

Sequential Integer Programming for Solving Curriculum-Based University Course Timetabling Problem

Mansour Hassani Abdalla, Joe Henry Obit, Rayner Alfred, Kuan Yik Junn

Abstract: This research propose two stages sequential integer programming (IP) approach for solving curriculum-based university course timetabling problems (CB-UTT) in University Malaysia Sabah, Labuan international campus (UMSLIC). Like other timetabling problems, CB-UTT in UMSLIC has its own rules and features. The problem involves several hard constraints which need to be fully satisfied and soft constraints which satisfaction are very highly desirable. In this research mathematical formulation and two stages sequential IP search methodology based on UMSLIC is proposed. The IP search methodology is tested over two real-world instances, semester 1, session 2016/2017 and semester 2, session 2016/2017. The objective of this research is to generate high quality feasible CB-UTT which satisfies all peoples affected by the timetable. The results show that, the IP formulation proposed in this research is able to produce feasible solution in the first stage, and further improve by 10.99% and 8.92% respectively by solving soft constraints in the second stage without violating any hard constraints solved in the first stage. This IP approach is applicable towards the CB-UTT in UMSLIC.

Index Terms: Curriculum Course Timetabling, Integer Programming, Mathematical Formulation.

I. INTRODUCTION

University timetabling represents a difficult optimisation problem and finding a high quality timetable is a challenging task [1]. With the university constantly growing and teaching program becoming modular [1], finding high quality timetable is a complicated and time consuming task. This is essentially due to the fact that, there is larger number of events involved and various hard and soft constraints that need to be satisfied. However, for every educational institution, the objective is always to construct an effective and satisfactory weekly timetable [2]. In this work, we propose two stages sequential IP for solving CB-UTT for the real-world problem instance based on UMSLIC. In general, trying to model IP in one stage is harder as when the problem is decomposed into several stages [3], [4]. In the underlying study, hard and soft constraints are not solved directly all together at once, rather we divided the problem into two stages wherein the first stage,

hard constraints are solved with pure 0-1 IP (to generate initial feasible solution) and in the second stage further improvement by employing simple local search for minimising soft constraints. This approach is, in fact, resulting in an easy IP formulation as well as making the proposed search methodology capable to solve large instances [5]. This paper is organized as follows. Section 2 presents problem description, section 3 presents background study, section 4 proposes curriculum based course timetabling problem integer program model, section 5 presents constraints for the proposed IP model, section 6 present this work IP search methodology, Section 7 presents numerical experiment on the proposed IP formulation for a fully defined timetabling problem of UMSLIC. The paper ends with recommendation and conclusions on section 8.

II. PROBLEM DESCRIPTION

The department of academic service division is always having the problem of producing course and examination timetable in every semester. The challenging task is to assigning the courses which offered by the institution for that semester into a given period and suitable lecture room such that either the lecturer, curriculum or lecture room will not be used more than once [6]. In addition, in every assignment, the capacity of the classroom should be taken into consideration by making sure that there is no room capacity is exceeded. In general, the timetabling problem is to fix the sequence of meetings between lecturers and students in prefixed periods of time (typically a week); satisfying a set of constraints of various types [3].

In this work, two stages sequential IP model that fulfill the requirements of zero hard constraints and minimum values for soft constraints is proposed. The model is applied in UMSLIC institution as summarized in table 1. The objective of this investigation is to find feasible solution with less soft constraints violations. The problem involves various hard and soft constraints. The hard constraints must be satisfied while soft constraint violations should be reduced to improve the quality timetable, indirectly satisfying the students or lecturers that affected by the changes in timetable [1]. Essentially, this study involves assigning courses offered into

Revised Manuscript Received on August 19, 2019.

Mansour Hassani Abdalla, Faculty of Computing and Informatic, Universiti Malaysia Sabah, F.T. Labuan, Malaysia.

Joe Henry Obit, Faculty of Computing and Informatics, Universiti Malaysia Sabah, F.T. Labuan, Malaysia.

Rayner Alfred, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia.

Kuan Yik Junn, Student of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia.

limited resources which consist of 35 timeslots with seven days a week and five timeslots in a day, following the schedule in UMSLIC. Some of the lecturer could teach more than one course in a particular semester and a lecture is consists of at least two hour a week. Besides, there are several categories of courses in this institution which divided into main courses (or compulsory) and elective courses which consist of center for promotion of knowledge and language learning (PPIB) and center for co-curriculum and student development (PKPP). This study takes into consideration of lecturer’s preference, optimizing the utilization of appropriate rooms and adjusting balance in the schedule for students and lecturers. Not only that, some of the requirements established by the institution such as main courses particularly faculty and program must not be having on weekends and have to be placed in the first or third timeslots in any weekdays while PKPP courses must be scheduled only on weekends and PPIB courses must be assign on second, fourth or fifth timeslot.

Therefore, this study focused on CB-UTT in UMSLIC. As a matter of fact, there are five variables identified in CB-UTT namely courses, curricula, periods, lecturers, and rooms. The main purpose is to assign all lecture of each course into available timeslot and room satisfying the hard and soft constraints based on UMSLIC study guidelines. This study works on implementation of sequential IP search methodology for solving course timetabling problem for this institution.

Table I. UMSLIC datasets summary

Type of Constraints	Semesters	
	Semester 1	Semester 2
	session 2016/2017	session 2016/2017
No. of students	2263	2224
No. of curriculum	65	49
No. of lecturers	108	92
No. of courses	134	117

III. BACKGROUND STUDY

The timetabling problems, similar to other research field of complex combinatorial optimisation, have been studied thoroughly by many popular techniques of computer science fields and operational research [2]. In the literature, many approaches have been proposed to tackle university course timetabling mainly heuristics, in particular, meta-heuristics algorithms fields [2], [7], [8] and recently asynchronous cooperative distribute heuristics and low-level heuristics [9]. In this few years, IP based techniques entered the scene, and actually quite successfully [10]. In addition, it is worth noticing IP is not a new approach for solving university timetabling problem. The researches on this topic exist since 1969, [11]. However, the approach was temporarily abandoned, due to low capabilities of the computers used in that time [12]. Furthermore, [13] also pointed out that, the effort needed for modelling complex operational rules and the computational challenges that result from the real world problems have discouraged practitioners and shifting their interest to other methods. Nevertheless [14] pointed out that, due to the technological advancements in computer software and hardware, both IP and MIP (Mixed Integer programming)

formulations have restarted as an applicable way for various complex combinatorial optimization problems. Moreover, the latest technologies in information systems, the availability of trustworthy software and the capability to handle relatively large problems in short period are the main reasons for making this modelling approach promising for the solution in real problems [2]. Interestingly, there are more researchers showing massive interest in IP in tackling combinatorial optimisation. This is because IP approach does not only proves to be able to tackle large problem instance but also very fast in generating the feasible solution. As [14] pointed out that, the capability to tackle relatively large problems in shorter time has made integer programming much more widespread. Indeed, IP is promising for many innovations in models as well as methods. In this paper, we propose two stages sequential IP for solving CB-UTT as it has been proposed before; see e.g. [2], [4], [5], [15], [16]. IP method also can be modified with act independently as intelligent agent [17] to achieve their own objectives. A new model of agent environment system [18] is dynamic to perform in complex problems. The framework performs using Multi-Agent System (MAS) whereby a centralize agent act as coordinator for other IP agents that cooperate and lead those agent to more promising search area which will improve the pool of global solutions. The performance can be compared using sequential IP and cooperative IP as agent to determine whether stand-alone is more effective than working in cooperative environment [19]. The results from [19] demonstrate that both parallel and sequential IP are able to generate feasible solution as well as improving the quality of solutions tremendously. However, the cooperative search among IP agent managed to outperform independent IP agent in the instances. Similarly, the approaches [20], [21] provide relaxation for the algorithms in order to diversify the searches in more promising areas. Comparing the algorithms in other domains such as course timetabling [22], [23] examination timetabling [24], hostel space allocation [25], the formulation model of the constraints are quite similar, hence potentially IP can be apply in other domains as well. The performance of IP can be competitive compared with other metaheuristics [26] such as Great Deluge [27], Simulated Annealing [28] and Genetic Algorithm [29]. While there are various evolutionary algorithms have been introduced in the past decades, some modified the existing meta-heuristics and enhance it to be more efficient towards specific domain [30]. This can be related with the connectivity of the search space of the particular domain [31]. When the density of the conflict matrices of a particular constraints are high, hence the more difficult to be scheduled or improve. Meanwhile, some may be using hyper-heuristics [32] approach to increase the effectiveness in the algorithms which is less domains specific in terms of parameter tuning. The selection of heuristics depends on the hyper-heuristics strategy to pick the best moves, such as add or remove the search while iterated [33].



IV. PROBLEM FORMULATION

Modelling university timetabling has been a difficult task mainly because of the variation of teaching guidelines from different universities [16]. This challenge has brought different models into literatures depending on the problem instances. Moreover, different universities and institutions have their own structure and different set of hard and soft constraints [1]. However, to the large extent, these models share some features. In this work, CB-UTT based on the real-world problem instances is addressed. The modeling is based on UMSLIC teaching guidelines shown in table 2. A number of formulations for the university course timetabling problems have been suggested in the literature and one of the formulations was introduced by [4]-[6]. However, in this work, new constraints are introduced according to UMSLIC teaching guidelines.

Sets. I: The set of offered courses; L: The set of lectures; R: The set of rooms; K: The set of curricula; T: The set of time slots; D: The set of days; T (d): The set of time slots that belongs to day $d \in D$; I (k): The set of courses that belongs to curriculum $k \in K$;

Parameters. Cr: The size of room $r \in R$; Si: The number of students having lecture $i \in I$; F(i,t) is one if course $i \in I$ is available in timeslot $t \in T$; zero otherwise; $\chi (i_1,i_2)$ is one if course $i_1 \in I$ is different from $i_2 \in I$ ($i_1 \neq i_2$) and conflicting i.e. are taught by the same lecturer or are part of the same curriculum while zero otherwise; Y(t1,t2) is one if t1 and t2 belongs to the same day and are adjacent to each other while zero otherwise.

Decision Variables. D c, t, r is one if course $i \in I$ is located to room $r \in R$ in timeslot $t \in T$ while zero otherwise; K t, c the number of times there is an overflow of seats more than 100 over the required room capacity $r \in R$ in the timeslot $t \in T$; V(t, r) the number of students the capacity of the room $r \in R$ is exceeded in timeslot $t \in T$.

Constraints. Every institution (and even every department) has its own rules, features, costs, and fixations [1], [16]. UMSLIC also have own rules and constraints. Basically, all soft and hard constraints are provided by UMSLIC's academic department. Hard constraints are the rules which must be fulfilled for the timetable to be feasible whereas soft constraints are desirable and its satisfaction leads to the quality of timetable.

A. Hard Constraints

- 1) Lectures: Each course has a predefined number of lectures which have to be given and must be scheduled in specific time slots. The total number of lectures cannot be more than the time slots given.
- 2) Room conflict: There must not be more than one lecture having in the same room at a time.
- 3) Main and PPIB courses: All the major (main) courses can only be schedule on weekdays. Examples of main courses are program and faculty courses, including a number of PPIB courses. This constraint is based on the guideline prepared by the UMSLIC institution.
- 4) Center for co-curriculum and student development (PKPP) courses: All PKPP courses cannot be scheduled on weekdays. Most of the courses in PKPP are

scheduled on weekend by default. Usually PKPP courses are offered for new students who enrolled into the institution.

- 5) Room Capacity: The total number of students taking lecture of a course must be smaller than the size of a room having that lecture. Hence, the room capacity must be large enough in order to accommodate those students who are taking that lecture for the particular course.
- 6) Curriculum and lecturer conflicts: Similar with lectures constraint, lecturers who taught more than one course must be schedule in different time slots in order to avoid conflict between the courses.

Soft Constraints

- 1) Lecturer preferences: The allocation of lecture rooms and time slots need to give the priority for lecturers' preference. For example, the lecturers can select the time slots which they are available for lecture. Aside from that, lecturers who taking more than one course for lecture should have a gap in between if those lectures are scheduled close with each other.
- 2) Appropriate room size utilization: There should not be a large difference between the room capacity and the size of lecture. The reason is to utilize the room as much as possible and give other lectures more suitable rooms. For example, the lectures consist of very few students should not be scheduled in large lecture rooms.
- 3) Evenly timetable: Each student should not be having consecutive lecture in any given day.

Table II. Summary UMSLIC teaching scheduling guidelines

Hours	Time Groups	
	Monday - Friday	Saturday & Sunday
08.00AM-10.00AM	1	3
10.00AM-12.00PM	2	3
02.00PM-04.00PM	1	3
05.00PM-07.00PM	2	3
07.00PM-10.00PM	2	3

Constraints for the IP Model

Constraint (1) represents each course can at most be assigned one room for any given timeslot and only if the course is available for that time slot.

$$\sum_{r \in R} X_{i,t,r} = F_{i,t} \forall i \in I, t \in T \quad (1)$$

Constraint (2) represents each course can at most be assigned one room for any given timeslot and only if the course is available for that time slot.

$$\sum_{i \in I} X_{i,t,r} \leq 1 \forall t \in T, r \in R \quad (2)$$

Constraint (3) represents conflicting courses are not allowed to be scheduled in the same time slot.



$$\sum_{r \in R} X_{i,t,r} + \sum_{r' \in R} X_{i,t,r'} \leq 1 \quad \forall i, i2 \in I, t \in T: X(i1, i2) = 1 \quad (3)$$

Constraint (4) represents calculation of room capacity overflow. i.e. How much the room is miss used by assigning course with less (example course with 30 students in room with capacity 200 seat) student in a room where more than 100 seats are left blanks.

$$\sum_{i \in I} C_r \cdot X_{i,t,r} - S_i \geq 100 \quad \forall t \in T, r \in R \quad (4)$$

Constraint (5) represents the (binary) indicator function $A_{q,t}(x)$ determine if a curriculum in a time slot has a scheduled lecture i.e. there is no adjacent lecture from the same curriculum. It is 1 if $X_{i,t,r}=1$ for some $i \in I(q), r \in R$ and there does not exist $t' \in T, i' \in C(q), r' \in R$ such that $X_{c',t',r'}=1$ where t and t' belong to the same day and adjacent to each other. More formality this can be stated as:

$$A_{q,t}(x) = \begin{cases} 1 & \text{if } \sum_{i \in I(q), r \in R} X_{c,t,r} = 1 \wedge \sum_{i' \in I(q), r' \in R, t' \in T: Y(t,t')=1} X_{c',t',r'} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Objective Function

The quality of the generated timetable is measured by the value of the objective function. The objective is the sum of all the objectives with their penalties and basically for the sake of completeness the entire mathematical formulation is given in the model expressed as follows:

$$f(i) = \sum_q 10 \cdot A_{q,t} + \sum_{t \in T, r \in R} 10 \cdot V_{r,t} + \sum_{i \in I, r \in R} 10 \cdot K_{t,i} \quad (6)$$

V. INTEGER PROGRAMMING SEARCH METHODOLOGY

The entire class of problems is referred as scheduling, routing, and sequencing. The proposed approach using sequential IP is developed to solve the problem in two different stages; the first stage focus on tackling the hard constraints (generation of feasible solution). This is essentially performed to satisfy the hard constraints as described this is yes or no scenario of IP (pure 0-1 IP) where the course is assigned only if all the criteria are satisfied else the course is not scheduled. In second stage, it handle the soft constraints while maintaining the solution feasibility (i.e. do not violate hard constraints). Whenever the algorithms try to minimize the soft constraints, the cost function of the hard constraints must be taken care or else this may lead to infeasibility of the timetable. The objective is to produce conflict free timetable solution as well as satisfying those who are affected by it [7], [9].

First Stage Algorithm

In the first stage, the IP formulation will handle the hard constraints and improving it in the second stage by minimizing the violation of soft constraints as much as possible. Initially, the search begins with an empty timetable. At the early phase, the problem instances formulation are

stored into the hash set (list) by reading the instances from text file and all the information are now in stored in data structure, the IP selects a course, room, day and period at random. Next, the algorithm test for feasibility i.e. the size of lecture room manage to accommodate a lecture, time slot is already occupied by other lecture, and if the course is scheduled before. If all the conditions stated in section 4 are met, then the course will be assign into timetable and the room is set as occupied at that period in a given day. The course is marked as scheduled while the time slot is set as occupied by the course and at the same time unscheduled courses are decremented. The algorithm repeats until all courses are assigned. The following pseudocode demonstrates the first stage of algorithm:

```

01: while all courses are not scheduled do
02:   Select randomly d ∈ D, r ∈ R, p ∈ P, c ∈ C from the problem instance
03:   if c ∈ C feasibly can be scheduled i.e. no hard constraint violation then
04:     Insert into timetable and update the number of course scheduled
06:     iterate++;
07:   else
08:     Remove c ∈ C from timetable and update number of course scheduled
09:   end if
10: end while
    
```

Fig. 1.First stage algorithm description.

Second Stage Algorithm

Meanwhile, in second stage two neighborhoods are implemented. The move tries to improve the solution constructed in the first stage while at the same time maintaining the feasibility of the solution. Recent researches shows the most common neighborhood structures are mostly implemented due to effectiveness in exploring and exploiting search space [34] while some suggest that hybridize the local search [30] is much more promising. The moves are:

- **Move Lecture (ML):** Change the period and the room of one lecture: Move a course from its currently assigned time slot and room to different time slot and/or another lecture room. In our case, we are giving restriction for it to move to a new room which is unoccupied in the new time slot. This is because it is been proved in the literature by [35] to be more effective as it avoids disturbing the current constructed feasible solution.
- **Swap Lectures (SL):** Swap the period and the room of two lectures of distinct courses: Select two lectures of distinct courses and perform swap based on their periods and rooms.

When the moves have performed, the feasibility of the timetable must be maintained. The move is randomly selected and the process is repeated until the termination condition is met. In this stage, we assign 300 seconds for each run. The larger the size of problem instance, the longer the time is needed to schedule [2] and the higher chance of obtaining



better solution. The following pseudocode is presented as follow:

```

01: Stop condition: is set 300 seconds
02: Best solution = initial solution
03: while stop condition is not met do
04:     Select ML: new solution S*, OR Select SL: new solution S*
06:     if cost Function(S*) < cost Function (best solution)
07:         Best solution = S*
08:         iterate ++;
09:     else
10:         Best solution = Best solution
11: end if
12: end while
    
```

Fig. 2. Second stage algorithm description.

Sequential IP Framework

The developed complete sequential IP search methodology is able to read and show all the necessary data in text file format. It includes the automatic computational operations, which perform all cost computation according to IP formulation model to generate the final timetables. In the end, the solution is evaluated based on the criteria provided. Data from the institution is used to test the proposed search methodology. The complete implemented search methodology can be well explained in term of modules where there are five modules listed as follow:

- The data module: This module includes the preliminary data and all the files in the system i.e. courses with their associated information (course, lecturer, number of lecture, minimum working days and number of student), unavailability (course, day which the course cannot be scheduled and the period in a day where the course cannot be scheduled), relation (relation between course and curriculum), room (capacity, rooms with special needs and requirements) and basic (courses, rooms, days, periods per day, curricula, number of constraints, lecturers).
- The control system module: This module includes the core of the proposed methodology where the data is read by the system from the data module into other hash map data structure. Generally, this module is the implementation of complex data structure which stores the variable obtained from the data. Apart from that, the search process is initiated at the control system. Simple search is used in order to assign course into feasible timeslot and room.
- The optimization module: This module includes the IP algorithm coding. The initial solution will change by modifications in the initial data or through the parameters setting. Finally, the problem is solved with the new data.
- The report generating module: This module generates some of the important reports such as the final course timetables, timetable quality and the detailed improvements from the initial timetable which measured by the final cost function (breakdown of the cost based on the list of hard constraints and soft constraints).
- The evaluation module: This module examines the timetable quality produced. The evaluation of the solution is performed automatically. The guideline practice provides a set of preferences which comply with some general guidelines in order to create a timetable. Based on the importance of the preference, the framework select a high or medium value

correspond to objective coefficient. In general practice, the solution produced often satisfies a high proportion of the preferences.

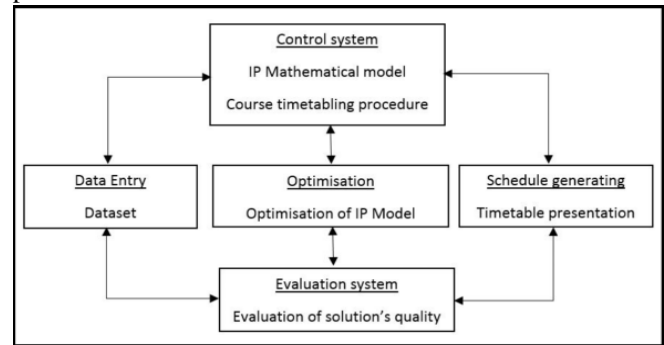


Fig. 3. Sequential IP Search methodology.

VI. EXPERIMENTS AND RESULTS

In order to evaluate the proposed IP formulation, two-semester problem instances of different difficulty were tackled. For each semester (session 2016/2017 and session 2016/2017) datasets, a fixed computation time was used to generate the initial solution and to farther to improve the solutions as the stopping condition. In average IP takes less than 5 seconds to generate the initial solution in the first stage, while 300 seconds is fixed to further improve the solution in the second stage. For each instance, 50 times experiment is run and average initial and final costs are recorded as shown in table 3. Generally, 50 runs are conducted to determine the consistency of the algorithms.

In all tests, the feasible solution is produced in less than five seconds. Results show that the particular problem with NP-complete in practice, it is not tough to solve. This is usually due to the fact that constraints are not very dense as the search methodology was applied on the small campus of UMSLIC. In addition, it is worth noticing that the decrement of the size of the problem by grouping subjects into groups (curriculums), the organization of the dataset, and UMSLIC's time grouping provide sustainable help in constructing a high quality timetabling solution. The percentage of improvement from initial to final cost function is 10.99% and 8.91 % for s1 2016/2017 and s2 2016/2017 respectively. The improvement in semester one is slightly higher than in semester 2, it can be assumed that semester two constraints are more compact as when compared to semester one. However, for each semester dataset, the approach was able to produce the feasible solution and further improve the solution significantly. It is important to note that:

- There are no conflicts among all the courses needed by the curriculum that appear in the timetable. Since all students in the early year should take all PPIB as well as curriculum courses and late year students should have the chance to select elective courses, there should be no conflict between courses which clearly reflected in the presented solution.

- Each course appears in different periods as required by the curriculum and assigning lectures in regular classrooms. Besides, the lecture period follows the requirements for each course. The search managed to schedule courses more than once in a week as specified i.e. courses with more than one sections.

- All lectures for the compulsory and elective courses are placed in morning sessions (08:00AM – 10:00AM) or afternoon session (02:00PM – 04:00PM). Meanwhile, the PPIB courses are scheduled during morning session (10:30AM – 12:30PM) or evening sessions (05:00PM – 07:00PM or 07:00PM – 10:00PM), following the guidelines for certain periods given by the academic department of UMSLIC.

Table III. Experimental results for two stages sequential Integer Programming search methodology

Experiments	Semesters	
	Semester 1 session 2016/2017	Semester 2 session 2016/2017
CELCAT/Manual UMSLIC	410.0	458.00
Average initial cost	368.04	377.30
Average final cost	326.94	343.69
Average improvement (%)	10.99	8.92
Lowest cost recorder in the second stage (lower bound)	318.00	3365.00
Highest cost recorded in the second stage (upper bound)	334.00	356.00

The existing manual solution of UMSLIC’s timetable was tested on the proposed frameworks in order to extract the value of soft constraints and identify if the schedule align with the UMSLIC’s teaching guidelines shown as explained in hard and soft constraints. It appears the timetable solution generated by CELCAT system does not follow the guidelines set by the academic department. According to academic department, in every semester they have to fix the arising issues such as trying to schedule courses base on the guideline. However, to schedule the courses exactly as the guidelines indicate manually by human being is a very hard task. This is well shown in the existing final generated timetable in UMSLIC as there quit number of teaching guideline (rules) which are not complied by the manual generated timetable in UMSLIC. Furthermore, in regards to soft constraints violation, we computed the objective values of the soft constraints for the existing UMSLIC timetable and we found that the penalty cost semester 1 2016/2017 and semester 2 2016/2017 are 410 and 458 respectively. These have also shown that, our proposed search methodology is not only able to schedule all the courses according to UMSLIC’s teaching guidelines but also minimised the soft constraint significantly as shown in Table 3.

Overall, this experiments evidence show that splinting IP into two stages by introducing simple local search, have been able to produce search methodology that is quite simple but

effective and able to solve real-life CB-UTT problem instances..

VII. CONCLUSION

In general, this paper focuses on investigating sequential IP for CB-UTT real-world problem in UMSLIC. The proposed search methodology managed to produce applicable solutions for UMSLIC two-stage sequential IP is developed. The proposed method focused on pure 0-1 IP and simple local search working together to solve real-world curriculum based course timetabling problem in semester 1 session 2016/2017 and semester 2 session 2016/2017. Similarly, the current study involve the problem tackle on hard constraints and soft constraints is done separately and this lead to producing much better solution. The constraints are solved in two stages wherein the first stage sequential IP which mainly focuses on solving hard constraints [7], the local search in the second stage is able to exploit the search space and minimize the soft constraints while preserving the feasibility of the generated solution.

A set of formal constraint mathematical models are presented based on IP approach that enables the identification of timetables which best meet the specific requirements of schedule construction in UMSLIC. Apart from that, the sequential IP framework is presented consisting of five different modules which related to one another. Experiments on sequential IP are conducted using two datasets from different semesters. Feasible solution is produced in average less than five seconds. By using the proposed sequential IP, an average improvement of ten percent from the first stage of algorithm is produced.

Future research will be aimed at enhancing the performance of the search methodology and incorporating into agent-based multi-agent systems, where there will be a parallel search and local inter-agents search and communication on trying to improve the global solution.

VIII. ACKNOWLEDGMENT

The authors would like to acknowledge Universiti Malaysia Sabah (UMS) for funding this research through Skim Geran Penyelidikan UMS (SGPUMS) SBK0362-2017.

REFERENCES

1. J.H. Obit, D. Ouelhadj, D. Landa-Silva, T.K. Vun, and R. Alfred, “Designing a multi-agent approach system for distributed course timetabling,” IEEE Hybrid Intelligent Systems (HIS), 10.1109/HIS, 2011, 6122088.
2. S. Daskalaki, T. Birbas, and E. Housos, “An integer programming formulation for a case study in university timetabling,” European Journal of Operational Research, vol. 153, no. 1, 2004, pp. 117-135.
3. J. Lee, S.P. Ma, L.F. Lai, N.L. Hsueh, and Y.Y. Fanjiang, “University timetabling through conceptual modeling,” International journal of intelligent systems, vol. 20, no. 11, 2005, pp. 1137-1160.
4. B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A.J. Parkes, and E.K. Burke, “Setting the research agenda in automated timetabling: The second international timetabling competition,” INFORMS Journal on Computing, vol. 22, no. 1, 2010, pp. 120-130.
5. G. Lach, and M.E. Lübbecke, “Curriculum based course timetabling: new solutions to Udine benchmark instances,” Annals of Operations Research, vol. 194, no. 1, 2012, pp. 255-272.



6. L. Di Gaspero, B. McCollum, and A. Schaerf, "The second international timetabling competition: Curriculum-based course timetabling," 2017. (track 3).
7. D. Landa-Silva, and J.H. Obit, "Comparing hybrid constructive heuristics for university course timetabling," In: VII ALIO/EURO Workshop on Applied Combinatorial Optimization, Porto, Portugal, May 4 - 6, 2011.
8. K.Y. Junn, J.H. Obit, and R. Alfred, "A Constraint Programming Approach to Solving University Course Timetabling Problem (UCTP)," *Advanced Science Letters*, vol. 23, no. 11, November 2017, pp. 11023-11026(4).
9. J.H. Obit, R. Alfred, and M.H. Abdalla, "A PSO Inspired Asynchronous Cooperative Distributed Hyper-Heuristic for Course Timetabling Problems," *Advanced Science Letters*, 2017, pp. 11016-11022(7).
10. M. Bergner, A. Caprara, A. Ceselli, F. Furini, M.E. Lübbecke, E. Malaguti, and E. Traversi, "Automatic Dantzig-Wolfe reformulation of mixed integer programs," *Mathematical Programming*, vol. 149, no. 1-2, 2015, pp. 391-424.
11. A.A.B. Pritsker, L.J. Waiters, and P.M. Wolfe, "Multiproject scheduling with limited resources: A zero-one programming approach," *Management science*, vol. 16, no. 1, 1969, pp. 93-108.
12. D. Sánchez-Partida, J.L. Flores, and E. Olivares-Benitez, "An integer linear programming model for a university timetabling problem considering time windows and consecutive periods," *Journal of Applied Operational Research*, vol. 6, 2014, pp. 158-173.
13. S. Daskalaki, and T. Birbas, "Efficient solutions for a university timetabling problem through integer programming," *European Journal of Operational Research*, vol. 160, no. 1, 2005, pp. 106-120.
14. E.L. Johnson, G.L. Nemhauser, and M.W. Savelsbergh, "Progress in linear programming-based algorithms for integer programming: An exposition," *Informatics journal on computing*, vol. 12, no. 1, 2000, pp. 2-23.
15. Y.Z. Ünal, and Ö. Uysal, "A new mixed integer programming model for curriculum balancing: Application to a Turkish university," *European Journal of Operational Research*, vol. 238, no. 1, 2014, pp. 339-347.
16. F. De Cesco, L. Di Gaspero, and A. Schaerf, "Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, and results. In E.K. Burke & M. Gendreau (Eds.)," *Proceedings of the 7th international conference on the practice and theory of automated timetabling, PATAT, Montréal, CA, 2010*.
17. M.H. Abdalla, J.H. Obit, R. Alfred, and J. Bolongkikit, "Agent based integer programming framework for solving real-life curriculum-based university course timetabling," In *Computational Science and Technology*, Springer, Singapore, 2019, pp. 67-76.
18. K.Y. Junn, J.H. Obit, R. Alfred, and J. Bolongkikit, "A Formal Model of Multi-agent System for University Course Timetabling Problems," In *Computational Science and Technology*, Springer, Singapore, 2019, pp. 215-225.
19. M. H. Abdalla, J.H. Obit, R. Alfred, and J. Bolongkikit, "Performance Comparison of Sequential and Cooperative Integer Programming Search Methodologies in Solving Curriculum-Based University Course Timetabling Problems (CB-UCT)," In *Computational Science and Technology*, Springer, Singapore, 2019, pp. 145-154.
20. J.H. Obit, D. Ouelhadji, D. Landa-Silva, and R. Alfred, "An evolutionary non-Linear great deluge approach for solving course timetabling problems," *International Journal of Computer Science Issues* 9 no. 4, pp. 1-13, 2012.
21. J.H. Obit, K.Y. Junn, and R. Alfred, "Performance comparison of linear and non-linear great deluge algorithms in solving university course timetabling problems," *Advanced Science Letters* 23, no. 11, pp. 11027-11030, 2017.
22. K.Y. Junn, J.H. Obit, and R. Alfred, "The Study of Genetic Algorithm Approach to Solving University Course Timetabling Problem," In *International Conference on Computational Science and Technology*, Springer, Singapore, 29 November 2017, pp. 454-463.
23. K.Y. Junn, J.H. Obit, and R. Alfred, "Comparison of simulated annealing and great deluge algorithms for university course timetabling problems (UCTP)," *Advanced Science Letters* 23, no. 11, pp. 11413-11417, 2017.
24. T.L. June, J.H. Obit, Y.B. Leau, and J. Bolongkikit, "Implementation of Constraint Programming and Simulated Annealing for Examination Timetabling Problem," In *Computational Science and Technology*, Springer, Singapore, 2019, pp. 175-184.
25. J.H. Obit, K.Y. Junn, R. Alfred, J. Bolongkikit, and O.Y. Sheng, "An Investigation towards Hostel Space Allocation Problem with Stochastic Algorithms," In *Computational Science and Technology*, Springer, Singapore, 2019, pp. 227-236.
26. J.H. Obit, K.Y. Junn, and R. Alfred, "A Performance Comparison of Metaheuristics Search for University Course Timetabling Problems," *Advanced Science Letters* 23, no. 11, pp. 11012-11015, 2017.
27. S. Abdullah, K. Shaker, B. McCollum, and P. McMullan, "Construction of course timetables based on great deluge and tabu search," In *Proceedings of MIC 2009: VIII Metaheuristic International Conference*, July 2009, pp. 13-16.
28. S. Ceschia, L. Di Gaspero, and A. Schaerf, "Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem," *Computers & Operations Research*, vol. 39, no. 7, 2012, pp.1615-1624.
29. S. Abdullah, and H. Turabieh, "Generating university course timetable using genetic algorithms and local search," In *2008 Third International Conference on Convergence and Hybrid Information Technology*, vol. 1, pp. 254-260, IEEE, November 2008.
30. R. Lewis, and B. Paechter, "New crossover operators for timetabling with evolutionary algorithms," In *5th International Conference on Recent Advances in Soft Computing*, Nottingham, UK, vol. 5, 2004, pp. 189-195.
31. R. Lewis, and J. Thompson, "Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem," *European Journal of Operational Research*, vol. 240, no. 3, 2015, pp.637-648.
32. E.K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems," *European Journal of Operational Research*, vol. 176, no. 1, 2007, pp.177-192.
33. 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," *Applied Soft Computing*, vol. 40, 2016, pp.581-593
34. Y. Nagata, "Random partial neighborhood search for the post-enrollment course timetabling problem," *Computers & Operations Research*, vol. 90, 2018, pp.84-96.
35. R. Bellio, L. Di Gaspero, and A. Schaerf, "Design and statistical analysis of a hybrid local search algorithm for course timetabling," *Journal of Scheduling*, vol. 15, no. 1, 2011, pp. 49-61. doi: 10.1007/s10951-011-0224-2.

AUTHORS PROFILE



Mr. Mansour Hassani Abdalla was a Master student at the Faculty of Computing and Informatics. His main research interest lies at the interface of Operational Research and Computer Science. In particular, the exploration and development of innovative Operational Research, Artificial Intelligence, and Distributed Artificial Intelligence models and methodologies for automatically producing high quality solutions to a wide range of real world combinatorial optimization and scheduling problems. Mr. Mansour obtained his master's in computer science from the Faculty of Computing and Informatics at Universiti Malaysia Sabah in 20018. and it was under the supervision of Dr. Joe Henry Obit and Dr. Rayner Alfred.



Dr. Joe Henry Obit is a Senior Lecturer at the Faculty of Computing and Informatics, Universiti Malaysia Sabah. His main research interest lies at the interface of Operational Research and Computer Science. In particular, the exploration and development of innovative Operational Research, Artificial Intelligence, and Distributed Artificial Intelligence models and methodologies for automatically producing high quality solutions to a wide range of real world combinatorial optimization and scheduling problems. Dr. Joe Obtained his PhD in Computer Science from the School of Computer Science at the University of Nottingham. His PhD thesis is developing a Novel Meta-heuristic, Hyper-heuristic and Cooperative Search.





Associate Professor Dr. Rayner Alfred is an Associate Professor in Software Engineering for the Faculty of Computing and Informatics at Universiti Malaysia Sabah. His main research interest lies at the Machine Learning in Knowledge Discovery. Dr. Rayner Alfred obtained his BSc in Computer Science at Polytechnic University of Brooklyn, New York, United States of America in 1994, an MSc in Computer Science from Western Michigan University, Michigan, United States of America in 1997 and a PhD in Computer Science from the School of Computer Science at the University of York, UK in 2008.



Mr. Kuan Yik Junn is A PhD student at the Faculty of Computing and Informatics under the supervision of Dr. Joe Henry and Dr. Rayner in Universiti Malaysia Sabah. His research focused on cooperative search in combinatorial optimization problems using Multi-agent framework.