

Research Article

A Heuristic Room Matching Algorithm in Generating Enhanced Initial Seed for the University Course Timetabling Problem

¹Teoh Chong Keat, ¹Habibollah Haron, ²Antoni Wibowo and ¹Mohd Salihin Ngadiman

¹Department of Computer Science, Faculty of Computing (FC), Universiti Teknologi Malaysia, Malaysia

²Department of Decision Sciences, School of Quantitative Sciences, UUM College of Arts and Sciences, Universiti Utara Malaysia, Malaysia

Abstract: The University Course Timetabling Problem (UCTP) such as the curriculum-based course timetabling problem is both an NP-hard and NP-complete scheduling problem. The nature of the problem concerns with the assignment of lecturers-courses to available teaching space in an academic institution. The Curriculum-Based University Course Timetabling Problem (CB-UCTP) has a high conflict-density and searching for an improved solution is not trivial. In this study, the authors propose a heuristic room matching algorithm which improves the seed of the CB-UCTP. The objective is to provide a reasonable search point to carry out any improvement phase and the results obtained indicate that the matching algorithm is able to provide very promising results as the fitness score of the solution is significantly enhanced in a very short period of time.

Keywords: Curriculum-based course timetabling problem, initial seed, room matching algorithm, university course timetabling problem

INTRODUCTION

The University Course Timetabling Problem (UCTP) is an NP-hard and NP-complete problem where there is no known polynomial time algorithm which guarantees in identifying the best solution (Bardadym, 1996; Ismayilova *et al.*, 2007). Inherently, the objective of the UCTP is to assign a set of resources which often comprises entities such as lecturers, courses and rooms to available timeslots whilst respecting the stipulated constraints of the institution. Generally, there are two types of constraints to adhere by which are hard constraints and soft constraints. The former constraints are vital and mandatory constraints in which they cannot be violated under any circumstances at all, lest the timetable is rendered infeasible. On the other hand, the latter constraints, usually specified preferences such as the unavailability of lecturers are secondary constraints which can be violated, but preferably not since the satisfaction of these constraints enhance the quality of the timetable. Since the problem is typically large-sized in nature and due to its large amount of variables and numerous constraints, it is also sometimes referred to as a constraint satisfaction problem (Teoh *et al.*, 2015).

The UCTP comprises several forms such as the Curriculum-Based University Course Timetabling Problem (CB-UCTP) and post-enrolment problem. In the literature, the UCTP has been reported to be solved

successfully through various metaheuristic algorithms. Metaheuristic algorithms are stochastic methods which are employed when either the search space becomes too vast or when there are no known algorithms which are capable of finding the optimal solutions (Rossi-Doria *et al.*, 2002). However, the parameters of the algorithms often require careful tuning prior to before they could be applied to solve the problem (Brownlee, 2011). Therefore one of the main concerns is how to produce an enhanced initial seed before employing the metaheuristic algorithms to further enhance the result. For example, a graph coloring heuristics has been employed to produce an enhanced initial seed in examination timetabling problem by (Burke *et al.*, 1995) and a two-point hybrid evolutionary algorithm is employed by (Rizam and Bakar, 2008) to produce an enhanced initial seeding in a similar examination timetabling problem. Additionally, the UCTP is also successfully solved by various metaheuristic algorithms such as Genetic Algorithm (Agustín-Blas *et al.*, 2009; Jain *et al.*, 2010; Kohshori and Abadeh, 2012; Suyanto, 2010), Ant Colony Optimization (Lutuksin and Pongcharoen, 2010; Thepphakorn *et al.*, 2009), Particle Swarm Optimization (Qarouni-Fard *et al.*, 2007; Shiau, 2011; Tassopoulos and Beligiannis, 2012), Artificial Bee Colony algorithm (Weng *et al.*, 2013), Simulated Annealing (Aycaan and Ayav, 2009; Frausto-Solis *et al.*, 2008; Zhang *et al.*, 2010) and Tabu Search (Alvarez-Valdes *et al.*, 2002; Lü and Hao, 2010) to name a few.

Corresponding Author: Teoh Chong Keat, Department of Computer Science, Faculty of Computing (FC), Universiti Teknologi Malaysia, Malaysia

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

However, most metaheuristic algorithms require a considerable initial seed in order to search for the optimal solution.

The works described in this study is inspired from the idea of employing graph coloring heuristics in the assignment of room to courses considering that once courses are scheduled to periods, the room assignment can be performed by simply solving a bipartite matching problem (Lü and Hao, 2010). In this study, the authors propose a heuristic room matching algorithm which seeks to improve the initial seed of the solutions.

MODEL DEFINITION

The CB-UCTP model is commonly adopted in many local universities as well as other international universities. The problem essentially deals with the scheduling of a set of lectures to various rooms and timeslots, where conflicts between courses are set according to the curriculums published by the University and not on the basis of enrolment data. The entities and their descriptions are adopted from the ITC-2007 technical report (Gaspero *et al.*, 2007) and are given as follows.

Days, timeslots and periods: A number of teaching days are given in the week (typically 5 or 6). Each day is split into a fixed number of timeslots, which is equal for all days. A period is a pair composed of a day and a timeslot. The total number of scheduling periods is the product of the days times the day timeslots.

Courses and lecturers: Each course consists of a fixed number of lectures to be scheduled in distinct periods; it is attended by given number of students and is taught by a lecturer. For each course there is a minimum number of days that the lectures of the course should be spread in, moreover there are some periods in which the course cannot be scheduled.

Rooms: Each room has a capacity, expressed in terms of number of available seats. All rooms are equally suitable for all courses (if large enough).

Curriculums: A curriculum is a group of courses such that any pair of courses in the group has students in common. Based on curriculums, we have the conflicts between courses and other soft constraints.

Based on the aforementioned entities, the solution to the problem is the complete assignment of a period (day and timeslot) and a room to every lecture for each course with respect to the stipulated constraints. The curriculum-based timetabling problem is composed of a set of n courses $C = \{c_1, \dots, c_n\}$ to be assigned to a set of m rooms $R = \{r_1, \dots, r_m\}$ and a set of p periods $T = \{t_1, \dots, t_p\}$. Every course c_i consists of l_i lectures to be

scheduled. A period consists of a day and a timeslot. A total of p periods are spread out over h daily timeslots and d days, i.e., $p = h \times d$. In addition, there is a set of w curricula $CR = \{Cr_1, \dots, Cr_w\}$ in each curriculum Cr_k is a group of courses that share common students. The nomenclature described in Table 1 is adopted from (Lü and Hao, 2010) whilst the properties of the dataset described in Table 2 is adopted from (Abdullah and Turabieh, 2012).

Constraints: The stipulation of constraints defines the search space of the solution and in this work; the matching algorithm only considers solutions which are already in its feasible form i.e., satisfied all of the hard constraints and only evaluates the penalty of the soft constraints. The hard and soft constraints of the problem are listed as follows.

H₁ (lectures): “All lectures of a course must be scheduled and assigned to distinct periods. A violation occurs if a lecture is not scheduled or two lectures within a course are scheduled in the same period.”

H₂ (conflicts): “All lectures of courses in the same curriculum or taught by the same lecturer must be scheduled in different periods. Two conflicting lectures in the same period represent one violation. Three conflicting lectures count as 3 violations: one for each pair.”

H₃ (room occupation): “Two lectures cannot be assigned to the same room at the same period. Two lectures in the same room at the same period represent one violation. Any extra lecture in the same period and room counts as one more violation.”

H₄ (availability): “If the teacher of the course is not available to teach that course at a given period, then no lecture of the course can be scheduled at that period. Each lecture scheduled in a period unavailable to that course is one violation.”

S₁ (room capacity): “The number of students that attend the course for each lecture must be less than or equal to the number of seats of the rooms hosting its lectures. Each student above the capacity counts as 1 violation.”

S₂ (minimum working days): “The lectures of each course must be spread over the given minimum number of days. Each day below the minimum, counts as 1 violation.”

S₃ (isolated lectures): “Lectures belonging to a curriculum should be adjacent to each other (i.e., in consecutive periods). For a given curriculum we

Table 1: Nomenclature in the curriculum-based university course timetabling problem

Symbols	Description
n_c	Total no of courses
m	Total no of rooms
d	Number of working days per week
h	Number of timeslots per working day
p	Total number of periods, $p = d \times h$
s	Total number of curricula
C	Set of courses, $C = \{c_1, \dots, c_n\}$, $ C = n$
R	set of rooms, $R = \{r_1, \dots, r_m\} = m$
T	set of periods, $T = \{t_1, \dots, t_p\}$, $ T = p$
CR	set of curricula, $CR = \{Cr_1, \dots, Cr_w\}$, $ CR = w$
CR_k	k^{th} curriculum including a set of courses
l_i	Number of lectures of course c_i
l	Total number of all lectures: $l = \sum_{i=1}^n l_i$
std_i	Number of students attending course c_i
tc_i	Lecturer instructing course c_i
md_i	Number of minimum working days of course c_i
cap_j	Capacity of room r_j
uav_{ij}	Whether course c_i is unavailable at period t_j , $uav_{ij} = 1$ if it is unavailable, $uav_{ij} = 0$ otherwise
con_{ij}	Whether course c_i and c_j are conflicting each other $con_{ij} = \begin{cases} 0, & \text{if } (tc_i \neq tc_j) \wedge (\forall Cr_q, c_i \notin Cr_q \wedge c_j \notin Cr_q) \\ 1, & \text{otherwise} \end{cases}$
X_{ij}	The course scheduled at room r_j and period t_i
$nr_i(X)$	Number of rooms occupied by course c_i for a candidate solution X : $nr_i(X) = \sum_{j=1}^m \sigma_{ij}(X)$ where,
$nd_i(X)$	$\sigma_{ij}(X) = \begin{cases} 1, & \text{if } \forall x_{k,j} \in X, x_{k,j} = c_i \\ 0, & \text{otherwise} \end{cases}$ Number of working days that course c_i takes place at in candidate solution X : $nd_i(X) = \sum_{j=1}^d \beta_{ij}(X)$ where:
$app_{k,i}(X)$	$\beta_{ij}(X) = \begin{cases} 1, & \text{if } \forall x_{u,v} \in X, x_{u,v} = c_i \wedge \left\lfloor \frac{u}{h} \right\rfloor = j \\ 0, & \text{otherwise} \end{cases}$ Whether Cr_k appears at t_i in X : $app_{k,i}(X) = \begin{cases} 1, & \text{if } \forall x_{i,j} \in X, \forall x_{i,j} = C_u \wedge C_u \in Cr_k \\ 0, & \text{otherwise} \end{cases}$

Table 2: Properties of the various problem instances featured in track 3 of the ITC-2007

Instance	Course	Total lectures	Rooms	Period/day	Days	Curricula	Min. and Max. lectures/day/curriculum
Comp01	30	160	6	6	5	14	2-5
Comp02	82	283	16	5	5	70	2-4
Comp03	72	251	16	5	5	68	2-4
Comp04	79	286	18	5	5	57	2-4
Comp05	54	152	9	6	6	139	2-4
Comp06	108	361	18	5	5	70	2-4
Comp07	131	434	20	5	5	77	2-4
Comp08	86	324	18	5	5	61	2-4
Comp09	76	279	18	5	5	75	2-4
Comp10	115	370	18	5	5	67	2-4
Comp11	30	162	5	9	5	13	2-6
Comp12	88	218	11	6	6	150	2-4
Comp13	82	308	19	5	5	66	2-3
Comp14	85	275	17	5	5	60	2-4
Comp15	72	251	16	5	5	68	2-4
Comp16	108	366	20	5	5	71	2-4
Comp17	99	339	17	5	5	70	2-4
Comp18	47	138	9	6	6	52	2-3
Comp19	74	277	16	5	5	66	2-4
Comp20	121	390	19	5	5	78	2-4
Comp21	94	327	18	5	5	78	2-4

Min.: Minimum; Max.: Maximum

account for a violation every time there is one lecture not adjacent to any other lecture within the same day.

Each isolated lecture in a curriculum counts as 1 violation.”

S₄ (room stability): “All lectures of a course should be delivered in the same room. Each distinct room used for the lectures counts as 1 violation.”

A list which details the various properties of the datasets is described in Table 2. The objective function of the problem therein is to minimize the violation of the hard constraints such that it produces a feasible timetable as given in Eq. (1). The mathematical notation for the constraints of the problem is adopted from Abdullah and Turabieh (2012) as follow:

$$\text{Min } F(X) = H_1 + H_2 + H_3 + H_4 + S_1 + S_2 + S_3 + S_4 \quad (1)$$

where,

$F(X)$ = Fitness value of candidate solution:

$$\begin{aligned} H_1: \text{Lectures: } & \forall c_k \in C, \\ & \sum_{i=1, \dots, p, j=1, \dots, m} \chi\{X_{i,j} = C_k\} = l_k \\ H_2: \text{Conflicts} & \\ & \forall x_{i,j}, x_{i,k} \in X, X_{i,j} = C_u, X_{i,k} = C_v \\ & \text{con}_{uv} = 0 \end{aligned}$$

where, χ is a truth indicator function which takes value of 1 if the given proposition is true and 0 otherwise.

H₃: Room occupation: $\forall x_{k,j} \in X, x_{k,j} = c_i$.

H₄: Availability: $\forall x_{i,j} \in C_k \in X_{uavk,i} = 0$.

S₁: Room capacity: $\forall x_{i,j} = c_k \in X$:

$$f_1(x_{i,j}) = \begin{cases} std_k - cap_j, & \text{if } std_k > cap_j \\ 0, & \text{otherwise} \end{cases}$$

S₂: Minimum working days: $\forall c_i = C$:

$$f_2(C_i) = \begin{cases} md_i - nd_i(X), & \text{if } nd_i(X) > md_i \\ 0, & \text{otherwise} \end{cases}$$

S₃: Isolated lectures (curriculum compactness): $\forall x_{i,j} = c_k \in X$:

$$f_3(x_{i,j}) = \sum_{Cr_q \in CR} \chi\{c +_k \in Cr_q\} \cdot iso_{q,i}(X)$$

where,

$$iso_{q,i}(X) = \begin{cases} 1, & \text{if } (i \bmod h = 1 \vee app_{q,i-1}(X) = 0) \wedge (i \bmod h = 0 \vee app_{q,i+1}(X) = 0) \\ 0, & \text{otherwise} \end{cases}$$

S₄: Room stability: $\forall c_i = C$:

$$f_4(c_i) = nr_i(x) - 1$$

Based on the mathematical model as stated above, the goal therein is to minimize the violation of the hard constraints such that it produces a feasible timetable. Upon feasibility, the algorithm then searches for a feasible solution, X^* such that $F(X^*) \leq F(X)$ for all X in the feasible search space region.

SOLUTION METHODOLOGY

Metaheuristic algorithms are employed when the size of the search space becomes too vast for exact methods and that no effective heuristic of identifying optimal solution is available (Rossi-Doria *et al.*, 2002). Due to their stochastic nature, a good initial seed is capable to aid the algorithm to converge to the global optimum within a shorter amount of time. Therefore one of the main concerns in the timetabling research community is how to

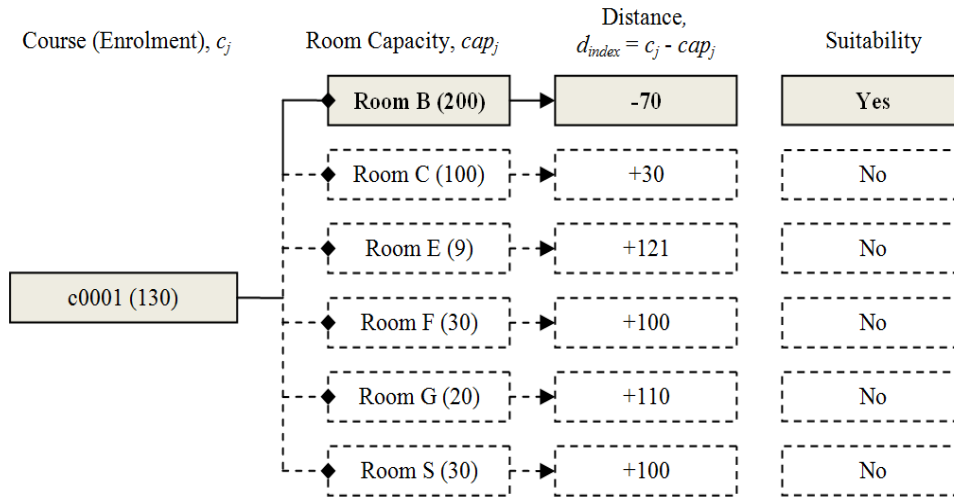


Fig. 1: Computation of the distance metric index, d_{index}

produce a high quality initial solution. For instance (Burke *et al.*, 1995) proposed a graph coloring heuristics to produce an initial seeding of higher quality as compared to randomly generated solutions while (Rizam and Bakar, 2008) has employed a Two-point Hybrid Evolutionary Algorithm (Tp_HEA) to capture the enhanced initial seed to be used in the sequential evaluation phase. Hence, this study proposes a heuristic room matching algorithm which is used to produce an enhanced initial seed for the curriculum-based course timetabling problem. Note that the initial feasible solutions used in this study are constructed using the constructive algorithm proposed by (Lü and Hao, 2010) and omits the explanation of the procedure as it is out of the scope of this study. Interested readers can find out more about constructive heuristics in generating a feasible timetable in the literature (Chiarandini *et al.*, 2006; Lü and Hao, 2010).

Heuristic room matching algorithm: In this study, the proposed heuristic room matching algorithm attempts to heuristically reassign the rooms of the solutions with the most ideal rooms in order to improve the initial seed of the candidate solution. Since is quite frequent in practice, for simplicity purpose, for rooms to be assigned at the end after the assignment of lectures to periods, the proposed algorithm is invoked after a feasible solution is found. In following Outlines the room matching algorithm pseudocode.

Phase 1: Construction of a feasible solution, X (omitted in this study): Output the feasible solution, $X_{feasible}$ to be utilized by the matching algorithm.

Phase 2: Heuristic room matching algorithm:
Construction of pre-processing files: Construct a Total_Timeslot matrix, TTS for each room.

Construct a Course_Room matrix, $crsroom$ which matches the usable rooms to each courses.

Invoke matching algorithm:

```

do while (j <= no. of lectures)
    select a usable room,  $r_{usable}$  which corresponds to
    period $_j$  from  $crsroom$  and assign to course,  $c_j$ 
    update the removal of  $r_{usable}$  from  $TTS$ 
end
    
```

Upon obtaining a feasible solution, the heuristic room matching algorithm is invoked to reassign the ideal rooms to all courses. The algorithm initializes with the construction of a Total_Timeslot matrix, TTS which lists every available timeslots for all rooms. Subsequently, a Course_Room matrix, $crsroom$ is constructed based on a distance metric index, d_{index} similar to that to the works of (Caldeira and Rosa, 1997) which measures the suitability of each room by subtracting the room capacity, cap_j from the student enrolment, std_j of a particular course. The assignment is only possible for all negative values of d_{index} and the suitability increases for $d_{index} \rightarrow 0$. In the computation of the d_{index} illustrated in Fig. 1, room B with a seating capacity of 200 can be assigned to course c0001 with the enrolment of 130 students with its d_{index} value -70. We only accept negative d_{index} as this denotes that the room capacity is bigger than the student enrolment. At the end of each iteration, the TTS updates with the removal of the assigned room to prevent any form of conflict.

The proposed algorithm is then tested on Track 3 of the Second International Timetabling Competition which consists of twenty-one problem instances which comprises the early, middle and late dataset of the curriculum-based course timetabling problem model.

RESULTS AND DISCUSSION

This section provides the results of the experiment. Table 3 and Fig. 2 depict the soft constraints improvement score of the initial feasible solution for

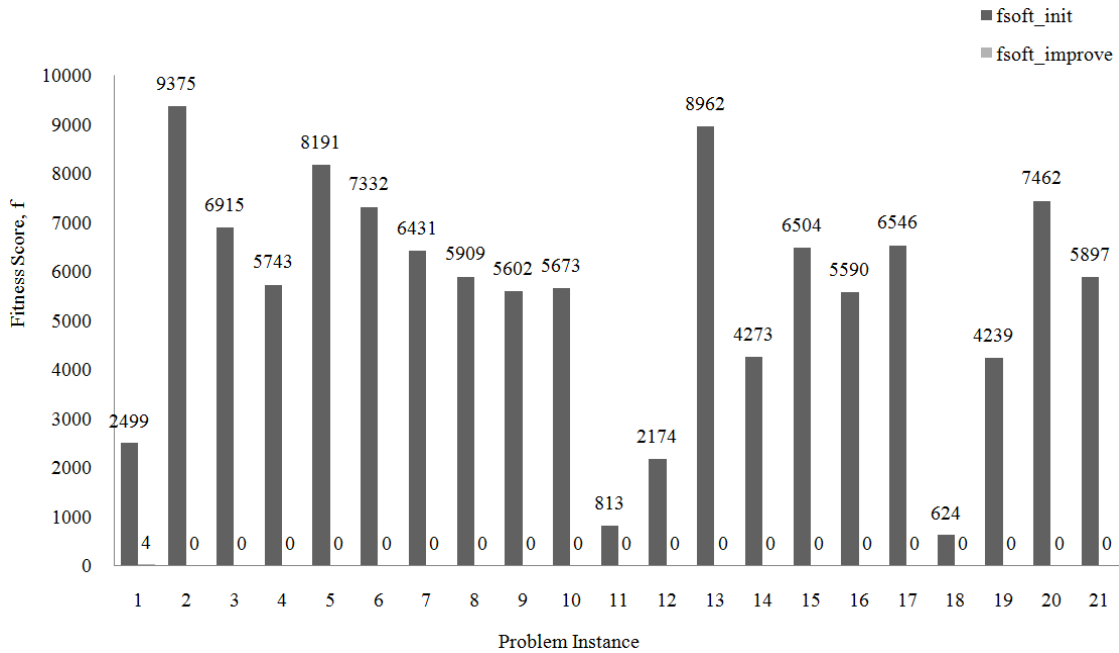


Fig. 2: Heuristic room matching algorithm fitness scores

Table 3: Heuristic room matching algorithm fitness scores

Instance	f _{soft_init}	f _{soft_improve}	%improve	t (sec)
1	2499	4	99.84	0.04
2	9375	0	100.00	0.05
3	6915	0	100.00	0.05
4	5743	0	100.00	0.05
5	8191	0	100.00	0.03
6	7332	0	100.00	0.07
7	6431	0	100.00	0.08
8	5909	0	100.00	0.06
9	5602	0	100.00	0.06
10	5673	0	100.00	0.07
11	813	0	100.00	0.03
12	2174	0	100.00	0.04
13	8962	0	100.00	0.06
14	4273	0	100.00	0.05
15	6504	0	100.00	0.05
16	5590	0	100.00	0.07
17	6546	0	100.00	0.06
18	624	0	100.00	0.02
19	4239	0	100.00	0.05
20	7462	0	100.00	0.07
21	5897	0	100.00	0.07

twenty-one problem instances of the curriculum-based course timetabling problem. The table and figure respectively describe the initial fitness score, f_{soft_init} , the improved fitness score, $f_{soft_improve}$, percentage of improvement, $\%improve$ and the computational time, t measured in seconds required by the matching algorithm to reassign the rooms to all courses.

Based on the results, for all of the tested instances, the proposed algorithm is able to enhance the quality of the initial seed significantly. In all of the tested problem instances, the matching algorithm exhibits an improvement in the fitness score, $f_{soft_improve}$ ranging from 99.04% to being able to successfully assign the

rooms to each lecture regardless of the search environment in less than one second, which is reflected in column t (sec) in Table 3. By reassigning a suitable room to each lecture, the matching algorithm has provided a considerable advantageous search point for the optimization of soft constraints as the optimization algorithm no longer needs to be concerned in the assignment of a suitable room and is able to focus in other regions of improvement.

CONCLUSION

The overall goal of the work presented in this study is to provide and generate an enhanced initial seed for the curriculum-based course timetabling problem which corresponds to Track 3 of the Second International Timetabling Competition. In order to acquire the enhanced initial seed for the problem, the authors have presented a heuristic room matching algorithm which is derived from the idea of graph coloring heuristics to solve a bipartite room matching problem. The proposed algorithm initializes with the construction of pre-processing files and then heuristically reassigns the most ideal rooms to each course until all courses have been reassigned with a suitable room. This approach is simple yet efficacious as it is able to render significant improvement to the candidate solutions in a significantly short amount of time and allows any future optimization to negate the aspect of room capacity. For future works, the authors intend to incorporate heuristics which take into account the other soft constraints collectively to provide a greater quality initial solution.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Universiti Teknologi Malaysia (UTM) and Research Management Center (RMC) for funding the research under the Research University Grant (RUG), number R.J130000.7828.4F497. Additionally, the authors would also like to thank the Ministry of Higher Education (MOHE) Malaysia for the my Brain scholarship and various anonymous referees, whose thoughtful and constructive criticisms contributed to the improvement of the study.

REFERENCES

- Abdullah, S. and H. Turabieh, 2012. On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems. *Inform. Sciences*, 191: 146-168.
- Agustín-Blas, L.E., S. Salcedo-Sanz, E.G. Ortiz-García, A. Portilla-Figueras and Á.M. Pérez-Bellido, 2009. A hybrid grouping genetic algorithm for assigning students to preferred laboratory groups. *Expert Syst. Appl.*, 36: 7234-7241.
- Alvarez-Valdes, R., E. Crespo and J.M. Tamarit, 2002. Design and implementation of a course scheduling system using tabu search. *Eur. J. Oper. Res.*, 137: 512-523.
- Aycan, E. and T. Ayav, 2009. Solving the course scheduling problem using simulated annealing. *Proceeding of IEEE International Advance Computing Conference (IACC, 2009)*, pp: 6-7.
- Bardadym, V.A., 1996. Computer-aided School and University Timetabling: A New Wave. In: Burke, V. and Ross (Eds.), *Practice and Theory of Automated Timetabling*. Lect. Notes Comput. Sc., 1153, Springer, Berlin, Heidelberg, pp: 22-45.
- Brownlee, J., 2011. *Clever Algorithms: Nature-inspired Programming Recipes*. Lulu Enterprises, pp: 401-402, ISBN 978-1-4467-8506-5.
- Burke, E.K., D.G. Elliman and R.F. Weare, 1995. A hybrid genetic algorithm for highly constrained timetabling problems. *Proceeding of the 6th International Conference on Genetic Algorithms*, pp: 605-610.
- Caldeira J.P. and A.C. Rosa, 1997. School timetabling using genetic search. *Proceeding of the 2nd International Conference on the Practice and Theory of Automated Timetabling*. Toronto, pp: 115-122.
- Chiarandini, M., M. Birattari, K. Socha and O. Rossi-Doria, 2006. An effective hybrid algorithm for university course timetabling. *J. Sched.*, 9(5): 403-432.
- Frausto-Solís, J., F. Alonso-pecina and J. Mora-vargas, 2008. An efficient simulated annealing algorithm for feasible solutions of course timetabling. In: Gelbukh, A. and E.F. Morales (Eds.), *MICAI 2008*. LNAI 5317, Springer-Verlag, Berlin, Heidelberg, pp: 675-685.
- Gaspero, L.D., B. McCollum and A. Schaerf, 2007. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (Track 3). *Proceeding of the 14th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion*, pp: 1-12.
- Ismayilova, N.A., M. Sağır and R.N. Gasimov, 2007. A multiobjective faculty-course-time slot assignment problem with preferences. *Math. Comput. Model.*, 46(7-8): 1017-1029.
- Jain, A., S. Jain and P.K. Chande, 2010. Formulation of genetic algorithm to generate good quality course timetable. *IJIMT 2010*, 1(3): 248-251.
- Kohshori, M. and M. Abadeh, 2012. Hybrid genetic algorithms for university course timetabling. *Int. J. Comput. Sci.*, 9(2): 446-455.
- Lü, Z. and J.K. Hao, 2010. Adaptive tabu search for course timetabling. *Eur. J. Oper. Res.*, 200(1): 235-244.
- Lutuksin, T. and P. Pongcharoen, 2010. Best-worst ant colony system parameter investigation by using experimental design and analysis for course timetabling problem. *Proceeding of 2nd International Conference on Computer and Network Technology*, pp: 467-471.
- Qarouni-Fard, D., A. Najafi-Ardabili and M.H. Moeinzadeh, 2007. Finding feasible timetables with particle swarm optimization. *Proceeding of 4th International Conference on Innovations in Information Technology (IIT '07)*, pp: 387-391.
- Rizam, M. and A. Bakar, 2008. Selecting quality initial random seed for metaheuristic approaches : A case of timetabling problem. *Int. J. Comput. Internet Manage.*, 16(1): 38-45.
- Rossi-Doria, O., M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L.M. Gambardella, J. Knowles *et al.*, 2002. A comparison of the performance of different metaheuristics on the timetabling problem. In: Burke, E. and P. De Causmaecker (Eds.), *PATAT 2002*. LNCS 2740, Springer-Verlag, Berlin, Heidelberg, pp: 329-351.
- Shiau, D.F., 2011. A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences. *Expert Syst. Appl.*, 38(1): 235-248.
- Suyanto, S., 2010. An informed genetic algorithm for university course and student timetabling problems. *Proceeding of the 10th International Conference on Artificial Intelligence and Soft Computing: Part II*. Berlin, Heidelberg, pp: 229-236.

- Tassopoulos, I.X. and G.N. Beligiannis, 2012. Solving effectively the school timetabling problem using particle swarm optimization. *Expert Syst. Appl.*, 39(5): 6029-6040.
- Teoh, C.K., A. Wibowo and M.S. Ngadiman, 2015. Review of state of the art for metaheuristic techniques in academic scheduling problems. *Artif. Intell. Rev.*, 44(1): 1-21.
- Thepphakorn, T., C. Hicks and P. Pongcharoen, 2009. An ant colony based timetabling tool. *Int. J. Prod. Econ.*, 149: 131-144.
- Weng, F.C., H. Asmuni and B. McCollum, 2013. Tackling university timetabling problems using hybridization of swarm based approach. *Proceeding of the 6th Multidisciplinary International Scheduling Conference: Theory and Applications*, Gent, Belgium, pp: 543-548.
- Zhang, D., Y. Liu, R. M'Hallah and S.C.H.H. Leung, 2010. A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *Eur. J. Oper. Res.*, 203(3): 550-558.