Received July 8, 2021, accepted July 22, 2021, date of publication July 27, 2021, date of current version August 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100613

A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities

MEI CHING CHEN[1], SAN NAH SZE[1], SAY LENG GOH[2], NASSER R. SABAR[3], AND GRAHAM KENDALL[4], (Senior Member, IEEE)

[1]Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak 94300, Malaysia
[2]Optimisation Research Group, Faculty of Computing and Informatics, Universiti Malaysia Sabah Kampus Antarabangsa Labuan, Labuan 87000, Malaysia
[3]Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC 3086, Australia
[4]School of Computer Science, University of Nottingham Malaysia Campus, Semenyih, Selangor 43500, Malaysia

Corresponding authors: San Nah Sze (snsze@unimas.my) and Say Leng Goh (slgoh@ums.edu.my)

This work was supported by Universiti Malaysia Sarawak (UNIMAS) Internal under Grant F08/RISE/2090/2021.

ABSTRACT The timetabling problem is common to academic institutions such as schools, colleges or universities. It is a very hard combinatorial optimisation problem which attracts the interest of many researchers. The university course timetabling problem (UCTTP) is difficult to address due to the size of the problem and several challenging hard and soft constraints. Over the years, various methodologies were proposed to solve UCTTP. The purpose of this survey paper is to provide the most recent scientific review of the methodologies applied to UCTTP. The paper unveils a classification of methodologies proposed in recent years based on chronology and datasets used. Perspectives, trends, challenges and opportunities in UCTTP are also presented. It is observed that meta-heuristic approaches are popular among researchers. This is followed closely by hybrid methodologies. Hyper-heuristic approaches are also able to produce effective results. Another observation is that the state-of-art methodologies in the scientific literature are not fully utilised in a real-world environment perhaps due to the limited flexibility of these methodologies.

INDEX TERMS Combinatorial optimisation problem, course timetabling problem, optimisation.

I. INTRODUCTION

Timetabling is defined as an optimization task of allocating a set of events (exams, courses, sporting events, surgeries) and resources (exam proctors, teachers, athletes, sport officials, nurses, medical doctors) to space (exam halls, classrooms, sport fields, operating theatres) and time [78]. It is a popular topic in operations research and is applied in a broad range of fields including education, transportation, hospitals, private enterprises, sports and many others [16]. The challenge of timetabling is common to academic institutions such as schools [71], [72], colleges or universities. It is a combinatorial optimisation problem which is of interest to many researchers. Addressing an optimisation problem involves searching for an optimal configuration of a given set of variables with the aim of achieving certain objectives [38]. This paper focuses on university course timetabling problem (UCTTP).

To date, there are several survey papers on UCTTP. Table 1 shows the scope of these papers. However, most of these papers only focus on presenting the methodologies that were applied to UCTTP. Also, no classification or discussion on the advantage and disadvantage of existing methodologies were provided. This paper aims to fill this gap and present a comprehensive survey on the UCTTP. The advantages and limitations of current studies are discussed. This paper will help researchers to understand the practical application of different methodologies applied to UCTTP. In addition, future research directions on UCTTP are also provided to promote further application of different methodologies.

This paper highlights the most recent approaches in addressing UCTTP. The mechanism of the approaches are briefly discussed and the achievement of the approaches are compared. The approaches are then categorised into OR based techniques, meta-heuristics (single-solution and population-based approaches), hyper-heuristic approaches, multi criteria/ objective and hybrid approaches. We discuss the advantages and disadvantages of each category. We present the methodologies in chronological order to show the trend in UCTTP. In addition, the methodologies are grouped according to benchmark datasets to identify not only the popular datasets but the state of the art methodologies

The associate editor coordinating the review of this manuscript and approving it for publication was Jerry Chun-Wei Lin.

TABLE 1. Summary of survey paper on timetabling problem.

Reference	Title	Scope of the paper
[26]	An introduction to timetabling	Educational timetabling models using graphs and networks.
[19]	Automated university timetabling: The state of the art	Approaches used in solving exam and course timetabling problem.
[64]	A survey of automated timetabling	Types of timetabling problem formulations with solution approaches.
[23]	Recent research directions in automated Timetabling	Recent approaches used in solving the timetabling problem explored by Automated Scheduling, Optimisation and Planning Research Group (ASAP) at the University of Nottingham.
[51]	A perspective on bridging the gap between theory and practice in university timetabling	New information that can help researchers to minimize the gap between theory in literature and real implementation at institutions.
[45]	A survey of meta-heuristic-based techniques for university timetabling problems	Meta-heuristic-based techniques as higher-level approaches that can be utilised to solve varieties of problem types.
[43]	A comprehensive study of educational timetabling-a survey	Emphasize on popular trends and accomplishments in student sectioning, university course timetabling, examination timetabling and high school timetabling.
[73]	Review of state of the art for meta-heuristic techniques in Academic Scheduling Problems	Meta-heuristic-based techniques and quality of solutions in terms of feasibility, optimality and computational costs.
[12]	A survey of approaches for university course timetabling problem	Classification of approaches into operational research (OR) based techniques, meta-heuristic, multi criteria and multi objective, intelligent novel and distributed multi agent systems.
[16]	An overview of curriculum-based course timetabling	Formulation of mathematical models, lower bounds, exact algorithms and heuristic algorithms proposed in solving curriculum-based course timetabling.
[58]	A review of hyper-heuristic for educational timetabling	Review on hyper-heuristic approaches in solving educational timetabling and research opportunities.
[11]	University course timetabling and the requirements: Survey in several universities in the east-coast of Malaysia	Information needed to generate mathematical model for university timetabling with the aim of closing the gap between theory in literature with real implementation at institution.
[57]	Practices in timetabling in higher education institutions: a systematic review	Identifying the similarities and differences between theory in literature with real implementation at higher education institutions.

for each dataset. Furthermore, We present case studies of real-world UCTTP. As far as we are aware, no previous survey of UCTTP covers this area. Constraints of different institutions are presented. Real world UCTTP instances are unique due to different policies set by the institutions.

This paper is organised as follows. Section II describes the UCTTP and the constraints involved. We discuss the approaches/methodologies in benchmark UCTTP in Section III. Section IV presents the approaches/methodologies in real-world UCTTP. The perspectives in UCTTP are provided in Section V. The trends in UCTTP are given in Section VI. We outline the limitations of the approaches/ methodologies in UCTTP in Section VII. Research opportunities in UCTTP are presented in Section VIII. Finally, conclusions are given in Section IX.

II. UNIVERSITY COURSE TIMETABLING PROBLEM

A. PROBLEM DEFINITION

The university course timetabling problem (UCTTP) is a variant of educational timetabling. Addressing UCTTP involves allocating a set of m courses, $C = \{c_1, \ldots, c_m\}$ to a set of n time-slots, $T = \{t_1, \ldots, t_n\}$ and a set of p venues, $V = \{v_1, \ldots, v_p\}$. Each university has its own unique timetabling problem, and therefore requirements, due to various reasons such as the policies set by the institution and the education system of the respective country and/ or region. Among the variants of UCTTP are the curriculum-based course timetabling problem (CB-CTTP) and the post-enrolment course timetabling problem (PE-CTTP) [3], [52]. UCTTP integrates several parameters such as courses offered each semester, lecturers assigned to teach the courses, number of students who registered for the courses and the locations where the lectures will be conducted [7]. A solution is a schedule that must fulfil all the hard constraints, but it is optional to satisfy soft constraints [2]. Aspects that need to be taken into consideration in generating a solution are computational speed, feasibility and quality. A feasible solution is a solution that satisfies all the hard constraints specified in the problem domain [2], [32]. For example, a student cannot attend two lectures at the same time while a lecturer cannot lecture more than one course simultaneously. The quality of the solution [22] is determined by the soft constraint violations. For example, students should not have only one lecture in a day and lecturers should not have to lecture after 5pm.

UCTTP is known to be NP-hard [7], [37], [68], [77], that is the problem cannot be solved exactly in polynomial time as the growth of the problem size and its complexity is exponential [12], [15]. Exact algorithms are guaranteed to provide optimal solutions but they are only applicable to small sized problems [64]. As an alternative, heuristic algorithms are often utilised to provide relatively good solutions in acceptable time [35].

B. PROBLEM CONSTRAINTS

The constraints involved in UCTTP are presented below. These constraints can be defined as hard or soft, depending on institution requirements.

- C1: Lectures taught by the same lecturer cannot be conducted at the same time.
- C2: Each venue can only be assigned to one lecture at one time.

Ideen für Szenarien

- C3: The rooms assigned to a certain lecture should be big enough to accommodate the students registered for the course.
- C4: All lectures should be scheduled in the timetable.
- C5: All the pre-assignments and blocked periods for classes must be taken into consideration.
- C6: A student can only attend one lecture at one time.
- C7: Lectures of each course are evenly spread in minimum working days.
- C8: Lectures for courses in the same set of curricula should be placed in the time-slot next to each other if scheduled in the same day.
- C9: No lecture can be allocated to lunch break time-slot.
- C10: The room features should match those required by the course.
- C11: Certain courses need to be scheduled in the correct order.
- C12: A student should attend more than one course in a day.
- C13: A student should attend less than three consecutive courses.
- C14: No course should be allocated to the last time-slot of the day.
- C15: Lectures for a course must be conducted in the same room.

III. APPROACHES/METHODOLOGIES IN BENCHMARK UCTTP

The methodologies utilised in UCTTP can be divided into six categories. The first category is operational research (OR) based techniques (graph colouring heuristics, integer/linear programming, mixed integer linear programming and constraint logic programming). The second category is single solution-based meta-heuristics (tabu search, variable neighbourhood search and simulated annealing). The third category is population-based meta-heuristics (genetic algorithms, ant colony optimisation and particle swarm optimisation). The fourth, fifth and sixth categories are hyper-heuristic, multi criteria/objective and hybrid approaches.

A. OPERATIONAL RESEARCH (OR) BASED TECHNIQUES

The graph colouring problem requires allocating minimal colours to vertices such that vertices connected by edges are allocated different colours. Timetabling and graph colouring are related such that events represent vertices, clashes between events correspond to edges and time slots denote colours [26]. All individual courses are referred as events. Due to the interconnection of graph colouring problem and UCTTP, earlier algorithms were derived from graph colouring heuristics [19], [45]. Lectures are assigned to periods (days and time) sequentially based on graph colouring heuristics for instance; largest degree, saturation degree, largest weighted degree and colour degree [23].

Conforming to the largest degree heuristic, events with the largest count of conflicts with other events should be assigned a time period first as it is difficult to find a valid time period for an event that has many clashes with other events. Based on largest weighted degree heuristic, events with the highest number of students are assigned to the time period first. In saturation degree heuristic, the next event to schedule is the one with the lowest remaining suitable time periods. In colour degree heuristic, priority is given to events with the highest number of conflicts with the scheduled events. Both the saturation degree and colour degree heuristics are calculated dynamically.

Reference [48] proposed a clique-based algorithm to generate feasible solutions for UCTTP. The clique refers to a set of courses that could be allocated in the same time-slot. Recombination and perturbation steps were taken to increase the size of the clique generated. The proposed algorithm was tested using hard benchmark datasets. The algorithm was comparable with other effective algorithms.

Reference [18] proposed a Graph Colouring (GC) approach to find feasible solutions for UCTTP. The proposed algorithm had two stages. In stage one, Least Saturation Degree First (LSDF) was used in finding feasible solutions. In stage two, the solution quality was improved using operators based on a column permutation. The algorithm was tested using Socha benchmark datasets. The algorithm managed to produce encouraging results.

Reference [47] proposed a Mixed Integer Linear Programming (MILP) approach to solve the CB-CTTP. Three bi-objective mixed-integer models were formulated. Problem instances from ITC-07 (Track 3) were used as testbeds. They found that the objectives (rooms, teaching periods and solution quality) affect one another and the relationships between these objectives are dependent on the problem instances.

Reference [14] proposed an Integer Programming (IP) relaxation to solve CB-CTTP. The model formulated was called pattern formulation, where a course was assigned to a set of periods on one day. The proposed model was tested using ITC-07 (Track 3) benchmark datasets. The proposed model managed to improve the lower bounds for three of the problem instances.

B. SINGLE SOLUTION-BASED META-HEURISTICS

Single solution-based meta-heuristics are defined as "a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies for developing heuristic optimisation algorithms" [69]. Single solution-based meta-heuristics are often known as local search algorithms. Local search algorithms start the search with single solution and then explore its neighbourhood areas to find a better one. Examples of local search algorithms are tabu search, simulated annealing, hill climbing and iterated local search. Reference [45] categorised single solution-based meta-heuristics into three types of optimisation algorithms which are one-stage, two-stage and the one that allows relaxation. One-stage optimisation algorithms satisfy both hard and soft constraints at the same time [45]. The best solution is determined by using a weighted sum function where each constraint is given a penalty value based on their importance. In two-stage optimisation algorithms, only hard constraints are considered in obtaining a feasible solution in the first stage. Meanwhile, only soft constraints are considered in getting a high quality solution in the second stage. For algorithms that allow relaxation, hard and soft constraint violations are addressed by relaxing some aspects of the problem instance. There are two types of relaxation. The first temporarily puts aside events that cannot be scheduled in a feasible solution. The second creates dummy or extra time slots to artificially accommodate the events to create a feasible solution [42].

1) TABU SEARCH

Tabu search (TS) uses a tabu list to avoid being stuck in a local optima. Whenever it is trapped in a local optima, the search continues with non-improving moves but solutions generated before will be rejected with the use of tabu list.

Reference [55] proposed a TS algorithm with the ability of changing the neighbourhood size called Random Partial Neighbourhood Search (RPNS) to solve the PE-CTTP. The proposed algorithm was tested using Socha, ITC-02 and ITC-07 (Track 2) benchmark datasets. The algorithm produced competitive results when compared to leading solvers.

2) SIMULATED ANNEALING

Simulated Annealing (SA) accepts non-improving moves using a probabilistic acceptance criteria. Its performance is dependent on the initial and end temperatures, cooling schedule and definition of neighbourhood structures.

Reference [44] proposed a time-dependent SA algorithm to solve the PE-CTTP. The algorithm had three distinguishable stages. Each stage had its defined time limit and the total must not exceed the full time limit. If one stage completed earlier than the specified time limit, the extra time could be utilised in the next stage. At each stage, constraints satisfied in the previous stages must not be violated. Stage 2 and stage 3 were implemented using SA. The proposed algorithm was tested using ITC-07 (Track 2) benchmark datasets.

Reference [24] proposed a SA algorithm to solve the PE-CTTP. Two neighbourhood structures were used moving one event and swapping two events. The proposed algorithm was tested using Socha, ITC-02, ITC-07 (Track 2) and hard benchmark datasets. A well engineered and finely tuned solver managed to outperform most of the algorithms in the scientific literature.

Reference [31] proposed Tabu Search with Sampling (TSSP) and Simulated Annealing with Reheating (SAR) to address the PE-CTTP. In stage one, TSSP was utilised to create feasible solutions. In stage two, SAR was used to improve the quality of the solutions. The algorithm was tested with Socha, ITC-02 and ITC-07 (Track 2) benchmark datasets. It managed to produce new best solutions for many instances.

Reference [33] proposed SA with Improved Reheating and Learning (SAIRL) in addressing the PE-CTTP. The method consisted of two stages. In the first stage, a feasible solution was generated, which was improved in the second stage. For the search to function effectively, a reinforcement learning-based methodology was proposed to obtain a suitable composition of neighbourhood structures. The proposed algorithm was tested using Socha, ITC-02 and ITC-07 (Track 2) datasets. The approach managed to generate six new best results.

Reference [34] proposed a two-phase hybrid local search algorithm to solve PE-CTTP. In the first phase, TSPP and Iterated Local Search (ILS) were used to generate a feasible solution. In second phase, SAR with two preliminary runs (SAR-2P) was used to improve the quality of the solution. Information gathered from the preliminary runs helped to improve the efficiency of SAR. The algorithm was tested with Hard, Socha, ITC-02 and ITC-07 (Track 2) benchmark datasets. It produced three new best results and seven new mean results.

3) ITERATED LOCAL SEARCH

Reference [68] proposed an Iterated Local Search (ILS) algorithm to address the UCTTP. It consisted of three stages which were initialisation, intensification and diversification. 60 instances from [46] were used. The proposed algorithm managed to find feasible solutions for 58 instances.

C. POPULATION-BASED META-HEURISTICS

Population-based meta-heuristics operate on a population of solutions and apply various operators and rules to evolve a new population of solutions in the neighbourhood areas of current ones. Examples of population-based meta-heuristics are genetic algorithms, ant colony optimisation and particle swarm optimisation.

1) GENETIC ALGORITHMS

Genetic Algorithm (GA) tackles optimisation problems using the concept of biological evolution. In each iteration, the algorithm selects solutions from the population using selective pressure to bias it towards choosing the best members as parents and these parent solutions are used to generate children solutions for the next generation [39]. This process is iterated until an optimal solution is generated or the time allowed has expired. In each iteration, the algorithm conforms to selection rules, crossover rules and mutation rules. Selection rules manage the selection of parents from the current population. Crossover rules are ways that parents are combined in generating children solutions for the next generation. Mutation is the way solutions are randomly changed to motivate diversity in the population.

Reference [7] proposed a GA approach in tackling the UCTTP. An initial population was generated by randomly assigning classes to periods by taking room capacity into considerations. The fitness value of the parents would determine their selection in generating children for the next generation through crossover and mutation operators. The authors used an array of classes in representing the chromosome where information such as the lecturers, rooms and periods were stored. This representation avoided conflict between courses. The approach was tested using their own generated data. Solutions were improved iteratively.

2) ANT COLONY OPTIMISATION

Ant Colony Optimisation (ACO) is an approach inspired from observing the foraging behaviour of ants [28]. To mark the shortest path in finding or transporting food, the ants deposit pheromone along the path as a guideline for other ants to follow. In optimisation problems, the artificial ants build their own solutions and share the information on the quality of their solution with other artificial ants.

Reference [56] proposed an ACO approach to tackle the PE-CTTP. Pheromone information was stored in two distinct matrices. Events were chosen randomly and allocated to time slots and rooms according to pheromone information. The solution was then further improved by an ejection chain. The pheromone information was updated accordingly based on the solutions with promising soft constraints penalty (SCP) and distance to feasibility (DTF) scores. The algorithm was tested using ITC-07 (Track 2) benchmark datasets. It outperformed many algorithms.

Reference [13] proposed an ACO approach to address the UCTTP by grouping students in mutually exclusive groups and then assigning each group to timeslots and venues accordingly. Three steps were iterated namely the development of initial solutions by the artificial ant, pheromone update and the execution of local search. The proposed method was tested using Socha datasets. The computational times were acceptable compared to other existing algorithms.

3) PARTICLE SWARM OPTIMISATION

Particle Swarm Optimisation (PSO) originated from social behaviour shown by collective species such as the flocking of birds, a group of tigers and a swarm of bees [73]. Every individual in a PSO model is called a particle and each particle position is equivalent to a candidate solution with a designated fitness function for the optimisation problem. This algorithm gives extra attention towards exploration and exploitation of the search space [30]. Efficiency of PSO can be improved by controlling the parameters such as the size of the swarm, inertia weight and acceleration coefficients.

Reference [25] proposed a constriction PSO to address the UCTTP. They worked on their own generated data. Among the advantages of PSO reported were fast convergence, less parameter settings and the ability to set dynamic environment characteristics. In generating a good quality solution, an interchange heuristic was applied to ensure better exploration of the solution space. The interchange heuristic prevented the particles from being trapped in local optima and allowed faster convergence to global optima. The method managed to generate acceptable solutions.

4) FISH SWARM INTELLIGENT

Reference [74] proposed fish swarm intelligent to solve the UCTTP. The proposed algorithm simulated movement shown by fish when searching for food. The search space was categorised into crowded, not crowded and empty areas. Each solution in the solution population was represented by a fish. Two local searches were used to improve the quality of the solution namely multi decay rate great deluge and steepest descent. The proposed algorithm was tested using Socha benchmark datasets. The algorithm produced best known results for some of the instances.

5) HONEY-BEE MATING

Reference [62] proposed a honey-bee mating algorithm to solve the PE-CTTP. The proposed algorithm simulated the behaviour shown by honey-bees when mating. This algorithm is effective in exploring and exploiting the search space at the same time. The proposed algorithm was tested using Socha benchmark datasets. Best known results were reported for some of the instances.

6) POPULATION BASED LOCAL SEARCH

Reference [1] proposed Population Based Local Search (PB-LS) to solve the UCTTP. They claimed it was good in exploring and exploiting the search space. Two operators were utilised for searching, namely single-direction and all-direction force. The proposed algorithm was tested using Socha benchmark datasets. The algorithm outperformed other approaches.

D. HYPER-HEURISTIC

Hyper-Heuristic approaches employ several heuristics in adaptive manner to solve the problem at hand [63], [75]. Reference [38] proposed Add-Delete Hyper-Heuristic (ADHH) to solve the UCTTP. The approach used an adaptive heuristic generation method through a variable-sized list of add and delete operations. The approach was tested on ITC-07 benchmark datasets. Its performance was better, on average, compared to other algorithms in the scientific literature.

E. MULTI CRITERIA/OBJECTIVE APPROACHES

Reference [36] proposed a Multi-Objective Simulated Annealing (MOSA) in tackling the UCTTP. They aimed to define a good Pareto front by taking into consideration the solution quality and the robustness of the solution. Algorithms were developed with single and multiple disruptions. Single disruption referred to disruption of only one lecture whereas multiple disruptions referred to disruption of more than one lectures. The ITC-07 (Track 3) benchmark dataset was used as testbed. The algorithm with multiple disruptions outperformed the one with single disruption.

F. HYBRID APPROACHES

Reference [66] proposed a Round Robin Scheduling Algorithm (RR) to control SA, Great Deluge (GD) and Hill Climbing (HC) in solving the UCTTP. It improved the quality of the initial solution generated using the least saturation degree heuristic. The algorithm was tested on Socha benchmark datasets. It managed to produce competitive results in a comparison to other state-of-the-art methods.

Reference [49] proposed an Adaptive Tabu Search (ATS) in addressing the CB-CTTP. The framework consisted of three stages. In the first initialisation stage, an initial feasible solution was generated using a greedy algorithm. A tabu search algorithm was used as a search intensification in the second stage. In the third stage, a perturbation operator from an iterated local search (ILS) was used as a diversification mechanism. Both intensification and diversification were employed to minimise soft constraint violations. They worked on ITC-07 (Track 3) datasets. The proposed method managed to produce better results compared to the TS and ILS run individually.

Reference [3] proposed a Hybrid Genetic Algorithm (HGA) in addressing the CB-CTTP. Hill climbing, simulated annealing and a genetic algorithm were hybridised in solving the formulated problem. Three moves were implemented during disruption; simple move, swap move and Kempe chain move. They tested the hybrid algorithm on ITC-07 benchmark datasets. The results produced showed high quality Pareto fronts.

Reference [41] proposed a network flow technique for the UCTTP. They generated a local solution by using Greedy Randomised Adaptive Search Procedure (GRASP) constructive algorithm together with a maximum flow partial solution. In their work, CB-CTTP was re-modelled using the maximum network flow technique. The local solution was then improved in terms of quality by using simulated annealing. The proposed method generated competitive solutions for the ITC-07 (Track 3) instances.

IV. APPROACHES/METHODOLOGIES IN REAL-WORLD UCTTP

A. OPERATIONAL RESEARCH (OR) BASED TECHNIQUES

Reference [60] proposed an Integer Programming (IP) model for the UCTTP at the Faculty of Applied Sciences, Wayamba University of Sri Lanka. The primary objective was to minimise the number of working hours per week. Decision variables were defined using a relational matrix. The model included completeness, uniqueness, consecutive and pre-assignment constraints. OpenSolver and Microsoft Excel were used as simulators. Results showed that UCTTP could be formulated effectively by using less decision variables and constraints. The model managed to reduce the number of working hours per week. The quality of the timetable could be further improved by considering the preferences of students and teachers.

Reference [8] proposed a Mixed Integer Programming (MIP) to solve the CB-CTTP at the Engineering Department of Sannio University. Two local branching heuristics were used. The first heuristic changed the room allocation but not the time-slot allocation. The second heuristic changed the day allocation but not the room and time-slot allocation. The proposed algorithm was tested on data derived from two semesters. It was shown to be efficient.

B. SINGLE SOLUTION-BASED META-HEURISTICS

1) SIMULATED ANNEALING

Reference [9] proposed a Simulated Annealing (SA) method in finding a feasible timetable for the Department of Computer Engineering in Izmir Institute of Technology. The authors investigated the performance of neighbourhood searching algorithms called swapping, simple search and their combinations. The performance of these algorithms were compared in terms of computational times and total costs. The datasets used were taken from 2007 to 2008. Results showed that the combination of simple search, swapping and simple search-swapping produced the most satisfactory timetable.

2) TABU SEARCH

Reference [4] proposed a Tabu Search (TS) algorithm in solving UCTTP for the Department of Statistics at Hacettepe University by utilising four neighbourhood structures. They were simple move, swap move and combination of both moves called Mixed_1 and Mixed_2. From experiments, simple move and Mixed_1 managed to generate the best timetables.

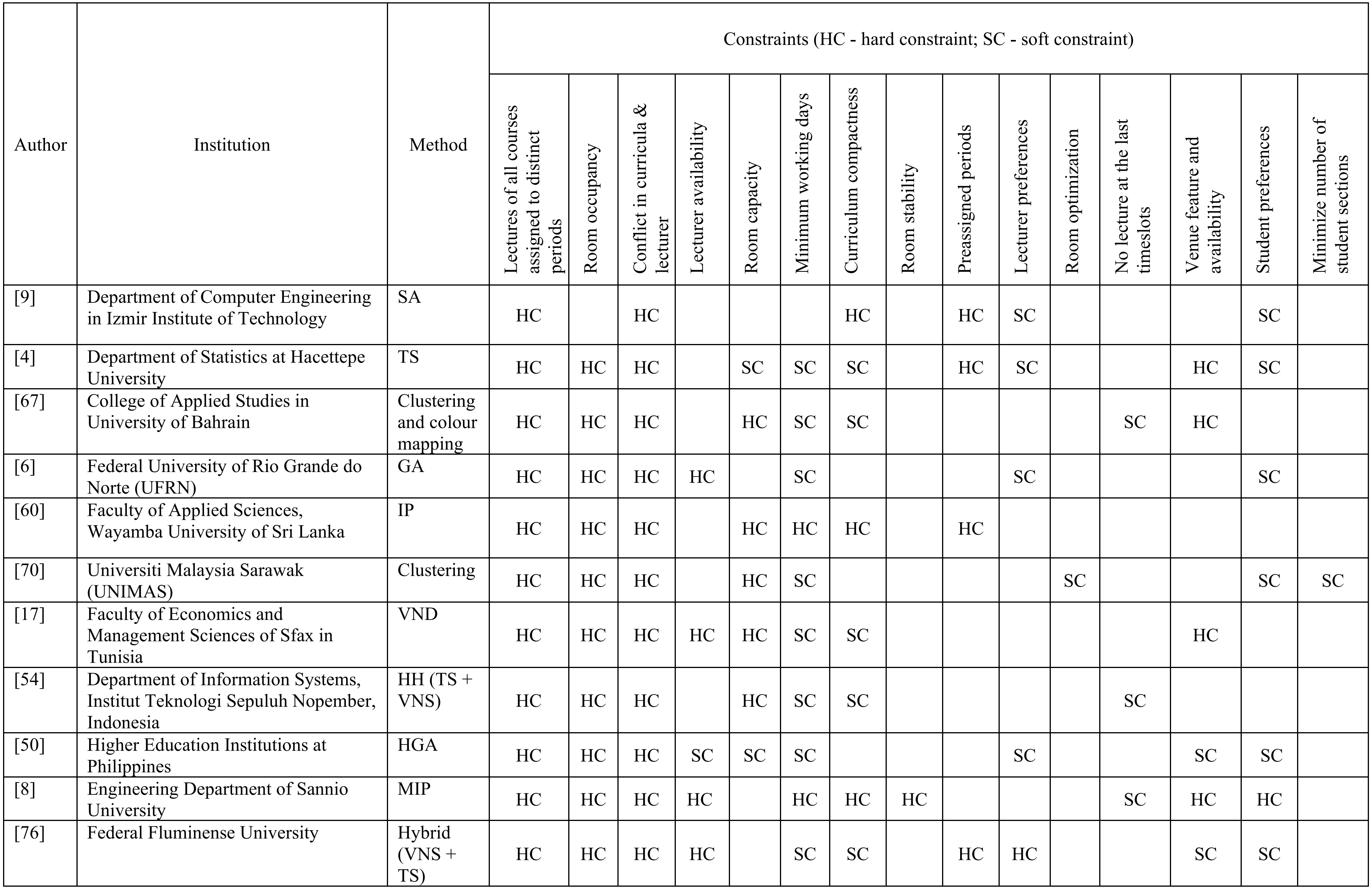
3) VARIABLE NEIGHBOURHOOD DESCENT

Reference [17] proposed a Variable Neighbourhood Descent (VND) approach in addressing the UCTTP for Faculty of Economics and Management Sciences of Sfax in Tunisia. The objectives were to minimise the total number of holes and the number of isolated lessons. Eleven neighbourhood structures were developed. Six neighbourhood structures to solve holes and five neighbourhood structures to solve isolated lessons. Six real datasets from 2012 to 2014 were used. Results showed that on average, the approach managed to eliminate 52.47% of the holes and isolated lessons. The quality of the feasible solution could be further improved by minimising working days and allocating lunch breaks for students.

C. POPULATION-BASED META-HEURISTICS

1) GENETIC ALGORITHMS

Reference [6] proposed a Genetic Algorithm (GA) approach to improve the quality of timetable for the Information Systems program of Federal University of Rio Grande do Norte (UFRN). Among the requirements were minimising time gaps between non-consecutive lectures for a group of students and avoiding scheduling all lectures for a course on the same day. Real datasets from the first semester of 2012 to the second semester of 2015 were used. Performance of the algorithm were determined by its fitness function. Results showed that solutions generated were better or equal to the ones generated manually.



Author	Institution	Method	Constraints (HC - hard constraint; SC - soft constraint)														
			Lectures of all courses assigned to distinct periods	Room occupancy	Conflict in curricula & lecturer	Lecturer availability	Room capacity	Minimum working days	Curriculum compactness	Room stability	Preassigned periods	Lecturer preferences	Room optimization	No lecture at the last timeslots	Venue feature and availability	Student preferences	Minimize number of student sections
[9]	Department of Computer Engineering in Izmir Institute of Technology	SA	HC		HC				HC		HC	SC				SC	
[4]	Department of Statistics at Hacettepe University	TS	HC	HC	HC		SC	SC	SC		HC	SC			HC	SC	
[67]	College of Applied Studies in University of Bahrain	Clustering and colour mapping	HC	HC	HC		HC	SC	SC					SC	HC		
[6]	Federal University of Rio Grande do Norte (UFRN)	GA	HC	HC	HC	HC		SC				SC				SC	
[60]	Faculty of Applied Sciences, Wayamba University of Sri Lanka	IP	HC	HC	HC		HC	HC	HC		HC						
[70]	Universiti Malaysia Sarawak (UNIMAS)	Clustering	HC	HC	HC		HC	SC					SC			SC	SC
[17]	Faculty of Economics and Management Sciences of Sfax in Tunisia	VND	HC	HC	HC	HC	HC	SC	SC						HC		
[54]	Department of Information Systems, Institut Teknologi Sepuluh Nopember, Indonesia	HH (TS + VNS)	HC	HC	HC		HC	SC	SC					SC			
[50]	Higher Education Institutions at Philippines	HGA	HC	HC	HC	SC	SC	SC				SC			SC	SC	
[8]	Engineering Department of Sannio University	MIP	HC	HC	HC	HC		HC	HC	HC				SC	HC	HC	
[76]	Federal Fluminense University	Hybrid (VNS + TS)	HC	HC	HC	HC		SC	SC		HC	HC			SC	SC	

FIGURE 1. Case studies of the real-world UCTTP.

D. HYPER-HEURISTICS

Reference [54] proposed a Hyper-Heuristic (HH) algorithm in addressing the UCTTP for the Department of Information Systems, Institut Teknologi Sepuluh Nopember, Indonesia. The objective was to generate a highly flexible optimal solution. The hyper-heuristic approach combined a TS and a VNS. Two datasets from 2017/2018 were used. Results produced were superior to that of manual timetable in terms of soft constraint violations.

E. HYBRID APPROACHES

Reference [67] proposed a clustering and colour mapping approach in producing a timetable for the College of Applied Studies in University of Bahrain. The objective was to enable students to register for their courses without any clashes. Data from official university registration system was extracted and processed in generating the timetable. The proposed algorithm was tested on problem instances featuring a total of 1270 students, 8 academic programs and 83 courses. Clusters of students were generated as an initial solution using data mining techniques. Next, they obtained the solutions using colour mapping algorithm. This was an improvement to the previous work [5] which comprised the data mining component only.

Reference [70] proposed a two-stage heuristic with clustering for Universiti Malaysia Sarawak (UNIMAS) CB-CTTP. The objectives were to automate course timetabling and increase venue utilisation. The algorithm included a prediction on course registration by students. In stage one, courses were divided into different groups. In stage two, courses in a group were assigned to the same timeslot but different venues. Real datasets from three semesters were used. Valid solutions with minimum unallocated courses were generated.

Reference [50] proposed a Hybrid Genetic Algorithm (HGA) with four neighbourhood operators to tackle the UCTTP for higher education institutions in the Philippines. The algorithm helped in managing teaching workload. When new staff were hired, it was necessary to make sure the new staff were assigned to classes that could utilise their skills and would not cause the timetable to be infeasible. The dataset consisted of 118 classes, 308 hours workload per week, 45 time-slots, five laboratories and five lecture rooms. The algorithm managed to generate feasible solutions and optimize teaching workloads.

Reference [76] proposed a hybrid of Variable Neighbourhood Search (VNS) and Tabu Search (TS) to address the UCTTP for Federal Fluminense University. The proposed algorithm was developed using the FINESS framework. The datasets used were derived from two undergraduate courses. Results indicated that the hybrid was better than the VNS and TS run individually.

Figure 1 shows the case studies of the real-world UCTTP. Constraints (hard and soft) are highly variable according to institutions. The same constraint may be hard/soft for one institution but soft/hard for the others.

V. PERSPECTIVES IN UCTTP

In this section, we provide some perspectives in UCTTP. Table 2 shows the approaches in addressing benchmark and real-world UCTTP. Figure 2 shows the classification of these approaches.

As evident from table 3, from the 35 papers surveyed, there are six OR methodologies, ten single solution-based meta-heuristics, eight population-based meta-heuristic approaches, two hyper-heuristics, one multi criteria/ objective and eight hybrid approaches.

For the benchmark UCTTP, single solution-based meta-heuristics(7) and population-based meta-heuristics(7) are the most popular approaches. Five out of the seven single solution-based meta-heuristic approaches are based on SA. Two out of the seven population-based meta-heuristic approaches are ACOs. It would be interesting to see the outcome of hybridising SA with population-based approaches such as ACO which are popular for their explorative capability. Approaches such as hyper-heuristic and multi criteria/objective are less popular perhaps due to their performance. However, they are less researched therefore providing opportunity for new studies.

For the real-world UCTTP, hybrid(4) approaches are the most popular. This is followed by single solution-based meta-heuristic(3), OR (2), population-based meta-heuristic(1) and hyper-heuristic(1). Two out of the four hybrid approaches are hybrids of VNS and TS.

From observation, state-of-the-art approaches in benchmark UCTTP are not fully utilised in real-world UCTTP. Researchers may adopt/adapt the state-of-the-art approaches in benchmark UCTTP to real-world UCTTP at academic institutions.

VI. TRENDS IN BENCHMARK UCTTP

The benchmark datasets and their respective state-of-the-art methodologies are discussed in this section. As evident from Table 4, the benchmark datasets utilised in the international timetabling competitions are the most popular testbeds among researchers in comparing algorithms.

A. SOCHA BENCHMARK DATASET

The Socha benchmark dataset is developed by utilising an algorithm created by Ben Paechter. It consists of 11 instances. The features of this dataset are shown in Table 5. In the last 10 years, 11 different approaches were proposed for this dataset. Variants of SA proposed by [34] are superior to others in terms of performance. Other state of the art method for this dataset is the TS based approach called random partial neighbourhood search (RPNS) by [55].

B. ITC-02 BENCHMARK DATASET

The International Timetabling Competition 2002 (ITC-02) is organized by the Meta-heuristic Network and sponsored by Practice and Theory of Automated Timetabling (PATAT). The benchmark dataset (20 instances) can be downloaded from

TABLE 2. Approaches in solving UCTTP (benchmark and real world problem datasets).

Year	Reference	Method	Category	Dataset
2010	[49]	ATS	Hybrid	ITC-07 (T3)
	[74]	Fish swarm intelligent	Population-based Meta-heuristic	Socha
	[66]	RR scheduling algorithm (HC, GD, SA)	Hybrid	Socha
	[44]	Time-dependent meta-heuristic	Single Solution-based Meta-heuristic	ITC-07 (T2)
2011	[48]	Clique-based	OR	Hard
2012	[18]	GC (LSDF)	OR	Socha
	[62]	Honey-bee mating	Population-based Meta-heuristic	Socha
	[24]	SA	Single Solution-based Meta-heuristic	Hard, Socha, ITC-02 and ITC-07 (T2)
	[56]	ACO	Population-based Meta-heuristic	ITC-07 (T2)
2013	[25]	PSO	Population-based Meta-heuristic	Own dataset
2014	[1]	PB-LS	Population-based Meta-heuristic	Socha
2016	[13]	ACO	Population-based Meta-heuristic	Socha
2017	[31]	TSSP and SAR	Single Solution-based Meta-heuristic	Socha, ITC-02 and ITC-07 (T2)
2018	[7]	GA	Population-based Meta-heuristic	Own dataset
	[68]	ILS	Single Solution-based Meta-heuristic	Hard
	[3]	HGA, SA and HC	Hybrid	ITC-07 (T3)
	[47]	MILP	OR	ITC-07 (T3)
	[38]	ADHH	Hyper-heuristic	ITC-07 (T3)
	[55]	TS(RPNS)	Single Solution-based Meta-heuristic	Socha, ITC-02 and ITC-07 (T2)
2019	[41]	Network flow, GRASP and SA	Hybrid	ITC-07 (T3)
	[14]	ILP	OR	ITC-07 (T3)
	[33]	SAIRL	Single Solution-based Meta-heuristic	Socha, ITC-02 and ITC-07 (T2)
2020	[36]	MOSA	Multi Criteria/ Objective	ITC-07 (T3)
	[34]	SAR, ILS and SAR-2P	Single Solution-based Meta-heuristic	Hard, Socha, ITC-02 and ITC-07 (T2)
2009	[9]	SA	Single Solution-based Meta-heuristic	Real-world dataset
	[4]	TS	Single Solution-based Meta-heuristic	Real-world dataset
2012	[67]	Clustering and color mapping	Hybrid	Real-world dataset
2016	[6]	GA	Population-based Meta-heuristic	Real-world dataset
2017	[60]	IP	OR	Real-world dataset
	[70]	Clustering	Hybrid	Real-world dataset
	[17]	VND	Single Solution-based Meta-heuristic	Real-world dataset
2019	[54]	HH (TS + VNS)	Hyper-heuristic	Real-world dataset
	[50]	HGA	Hybrid	Real-world dataset
	[8]	MIP	OR	Real-world dataset
2020	[76]	VNS + TS	Hybrid	Real-world dataset

TABLE 3. Summary of approaches in UCTTP.

UCTTP	Category						Total
	OR	Single Solution-based Meta-heuristic	Population-based Meta-heuristic	Hyper-heuristic	Multi Criteria/ Objective	Hybrid	
Benchmark	4	7	7	1	1	4	24
Real-world	2	3	1	1	0	4	11
Total	6	10	8	2	1	8	35

the ITC-02 website.[1] The dataset is produced using an algorithm by Ben Paechter. There is a time limit requirement for this dataset which is dictated by running a program on the host computer. Over the last 10 years, five different approaches have been proposed for this dataset. TSSP, ILS and SAR-2P proposed by [34] performed better than the other four. The features of this dataset are presented in Table 6.

C. ITC-07 (TRACK 2) BENCHMARK DATASET

The PE-CTT variant benchmark dataset (24 instances) from the International Timetabling Competition 2007 (ITC-07) can be downloaded.[2] In the last 10 years, seven different approaches were proposed for this dataset. The current state of the art methods are RPNS [55], SAR-2P [34] and SA [24]. The features of the dataset are given in Table 7.

D. ITC-07 (TRACK 3) BENCHMARK DATASET

The CB-CTT variant dataset (21 instances) for International Timetabling Competition (ITC-07) can be downloaded.[3] In the last 10 years, seven different approaches were proposed for this dataset. A network flow methodology (GRASP + SA) [41] is superior than ATS [49]. IP relaxation proposed by [14] outperformed the other six methodologies by improving the lower bounds for three of the problem instances. The features of the dataset are given in Table 8.

[1] http://sferics.idsia.ch/Files/ttcomp2002/oldindex.html. Last accessed: Nov 26, 2020

[2] http://www.cs.qub.ac.uk/itc2007/index.htm. Last accessed: Nov 26, 2020

[3] http://www.cs.qub.ac.uk/itc2007/index.htm. Last accessed: Nov 26, 2020

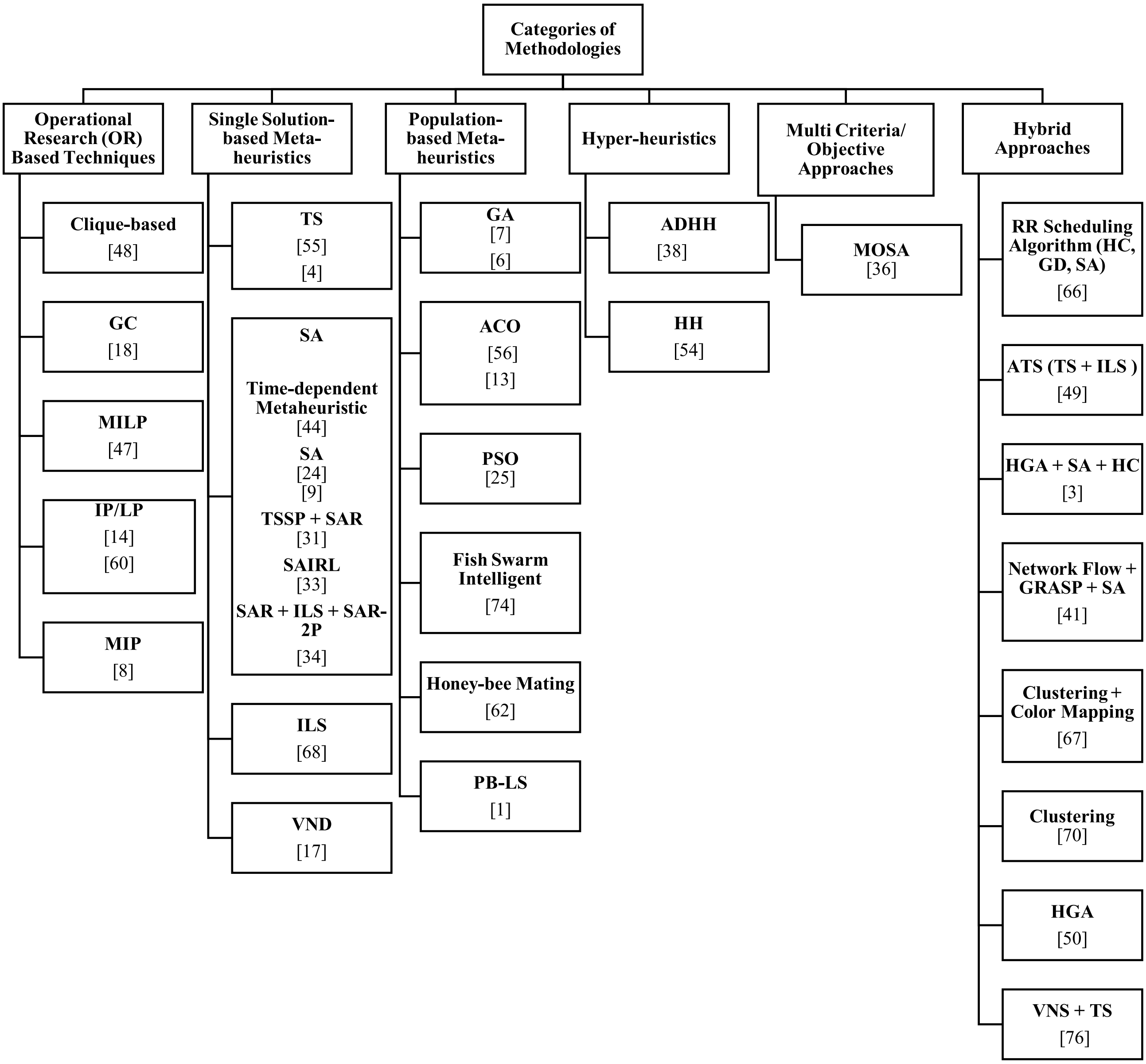

Categories of Methodologies
Operational Research (OR) Based Techniques
Clique-based [48]
GC [18]
MILP [47]
IP/LP [14] [60]
MIP [8]
Single Solution-based Meta-heuristics
TS [55] [4]
SA
Time-dependent Metaheuristic [44]
SA [24] [9]
TSSP + SAR [31]
SAIRL [33]
SAR + ILS + SAR-2P [34]
ILS [68]
VND [17]
Population-based Meta-heuristics
GA [7] [6]
ACO [56] [13]
PSO [25]
Fish Swarm Intelligent [74]
Honey-bee Mating [62]
PB-LS [1]
Hyper-heuristics
ADHH [38]
HH [54]
Multi Criteria/ Objective Approaches
MOSA [36]
Hybrid Approaches
RR Scheduling Algorithm (HC, GD, SA) [66]
ATS (TS + ILS) [49]
HGA + SA + HC [3]
Network Flow + GRASP + SA [41]
Clustering + Color Mapping [67]
Clustering [70]
HGA [50]
VNS + TS [76]

FIGURE 2. Classification of approaches in benchmark and real-world UCTTP.

E. HARD BENCHMARK DATASET

The 60 instances (20 smalls, 20 mediums and 20 larges) proposed by [46] can be downloaded from the Centre for Emergent Computing website.[4] The current state-of-the-art method for this dataset is ILS proposed by [68] and TSSP-ILS proposed by [34]. They managed to find feasible solutions for 58 and 57 instances respectively. Table 9 shows the features of the dataset.

F. ITC-2019 BENCHMARK DATASET

The International Timetabling Competition (ITC-2019) is the latest timetabling competition. Its benchmark dataset (30 instances) can be downloaded.[5] Student sectioning is considered in these problem instances. Table 10 shows the features of the dataset.

VII. LIMITATIONS OF THE APPROACHES/ METHODOLOGIES IN UCTTP

A total of 35 approaches in solving UCTTP are surveyed in this paper. Each approach has their own unique advantages and limitations.

Operational research (OR) based techniques such as constraint logic programming and graph colouring are effective in generating feasible solutions but are lacking in producing good quality solutions compared to other approaches [23].

[4] http://www.rhydlewis.eu/hardTT/. Last accessed: Nov 26, 2020

[5] https://www.itc2019.org/. Last accessed: Nov 26, 2020

TABLE 4. Benchmark datasets and their respective state-of-the-art methodologies.

Dataset	Year	Methods
Socha	2010	Fish Swarm Intelligent [74]
		RR scheduling algorithm (HC, GD, SA) [66]
	2012	Honey-bee mating [62]
		SA [24]
		GC(LSDF) [18]
	2014	PB-LS [1]
	2016	ACO [13]
	2017	TSSP and SAR [31]
	2018	TS(RPNS) [55]
	2019	SAIRL [33]
	2020	SAR, ILS and SAR-2P [34]
ITC-02	2012	SA [24]
	2017	TSSP and SAR [31]
	2018	TS(RPNS) [55]
	2019	SAIRL [33]
	2020	SAR, ILS and SAR-2P [34]
ITC-07 (Track 2)	2010	Time-dependent meta-heuristic [44]
	2012	SA [24]
		ACO [56]
	2017	TSSP and SAR [31]
	2018	TS(RPNS) [55]
	2019	SAIRL [33]
	2020	SAR, ILS and SAR-2P [34]
ITC-07 (Track 3)	2010	ATS [49]
	2018	HGA, SA, HC [3]
		MILP [47]
		ADHH [38]
	2019	ILP [14]
		Network flow, GRASP and SA [41]
	2020	MOSA [36]
Hard	2011	Clique-based [48]
	2012	SA [24]
	2018	ILS [68]
	2020	SAR, ILS and SAR-2P [34]

TABLE 5. The features of the Socha benchmark dataset.

Instances	Students	Events	Rooms	Features
S01	80	100	5	5
S02	80	100	5	5
S03	80	100	5	5
S04	80	100	5	5
S05	80	100	5	5
M01	200	400	10	5
M02	200	400	10	5
M03	200	400	10	5
M04	200	400	10	5
M05	200	400	10	5
L	400	400	10	10

TABLE 6. The features of the ITC-02 benchmark dataset.

Instances	Students	Events	Rooms	Features
01	200	400	10	10
02	200	400	10	10
03	200	400	10	10
04	300	400	10	5
05	300	350	10	10
06	300	350	10	5
07	350	350	10	5
08	250	400	10	5
09	220	440	11	6
10	200	400	10	5
11	220	400	10	6
12	200	400	10	5
13	250	400	10	6
14	350	350	10	5
15	300	350	10	10
16	220	440	11	6
17	300	350	10	10
18	200	400	10	10
19	300	400	10	5
20	300	350	10	5

Moreover, some of the early heuristics are not efficient in solving large problems [22].

Single solution-based meta-heuristics such as SA, are effective in achieving high quality solutions. However, researchers need to consider parameter tuning when choosing meta-heuristic approaches. Researchers are working on designing an optimisation algorithm that is not only effective but requires less manual parameter setting [31], [33], [34].

Population-based meta-heuristics such as GA, PSO and ACO are superior compared to others in terms of solution space exploration [73]. However, one of the drawbacks of these approaches are the computational times required in finding good quality solutions.

Due to the limitation of meta-heuristic approaches which require intensive parameter tuning, hyper-heuristic approaches were introduced [61]. Hyper-heuristics are general, simple and fast algorithms applicable to variety of problem domain and can event adapt to different instances of a given benchmark dataset. Hyper-heuristics are heuristics to choose heuristics (algorithms), working on a search space of heuristics (algorithms) instead of a search space of solutions [21]. The challenges in hyper-heuristic approaches are balancing information exchange and maintaining a problem domain barrier between the low level heuristics and the high level search methodology [29].

Addressing multiple objectives is often challenging when tackling an optimisation problem. This is because when the number of objectives increases, the proportion of non-dominated solutions in a randomly chosen set of objective vectors becomes exponentially large [27]. Multi criteria/objective approaches gather much interest from researchers in finding the optimal Pareto front, a set of optimal compromise solutions. However, large computational effort is often required in finding the Pareto front, even more so when it is desirable to have the solutions evenly spread along the Pareto fronts [47].

It is believed that population-based approaches and local search algorithms are suited in solution space exploration and exploitation respectively. Therefore, attempts have been made to achieve the synergy of both capabilities required in addressing optimisation problems. However, hybrid methods are more complex to implement and require greater computational cost [40].

VIII. RESEARCH OPPORTUNITIES IN UCTTP

Addressing UCTTP is important for academic institutions yet challenging due to the size and the number of hard and soft constraints involved.

Heuristic approaches help to generate feasible solutions in reasonable computational time but they are lacking compared to meta-heuristic approaches in terms of optimisation [23]. Reference [59] noted that constructive heuristics are important in addressing combinatorial optimisation problems as

TABLE 7. The features of the ITC-07 (Track 2) benchmark dataset.

Instances	Students	Events	Rooms	Features
01	500	400	10	10
02	500	400	10	10
03	1000	200	20	10
04	1000	200	20	10
05	300	400	20	20
06	300	400	20	20
07	500	200	20	20
08	500	200	20	20
09	500	400	10	20
10	500	400	10	20
11	1000	200	10	10
12	1000	200	10	10
13	300	400	20	10
14	300	400	20	10
15	500	200	10	20
16	500	200	10	20
17	500	100	10	10
18	500	200	10	10
19	1000	300	10	10
20	1000	400	10	10
21	300	500	20	20
22	500	600	20	20
23	1000	400	20	30
24	1000	400	20	30

TABLE 8. The features of the ITC-07 (Track 3) benchmark dataset.

Instances	Rooms	Courses	Curricula	Constraints
01	6	30	14	53
02	16	82	70	513
03	16	72	68	382
04	18	79	57	396
05	9	54	139	771
06	18	108	70	632
07	20	131	77	667
08	18	86	61	478
09	18	76	75	405
10	18	115	67	694
11	5	30	13	94
12	11	88	150	1368
13	19	82	66	468
14	17	85	60	486
15	16	72	68	382
16	20	108	71	518
17	17	99	70	548
18	9	47	52	594
19	16	74	66	475
20	19	121	78	691
21	18	94	78	463

TABLE 9. The features of hard benchmark dataset.

Instances	Students	Events	Rooms	Features
01	1000	1000	28	20
02	1000	1000	25	20
03	900	1000	25	20
04	800	1050	25	20
05	1000	1075	25	20
06	1000	1075	25	20
07	1100	1050	25	20
08	1000	1025	25	20
09	800	1050	25	20
10	1000	1075	25	20
11	1000	1075	25	20
12	1000	1000	26	25
13	1000	1000	25	25
14	1000	1000	25	25
15	1000	1000	25	25
16	1000	1000	25	10
17	1200	1000	25	10
18	1000	1000	25	10
19	1000	1000	25	10
20	1000	1000	25	10
21	400	400	10	10
22	400	390	10	10
23	400	390	10	10
24	400	410	10	9
25	450	410	10	9
26	450	410	11	10
27	450	410	11	10
28	400	400	10	10
29	400	400	10	10
30	500	400	10	8
31	800	400	10	8
32	800	400	10	8
33	800	400	10	8
34	1000	400	10	8
35	500	425	10	8
36	1000	400	10	8
37	800	400	10	8
38	1000	400	10	8
39	1000	410	10	8
40	1000	410	10	8
41	200	200	5	5
42	400	210	6	5
43	400	200	6	5
44	500	200	5	8
45	500	200	5	8
46	1000	200	5	3
47	800	200	5	3
48	1000	225	5	10
49	900	225	5	10
50	1000	220	5	10
51	1000	200	5	4
52	1000	225	5	10
53	1000	225	5	10
54	1000	225	5	3
55	900	200	5	3
56	900	200	5	3
57	900	200	5	3
58	1000	225	5	3
59	1000	225	5	3
60	1000	225	5	3

they are usually used to create initial solutions which would then improved by other approaches.

Local search (Single solution-based meta-heuristics) is promising as it is easy to implement and capable of addressing large sized problems in reasonable computational times [20]. It will be interesting to test relatively new meta-heuristics such as grey wolf optimizer (GWO) [53] and elitist self-adaptive step-size search (ESASS) [10] in addressing the UCTTP. The GWO algorithm was inspired by the leadership hierarchy and hunting mechanism of grey wolves in nature. Meanwhile, ESASS was first utilised in steel frame structural design optimization. Both of these algorithms provided very competitive results in their respective problem domains.

Hybridisation of approaches appears to be the best methodology to adopt. It has shown good quality results in previous research [40]. These earlier findings have been validated by more recent work [3]. Hybrid methodologies are suitable in exploiting the strength of individual approaches.

As UCTTP is unique across institutions due to policies and regulatory requirements, it is difficult if not impossible, to compare solution approaches objectively [36]. This has led to the introduction of international timetabling

TABLE 10. The features of the ITC-2019 benchmark dataset.

Instance	Size (MB)	Courses	Classes	Rooms	Students
1	14.55	340	1239(543 fixed)	80	1641
2	5.82	272	1852(332 fixed)	44	2116
3	3.88	353	983(79 fixed)	62	3018
4	12.60	1206	2641(530 fixed)	214	0
5	2.94	544	882(63 fixed)	90	3666
6	1.41	228	575(128 fixed)	35	1543
7	1.48	226	561(191 fixed)	44	865
8	15.92	1089	2526(1132 fixed)	70	2938
9	4.69	687	1001(318 fixed)	75	27018
10	1.94	36	711(74 fixed)	15	0
11	11.18	406	1144(97 fixed)	84	2254
12	10.74	234	460(2 fixed)	39	1988
13	3.03	313	487(3 fixed)	73	0
14	1.39	186	516(63 fixed)	35	1469
15	7.70	116	650(32 fixed)	29	395
16	6.71	881	1515(159 fixed)	83	3443
17	3.53	404	782(41 fixed)	67	2293
18	2.82	212	1061(115 fixed)	84	13497
19	32.11	2839	8813(2809 fixed)	768	38437
20	2.05	91	417(14 fixed)	28	821
21	42.84	1363	5081(341 fixed)	327	6925
22	4.16	357	1083(97 fixed)	63	2921
23	12.31	1290	2782(838 fixed)	208	0
24	3.10	328	502(8 fixed)	73	0
25	2.49	540	951(186 fixed)	93	5051
26	1.36	188	535(60 fixed)	36	1685
27	11.52	515	1623(443 fixed)	33	1152
28	27.25	1635	3717(312 fixed)	86	5651
29	10.83	1154	2798(449 fixed)	224	35213
30	1.74	44	676(47 fixed)	18	0

competitions. The winners are selected based on the quality of the solution (evaluated by a cost function) [2]. A solution that violates any hard constraints is considered as worthless [45]. Recently, the benchmark dataset for the International Timetabling Competition 2019 (ITC-19) has been made available to the public. It is interesting to justify the performance of methodologies using this dataset.

Benchmark and real-world UCTTP vary in terms of the number and type of hard and soft constraints. Benchmark UCTTP are usually oversimplified and meant for objective comparison of methodologies. Meanwhile, real-world UCTTP focus on the practicality of solution at academic institutions. Even real-world UCTTP vary between them in terms of requirements due to different policies, education systems and cultures. Therefore, a general solution (particularly the underlying mathematical model) that fits all does not exist. Reference [57] conducted a systematic review on practices in timetabling in higher education. The aim is to identify the gap (similarities and differences) between theory in timetabling problems and the practicality in real-world environments of higher education institutions. McCollum highlighted the importance of generating robust and flexible techniques that can cope with complexities that arise in real-world implementations [51]. It is imperative to design an optimization algorithm that is not only effective but simple to use and adaptable to a range of real-world UCTTP. This will allow researchers to adapt/adopt the implementation of state-of-the-art methods on real-world UCTTP at academic institutions.

From observation, researchers are more interested in the operational than the strategic perspective of timetabling problem. Operational level refers to allocation process of lectures to rooms and time slots. Meanwhile, strategic level refers to management decisions such as room number and capacity. Reference [47] conducted research from the aspect of strategic planning of academic institutions. The authors focused on two strategic components, namely the number of rooms required and the available time slots. Future work may focus on room features and location.

In real-world UCTTP, student sectioning is a method to improve room utilisation and timetable feasibility especially for large sized problems. Schindl investigated student sectioning with the aim of dividing students into optimal sections [65]. The optimal number of sections for each course depends on various factors such as room capacity, institution budget and pedagogical constraints. Currently, the author managed to achieve optimal sectioning with equal section size. Optimal sectioning with non-equal section size is an open research issue.

IX. CONCLUSION

The university course timetabling problem is an active and important research area based on the sizeable amount of papers found in the scientific literature. The introduction of international timetabling competitions continues to motivate research in this area. This paper surveys the approaches in addressing university course timetabling problem (benchmark and real-world) proposed in the last 10 years. The approaches are classified according to category. In addition, they are sorted chronologically according to publication year thus giving an overview of the current trend in this domain. Features of benchmark datasets are detailed in tabular form. The origin, links, state of the art methodologies for each dataset are presented. In addition, this paper provides limitations of each category of methodologies. Research opportunities in university course timetabling problem are also discussed.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Universiti Malaysia Sarawak (UNIMAS). This paper would not have been published without the support of UNIMAS.

REFERENCES

[1] A. Abuhamdah, M. Ayob, G. Kendall, and N. R. Sabar, "Population based local search for university course timetabling problems," *Int. J. Speech Technol.*, vol. 40, no. 1, pp. 44–53, Jun. 2013.

[2] L. N. Ahmed, E. Özcan, and A. Kheiri, "Solving high school timetabling problems worldwide using selection hyper-heuristics," *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5463–5471, Aug. 2015.

[3] C. Akkan and A. Gülcü, "A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem," *Comput. Oper. Res.*, vol. 90, pp. 22–32, Feb. 2018.

[4] C. H. Aladag, G. Hocaoglu, and M. A. Basaran, "The effect of neighborhood structures on Tabu search algorithm in solving course timetabling problem," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12349–12356, Dec. 2009.

[5] F. Albalooshi and S. Shatnawi, "Online Academic advising support," in *Technological Developments in Networking, Education and Automation*. Dordrecht, The Netherlands: Springer, 2010, pp. 25–29.

[6] M. W. S. Almeida, J. P. S. Medeiros, and P. R. Oliveira, "Solving the academic timetable problem thinking on Student needs," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 673–676.

[7] M. Assi, B. Halawi, and R. A. Haraty, "Genetic algorithm analysis using the graph coloring method for solving the university timetable problem," *Procedia Comput. Sci.*, vol. 126, pp. 899–906, Jan. 2018.

[8] P. Avella, M. Boccia, S. Viglione, and I. Vasilyev, "A local branching MIP heuristic for a real-world curriculum-based course timetabling problem," in *Mathematical Optimization Theory and Operations Research*. Cham, Switzerland: Springer, 2019, pp. 438–451.

[9] E. Aycan and T. Ayav, "Solving the course scheduling problem using simulated annealing," in *Proc. IEEE Int. Advance Comput. Conf.*, Mar. 2009, pp. 462–466.

[10] S. K. Azad and O. Hasançebi, "An elitist self-adaptive step-size search for structural design optimization," *Appl. Soft Comput.*, vol. 19, pp. 226–235, Jun. 2014.

[11] N. L. A. Aziz and N. A. H. Aizam, "University course timetabling and the requirements: Survey in several universities in the east-coast of Malaysia," *AIP Conf. Proc.*, vol. 1870, Aug. 2017, Art. no. 040013.

[12] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Comput. Ind. Eng.*, vol. 86, pp. 43–59, Aug. 2015.

[13] R. P. Badoni and D. K. Gupta, "A new algorithm based on students groupings for university course timetabling problem," in *Proc. 2nd Int. Conf. Recent Adv. Eng. Comput. Sci. (RAECS)*, Dec. 2015, pp. 1–5.

[14] N.-C.-F. Bagger, G. Desaulniers, and J. Desrosiers, "Daily course pattern formulation and valid inequalities for the curriculum-based course timetabling problem," *J. Scheduling*, vol. 22, no. 2, pp. 155–172, Apr. 2019.

[15] A. V. Bardadym, "Computer-aided school and university timetabling: The new wave," in *Practice and Theory of Automated Timetabling* (Lecture Notes in Computer Science), vol. 1153. Berlin, Germany: Springer, 1996, pp. 22–45.

[16] A. Bettinelli, V. Cacchiani, R. Roberti, and P. Toth, *An Overview Curriculum-Based Course Timetabling*, vol. 23. Berlin, Germany: Springer, 2015.

[17] R. Borchani, A. Elloumi, and M. Masmoudi, "Variable neighborhood descent search based algorithms for course timetabling problem: Application to a Tunisian university," *Electron. Notes Discrete Math.*, vol. 58, pp. 119–126, Apr. 2017.

[18] T. A. Budiono and K. Wai Wong, "A pure graph coloring constructive heuristic in timetabling," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Jun. 2012, pp. 307–312.

[19] E. Burke, "Automated university timetabling: The state of the art," *Comput. J.*, vol. 40, no. 9, pp. 565–571, Sep. 1997.

[20] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," *Eur. J. Oper. Res.*, vol. 258, no. 1, pp. 70–78, 2017.

[21] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *J. Oper. Res. Soc.*, vol. 64, no. 12, pp. 1695–1724, 2013.

[22] E. K. Burke, B. Mccollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems," *Eur. J. Oper. Res.*, vol. 176, no. 1, pp. 177–192, 2007.

[23] E. K. Burke and S. Petrovic, "Recent research directions in automated timetabling," *Eur. J. Oper. Res.*, vol. 140, no. 2, pp. 266–280, Jul. 2002.

[24] S. Ceschia, L. D. Gaspero, and A. Schaerf, "Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1615–1624, Jul. 2012.

[25] R.-M. Chen and H.-F. Shih, "Solving university course timetabling problems using constriction particle swarm optimization with local search," *Algorithms*, vol. 6, no. 2, pp. 227–244, Apr. 2013.

[26] D. de Werra, "An introduction to timetabling," *Eur. J. Oper. Res.*, vol. 19, no. 2, pp. 151–162, Feb. 1985.

[27] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach—Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[28] M. Dorigo, M. Birattari, and T. Stutzle, "Artificial ants as a computational intelligence technique," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Sep. 2006.

[29] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, "Recent advances in selection hyper-heuristics," *Eur. J. Oper. Res.*, vol. 285, no. 2, pp. 405–428, Sep. 2020.

[30] M. B. Ghalia, "Particle swarm optimization with an improved exploration-exploitation balance," in *Proc. 51st Midwest Symp. Circuits Syst.*, Aug. 2008, pp. 759–762.

[31] S. L. Goh, G. Kendall, and N. R. Sabar, "Improved local search approaches to solve the post enrolment course timetabling problem," *Eur. J. Oper. Res.*, vol. 261, no. 1, pp. 17–29, Aug. 2017.

[32] S. L. Goh, G. Kendall, and R. Nasser Sabar, "Monte Carlo tree search in finding feasible solutions for course timetabling problem," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 9, no. 6, pp. 1936–1943, 2019.

[33] S. L. Goh, G. Kendall, and N. R. Sabar, "Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem," *J. Oper. Res. Soc.*, vol. 70, no. 6, pp. 873–888, Jun. 2019.

[34] S. L. Goh, G. Kendall, R. Nasser Sabar, and S. Abdullah, "An effective hybrid local search approach for the post enrolment course timetabling problem," *Opsearch*, vol. 57, no. 4, pp. 1131–1163, Jun. 2020.

[35] W. Gora, G. Lach, J. Lubbe, O. Pfeiffer, E. Zorn, and S. Jeschke, "Management and optimal distribution of large student numbers," in *Proc. IEEE EDUCON Conf.*, Apr. 2010, pp. 1891–1896.

[36] A. Gülcü and C. Akkan, "Robust university course timetabling problem subject to single and multiple disruptions," *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 630–646, Jun. 2020.

[37] A. Gunawan, K. M. Ng, and K. L. Poh, "A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3074–3088, Dec. 2012.

[38] S. S. Habashi, C. Salama, A. H. Yousef, and H. M. A. Fahmy, "Adaptive diversifying hyper-heuristic based approach for timetabling problems," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 259–266.

[39] T. Harada and E. Alba, "Parallel genetic algorithms: A useful survey," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–39, Sep. 2020.

[40] R. Ilyas and Z. Iqbal, "Study of hybrid approaches used for university course timetable problem (UCTP)," in *Proc. IEEE 10th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2015, pp. 696–701.

[41] H. E. Kampke, M. L. Scheideger, R. G. Mauri, and C. S. M. Boeres, "A network flow based construction for a GRASP + SA algorithm to solve the university timetabling problem," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2019, pp. 215–231.

[42] A. Kiefer, R. F. Hartl, and A. Schnell, "Adaptive large neighborhood search for the curriculum-based course timetabling problem," *Ann. Oper. Res.*, vol. 252, no. 2, pp. 255–282, May 2017.

[43] S. Kristiansen and T. R. Stidsen, "A comprehensive study of educational timetabling, a survey," in *A Comprehensive Study of Educational Timetabling—A Survey*, Nov. 2013.

[44] R. Lewis, "A time-dependent Metaheuristic algorithm for post enrolment-based course timetabling," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 273–289, Jan. 2010.

[45] R. Lewis, "A survey of Metaheuristic-based techniques for university timetabling problems," *OR Spectr.*, vol. 30, no. 1, pp. 167–190, Nov. 2007.

[46] R. Lewis and B. Paechter, "Finding feasible timetables using group-based operators," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 397–413, Jun. 2007.

[47] M. Lindahl, A. J. Mason, T. Stidsen, and M. Sørensen, "A strategic view of university timetabling," *Eur. J. Oper. Res.*, vol. 266, no. 1, pp. 35–45, Apr. 2018.

[48] Y. Liu, D. Zhang, and F. Y. L. Chin, "A clique-based algorithm for constructing feasible timetables," *Optim. Methods Softw.*, vol. 26, no. 2, pp. 281–294, Apr. 2011.

[49] Z. Lü and J.-K. Hao, "Adaptive Tabu search for course timetabling," *Eur. J. Oper. Res.*, vol. 200, no. 1, pp. 235–244, Jan. 2010.

[50] J. B. Matias, A. C. Fajardo, and R. P. Medina, "A hybrid genetic algorithm for course scheduling and teaching workload management," in *Proc. IEEE 10th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2018, pp. 1–6.

[51] B. McCollum, "A perspective on bridging the gap between theory and practice in university timetabling," in *Practice and Theory of Automated Timetabling VI* (Lecture Notes in Computer Science), vol. 3867. Berlin, Germany: Springer, 2006, pp. 3–23.

[52] I. Méndez-Díaz, P. Zabala, and J. J. Miranda-Bront, "An ILP based heuristic for a generalization of the post-enrollment course timetabling problem," *Comput. Oper. Res.*, vol. 76, pp. 195–207, Dec. 2016.

[53] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[54] A. Muklason, R. G. Irianti, and A. Marom, "Automated course timetabling optimization using tabu-variable neighborhood search based hyper-heuristic algorithm," *Procedia Comput. Sci.*, vol. 161, pp. 656–664, Jan. 2019.

[55] Y. Nagata, "Random partial neighborhood search for the post-enrollment course timetabling problem," *Comput. Oper. Res.*, vol. 90, pp. 84–96, Feb. 2018.

[56] C. Nothegger, A. Mayer, A. Chwatal, and G. R. Raidl, "Solving the post enrolment course timetabling problem by ant colony optimization," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 325–339, Feb. 2012.

[57] 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, Apr. 2019.

[58] N. Pillay, *A Review of Hyper-Heuristics for Educational Timetabling*, vol. 239. Cham, Switzerland: Springer, 2016.

[59] N. Pillay and E. Özcan, "Automated generation of constructive ordering heuristics for educational timetabling," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 181–208, Apr. 2019.

[60] R. A. R. Prabodanie, "An integer programming model for a complex university timetabling problem: A case study," *Ind. Eng. Manage. Syst.*, vol. 16, no. 1, pp. 141–153, Mar. 2017.

[61] P. Ross and J. G. Marin-Blazquez, "Constructive hyper-heuristics in class timetabling," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2005, pp. 1493–1500.

[62] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A honey-bee mating optimization algorithm for educational timetabling problems," *Eur. J. Oper. Res.*, vol. 216, no. 3, pp. 533–543, Feb. 2012.

[63] N. R. Sabar, A. Turky, A. Song, and A. Sattar, "An evolutionary hyper-heuristic to optimise deep belief networks for image reconstruction," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 105510.

[64] A. Schaerf, "Survey of automated timetabling," *Artif. Intell. Rev.*, vol. 13, no. 2, pp. 87–127, 1999.

[65] D. Schindl, "Optimal student sectioning on mandatory courses with various sections numbers," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 209–221, Apr. 2019.

[66] K. Shaker and S. Abdullah, "Controlling multi algorithms using Round Robin for university course timetabling problem," in *Database Theory and Application, Bio-Science and Bio-Technology*. Berlin, Germany: Springer, 2010, pp. 47–55.

[67] S. M. Shatnawi, F. Albalooshi, and K. Rababa, "Generating timetable and students schedule based on data mining techniques," *Int. J. Eng. Res. Appl.*, vol. 2, no. 4, pp. 1638–1644, 2012.

[68] T. Song, S. Liu, X. Tang, X. Peng, and M. Chen, "An iterated local search algorithm for the university course timetabling problem," *Appl. Soft Comput.*, vol. 68, pp. 597–608, Jul. 2018.

[69] K. Sörensen and W. F. Glover, *Metaheuristics*. Boston, MA, USA: Springer, 2013, pp. 960–970.

[70] S.-N. Sze, C.-L. Bong, K.-L. Chiew, W.-K. Tiong, and N. A. Bolhassan, "Case study: University lecture timetabling without pre-registration data," in *Proc. Int. Conf. Appl. Syst. Innov. (ICASI)*, May 2017, pp. 732–735.

[71] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113943.

[72] J. S. Tan, S. L. Goh, S. Sura, G. Kendall, and N. R. Sabar, "Hybrid particle swarm optimization with particle elimination for the high school timetabling problem," *Evol. Intell.*, pp. 1–16, Aug. 2020.

[73] C. K. Teoh, A. Wibowo, and M. S. Ngadiman, "Review of state of the art for metaheuristic techniques in Academic scheduling problems," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 1–21, Jun. 2015.

[74] H. Turabieh, S. Abdullah, B. McCollum, and P. McMullan, "Fish swarm intelligent algorithm for the course timetabling problem," in *Rough Set and Knowledge Technology*. Berlin, Germany: Springer, 2010, pp. 588–595.

[75] A. Turky, N. R. Sabar, S. Dunstall, and A. Song, "Hyper-heuristic local search for combinatorial optimisation problems," *Knowl.-Based Syst.*, vol. 205, Oct. 2020, Art. no. 106264.

[76] D. S. Vianna, C. B. Martins, T. J. Lima, M. D. F. D. Vianna, and E. B. M. Meza, "Hybrid VNS-TS heuristics for university course timetabling problem," *Brazilian J. Oper. Prod. Manage.*, vol. 17, no. 1, pp. 1–20, 2020.

[77] B. Wang, Y. Geng, and Z. Zhang, "Applying genetic algorithm to university classroom arrangement problem," *J. Phys., Conf. Ser.*, vol. 1325, Oct. 2019, Art. no. 012157.

[78] D. Zhang, Y. Liu, R. M. Hallah, and S. C. H. Leung, "Discrete Optimization A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems," *Eur. J. Oper. Res.*, vol. 203, no. 3, pp. 550–558, 2010.

MEI CHING CHEN received the Bachelor of IT degree in computational science and the Master of Education degree in technical and vocational study. She is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak. Her current research interest includes hybrid algorithms with specific interest in timetabling problems.

SAN NAH SZE received the Doctor of Philosophy (Ph.D.) degree from the University of Sydney, in 2011. She is currently working as a Senior Lecturer with the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS). She has more than ten years of experience in research and development (R&D). She had presented at numerous conferences and published her work in ISI and Scopus publications. Her research interests include educational timetabling, vehicle routing, heuristics, and meta-heuristics.

SAY LENG GOH received the B.I.T. degree (Hons.) from Universiti Malaysia Sabah, Malaysia, the M.Sc. degree from Imperial College London, and the Ph.D. degree from the University of Nottingham. He is currently an Academic with the Faculty of Computing and Informatics, Universiti Malaysia Sabah. He has published in various top-tier journals, such as *European Journal of Operational Research*, *Journal of the Operational Research Society*, and *Expert Systems with Applications*. His current research interests include artificial intelligence, operational research, discrete optimisation, meta-heuristics, timetabling, and scheduling.

NASSER R. SABAR is currently a Lecturer with the Computer Science and IT Department, La Trobe University, Australia. He has published over 76 papers in international journals and peer-reviewed conferences. His current research interests include hyper-heuristic frameworks, evolutionary computation, and hybrid algorithms with a specific interest in big data optimisation problems, cloud computing, dynamic optimisation, and scheduling problems.

GRAHAM KENDALL (Senior Member, IEEE) is currently the Senior Vice President of the PETRA Group and the Chief Executive of the Good Capitalism Forum (GCF). Prior to this, he worked at the University of Nottingham for 21 years and prior to this, he worked in the IT sector, holding several senior positions. Prior to joining the PETRA Group, he worked in Malaysia for ten years, being based at the University of Nottingham's campus, Semenyih, where he was the CEO, a Provost, and a Pro-Vice Chancellor of Global University. He is an Emeritus Professor at the University of Nottingham. He is a world recognised academic, with his research focussing solving real world problems through the use of operations research and utilising artificial intelligence methods which are based on Charles Darwin's principles of natural evolution.

Dr. Kendall is (or has been, prior to leaving academia) a fellow of the British Computer Society and the Operational Research Society. He holds honorary and distinguished professorial positions at universities in Hong Kong and India. He has written over 250 peer-reviewed papers and has served on several editorial boards, including being the Editor-in-Chief of the IEEE Transactions of Computational Intelligence and AI in Games and an Associate Editor of the *Journal of the Operational Research Society*.