

A Survey of Computational Intelligence in Educational Timetabling

Kaixiang Zhu, Lily D Li, and Michael Li

Abstract—Timetabling problems have been widely studied, of which Educational Timetabling Problem (ETP) is the biggest section. Generally, ETP can be divided into three modules, namely, course timetabling, school timetabling, and examination timetabling. For solving ETP, many techniques have been developed including conventional algorithms and computational intelligence approaches. Several surveys have been conducted focusing on those methods. Some surveys target on particular categories; some tend to cover all types of approaches. However, there are lack of reviews specifically focusing on computational intelligence in ETP. Therefore, this paper aims at providing a reference of selecting a method for the applications of ETP by reviewing popular computational intelligent algorithms, such as meta-heuristics, hyper-heuristics, hybrid methods, fuzzy logic, and multi-agent systems. The application would be categorised and described into the three types of ETP respectively.

Index Terms—Computational intelligence, educational timetabling, heuristics, fuzzy logic.

I. INTRODUCTION

Timetabling problem is known as an NP-complete problem, meaning that it is difficult to provide a general optimal solution for a wide range of cases. In the past decade, there has been a large interest in researching timetabling problems. Around three thousand studies are published every year in this field, within which university timetabling is the most popular section occupying over 85% publication volume (Fig. 1). Since Appleby, Blake and Newman [1], which may be the first timetabling study in the computer field, computational timetabling has been developed over 50 years, and some surveys have been done to review those techniques [2]-[10]. However, according to the definition given by Engelbrecht [11], some of those techniques would not be able to learn, discover, reason and adapt to new situations, in other words, they are not computationally intelligent. Although some surveys focused on computational intelligence, they specifically paid attention to a single category [6]. Therefore, this paper aims at reviewing the current computational intelligent approaches in ETP and giving an overview for further research in solution model construction and algorithm selection.

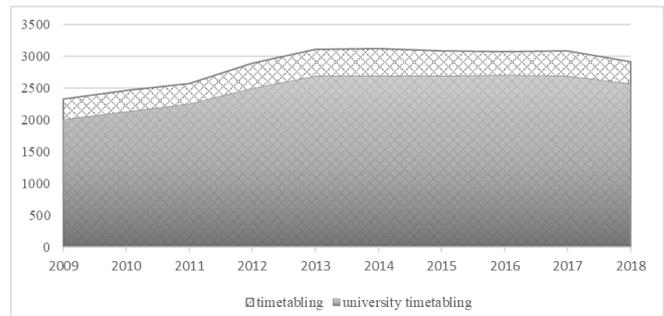


Fig. 1. Volume of publication in timetabling (2009-2018).

Educational timetabling is a significant task for ensuring every aspect runs smoothly and collaboratively. Its efficiency and reasonableness would significantly impact the operations between faculties, teaching implementations, utilisation of limited educational resources and satisfaction of educators and students. Wren [12] defined “Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives”. For educational management purpose, timetabling can be described as to distribute a number of activities, such as lectures, tutorials, exams and meetings, into a limited number of timeslots and/or room-slots [13]. Each activity has its unique conditions that need to be satisfied. Those conditions generally fall into two categories: soft constraints and hard constraints. Hard constraints are those factors of a problem cannot be violated to ensure that the timetable is feasible. Generally, hard constraints include that 1) no person can be allocated to be in more than one room in any timeslot; 2) resource requirement cannot exceed its availability in any timeslot. Soft constraints are those factors that are not necessarily to be met, but the less the violation of soft constraints the better the quality of a timetable is [14]. The sets of hard constraints and soft constraints are significantly different from university to university depending on how a university treats the importance of a constraint. Comprehensive studies for university timetabling constraints have been published in [7], [10], [15].

ETP can be classified as course timetabling, school timetabling and examination timetabling [9].

- Course timetabling, known as post-enrolment-based timetabling, is to assign lecturers and tutorials to timeslots, rooms or other facilities, against an individual student. The objective of it is to ensure that no student takes classes overlapped in timeslot.
- School timetabling, also called curriculum-based timetabling, focuses on teaching resource allocation. To some extent, school timetabling is similar to curriculum-based course timetabling [3]. However,

Manuscript received November 21, 2019; revised July 20, 2020.

The authors are with the School of Engineering and Technology, CQUniversity, Rockhampton, Q4702 Australia (e-mail: k.zhu@cqu.edu.au, l.li@cqu.edu.edu, m.li@cqu.edu.edu).

school timetabling would take the availability and specialisation (also level of specialisation) of an educator as hard constraints [16].

- Examination timetabling is to avoid students taking two or more exams simultaneously and to level resource usage over the examination period [16]. Uniquely, a room (exam place) or an educator (examiner) can be assigned to multiple courses.

ETP is idiosyncratic. The parameters, such as teachers, students, courses, rooms and timeslots, to be considered are different from university to university. The weigh, priorities of those parameters are different as well. Besides, the local educational policies and institute structures also will significantly affect the timetabling solution. Perhaps, that is one of the reasons that there was no literature providing a general approach or guide for every instance. This paper will review the three categories of the ETP and the current approaches for each of them respectively. Due to the uniqueness of ETP, many solutions were designed for a specific case and tested locally. Even some approaches tended to solve general problems, they had not chosen the same benchmark data to examine the results. Consequently, it is very difficult to compare those methodologies. Thus, this paper outlines and briefly analyses each selected method as references for further research.

The rest of the paper is organised as follows: Section II reviews the major algorithms currently being applied to ETP. Section III is divided into three subsections based on the three types of timetabling problems. Each subsection will review approaches evolved from the major algorithms. The last section concludes this paper.

II. MAJOR ALGORITHMS

The computational intelligent approaches solving ETP can be classified as heuristic approaches, novel methods and distributed multi-agent systems [2], [5].

A. Heuristic Approaches

Heuristic approaches include meta-heuristics and hyper-heuristics. Meta-heuristics is the interface of Artificial Intelligence and Operational Research using a particular technique to tackle specific problem, while hyper-heuristics is the higher-level meta-heuristics employing meta-heuristics to select meta-heuristics for general issues [17]. It could be said that meta-heuristics are specific and hyper-heuristics are general.

1) Meta-heuristics

In the nature, there are many successful mechanisms solving multi-objective and combinatorial optimisation problems, including biological systems, physical and chemical processes. These natural mechanisms inspire people to research their mechanism and lead to the popularity of meta-heuristics algorithms [18]. Timetabling problems are known as a NP-complete issue in almost all university cases [9]. As timetabling problems share the characteristics with Graph Colouring problems [13], traditionally, many timetabling algorithms were developed based on graphic colouring-based heuristics [4]. Given that metaheuristics can be applied to a set of optimisation

problem without many modifications and take both soft and hard constraints into its formulations, metaheuristics have been paid a lot of attention to tackling timetabling problems. Based on the importance of soft and hard constraints for a particular case, metaheuristic algorithms can be classified into three types: one-stage, two-stage and relaxation-allowed algorithms [6].

One-stage optimisation tolerates both soft and hard constraints to be violated in order to seek a workable solution. In implementation, soft and hard constraints will be given corresponding weights depending on their importance in different scenarios. An example of this type is Abdullah and Turabieh [19] set penalties for both soft and hard constraints and used a Tabu-based memetic algorithm to solve university course timetabling.

Two-stage optimisation looks for a solution in two steps. In step one, the algorithm searches for a feasible solution to satisfy hard constraints without considering the soft constraints. Once a feasible solution is found, soft constraint violations then will be attempted to minimise. Unlike one-stage approach, this algorithm does not require weightings. Based on this model, Yasari, Ranjbar, Jamili and Shaelaie [20] developed a stochastic approach successfully solving university timetabling problem with course cancellation risk. This method allows a pre-defined timetable to be changed and minimises the impact of such uncertainties.

Relaxation-allowed algorithm keeps hard constraints from violations and try to satisfy soft constraints under the conditions that some factors of the problem can be relaxed, such as timeslot and location [6].

As meta-heuristics have been widely studied for many years, a lot of algorithms have been introduced. Depending on the characteristics, they can be classified into several overlapping categories: (Fig. 2 listed typical algorithm being described in this paper). Swarm intelligence might be the most favoured one. Inspired by the nature, multiple swarm algorithms have emerged, including bird flocks, fish schools [21], ant colony [22], cat swarm [23] and Artificial Bee Colony (ABC). Among those approaches, ABC might be the most popular one widely studied and applied to practical ETP [24].

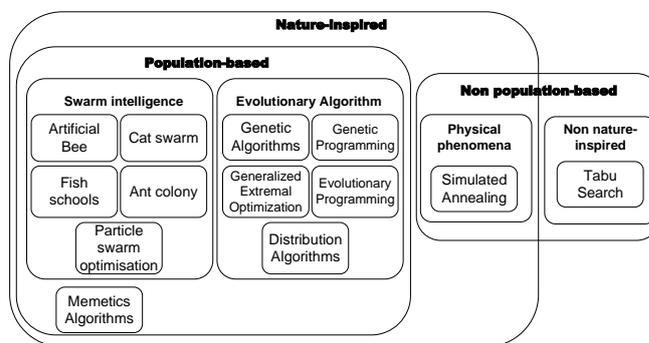


Fig. 2. Classification of meta-heuristics algorithms.

2) Hyper-heuristics

Hyper-heuristics is described as “heuristics to choose heuristics” [25]. It aims to solve combinatorial optimisation problems with more generalised solutions instead of a technique derived from specific scenarios [26]. This idea was motivated by the fact that many heuristics approaches

could only get good results for a few cases within the scope. For solving this weakness, hyper-heuristics automatically searches low-level heuristics space rather than solution space [27]-[29].

Compared to hyper-heuristics, other heuristics can be called low-level heuristics. Low-level heuristics can be classified as construction and perturbation [30]. In the implementation, hyper-heuristics will be fed with a number of low-level heuristics with particular problems, and then an adequate combinatorial solution would be produced. The product can be a low-level heuristic either selected from existing ones or newly generated [26]. Constructive heuristics play a role to construct an initial solution for later optimisation while perturbative heuristics are used to improve the initial solution generated randomly or from a constructive heuristic. Respectively, the hyper-heuristics based on those two types of low-level heuristics abovementioned are named as selection constructive, selection perturbative, generation constructive or generation perturbative [31].

Selection constructive hyper-heuristics apply a heuristic to explore the space of low-level constructive heuristics. Mainly, Case-Based Reasoning (CBR), local search and population-based methods are used to select a reasonable low-level constructive heuristic. CBR solves problems taking previous similar cases as data sources [32]-[34]. Local search methods look for a solution from the neighbourhood around the initial point and then move to another point with Tabu search [19], [35], [36] and neighbourhood search [37]. Albeit local search focuses on one point each time, population-based methods explore multiple points simultaneously [38].

Selection perturbative hyper-heuristics choose low-level perturbative heuristics to improve each point originated from the initial solution. Selection perturbative hyper-heuristics designate single-point or multipoint search to select a solution from the low-level perturbative pool. For single-point search, heuristic selection and move acceptance [39], [40] techniques are used; in multipoint search, population-based methods are employed [26].

Generation constructive hyper-heuristics can automatically provide a low-level constructive heuristic as an initial solution which normally is derived manually and intuitively. Generation constructive hyper-heuristics automatically generating a starting solution can help reduce time-cost but also would create a new constructive heuristic [28]. Genetic programming [38], [41] is the chief method used for Generation constructive hyper-heuristics.

Generation perturbative hyper-heuristics works to create a new low-level perturbative heuristic for solution improvement. Genetic programming is its major method to combine or configure a new heuristic from a low-level perturbative heuristic pool [26].

B. Novel Methods

1) Hybrid approach

A number of computational intelligent approaches have been emerging to solve university timetabling problems [2], [4]-[6], [28]. However, each individual approach has some weaknesses. Therefore, people combine different methods to mitigate the weaknesses and generate a preferable approach to a problem. The form approached being hybridised

includes integrating multiple methods for one step and employing different method on different steps of a whole process. For example, Ishak, Lee and Ibragimov [42] hybridised three techniques to reduce cost penalty in timeslot arrangement stage. In the study conducted by Fong, Asmuni and McCollum [21], Particle Swarm Optimisation (PSO) was used for exploration step and, in second step, Nelder-Mead simplex search was adopted for exploitation. This hybridised model offers a better balance between exploration and exploitation, which the former researches could not achieve.

2) Fuzzy logic approaches

In the real world, many objects do not have a precise classification and thus are hard to be quantitated and modelled. In other words, they are fuzzy. Therefore, a conceptual framework for those ambiguous problems is required, which was introduced by Zadeh [43]. In the computational intelligent field, fuzzy logic was proved more capable over traditional methods in dealing with ill-defined problems [44], such as impreciseness, uncertainties or unreliability. Those features match the characteristics of ETP which contains multiple uncertain variables, parameters and constraints. To prove the importance of applying fuzzy logic for school timetabling, Gorka and Thipwiwatpotjana [45] chose teaching preference as the weight in their research.

C. Multi-agent Systems

Multi-agent systems are artificial intelligence techniques involving many agents to achieve a common goal collaboratively [46]. Each agent within the system is a program entity, sensitive and interactive to its environment, able to communicate with each other with a common language, autonomous, able to do a task incompletely [47]. Generally, each agent entity consists of searcher who is to search a local solution, negotiator responsible for negotiating resources with other agents, and manager who is in charge the related information [2]. Since agents are cooperative, multi-agent systems can reach an optimal or sub-optimal solution [48].

III. RELATED APPLICATIONS IN ETP

The computational intelligences in ETP have been evolved and varied into many applications. These applications could only cover one or two ETP categories. Therefore, this section separates recent ETP computational intelligent applications into three ETP categories to describe and summaries them in Table III.

A. Course Timetabling

1) Heuristics approaches

Van den Broek and Hurkens [49] employed an integer-programming-based (IP-based) with column generation to solve post-enrolment timetabling problems. The proposed approach was tested with the Track 2 of second international timetabling competition (ITC2007) dataset with timeslots and room-slot satisfaction without hard constraints violation. Soria-Alcaraz, Ochoa, Swan, Carpio, Puga and Burke [39] combined Iterated Local Search (ILS) with two types of learning mechanism: static (offline) and dynamic (online)

learning. This algorithm could automatically generate or select new improved solution based on the previous stage on each iteration, which helped it obtained a competitive result from Track 2 of ITC2007. A year later, Soria-Alcaraz, Özcan, Swan, Kendall and Carpio [50] optimised ILS approach integrating with generation perturbative hyper-heuristic approach for post-enrolment timetabling and curriculum-based timetabling problems. Add-delete operators were used to improve the selected processes. The add or delete symbol was for indicating an event being temperately removed or reassigned from or to the timetable on the event list. This approach can be applied to both Track 2 and Track 3 of ITC 2007.

Table I shows the results of abovementioned heuristics being tested with Track 2 of ITC2007. The solution timetable they generated were all feasible (i.e. hard constraints violation is 0). The data represents the number of soft constraint violation. It is obvious that the Soria-Alcaraz, Özcan, Swan, Kendall and Carpio [50] had the best performance in this dataset.

TABLE I: TRACK 2 TEST RESULTS OF SOFT CONSTRAINT VIOLATION (THE BEST RESULT IS IN BOLD)

ITC-2007	VH	SOSCPB	SOSKC
1	1636	650	630
2	1634	470	450
3	355	290	300
4	644	600	602
5	525	35	6
6	640	20	0
7	0	30	0
8	241	0	0
9	1889	630	640
10	1677	2349	663
11	615	350	344
12	528	480	198
13	485	46	0
14	739	80	35
15	330	0	0
16	260	0	140
17	35	0	0
18	503	20	0
19	963	360	400
20	1229	150	150
21	670	0	0
22	1956	33	32
23	2368	1007	238
24	945	0	640

Note: on Table I VH refers to Van den Broek and Hurkens [49], SOSCPB means Soria-Alcaraz, Ochoa, Swan, Carpio, Puga and Burke [39] and SOSKC stands for Soria-Alcaraz, Özcan, Swan, Kendall and Carpio [50]. The best results are in bold.

2) Hybrid approaches

Bolaji, Khader, Al-Betar and Awadallah [51] integrated ABC, and local search algorithm, Hill Climbing (HC), together for post-enrolment course timetabling problems. ABC works as its classical procedure but in neighbourhood food source search stage, HC will help to seek for the best solution in objective cost value. As this hybridisation had good structured exploitation to balance global exploration and local exploitation, it achieved good performance in small to large data instances. Besides room features and room capacity constraints were satisfied. Recently, Akkan and Gülcü [52] modelled bi-criteria solution by hybridising Hill Climbing and Simulated Annealing algorithms. This approach populates a solution by genetic algorithm and

improves the solution in each iteration by two hill-climbing operators. This approach was tested with ITC 2007 dataset and gained high robustness and created high-performance solutions.

3) Fuzzy logic approaches

Kohshori, Abadeh and Sajedi [53] proposed a fuzzy genetic algorithm with a randomised iterative algorithm for local search. During the process, solutions were initialised and selected randomly. Crossover and mutation operators were used for improvement. Fuzzy sets were the evaluators for soft constraints violation. The simulation experiment results were obviously better than conventional genetic algorithms in many different constraint situations. A similar approach was applied by Perzina and Ramik [54] two year later. Uniquely, the formulation was constructed with self-learning genetic algorithm along with event priority constraints, in which deletion and duplication operators were used to control polyploidy. Performance of this formulation was compared to a manual feasible timetable showing that it satisfied every soft constraint with high quality. The drawback of this approach is its computational complexity.

4) Multi-agent systems

Pedroso [55] established a Multi-Agent System (MAS) for the University of Porto to tackle course timetabling problem, specifically for room sharing. In the case, rooms were shared to different faculty and requirement of timeslot length was different from department to department. A greedy algorithm was used to search for best solution and a dispatch method was employed to assign the rooms. The computational result showed that this system could tackle clashes between rooms, events and students. While Kaplansky and Meisels [56] built up a MAS model to solve the shared-course problem for Ben-Gurion University, University of Udine did not have a centralised timetable and each faculty had its private room. Besides, those faculties maintained their timetable autonomously and were reluctant to share the information of room availability. However, they needed external room resources to mitigate their room-slot conflict. Therefore, Di Gaspero, Mizzaro and Schaerf [57] proposed a new architecture for those issues. The architecture treated each faculty as an independent entity of the system. Each unit employed three characters: solver for searching local solution, negotiator to communicate with other entities, and manager in charge of information management and updating. This architecture helped reach high satisfaction in practice. Based on the abovementioned technique, Yang and Paranjape [58] introduced mobile agent. In the study, a timetable was divided into five platform from Monday to Friday. Each department entity assigns solvers and managers to every platform. The negotiator can move among platforms to negotiate a compromise solution. Recently, Houhamdi, Athamena, Abuzaineddin and Muhairat [59] developed a MAS application to support timetable generation in reality by imitating what a human planner will do, which get preferences of students and faculty members involved.

B. School Timetabling

1) Heuristics methods

Beligiannis, Moschopoulos and Likothanassis [60]

proposed an adaptive genetic algorithm to help Greece high schools generate a workable and efficient timetable. The methods mainly focused on teacher allocation with a “teacher-course-class” circuit. A tight chromosome encoding scheme was adopted to avoid solution seeking agent diving too deep in a wrong search space. The methods had been exhaustively tested with data collecting from many high schools in the city of Patras. Although the soft constraint violation could not be well minimised, the test result showed the algorithm was efficient and effective with low implementation cost. Also, for solving high school timetabling problems, Odeniyi, Omidiora, Olabiyisi and Aluko [61] developed a Modified Simulated Annealing (MSA) approach. Simulated annealing algorithm is efficient for nonlinear combinatorial optimisation problems, but it takes considerable time for convergence in large search spaces. Hence, a temperature reduction parameter was introduced to make the cooling schedule parabolic. This approach was successfully implemented in Fakunle Comprehensive High School in Nigeria with the result of reducing convergence time and computational cost.

2) *Hybrid methods*

Skoullis, Tassopoulos and Beligiannis [23] applied Cat Swarm Optimisation (CSO) to specifically solve school timetabling problem for high schools in Greece. CSO belongs to nature-inspired swarm algorithm category. It mimics the seeking food behaviours of cats in two steps: seeking and chasing. In [23], seeking stage is introduced in global variable to find out the best cat releasing for food sources in each iteration; and a swap operator is integrated to change the status of a cat between seeking and chasing. These two modifications helped reduce the computational time and achieved high performance in Beligiannis benchmark dataset test. On the other hand, Sutar and Bichkar [62] synthesised conversional algorithms, tabu and genetic, to generate a faster solution. However, a workload preserving crossover operator was used to avoid overlaps. This hybridised genetic algorithm could generate a solution in a few seconds.

3) *Fuzzy logic*

With the purpose of increasing lecturer satisfaction and minimising resource usage, Babaei, Karimpour and Oroji [63] employed fuzzy c-means clustering algorithm for Islamic Azad University. To some extent, this method also falls into MAS as four agents were assigned to work for local information management, global information clustering, negotiating and global information management. Fuzzy c-means helped to optimise the timetable with feature weight applied to soft constraints. The proposed approach centrally managed resource over every faculty led to resource redundancy reduction. Besides, the preferences of lecturers were considered.

4) *Multi-agent system*

Oprea [64] noticed the lack of research on computational intelligence in educational administration which involves many communication, cooperation and negotiation processes. In [64], it was proved that MAS_UP-UCT could potentially deal with negotiations between different faculties(agents) and minimised conflicts as the algorithm was able to analyse and exchange information in the

computational process. Initially, each faculty creates a timetable according to courses’ characteristics and lecturers’ interest. This task will be done by a faculty scheduler agent. After lecturer assignment, rooms will be allocated based on a faculty timetable. The arrangement will be done by a university scheduler agent. If there is a conflict toward room assignment. Faculty scheduler agents will enforce negotiation strategy. However, this approach could not satisfy all the lecturers’ preferences. Tkaczyk, Ganzha and Paprzycki [65] based on actual case of University of Gdansk mimicked real workflow to develop a MAS. Each node of the workflow was programmatically defined as an agent including boot agent, database agent, room agent, teacher agent and scheduler. Those agents work as their names suggest. The test result of this approach satisfies the need of University of Gdansk.

C. Examination Timetabling

1) *Heuristic methods*

Kasm, Mohandes, Diabat and El Khatib [66] combined constructive heuristic along with novel colour graphing algorithm to solve the exam timetabling problem of Masdar Institute. This approach overcome the limitation of Integer Programming that only can handle small size problems and optimised two constraints that arranging two course exams to separated days and exam room capacity. Burke and Bykov [67] introduced a new local search technique for meta-heuristics for solving exam timetabling problems and named it as Late Acceptance Strategy. This algorithm is a subclass of Hill Climbing which accepts the better solution from the previous few iterations. Due to its simplicity and easiness of implementation, this algorithm gained popularity. Hence, Bykov and Petrovic [68], Bykov and Petrovic [69] developed it further to be Step Counting Hill Climbing (SCHC). SCHC remains all the merits of Hill Climbing. Additionally, the counting mechanism is easy to be implemented and to be implemented in many steps of the solution selection process.

2) *Hybrid approaches*

Bolaji, Khader, Al-Betar and Awadallah [70] hybridised ABC with Simple Local Search (SLS) and Harmony Search (HS). The SLS improved local exploitation while HS controlled diversity and fasted convergence. In the same year, Alzaqebah and Abdullah [71] introduced several algorithms to modify the original ABC. On selection stage, tournament selection, rank selection and disruptive selection were proposed to diversify the population; on neighbourhood search phase, self-adaptive mechanism was employed to maintain the useful neighbourhood structure to guide the search; on local search step, simulated annealing and late acceptance hill climbing algorithms were adopted to filter improved solutions and demolish the worse ones. Inspired by PSO, Fong, Asmuni and McCollum [21] implemented a “global best” model into ABC to improve the search process. Meanwhile, Nelder-Mead simplex search (NMSS) and Great Deluge (GD) algorithms were integrated to enhance exploitation. By combining these two algorithm sets, both exploration and exploitation were optimised. Three of abovementioned approaches were tested on Carter un-capacitated examination timetabling dataset. In Table II, it can be seen that the three approaches are competitive to

the best known result produced by Burke and Bykov [67]. The reason to choose this result as the reference is its capability and feasibility was verified in [8].

TABLE II: RESULT COMPARISON OF HYBRID APPROACHES FOR EXAMINATION TIMETABLING

Carter	A.A		BKABA		FAMC		BB	
	Best	Avg	Best	Avg	Best	Best	Avg	Best
car91I	4.38	4.52	5	5.05	3.89	4.27	4.58	4.68
car92I	3.88	4.09	4.22	4.29	4.79	4.85	3.81	3.92
ear83 I	33.34	33.66	34.52	34.86	33.43	34.48	32.65	32.91
hec92 I	10.39	10.9	10.68	10.78	10.49	10.61	10.06	10.22
kfu93	13.23	13.46	14.02	14.17	13.72	13.76	12.81	13.02
lse91	10.52	10.82	11.04	14.17	10.29	10.39	9.86	10.14
rye92	8.92	9.26	9.28	9.49	N/A	N/A	7.93	8.06
sta83 I	157.06	157.16	157.04	157.13	157.07	157.37	157.03	157.05
tre92	7.89	8.09	8.38	8.47	7.86	8.04	7.72	7.89
uta92 I	3.13	3.23	3.4	3.45	3.1	3.31	3.16	3.26
ute92	25.12	25.33	25.8	26.17	25.33	26.04	24.79	24.82
yor83 I	35.49	35.69	37.53	37.69	36.12	36.67	34.78	35.16

Note: on Table II, AA stands for Alzaqebah and Abdullah [71], BKABA represents Bolaji, Khader, Al-Betar and Awadallah [70] and FAMC refers to Fong, Asmuni and McCollum [21], BB: Burke and Bykov [67]. The best results are in bold

3) Fuzzy logic approaches

Asmuni [72] proved that fuzzy approach would have significant potential to evaluate examination timetabling solutions. Five criteria, largest degree, saturation degree, largest enrolment, largest coloured degree and weighted largest degree, had been examined in the studied. The experimental results showed fuzzy multiple heuristic orderings suppressed wide range of algorithms in examination timetabling research filed. Chaudhuri and De [73] applied fuzzy logic to solve a real-world problem. A fuzzy integer linear programming was employed to Netaji Subhas Open University in India, which not only tackled the examination timetabling problem in the university in high quality but also obtained good result when being tested in Carter dataset. Cavdur and Kose [74] introduced a fuzzy-criticality methodology to identify the exam criticalities and then generated a balanced-exam timetabling solution. This approach could satisfy the scenario integrated with different seniority and examiner distribution, and had been applied to Uludag University gaining a better solution in satisfaction than the one generated by the human expert.

TABLE III: SUMMARY OF RELATED APPLICATIONS IN ETP

Categories	Approaches	Methodologies	Performances	References
Course timetabling	Heuristics	IP-based with column generation	Satisfied time-slot and room-slot without hard constraints violations	[49]
		ILS with static and dynamic learning	Automatically generate or select new improved solution on each iteration	[39]
		ILS with generation perturbative hyper-heuristic approach	Satisfied both Track 2 and Track 3 of ITC2007	[50]
	Hybrid	ABC + LS + HC	Had good performances in small to large data instances and satisfied room constraints	[51]
		HC + SA + GA	Gained high performance in ITC 2007	[52]
	Fuzzy Logic	Fuzzy GA with randomised iterative algorithm for local search	Gained better results in many different constraints situations	[53]
	MAS	Self-learning genetic algorithm	Satisfied every soft constraint with high quality	[54]
		Greedy algorithm to search best solution, dispatch method to assign rooms	Tackled rooms, events and students conflicts	[55]
		Treat every faculty as an entity	Reach high performance in practice	[57]
		Treat every workday as an entity	Gained effective and flexible solution for the practical problem	[58]
		Imitates human planner	Get preferences of students and faculty members involved	[59]
	School timetabling	Heuristics	With tight chromosome encoding scheme	Low implementation cost
Modified SA with temperature reduction parameter			Reduce convergence time and computational cost	[61]
Hybrid		Modified CSO with swap operator	Gained good result in Beligiannis benchmark dataset	[23]
		Tabu + GA	Could generate a solution in a short time	[62]
Fuzzy Logic		Employed fuzzy c-means clustering	Reduce resource redundancy	[63]
MAS		Uses MAS_UP-UCT	Enhance negotiation between faculties and minimise conflicts.	[64]
		Treat each workflow of the university as an agent	Satisfied the need of University of Gdansk	[65]
Examination timetabling	Heuristics	Integrated colour graphing algorithm	Can handle big size problems	[66]
		Late Acceptance Strategy	Simple and easy to implement	[67]
		SCHC	Easy to be implemented in many steps of solution selection process	[68, 69]
	Hybrid	ABC + SLS + HS	Improve local exploitation and fast convergence	[70]
		ABC + tournament, rank and disruptive selection + self-adaptive mechanism + SA + late acceptance HC	Gained good result in Carter dataset	[71]
		ABC + NMSS + GD		[21]
		Fuzzy multiple heuristic orderings	Gained better results than many algorithms	[72]
	Fuzzy Logic	Fuzzy interger linear	Gained high performance in the university and good result in Carte dataset	[73]
		Fuzzy-criticality	Gained good result in the university	[74]

IV. CONCLUSION

This paper reviewed the computational intelligence applied to educational timetabling problems. After introducing major intelligent algorithms being used in

educational scheduling section, those approaches evolved from major algorithms were classified into three categories, namely, course timetabling, school timetabling and examination timetabling, with briefly descriptions. Due to the idiosyncratic traits, timetabling problems do not have a

general solution. Each method only tackles a specific scenario or a narrow range of instances. Besides, those approaches described in this paper had not been tested with a general dataset resulting in a difficulty to compare them.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this paper.

AUTHORS CONTRIBUTIONS

Kaixiang Zhu conducted the research, analysed the data and wrote the paper. Dr Lily Li guided analysis and revised the article. Dr Michael Li participated in the discussion and provided advices to the research.

REFERENCES

- [1] J. Appleby, D. Blake, and E. Newman, "Techniques for producing school timetables on a computer and their application to other scheduling problems," *The Computer Journal*, vol. 3, no. 4, pp. 237-245, 1961.
- [2] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Computers & Industrial Engineering*, vol. 86, pp. 43-59, 2015.
- [3] A. Bettinelli, V. Cacchiani, R. Roberti, and P. Toth, "An overview of curriculum-based course timetabling," *TOP*, vol. 23, no. 2, pp. 313-349, July 1 2015.
- [4] M. W. Carter, "OR practice—A survey of practical applications of examination timetabling algorithms," *Operations Research*, vol. 34, no. 2, pp. 193-202, 1986.
- [5] M.-R. Feizi-Derakhshi, H. Babaei, and J. Heidarzadeh, *A Survey of Approaches for University Course Timetabling Problem*, 2012.
- [6] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," *OR Spectrum*, vol. 30, no. 1, pp. 167-190, 2008.
- [7] N. Pillay, "A survey of school timetabling research," *Annals of Operations Research*, vol. 218, no. 1, pp. 261-293, July 1, 2014.
- [8] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee, "A survey of search methodologies and automated system development for examination timetabling," *Journal of Scheduling*, vol. 12, no. 1, pp. 55-89, February 1, 2009.
- [9] A. Schaefer, "A survey of automated timetabling," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 87-127, 1999.
- [10] N. Pillay, *An Overview of School Timetabling Research*, 2010.
- [11] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, 2007.
- [12] A. Wren, *Scheduling, Timetabling and Rostering — a Special Relationship?* Springer, 1995.
- [13] R. Lewis, B. Paechter, and O. Rossi-Doria, *Metaheuristics for University Course Timetabling*, Springer, 2007.
- [14] S. Petrovic and E. K. Burke, *University Timetabling*, 2004.
- [15] S. Kristiansen and T. R. Stidsen, *A Comprehensive Study of Educational Timetabling — A Survey*, 2013.
- [16] C. Valouxis, C. Gogos, P. Alefragis, and E. Housos, "Decomposing the high school timetable problem" *Practice and Theory of Automated Timetabling (PATAT 2012)*, *Son, Norway*, vol. 61, 2012.
- [17] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg, *Hyper-Heuristics: An Emerging Direction in Modern Search Technology*, Springer, 2003.
- [18] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic algorithms in modeling and optimization," *Metaheuristic Applications in Structures and Infrastructures*, pp. 1-24, 2013.
- [19] S. Abdullah and H. Turabieh, "On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems," *Information Sciences*, vol. 191, pp. 146-168, 2012.
- [20] P. Yasari, M. Ranjbar, N. Jamili, and M.-H. Shaelaie, "A two-stage stochastic programming approach for a multi-objective course timetabling problem with courses cancelation risk," *Computers & Industrial Engineering*, vol. 130, pp. 650-660, 2019.
- [21] C. W. Fong, H. Asmuni, and B. McCollum, "A hybrid swarm-based approach to university timetabling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 870-884, 2015.
- [22] K. Patrick and Z. Godswill, "Greedy ants colony optimization strategy for solving the curriculum based university course timetabling problem," *Journal of Advances in Mathematics and Computer Science*, pp. 1-10, 2016.
- [23] V. I. Skoullis, I. X. Tassopoulos, and G. N. Beligiannis, "Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm," *Applied Soft Computing*, vol. 52, pp. 277-289, 2017.
- [24] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21-57, 2014.
- [25] P. Cowling, G. Kendall, and E. Soubeiga, *A Hyperheuristic Approach to Scheduling a Sales Summit*, Springer Berlin Heidelberg, 2001.
- [26] N. Pillay and R. Qu, *Hyper-Heuristics: Theory and Applications*, Springer, 2018.
- [27] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, "Recent advances in selection hyper-heuristics," *European Journal of Operational Research*, 2019.
- [28] 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," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695-1724, December 1, 2013.
- [29] E. Burke, M. Dror, S. Petrovic, and R. Qu, *Hybrid Graph Heuristics within a Hyper-Heuristic Approach to Exam Timetabling Problems*, Springer, 2005.
- [30] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, *A Classification of Hyper-Heuristic Approaches*, Springer, 2010.
- [31] E. Burke, T. Curtois, and M. Gendreau, *Iterated Local Search vs. Hyper-Heuristics: Towards General-Purpose Search Algorithms*, IEEE, 2010.
- [32] E. K. Burke, B. MacCarthy, S. Petrovic, and R. Qu, *Case-based Reasoning in Course Timetabling: An Attribute Graph Approach*, Springer, 2001.
- [33] E. K. Burke, B. L. MacCarthy, S. Petrovic, and R. Qu, "Multiple-retrieval case-based reasoning for course timetabling problems," *Journal of the Operational Research Society*, vol. 57, no. 2, pp. 148-162, 2006.
- [34] T. Zhe et al., *The Application of Case-Based Reasoning Retrieve in Course Timetabling*, IEEE, 2015.
- [35] T. Islam, Z. Shahriar, M. A. Perves, and M. Hasan, "University timetable generator using Tabu search," *Journal of Computer and Communications*, vol. 4, no. 16, p. 28, 2016.
- [36] S. Jaengchuea and D. Lohpetch, "A hybrid genetic algorithm with local search and tabu search approaches for solving the post enrolment based course timetabling problem: Outperforming guided search genetic algorithm," IEEE, 2015.
- [37] N. Pillay and R. Qu, *Examination Timetabling Problems*, Springer, 2018.
- [38] N. Pillay, "Evolving hyper-heuristics for the uncapacitated examination timetabling problem," *Journal of the Operational Research Society*, vol. 63, no. 1, pp. 47-58, 2012.
- [39] J. A. Soria-Alcaraz, G. Ochoa, J. Swan, M. Carpio, H. Puga, and E. K. Burke, "Effective learning hyper-heuristics for the course timetabling problem," *European Journal of Operational Research*, vol. 238, no. 1, pp. 77-86, 2014.
- [40] E. Ozcan, Y. Bykov, M. Birben, and E. K. Burke, *Examination Timetabling Using Late Acceptance Hyper-Heuristics*, IEEE, 2009.
- [41] M. Bader-El-Den, R. Poli, and S. Fatima, "Evolving timetabling heuristics using a grammar-based genetic programming hyper-heuristic framework," *Memetic Computing*, vol. 1, vol. 3, p. 205, 2009.
- [42] S. Ishak, L. Lee, and G. Ibragimov, "Hybrid genetic algorithm for University examination timetabling problem," *Malaysian Journal of Mathematical Sciences*, vol. 10, no. 2, pp. 145-178, 2016.
- [43] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338-353, 1965.
- [44] C. P. Pappis and C. I. Siettos, *Fuzzy Reasoning*, Springer US, 2005.
- [45] A. Gorka and P. Thipwiwatpotjana, "The importance of fuzzy preference in course assignment problem," *Mathematical Problems in Engineering*, 2015.
- [46] K. Y. Junn, J. H. Obit, R. Alfred, and J. Bolongkikit, *A Formal Model of Multi-agent System for University Course Timetabling Problems*, Springer Singapore, 2019.
- [47] S. Srinivasan, J. Singh, and V. Kumar, "Multi-agent based decision support system using data mining and case based reasoning," *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 4, p. 340, 2011.

- [48] P. Wangmaeteekul, *Using Distributed Agents to Create University Course Timetables Addressing Essential & Desirable Constraints and Fair Allocation of Resources*, Durham University, 2011.
- [49] J. V. den Broek and C. A. Hurkens, "An IP-based heuristic for the post enrolment course timetabling problem of the ITC2007," *Annals of Operations Research*, vol. 194, no. 1, pp. 439-454, 2012.
- [50] J. A. Soria-Alcaraz, E. Özcan, J. Swan, G. Kendall, and M. Carpio, "Iterated local search using an add and delete hyper-heuristic for university course timetabling," *Applied Soft Computing*, vol. 40, pp. 581-593, 2016.
- [51] A. L. A. Bolaji, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "University course timetabling using hybridized artificial bee colony with hill climbing optimizer," *Journal of Computational Science*, vol. 5, no. 5, pp. 809-818, 2014.
- [52] C. Akkan and A. Güllü, "A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem," *Computers & Operations Research*, vol. 90, pp. 22-32, 2018.
- [53] M. S. Kohshori, M. S. Abadeh, and H. Sajedi, *A Fuzzy Genetic Algorithm with Local Search for University Course Timetabling*, IEEE, 2011.
- [54] R. Perzina and J. Ramik, "Timetabling problem with fuzzy constraints: a self-learning genetic algorithm," *Constraints*, vol. 3, pp. 105-113, 2013.
- [55] J. P. Pedroso, "A multi-agent system for automated timetabling with shared resources," *Faculdade de Ciências da Universidade do Porto Departamento de Ciência de Computadores Rua do Campo Alegre*, 2003, pp. 8234150-8234180.
- [56] E. Kaplansky and A. Meisels, *Negotiation among Scheduling Agents for Distributed Timetabling*, Citeseer, 2004.
- [57] Di Gaspero, S. Mizzaro, and A. Schaerf, *A Multiagent Architecture for Distributed Course Timetabling*, 2004.
- [58] Y. Yang and R. Paranjape, "A multi-agent system for course timetabling," *Intelligent Decision Technologies*, vol. 5, no. 2, pp. 113-131, 2011.
- [59] Z. Houhamdi, B. Athamena, R. Abuzaineddin, and M. Muhairat, *A Multi-Agent System for Course Timetable Generation*, 2019.
- [60] G. N. Beligiannis, C. Moschopoulos, and S. D. Likothanassis, "A genetic algorithm approach to school timetabling," *Journal of the Operational Research Society*, vol. 60, no. 1, pp. 23-42, 2009.
- [61] O. Odeniyi, E. Omidiora, S. Olabiyisi, and J. Aluko, "Development of a modified simulated annealing to school timetabling problem," *International Journal of Applied Information Systems*, vol. 8, no. 2, pp. 16-24, 2015.
- [62] S. R. Sutar and R. S. Bichkar, "High school timetabling using Tabu search and partial feasibility preserving genetic algorithm," *International Journal of Advances in Engineering & Technology*, vol. 10, no. 3, p. 421, 2017.
- [63] H. Babaei, J. Karimpour, and H. Oroji, *Using Fuzzy C-Means Clustering Algorithm for Common Lecturers Timetabling among Departments*, IEEE, 2016.
- [64] M. Oprea, "MAS_UP-UCT: A multi-agent system for university course timetable scheduling," *International Journal of Computers Communications & Control*, vol. 2, no. 1, pp. 94-102, 2007.
- [65] R. Tkaczyk, M. Ganzha, and M. Paprzycki, "Agent planner-agent-based timetabling system," *Informatica*, vol. 40, no. 1, 2016.
- [66] O. A. Kasm, B. Mohandes, A. Diabat, and S. El Khatib, "Exam timetabling with allowable conflicts within a time window," *Computers & Industrial Engineering*, vol. 127, pp. 263-273, 2019.
- [67] E. K. Burke and Y. Bykov, *A Late Acceptance Strategy in Hill-Climbing for Exam Timetabling Problems*, 2008.
- [68] Y. Bykov, and S. Petrovic, "A step counting hill climbing algorithm applied to university examination timetabling," *Journal of Scheduling*, vol. 19, no. 4, pp. 479-492, August 1, 2016.
- [69] Y. Bykov and S. Petrovic, *An Initial Study of a Novel Step Counting Hill Climbing Heuristic Applied to Timetabling Problems*, 2013.
- [70] A. L. A. Bolaji, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "A hybrid nature-inspired artificial bee colony algorithm for uncapacitated examination timetabling problems," *Journal of Intelligent Systems*, vol. 24, no. 1, pp. 37-54, 2015.
- [71] M. Alzaqebah and S. Abdullah, "Hybrid bee colony optimization for examination timetabling problems," *Computers & Operations Research*, vol. 54, pp. 142-154, 2015.
- [72] H. Asmuni, *Fuzzy Methodologies for Automated University Timetabling Solution Construction and Evaluation*, University of Nottingham, 2008.
- [73] A. Chaudhuri and K. De, *Fuzzy Integer Linear Programming Mathematical Models for Examination Timetable Problem*, arXiv preprint arXiv:1307.1900, 2010.
- [74] F. Cavdur and M. Kose, "A fuzzy logic and binary-goal programming-based approach for solving the exam timetabling problem to create a balanced-exam schedule," *International Journal of Fuzzy Systems*, vol. 18, no. 1, pp. 119-129, February 1, 2015.

Copyright © 2021 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.



Kaixiang Zhu is a research student and a sessional lecturer of CQUniversity Australia. He completed his MIT in 2019. Zhu's educational background includes five years in clinical medicine (Guangzhou Medical University), multiple certificates in IT for ongoing professional development (PRC's Ministry of Labour and Social Security, Ministry of Personnel), and over fifteen years of working in IT with experiences in China and in Australia. In 2019,

Zhu was inducted as a member of the Golden Key International Honour Society.



Lily Li received a B Eng. from Xi'an Jiaotong University (China), a MIT from the University of Newcastle (Australia), a PhD and a Grad. Cert. in Tertiary Education from CQUniversity (Australia). Dr Li has been a full time academic at CQUniversity since 2002. Her experiences consist of course coordination, teaching and learning management, curriculum design and research. Dr Li's research interests include computational intelligence, evolutionary optimisation and software engineering. She is a member of the Australia Computer Society (ACS).



Michael Li received the PhD degree in 2003 from the University of Newcastle, Australia. Currently he is a senior lecturer with the School of Engineering and Technology, Central Queensland University, Australia. His research interests include optimization algorithms, neural networks, probabilistic modelling, and Markov Chain Monte Carlo simulation.