

- There is no identical lecturer in each group.
- Teaching slot type of the course

Besides that, for lecture sessions that are conducted on either Saturday or Sunday, each group is allowed to be assigned two courses for the same cohort but on the condition of no identical lecturer in the same group. Hence, each course that is conducted in one day can be assigned to two of the groups. To illustrate the said situation, Algorithm 1 explains the lecturer's grouping stage in the first stage of the proposed heuristic algorithm. Figure 3 shows the example of group assignment based on the teaching slot type of the courses.

C. Stage II. Group Allocation Stage

The group allocation stage is ready to allocate the group of lecturers into a set of academic weeks throughout the semester. In this stage, the timeslot for each course can be allocated, and the team-teaching slot for the lecturers can be assigned. This stage starts from weighting each of the groups with unavailability dates of the lecturers in a set of academic weeks, respectively. Next, a matching matrix is used to identify the availableness of the groups with n number of

academic weeks, *week* in this group allocation stage. This stage generates a $week \times g$ matrix with elements of 0 and 1. Element 0 represents that the group is available and able to conduct a lecture session without conflict of unavailability dates of the lecturers. On the other hand, element 1 represents that the group is unavailable on that particular week and should not be assigned into that week. Figure 4 shows the example of the matrix performed when comparing the unavailability dates of the lecturers with the list of lecturer groups.

Algorithm 1: Lecturer Grouping Stage

```

Find all critical lecturer,  $l$  ranking
Compute each cohort,  $c$  ranking
Arrange  $c$  ranking in descending order
FOR  $i = c$  to 1
    Arrange  $l$  ranking in descending order
    FOR group,  $g = 1$  to  $n$ 
        Check group with  $l$ , set found
        IF found = false then
            UPDATE the  $g$  list with  $l$ 
        ELSE
            CONTINUE to next  $g$ 
    ENDFOR
ENDFOR
ENDFOR

```

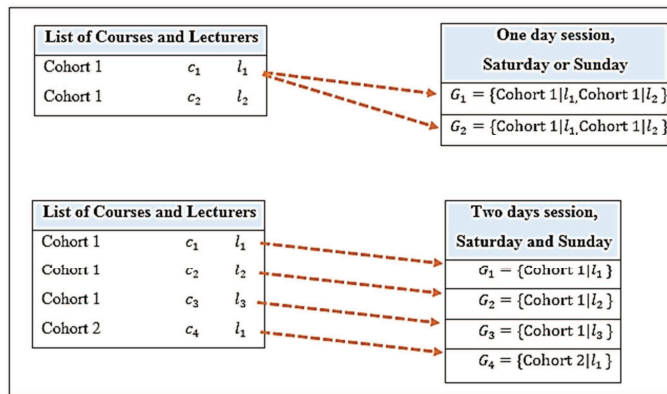


Fig. 3 Group assignment based on teaching slot types.

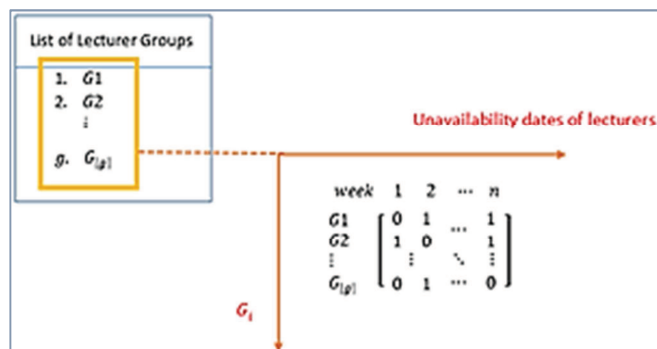


Fig. 4 Generating a matrix in Group Allocation Stage.

Each group is then weighted in a total number of unavailability dates of the lecturers and rearranged in weight size descending order. Within a set of academic weeks throughout the semester, the groups are assigned into the

weeks by round-robin order. By this step, the group allocation stage is ready to generate the timeslot allocation with the matrix from the unavailability dates of the lecturers. It starts with the group with the most unavailability dates of the lecturers. Then, the algorithm goes through the round-robin order with the matrix to allocate the groups.

The criteria to assign the timeslot to a lecturer group are listed as follow:

- The group allocation is in a round-robin order.
- The available timeslot can be assigned to anyone of the lecturer groups.
- The unavailable timeslot should not be assigned to the lecturer groups, which have unavailability dates on it.
- The available timeslot can only be assigned to one of the lecturer groups.
- Due to the round-robin order, the lecturer group with less timeslot allocation needs to be considered first before assigning another lecturer group.

Algorithm 2 explains the steps in the group allocation stage on a set of academic weeks for a semester.

Algorithm 2: Groups Allocation Stage

```

 $lct$  = list of lecturers in a group,  $g$ 
FOR  $week = 1$  to 14
    FOR  $g = 1$  to  $n$ 
        Compare unavailability date of  $lct$  with  $week$ 
        IF found = true then
            UPDATE unavailable date to  $g$  respectively to  $week$ 
        ENDFOR
    ENDFOR
    Compute total unavailable date in each  $g$ 
    Arrange  $g$  total unavailable date in descending order
    FOR  $week = 1$  to 14
        SET  $g = week \% n$ 
        Compare unavailable date to  $week$ , set found
        IF found = false then
            UPDATE  $week$  with the  $g$ 
        ELSE
            CONTINUE to next  $g$ 
    ENDFOR

```

After the group allocation stage, the stage proceeds with team teaching slot assignment process for all the courses and the lecturers. From this process, one day or two consecutive days of lecture sessions are assigned for each course based on their teaching slot types. Hence, a feasible timetable is generated at the end of the two-stage heuristic algorithm.

III. RESULTS AND DISCUSSION

In this section, computational analysis on timetable solutions generated by the simulator is carried out. The analysis includes a comparison between the timetable solutions produced by the two-stage heuristic algorithm and current timetabling practices. The simulator is coded with PHP web-based programming language and MYSQL database. The specification of the computer is a processor with i5-3210m CPU @ 2.50GHz and 6GB RAM in 64-bit Windows 7.

Two real semester datasets were collected from FCSHD for validation. The real timetables compared with the timetable generated by the simulator. The size of the problem for dataset 2017/2018-1 is relatively smaller as compared to others. Starting from 2017/2018-1, the MSc HRD courses are offered to off-UNIMAS campuses such as Kuala Lumpur and Sibul. Therefore, a new cohort would begin their study at these new off-UNIMAS campuses. Hence, the number of courses and lecturers to conduct the lecture sessions are increased in the following semester 2017/2018-2.

Due to the traveling budget consideration, course lectures at off-UNIMAS campuses are conducted for two consecutive days per weekend or continuous 14 hours over a weekend. These datasets have all lecturers that are available throughout the semesters. Table 3 shows the general information of each dataset.

TABLE III
DATASET GENERAL INFORMATION

Datasets	2017/2018-1			2017/2018-2		
	KL	Sibu	UNIMAS	KL	Sibu	UNIMAS
Number of Lecturers	17			20		
Number of Cohorts	1	1	4	2	2	4
Number of Courses	3	3	8	4	3	11
Number of Locations	3			3		
Lecture Hours Per Sessions	6 (UNIMAS), 14 (KL, Sibu)					

All constraints defined in this paper are fulfilled, and no clash solution is generated from the simulator. Table 4 and Table 5 show the comparison of the maximum week break between the real solution and two-stage heuristic for dataset 2017/2018-1 and 2017/2018-2. The ideal maximum week break between lecture sessions in UNIMAS is one to two weeks; meanwhile, in off-UNIMAS campuses, the breaks are three to four weeks. This is to ensure the maximum teaching effectiveness of the courses in all teaching locations.

TABLE IV
MAXIMUM WEEK BREAK COMPARISON FOR DATASET 2017/2018-1

Location	Max week break between lecture sessions	Current practices	Ideal ratio	Two-stage	Ideal ratio
UNIMAS	1	0	25%	0	37.5%
	2	2		3	
	3	6		5	
Off-UNIMAS	2	1	50%	0	100%
	3	2		0	
	4	1		6	
Total ideal ratio			35.7%	+28.5	64.2%

Legend: Ideal max week break

TABLE V
MAXIMUM WEEK BREAK COMPARISON FOR DATASET 2017/2018-2

Location	Max week break between lecture sessions	Current practices	Ideal ratio	Two-stage	Ideal ratio
UNIMAS	1	3	100%	4	100%
	2	8		7	
	3	0		0	
Off-UNIMAS	2	0	100%	0	100%
	3	3		7	
	4	4		0	
Total ideal ratio			100%	0	100%

Legend: Ideal max week break

The two-stage heuristic algorithm allocates a better ratio of ideal distribution on lecture sessions in dataset 2017/2018-1 and dataset 2017/2018-2. By comparing the different percentages in both solutions, it can be observed that the two-stage heuristic algorithm offers a better distribution allocation on lecture sessions. Hence, there is an increment of 28.5% in the dataset 2017/2018-1, as shown in Table 4. Some courses in UNIMAS have three maximum week break is due to the addition of mid-semester week between the lecture sessions.

In Table 5, the comparison between the real solution and two-stage heuristic algorithm shows that the method of distribution is very close to the real solution. However, the solution from the two-stage heuristic algorithm achieves a higher number of courses in smaller "max week" in dataset 2017/2018-2. The timetable solution generated by the simulator shows that the lower the maximum week break, the better the quality of the timetable. From the result for dataset 2017/2018-2, the ideal gap has increased to 9% for on-UNIMAS campus and 44.4% for off-UNIMAS campuses.

Figure 5 and Figure 6 show the comparison of the average week break between the real solution and two-stage heuristic for dataset 2017/2018-1 and 2017/2018-2. From the analysis of the average week break, the two-stage heuristic algorithms are subject to distribute the lecture duty for both datasets evenly. From the analysis on the distribution of lecture duty, the courses at UNIMAS campus and off-UNIMAS campuses should have a consistent average week break among each other.

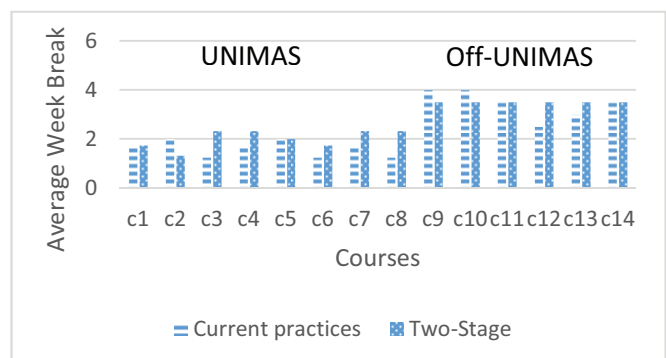


Fig. 5 Graph on the average week break on dataset 2017/2018-1

In Figure 5, the timetable solution that is generated by two-stage has a higher consistency of 2.33 average week break for the courses at UNIMAS campus than current practices. Meanwhile, the two-stage produces a consistent of the 3.5-week break for all the courses in off-UNIMAS campuses.

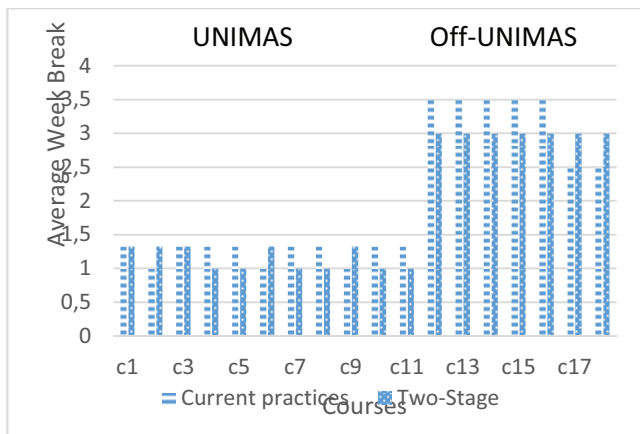


Fig. 6 Graph on the average week break on dataset 2017/2018-2.

In Figure 6, a comparison between the real solution and two-stage shows that the average week break is very close to a real solution, which both solutions have a consistent range of average week break between 1 to 1.33. Besides, a constant of a 3-week break for all the courses in off-UNIMAS campuses. It can be observed that the two-stage heuristic algorithm able to offer a better distribution of lecture duty with a consistent average week break for the courses. The current practices may take up to two weeks to produce a quality timetable solution. Meanwhile, the computational time taken by the proposed two-stage heuristic algorithm is less than a minute to generate a timetable solution. Depending on the problem size and the number of unavailability dates of lecturers for the requirement of timeslot allocation, the computational time varies for the timetabling problems from different semesters. Generally, the lecturer grouping stage takes a longer time than the timeslot allocation stage. It is time-consuming when the lecturer grouping stage loops through several possible groups and uses priority rules to allocate the “busy” lecturers in groups first.

IV. CONCLUSION

In this paper, a two-stage heuristic algorithm with the lecturer grouping stage and group allocation stage is proposed for multi-location lecture timetabling problems in FCSHD, UNIMAS. This paper concludes that the heuristic algorithm can minimize the clashes of lecturers when conducting the courses at different teaching locations. Besides that, the analyzed results display an even distribution in week break for all the courses, and it is also time-efficient to generate a feasible timetable. Besides, this algorithm can fulfill all operational constraints compared to current practice. This paper has proven that this heuristic algorithm in lecturer timetabling can help, indeed that eliminate tedious processes from current practice and takes a shorter time to generate a feasible and quality timetable solution.

ACKNOWLEDGMENT

We would like to thank Mr. Heng Chin Siong for permission to use and revise the timetable solutions on the dataset provided by the Faculty of Cognitive Sciences and Human Development (FSCHD), UNIMAS.

REFERENCES

- [1] A. Chowdhary, P. Kakde, S. Dhoke, S. Ingle, R. Rushiya, and D. Gawande, “Timetable generation system,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 2, 2014.
- [2] A. Dammak, A. Elloumi, H. Kamoun, and J. A. Ferland, “Course Timetabling at a Tunisian University: a case study,” *J. Syst. Sci. Syst. Eng.*, vol. 17, no. 3, p. 334, 2008.
- [3] M. Lindahl, T. J. R. Stidsen, and M. Sørensen, “Strategic, Tactical and Operational University Timetabling,” 2017.
- [4] M. W. S. Almeida, J. P. S. Medeiros, and P. R. Oliveira, “Solving the Academic Timetable Problem Thinking on Student Needs,” in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 673–676.
- [5] N. Pillay, “A review of hyper-heuristics for educational timetabling,” *Ann. Oper. Res.*, vol. 239, no. 1, pp. 3–38, 2016.
- [6] M. V. Padmini and K. Athre, “Efficient design of university timetable,” in *2010 IEEE International Conference on Electro/Information Technology*, 2010, pp. 1–5.
- [7] J. Pandey and A. K. Sharma, “Survey on university timetabling problem,” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 160–164.
- [8] E. Burke and W. Erben, *Practice and Theory of Automated Timetabling III: Third International Conference, PATAT 2000 Konstanz, Germany, August 16-18, 2000 Selected Papers*, vol. 2079. Springer, 2003.
- [9] A. E. Phillips, C. G. Walker, M. Ehr Gott, and D. M. Ryan, “Integer programming for minimal perturbation problems in university course timetabling,” *Ann. Oper. Res.*, vol. 252, no. 2, pp. 283–304, 2017.
- [10] G. Woumans, L. De Boeck, J. Beliën, and S. Creemers, “A column generation approach for solving the examination-timetabling problem,” *Eur. J. Oper. Res.*, vol. 253, no. 1, pp. 178–194, 2016.
- [11] J. A. Soria-Alcaraz, E. Özcan, J. Swan, G. Kendall, and M. Carpio, “Iterated local search using an add and delete hyper-heuristic for university course timetabling,” *Appl. Soft Comput.*, vol. 40, pp. 581–593, 2016.
- [12] T. A. Nugraha, K. T. Putra, and N. Hayati, “University Course Timetabling with Genetic Algorithm: A Case Study,” *J. Electr. Technol. UMY*, vol. 1, no. 2, pp. 100–105, 2017.
- [13] E. A. Abdelhalim and G. A. El Khayat, “A utilization-based genetic algorithm for solving the university timetabling problem (uga),” *Alexandria Eng. J.*, vol. 55, no. 2, pp. 1395–1409, 2016.
- [14] R. A. O. Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, “Practices in timetabling in higher education institutions: a systematic review,” *Ann. Oper. Res.*, vol. 275, no. 1, pp. 145–160, 2019.
- [15] M. Davoudzadeh, R. Rafeh, and R. Rashidi, “A linear solution for the university timetabling problem,” in *2009 Second International Conference on Computer and Electrical Engineering*, 2009, vol. 2, pp. 54–57.
- [16] T. Ferdoushi, P. K. Das, and M. A. H. Akhand, “Highly constrained university course scheduling using modified hybrid particle swarm optimization,” in *2013 International Conference on Electrical Information and Communication Technology (EICT)*, 2014, pp. 1–5.
- [17] S. Parera, H. T. Sukmana, and L. K. Wardhani, “Application of genetic algorithm for class scheduling (Case study: Faculty of science and technology UIN Jakarta),” in *2016 4th International Conference on Cyber and IT Service Management*, 2016, pp. 1–5.
- [18] R. Lewis, “A survey of metaheuristic-based techniques for university timetabling problems,” *OR Spectr.*, vol. 30, no. 1, pp. 167–190, 2008.
- [19] K. Schimmelpfeng and S. Helber, “Application of a real-world university-course timetabling model solved by integer programming,” *Or Spectr.*, vol. 29, no. 4, pp. 783–803, 2007.
- [20] J. Wahid and N. M. Hussin, “Combination of graph heuristics in producing initial solution of curriculum based course timetabling problem,” in *AIP Conference Proceedings*, 2016, vol. 1761, no. 1, p. 20104.